Hybrid Travel Recommendation System: Integrating Collaborative and Content-Based Filtering Techniques

A Project Report submitted by

Sambit Mohanty

M22AI622

in partial fulfilment of the requirements for the award of the degree of

Master of Technology
in
Data and Computational Science

॥ त्वं ज्ञानमयो विज्ञानमयोऽसि ॥

Indian Institute of Technology, Jodhpur

School of Artificial Intelligence and Data Science

January 2024

DECLARATION

I hereby declare that the work presented in this Project Report titled Hybrid Travel Recommendation System: Integrating Collaborative and Content-Based Filtering Techniques -Master of Technology in Data and Computational Science submitted to the Indian Institute of Technology Jodhpur in partial fulfilment of the requirements for the award of the degree of Master of Technology in Data and Computational Science submitted., is a bonafide record of the research work carried out under the supervision of Professor Dr. Sandeep Kumar Yadav. The contents of this Project Report in full or in parts, have not been submitted to, and will not be submitted by me to, any other Institute or University in India or abroad for the award of any degree or diploma.

Signature

Mr. Sambit Mohanty

M22AI622

CERTIFICATE

This is to certify that the Project Report titled Hybrid Travel Recommendation System: Integrating Collaborative and Content-Based Filtering Techniques, submitted by Mr. Sambit Mohanty (M22AI622) to the Indian Institute of Technology Jodhpur for the award of the degree of Master of Technology in Data and Computational Science is a bonafide record of the research work done by him under my supervision. To the best of my knowledge, the contents of this report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Signature

Dr. Sandeep Kumar Yadav
Department of Electrical Engineering
Indian Institute of Technology Jodhpur
Karwar, 342037, India

ABSTRACT

Within the sphere of contemporary travel, personalisation is an essential component in increasing the user experience and ensuring that the customer is satisfied. The goal of this project is to develop a novel hybrid recommendation algorithm that is specifically designed for the travel industry. This algorithm combines the benefits of collaborative filtering approaches with content-based filtering techniques. The suggested algorithm generates personalised travel recommendations by taking into account not only the preferences of the user but also their previous interactions and the content characteristics of the destinations they visit. We have demonstrated that our technique is superior in terms of suggestion accuracy and coverage by conducting comprehensive experiments and evaluations on real-world datasets. This approach addresses the intrinsic challenges that are associated with the travel industry, which include limited data and various consumer preferences. The results of this study not only help in the development of a reliable recommendation system for travellers, but they also pave the way for additional developments in hybrid algorithmic designs across a variety of different fields.

Keywords: Content-based filtering algorithm, Collaborative filtering-based algorithm, Utility matrix, Sparse Density, Cosine Similarity.

TABLE OF CONTENTS

CHAPTER-1
INTRODUCTION

In recent years, the field of recommendation systems has undergone tremendous development. These advancements have been driven by the ever-increasing desire for personalised user experiences across a variety of disciplines. These kinds of technical advancements have the potential to be of tremendous advantage to the travel industry, which is a significant industry that feeds on user-centric solutions. A deeper connection between travellers and their locations can be fostered through the capacity to recommend destinations, lodgings, and activities that are personalised to individual tastes. This not only improves the user experience but also serves to enrich the user experience.

Collaborative filtering, rooted in the idea of leveraging user interactions and preferences, offers insights into patterns and trends by examining user-item interactions. However, it often faces challenges in situations where user data is sparse or when new items are introduced into the system. On the other hand, content-based filtering, which focuses on the intrinsic attributes of items and user profiles, provides recommendations based on similarity metrics. Yet, it may overlook serendipitous recommendations or fail to capture evolving user preferences over time.

Recognising these challenges, this research introduces a hybrid travel-based recommendation algorithm that synergistically combines both collaborative and content-based filtering approaches. By harnessing the complementary strengths of these methodologies, the proposed algorithm aims to provide users with a more comprehensive and accurate set of travel recommendations. This introduction sets the stage for a deeper exploration of the hybrid algorithm, its design principles, implementation details, and the potential impact on enhancing the travel planning and booking experience for users worldwide.

