ENHANCED TRAVEL RECOMMENDATION SYSTEM USING HYBRID FILTERING

A MINIPROJECT REPORT

Submitted by

DEVIKA C (221801009)

KEERTHIKA P(221801027)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY IN ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

RAJALAKSHMI ENGINEERING COLLEGE
DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

ANNA UNIVERSITY, CHENNAI

Nov 2024

ANNA UNIVERSITY, CHENNAI

BONAFIDE CERTIFICATE

Certified that this Report titled "**ENHANCED TRAVEL RECOMMENDATION SYSTEM USING HYBRID FILTERING** "is the bonafide work **DEVIKA C (221801009) , KEERTHIKA P (221801027)** who carried out the work under my supervision .

SIGNATURE

Dr. J. M. GNANASEKAR, M.E., Ph.D.,
Professor and Head,
Department of Artificial Intelligence and Data Science,
Rajalakshmi Engineering College,
Chennai – 602 105.

SIGNATURE

Dr. J. M. GNANASEKAR, M.E., Ph.D.,
Professor and Head,
Department of Artificial Intelligence and Data Science,
Rajalakshmi Engineering College,
Chennai – 602 105.

Submitted for the project viva-voce examination held on______________________.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman Mr. S. MEGANATHAN, B.E, F.I.E., our Vice Chairman Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S., and our respected Chairperson Dr. (Mrs.) THANGAM MEGANATHAN, Ph.D., for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution.

Our sincere thanks to Dr. S.N. MURUGESAN, M.E., Ph.D., our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to Dr. J. M. GNANASEKAR., M.E., Ph.D., Head of the Department, Professor and Head of the Department of Artificial Intelligence and Data Science for his guidance and encouragement throughout the project work. We are glad to express our sincere thanks to our supervisor Dr. J. M. GNANASEKAR., M.E., Ph.D., Head of the Department, Department of Artificial Intelligence and Data Science, Rajalakshmi Engineering College for his valuable guidance throughout the course of the project.

Finally we express our thanks for all teaching, non-teaching, faculty and our parents for helping us with the necessary guidance during the time of our project.

ABSTRACT

The travel industry has seen a transformative shift with digital platforms and the growing demand for personalized recommendations. Traditional systems, like collaborative filtering and content-based filtering, face challenges such as data sparsity and the cold-start problem, limiting their ability to offer accurate and diverse travel suggestions. Collaborative filtering struggles with insufficient user interaction data, while content-based filtering often lacks novelty and fails to account for nuanced user preferences.

To address these limitations, this project proposes a hybrid travel recommendation system that combines the strengths of both methods. Collaborative filtering predicts user preferences based on the behavior of similar users, while content-based filtering utilizes destination attributes like location, activities, and climate to tailor suggestions. The hybrid system mitigates the cold-start problem by generating initial recommendations using content-based filtering and improves over time as more user interaction data becomes available for collaborative filtering.

By balancing personalization with novelty, the hybrid system ensures relevant and engaging recommendations, encouraging users to explore new destinations while satisfying their preferences. This innovative approach enhances user satisfaction, improves booking conversions, and offers a more dynamic travel planning experience. The insights from this project also contribute to the broader field of recommendation systems, demonstrating applicability in domains such as e-commerce and entertainment. This study aims to redefine personalized travel planning, fostering deeper user engagement and exploration.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER -I

INTRODUCTION

1.1 GENERAL

In recent years, the research of recommender system has developed rapidly. Various recommendation systems are also widely used in e-commerce, social networking sites, e-tourism, Internet advertising, and many other fields, and these recommendation systems show superior effects and prospects . With the rise of more and more online travel websites (such as Expedia, Travelzoo, tuniu), more and more online data can describe users' interests and preferences

This makes tourism product recommendation become one of the hotspots of recommendation system research. Providing personalized recommendations is essential in the travel industry to improve user engagement and satisfaction on digital platforms in today's digital world. Common challenges like data sparsity, cold start problems, and a lack of personalized suggestions limit traditional recommendation systems like collaborative filtering (CF) and content-based filtering (CBF). Often, these constraints lead to general suggestions that do not cater to users' individual travel requirements, diminishing the overall efficiency of these platforms.With the increasing complexity and competitiveness of the travel industry, there is a higher demand for more precise, and tailored travel suggestions.