.

CHAPTER-2
LITERATURE SURVEY

2.1 A systematic review and research perspective on recommender systems [1]

Deepjyoti Roy et al. (2022) classified the recommender systems into 3 categories: content-based recommender systems, collaborative filtering-based recommender systems, and optimization-based recommender systems. However, the scope of our research is limited to content-based and collaborative filtering-based algorithms only. Further categorization is done in Figure 1.

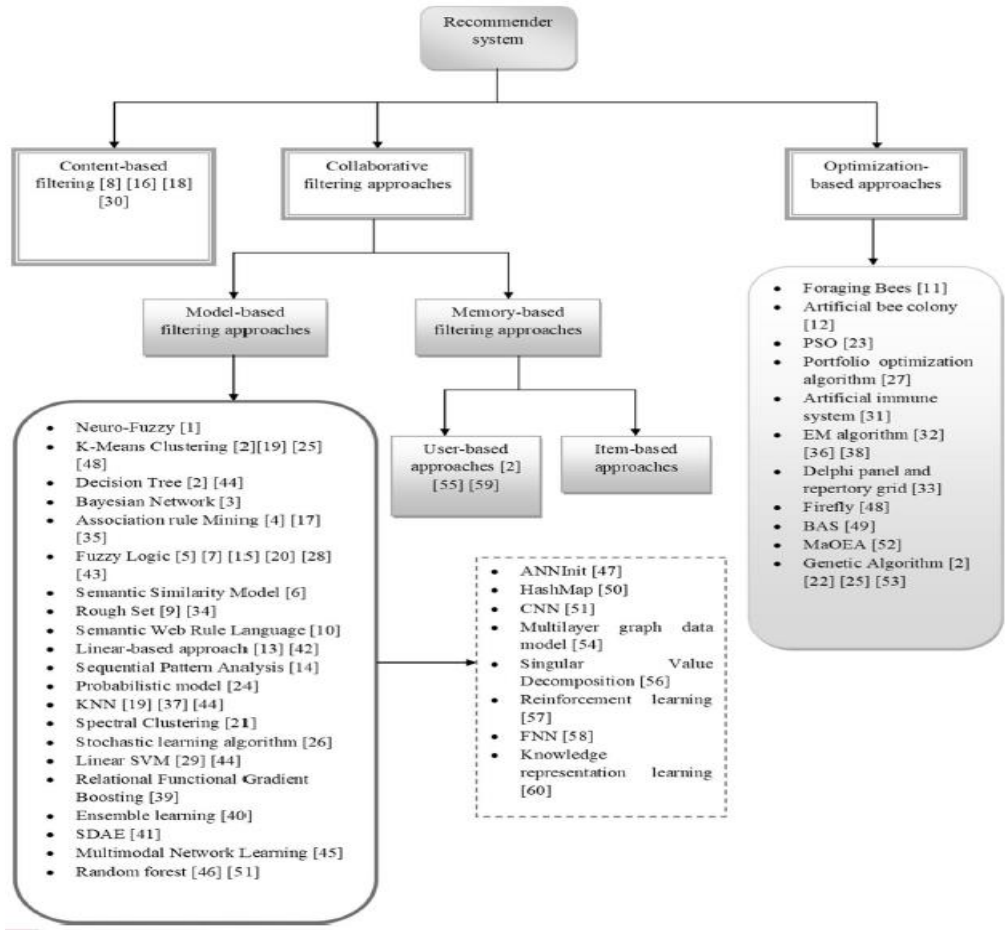

Figure 1: Categorization of Recommender Systems

In our research, for model-based approaches, we have covered K-Means Clustering, KNN, Singular Value Decomposition (SVD), Normal Predictor (based on normal distribution), and some other matrix factorization algorithms.

2.2 Netflix Recommendation System based on TF-IDF and Cosine Similarity Algorithms [2]

Mohamed Chiny et al. (2022) implemented a recommender system using TF-IDF and cosine similarity, often used in natural language processing (NLP). TF-IDF stands for term frequency (inverse document frequency). Term frequency is the ratio of the number of times a term appears in a document to the total number of terms in the document. Inverse document frequency is the logarithm of the ratio of the total number of documents to the number of documents containing the term. Fig. 2 shows the formula.

$$tf_i = \frac{n_i}{\sum_k n_k} \quad (1) \qquad idf_i = \log \frac{|D|}{|\{d_j : t_i \in d_j\}|} \quad (2)$$

Figure 2: Formula for TF and IDF

The TF-IDF score is the product of the TF and IDF values. Cosine similarity is the angle between the two movie vectors, where each vector has a TF-IDF component.

$$\cos\theta = \frac{A.B}{\|A\|\ \|B\|}$$

Figure 3: Cosine Similarity Formula

2.3 Model-based Approach for Collaborative Filtering [3]

Minh-Phung Thi Do et al. (2010) explained collaborative filtering as a technique where user's choices on items are predicted based on the choices of other users. To do this, we need a rating matrix (also known as a utility matrix), where rows represent the users, items represent the items (destinations in our case), and the intersection of these two represents the rating. Collaborative filtering is divided into a memory-based approach and a model-based approach.

	Item 1	Item 2	Item 3	Item 4
User 1	$r_{11} = 1$	$r_{12} = 2$	$r_{13} = 1$	$r_{14} = 5$
User 2	$r_{21} = 2$	$r_{22} = 1$	$r_{23} = 2$	$r_{24} = 4$
User 3	$r_{31} = 4$	$r_{32} = 1$	$r_{33} = 5$	$r_{34} = 5$
User 4	$r_{41} = 1$	$r_{42} = 2$	$r_{43} = ?$	$r_{44} = ?$

Figure 4: Structure of a Utility Matrix

A memory-based approach uses cosine similarity between the user vectors in order to predict their closeness. The closer the value of cosine similarity to zero, the more similar the two user vectors are.

A model-based approach uses machine learning models to predict user behaviour. In this paper, the models explained are clustering models, classification models, latent class models, Markov decision-based models, and matrix factorization-based models. In our research, we have used clustering and matrix factorization-based models.

2.4 Movies recommendation system using collaborative filtering and k-means [4]

Phongsavanh Phorasim et al. (2017) implemented K-means clustering in the movie recommendation system. The centroid, in this case, is considered the user rating vector, and updates to the centroid take place based on the users added to the cluster. The addition of the user to the cluster depends upon the Euclidean distance between the user and cluster centroid.

$$distance_{Euclidian}(u_1, u_2) = \sqrt{\sum_j (r_{1j} - r_{2j})^2}$$

Figure 5: Formulae for Euclidean Distance

2.5 Improved Ranking-Based Collaborative Filtering Using SVD and Borda Algorithm [5]

Muhammad Iqbal Ardiansyah et al. (2019) proposed a method based on single value decomposition (SVD). In SVD, the utility matrix is formulated as product of three matrices, i.e

$$R = U\Lambda V^T$$

Where R is the utility matrix, U is the user matrix, V is the item matrix, and Λ is a diagonal matrix where entries in the diagonal represent the eigenvalue of R. Both U and V are orthonormal matrices.

The formula is as follows to forecast the rating of a user U on an item i:

$$\hat{r}_{ui} = p_u . q_i^T + \mu + b_u + b_i$$

where $\hat{r}_{ui}$ the predicted p_u is the user vector, $p_u \in$ U, q_i^T is the transposed item vector where $q_i \in$ V, μ is the average rating of all items, b_u and b_i are biases used to minimise the prediction error.