A recommendation system that merges the benefits of collaborative and content-based filtering methods provides a hopeful answer to these challenges. Combining CF's user interaction leveraging and CBF's item feature analysis can address data sparsity and cold start issues, offering more detailed and personalized recommendations called as hybrid filtering approch..

Moreover, hybrid systems have the capability to make adjustments in real-time according to recent user input, enabling a more flexible and interactive experience. In the fast-moving and cutthroat travel sector, giving personalized travel suggestions in real-time can greatly enhance user retention, boost engagement, and nurture customer loyalty, ultimately leading to higher conversion rates and overall business success

1.2NEED FOR THE STUDY

The rise in competition within the travel sector and the greater need for customized user experiences have exposed the shortcomings of conventional recommendation systems. Consumers are looking for more personalized recommendations, particularly when it comes to planning trips. However, current systems that rely solely on collaborative filtering or content-based filtering often fall short of meeting these expectations. Traditional methods face challenges like data scarcity, difficulties with new items, and lack of personalization, resulting in recommendations that are not tailored to individual user preferences or travel requirements.

Because travel preferences vary greatly, including where to go, how much to spend, and what to do, there is a high demand for a sophisticated recommendation system that can handle these intricacies. Blending collaborative and content-based filtering can improve recommendation accuracy and relevance by utilizing various data sources and techniques. This approach offers improved answers for issues like inexperienced users travel spots, despite limited data in conventional systems.

Furthermore, in a data-driven travel industry, providing real-time, dynamic, and customized travel suggestions has become crucial. An adaptive recommendation system that combines different methods can significantly boost user interaction, improve customer contentment, and elevate conversion rates. The approach Hybrid filtering provides good high-level user recommendtion by met the user needs and the booking history.

The study is motivated by the necessity for travel suggestions that are more precise and tailored to the specific context, in order to meet changing user needs and industry developments.

1.3 OVERVIEW OF THE PROJECT

The Travel Recommendation System is designed to simplify the process of finding travel destinations by offering personalized suggestions based on users' preferences and interactions. With the rise of travel platforms and an overwhelming amount of information online, it can be challenging for users to identify places that match their tastes. This system addresses that issue by utilizing a hybrid recommendation approach, combining both collaborative and content-based filtering. By leveraging data on user behaviours and destination features, the system can generate tailored suggestions, helping users discover new and relevant travel destinations. The ultimate goal is to enhance the user experience and increase engagement by offering recommendations that resonate with individual preferences.

Collaborative filtering is at the heart of the system, focusing on user interactions to make recommendations. It uses patterns from a large number of users to suggest destinations based on how similar users have rated or interacted with them. For example, if a user has shown interest in beach destinations, the system identifies other users with similar preferences and recommends destinations they have explored, even if the first user hasn't directly interacted with them. Techniques such as Matrix Factorization (e.g., Singular Value Decomposition) are used to factorize user-item interaction matrices, uncovering hidden patterns in user preferences. This allows the system to predict which places a user might enjoy, even if they haven’t explicitly expressed interest in them before.

To enhance the accuracy of the recommendations, the system also implements content-based filtering. This method analyzse the attributes of each travel destination, such as geographical features, amenities, or types of activities, and compares them to places a user has shown interest in. For instance, if a user enjoys cultural landmarks or hiking trails, the system looks for similar destinations by assessing their features. One of the key techniques used here is TF-IDF (Term Frequency-Inverse Document Frequency), which highlights important keywords from the descriptions of destinations. By calculating how often these words appear in the description relative to how frequently they appear in other destination descriptions, the system can recommend places that align with the user's travel style and interests.

The combination of collaborative and content-based filtering addresses several limitations that traditional recommendation systems face. One major challenge is the cold start problem, where limited data on new users or places makes it difficult to generate recommendations.

The hybrid filtering system ultimately creates a robust travel recommendation platform that provides more personalized, accurate, and diverse suggestions to users. It does this by analyzing both user behaviour (through collaborative filtering) and destination content (through content-based filtering). This ensures users discover destinations that truly align with their preferences and travel styles. By making the experience more engaging, travel platforms can benefit from improved user satisfaction, retention, and higher booking rates. Whether it's recommending hidden gems in unexplored regions or well-known landmarks that match a user's specific criteria, this hybrid system offers users a complete and enriched travel planning experience.