2.6 Surprise Documentation [6]

Surprise is an easy-to-use Python library used for a recommendation system. It has the **prediction_algorithms** package, from which we can use in-built method for basic algorithms like Normal Predictor, Baseline, etc, KNN algorithms and Matrix factorization methods like SVD, SVDpp, NMF, etc. The **Dataset module** is used to convert the utility matrix into a dataset of schema (user_id, item_id, rating). The **accuracy** module has performance metrics like MAE, RMSE, etc. that can be used for performance analysis.

CHAPTER-3
IMPLEMENTATION OF EXISTING ALGORITHMS

3.1 Content-Based Filtering

Dataset: City.csv from Kaggle.

	City	Ratings	Ideal_dura	Best_time_	City_desc
1	City	Ratings	Ideal_dura	Best_time_	City_desc
2	Manali	4.5	02-Apr	October-Ju	[' One of the most popular hill stations in Himachal, Manali offers the most magnificent views of the Pir Panjal and the Dhauladhar ranges covered with snow for most parts of the year. ', '', " With the Covid-19 pa
3	Leh Ladakl	4.6	05-Jul	JulyOctobe	[" Ladakh is a union territory in the Kashmir region of India. Formerly falling in the state of Jammu & Kashmir, Ladakh was administered a union territory on 31st October 2019. Extending from the Siachen Glacier
4	Coorg	4.2	02-Mar	Septembe	[' Located amidst imposing mountains in Karnataka with a perpetually misty landscape, Coorg is a popular coffee producing hill station. It is popular for its beautiful green hills and the streams cutting right throug
5	Andaman	4.5	04-Jun	October-M	[' Replete with turquoise blue water beaches and a bit of history, Andaman & Nicobar Islands is a little slice of paradise tucked around 1,400 km away from the east coast of mainland India. Port Blair, the capital
6	Lakshadwe	4	04-Jun	Septembe	[" Formerly known as Laccadive Islands, Lakshadweep translates to 'one hundred thousand islands' in Malayalam. Home to a few of the most beautiful and exotic islands and beaches of India, Lakshadweep lies 4
7	Goa	4.5	03-Jul	November	[" Lying on the western coast, Goa is India's smallest state and unlike any other, known for its endless beaches, stellar nightlife, eclectic seafood, world-heritage listed architecture. Spread across just 3,702 km, Go
8	Udaipur	4.3	02-Mar	October-M	[' Udaipur, also known as the City of Lakes, is one of the most visited tourist places in Rajasthan. Located around stunning water lakes and enveloped by the Aravalli Hills in all directions, Udaipur is known for its a
9	Srinagar	4.5	03-May	April-Octo	[" Famously known as 'Heaven on Earth, Srinagar is located in the union territory of Jammu & Kashmir, on the banks of river Jhelum. Srinagar is known for the stationary houseboats and gondola-type rowboats- S
10	Gangtok	4.4	02-Apr		[' Incredibly alluring, pleasantly boisterous and wreathed in clouds - Gangtok, the capital of Sikkim, is one of the most popular hill stations in India. Lying at the height of 1650 m above sea level, the town during its
11	Munnar	4.5	02-Mar	Septembe	[' Popular among honeymooners, Munnar is a hill station in Kerala, located in the Idukki district. Lying in the Western Ghats at 1600 metres, it is one of the most sought after and visited travel destinations globally
12	Varkala	4.5	01-Feb		[" Varkala is a coastal town in the southern part of Kerala known for the unique 15m high 'Northern Cliff' adjacent to the Arabian Sea. It is popular for its hippie culture, shacks on the cliff serving great seafood an

Figure 6: City.csv dataset

Dataset preprocessing

Ratings, Ideal_duration, Best_time_to_visit columns are dropped from the dataset.

Implementation steps

- *TfidfVectorizer* method is used to create a matrix with columns as keywords in the document and rows would be the same as that of initial dataset.
- The initial challenge is numerical terms would also be considered as a keyword as they are in the form of string in the document.
- So, we had to remove these numerical terms using regular expression for alphabet characters as token pattern.
- The *fit_transform* method would be used to calculate the TF-IDF score of each keyword in each row of the dataset. A sample tf-idf matrix is shown in figure 7.

```
    zenith  zones  zoo   zorbing
0      0.0    0.0  0.0  0.092522
1      0.0    0.0  0.0  0.000000
2      0.0    0.0  0.0  0.000000
3      0.0    0.0  0.0  0.000000
4      0.0    0.0  0.0  0.000000
..     ...    ...  ...       ...
95     0.0    0.0  0.0  0.000000
96     0.0    0.0  0.0  0.000000
97     0.0    0.0  0.0  0.000000
98     0.0    0.0  0.0  0.000000
99     0.0    0.0  0.0  0.000000
```

-

Figure 7: TF-IDF Matrix sample

- Cosine similarity is calculated which would give us the similarity score of the cities.
- *recommend_destination* method would return top 5 cities based on the input city. The sorting in this method would be based on the similarity score of the input city with the remaining cities of dataset in the descending order.
- Higher the similarity score, more is the similarity between the cities.

3.2 Collaborative-Filtering based Algorithm

Dataset Creation:

- A utility matrix is created by generating the ratings using custom algorithm.
- Rows consist of users, where user_id is of format 'user_{index_of_row-1}'. The number of users is considered at 5000.
- Column consists of the cities which are retrieved from the **'City.csv'** file.
- Intersection of rows and columns is the rating given by user in that row to the City in that column.
- All the ratings are initialised to zero.
- Integer values in the range of [1,5] are assigned randomly based upon the spare density.
- For eg: If sparse density is 0.1, which means only 10% of in the utility matrix is non zero.
- After matrix generation, it is stored in .csv file. Naming convention of file is given by **'Utility_Matix_{no of users}_{sparse density}.csv'**.
- Sparse density is increased by 0.1 and the above steps are repeated till the sparse density value is 1.

3.2.1 Memory based Algorithm

- For user based collaborative filtering, cosine similarity score is calculated between the users. The more closure the similarity score to zero, the more similar the users are.
- Similarly for item based collaborative filtering, cosine similarity score is calculated between the items (Cities in our case).

3.2.2 Model Based Algorithm

SVD

- After reading utility matrix, the dataframe is converted into **Surprise Dataset (**defined in the **Surprise** Library of python).

	user	location	ratings
	user_4990	Poovar	2.0
	user_4991	Poovar	5.0
	user_4992	Poovar	4.0
	user_4993	Poovar	1.0
	user_4994	Poovar	2.0
	user_4995	Poovar	5.0
	user_4996	Poovar	3.0
	user_4997	Poovar	5.0
	user_4998	Poovar	3.0
	user_4999	Poovar	1.0

Figure 8: Structure of a Surprise Dataset

- The dataset is divided into a trainset and a testset (80:20).
- After training the data, predictions were made on the test set, and these predictions were used to calculate performance metrics like RMSE and MAE.
- Finally, 5-fold cross-validation is performed to generate avg rsme and avg mae
- The above process is repeated for all the files that were generated in the dataset creation step.
- The performance of SVD model with different sparse densities is compared by plotting avg rsme and avg mae against the sparse densities.

Other models like NMF (similar to SVD with non-negative predictions), Normal Predictor, BaseLine, KNN, etc. are trained, and performance comparison is done with SVD.

CHAPTER-4
IMPLEMENTATION OF HYBRID ALGORITHM

Dataset used: ‘City.csv’, ‘Utility_Matrix_5000_0.3.csv’

A hybrid algorithm is implemented by combining the content-based and collaborative-based filtering algorithms. It leverages the strengths of both approaches in order to enhance the performance and relevance of travel recommendations. Currently, we have implemented a draft version of the hybrid algorithm, and there’s still scope for improvement in performance metrics and relevant recommendations.

Steps:

- Initially, we’ll be implementing a content-based algorithm on the **‘City.csv’** dataset. This would give us a set of cities based on the similarity of TF-IDF scores.
- 20 cities are retrieved, including the city that is given as input to the algorithm.
- Before implementing collaborative filtering, only the columns with the same column heading as those of 20 cities retrieved in previous step were fetched from **‘Utility_Matrix_5000_0.3.csv’** dataset.
- Some rows were also eliminated based on the logic that the users who have liked the input city should be in the final data frame, i.e., rating > 3 given by the users for the input city.
- So collaborative filtering would be applied on a matrix (dataframe) whose dimensions are m x n, where m (no of users) < 5000 and n (no of cities) = 20

Users	Jammu	Vaishno Devi	Amarnath	Chandigarh	Almora	Gulmarg	Srinagar	Ajanta and Ellora Caves	Tirupati	Auli	Bhubaneswar	Leh Ladakh	Kalimpong	J
user_8	4.0	1.0	0.0	0.0	4.0	1.0	0.0	3.0	0.0	5.0	4.0	3.0	1.0	
user_15	4.0	0.0	0.0	2.0	4.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	2.0	
user_20	3.0	0.0	5.0	0.0	0.0	1.0	0.0	0.0	2.0	0.0	0.0	0.0	4.0	
user_43	4.0	3.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
user_46	3.0	0.0	0.0	5.0	2.0	0.0	0.0	3.0	1.0	0.0	0.0	1.0	0.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
user_4987	4.0	0.0	0.0	0.0	0.0	0.0	3.0	0.0	2.0	2.0	0.0	4.0	0.0	
user_4989	5.0	0.0	0.0	0.0	5.0	0.0	0.0	3.0	0.0	0.0	0.0	5.0	0.0	
user_4992	5.0	0.0	0.0	2.0	2.0	0.0	0.0	4.0	1.0	0.0	5.0	0.0	0.0	
user_4993	3.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	0.0	1.0	
user_4999	4.0	5.0	2.0	5.0	0.0	0.0	0.0	5.0	2.0	0.0	3.0	0.0	0.0	

823 rows × 20 columns

Figure 8: Structure of a Final DataFrame (input City = Jammu)

- SVD model is preferred for training the data. The process of splitting the data, converting to surprise dataset, cross-validation, and performance metrics calculation remains the same as that explained under **“SVD”** section.
- To get the recommendations, user_id is taken as input.
- Cities that are not visited by the user are recommended and are sorted based on the ratings predicted by the model.

RESULTS

5.1 Results of Content-Based Filtering.

	Prefered Place	Recommendations
0	Jammu	[Vaishno Devi, Amarnath, Chandigarh, Almora, Gulmarg]
1	Puri	[Digha, Mathura, Alibaug, Bhubaneswar, Lavasa]
2	Manali	[Kasol, Gulmarg, Shimla, Nainital, Jaisalmer]
3	Jodhpur	[Jaipur, Bikaner, Udaipur, Mathura, Kolkata]
4	Jaipur	[Agra, Kolkata, Delhi, Gwalior, Jodhpur]
5	Kolkata	[Jaipur, Bangalore, Delhi, Mumbai, Hyderabad]
6	Delhi	[Jaipur, Bhubaneswar, Agra, Kolkata, Chandigarh]
7	Mumbai	[Kolkata, Mahabaleshwar, Lavasa, Lonavala, Gwalior]
8	Lonavala	[Khandala, Mahabaleshwar, Pune, Matheran, Lavasa]
9	Pune	[Ahmedabad, Alibaug, Lavasa, Lonavala, Mahabaleshwar]
10	Bangalore	[Hogenakkal, Kolkata, Chandigarh, Jaipur, Udaipur]
11	Gwalior	[Jaipur, Bhubaneswar, Kolkata, Delhi, Ahmedabad]
12	chennai	[Alibaug, Pune, Kanyakumari, Kovalam, Madurai]

5.2 Comparison of the performance of SVD with different sparse density of datasets.

Density of data in percentage	Avg. RMSE	Avg. MAE
10.0	1.225714	1.100604
20.0	1.416701	1.201519
30.0	1.585744	1.310095
40.0	1.722080	1.470018
50.0	1.807302	1.599886
60.0	1.839857	1.634412
70.0	1.819014	1.601559
80.0	1.749215	1.536891
90.0	1.620198	1.403917
100.0	1.419017	1.218617

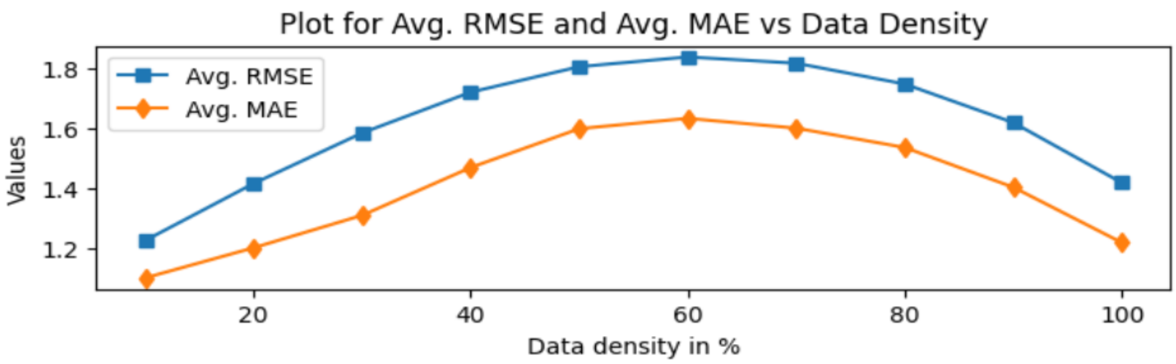

Figure 9: Performance metrics Comparison

5.3 Comparison of performance of SVD with other models.

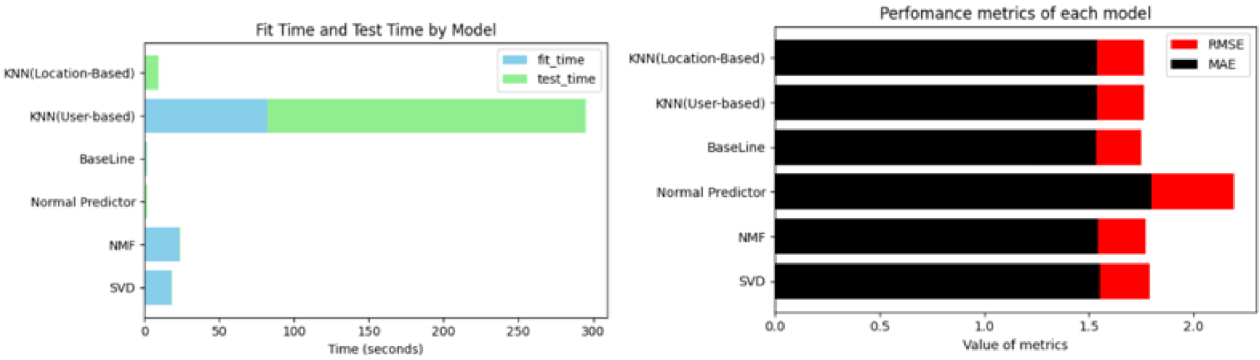

Figure 10: Performance metrics Comparison of different models

5.4 Impact of number of users on SVD model.

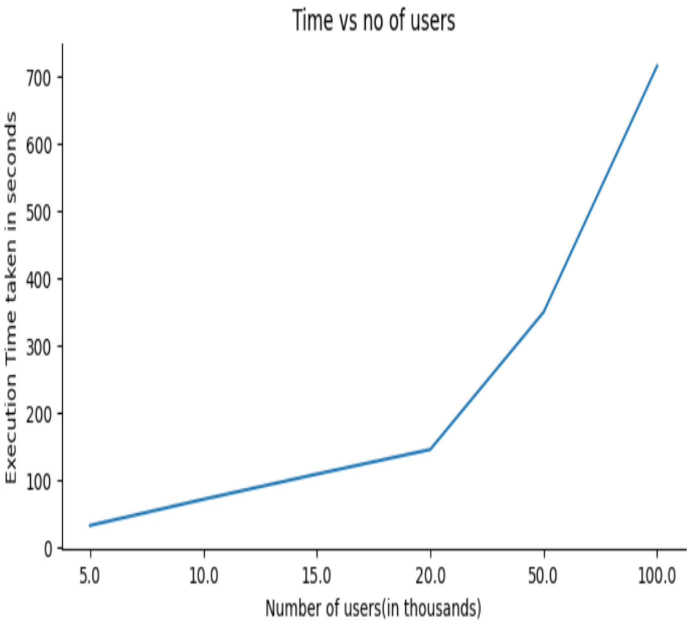

Figure 11: Increase in number of users increases the training time

No of Users(in thousand)	RMSE	MAE
5.0	1.753108	1.541626
10.0	1.745633	1.534011
15.0	1.746292	1.535300
20.0	1.747603	1.537025
50.0	1.745872	1.536578
100.0	1.745449	1.536092

Figure 12: Increase in number of users has no significant impact on performance metrices

5.5 Results of Hybrid approach

Input location: Jammu, Input User id = user_20
RMSE: 1.7096
MAE: 1.4868

Recommendations based on users who liked Jammu
Amarnath
Tirupati
Dalhousie
Srinagar
Chandigarh

Input location: Chennai, Input user = user_23
RMSE: 1.7062
MAE: 1.4840

Recommendations based on users who liked Chennai
Thanjavur
Madurai
Ahmedabad
Lucknow
Kovalam

Input Location: Tirupati

Input user = user_31
RMSE: 1.7231
MAE: 1.5009

Recommendations based on users who liked Tirupati
Kasauli
Visakhapatnam
Rameshwaram
Haridwar
Amritsar

Input user = user_32
RMSE: 1.7176
MAE: 1.4951

Recommendations based on users who liked Tirupati
Vrindavan
Kasauli
Udaipur
Haridwar

CHAPTER-6
CONCLUSION AND FUTURE WORK

We have presented a comprehensive study and the development of a hybrid travel recommendation algorithm in this research paper. The purpose of this algorithm is to improve the personalisation and effectiveness of travel recommendations. The proposed algorithm takes advantage of the strengths of multiple recommendation techniques to provide a comprehensive and user-centric approach to travel planning and exploration. This is accomplished by integrating collaborative filtering, content-based filtering, and contextual information.

The scope of future work is open to all three algorithms. For content-based, we can implement reinforcement learning in order to add the cities that are not present in the 'City.csv' dataset. For collaborative filtering, models other than SVD need to be explored, which may give us better performance. This can also include deep learning models. Hybrid implementation is still a work in progress in our current research. The scope of work in hybrid implementation would be to focus on more relevant outputs.

REFERENCES

[1]. A systematic review and research perspective on recommender systems (Luca Guarnera et al.)

[2]. Netflix Recommendation System based on TF-IDF and Cosine Similarity Algorithms (Mohamed Chiny et al., (2022)).

[3]. Model-based Approach for Collaborative Filtering (Minh-Phung Thi Do et al. (2010)).

[4]. Movies recommendation system using collaborative filtering and k-means (Phongsavanh Phorasim et al. (2017))

[5]. Improved Ranking Based Collaborative Filtering Using SVD and Borda Algorithm Muhammad Iqbal Ardiansyah et al. (2019).

[6]. Surprise Documentation (https://surprise.readthedocs.io/en/stable/)