1.4 OBJECTIVES OF THE STUDY

Enhancing Personalization in Travel Recommendations

The primary objective of this study is to design and implement a personalized travel recommendation system tailored to individual user preferences. In an era where personalization drives user engagement, this system employs a hybrid filtering approach by combining collaborative and content-based methods. The aim is to deliver recommendations that are not only relevant but also diverse, significantly enhancing the user experience. By analyzing historical data and understanding user preferences such as preferred destinations, activities, or accommodations, the system suggests travel options aligned with user interests. This approach explores the impact of personalized recommendations on user satisfaction, travel decision-making, and conversion rates for booking platforms.

Mitigating the Cold Start Problem and Data Sparsity Issues

Another key focus of this study is addressing challenges commonly encountered in recommendation systems: the cold start problem and data sparsity. Collaborative filtering methods often struggle when insufficient interaction data exists, as with new users or poorly reviewed destinations. Meanwhile, content-based filtering, while effective for individual preferences, might restrict users from discovering novel or unexpected destinations. By integrating collaborative and content-based filtering into a hybrid model, this research seeks to mitigate these issues, ensuring accurate recommendations for all users, even in sparse datasets. The study evaluates how the hybrid approach balances user exploration of new options while exploiting existing preferences, enhancing system performance and user satisfaction.

Evaluating the Effectiveness of Hybrid Filtering Models

This study will assess the hybrid filtering model's performance in generating travel recommendations, comparing its outcomes with traditional collaborative and content-based approaches. Metrics such as precision, recall, and Root Mean Square Error (RMSE) will be used to measure the system's accuracy, relevance, and diversity. Real-world datasets and user feedback will provide practical insights into the benefits of combining these techniques. Beyond technical evaluation, this research will analyze the hybrid model's influence on user satisfaction and engagement, illustrating how a well-designed system can outperform conventional approaches in meeting diverse user needs.

Improving User Engagement and Platform Retention

Enhanced travel recommendations aim to boost user engagement and platform retention, which are critical for the success of travel platforms. Personalized suggestions increase the likelihood of users exploring new destinations, engaging deeply with the platform, and returning for future recommendations. This study examines how tailored recommendations affect user behavior, such as repeat visits, travel bookings, and system interactions. It will also investigate the role of trust in recommendations, exploring whether aligning suggestions with user preferences fosters loyalty and confidence in the platform.

Contributing to the Field of Recommendation Systems in the Travel Industry

This study contributes to the growing field of recommendation systems, focusing on their application in the travel industry. The research addresses unique challenges in this domain, such as dynamic travel preferences, diverse user profiles, and an expansive range of destinations and activities. By presenting a comprehensive framework for hybrid filtering, this work offers valuable insights for researchers and developers aiming to enhance recommendation systems across industries. Findings will be disseminated through academic publications, conferences, and industry reports, fostering advancements in artificial intelligence-driven personalization and its application in travel technology.

CHAPTER II

REVIEW OF LITERATURE

2.1 INTRODUCTION

1. Background and Context

The travel industry has been significantly transformed by digital platforms, driven by the increasing demand for personalized travel experiences. Modern travelers expect tailored recommendations rather than generic suggestions, making recommendation systems essential in this domain. Companies like Airbnb, TripAdvisor, and Expedia leverage artificial intelligence and big data to enhance user experiences. Despite their advancements, these systems face challenges such as data sparsity and the cold-start problem.

Collaborative filtering, a common technique in recommendation systems, relies on user-item interactions, which may be incomplete or unavailable for new users or destinations. Content-based filtering, while useful, often fails to capture the nuanced preferences of users. Thus, there is a pressing need for hybrid approaches that combine the strengths of both methods.

This project introduces a hybrid travel recommendation system that integrates collaborative filtering with content-based filtering. This approach aims to mitigate data sparsity and cold-start issues while delivering personalized and diverse travel suggestions. By considering both user preferences and destination attributes, the system ensures relevant recommendations, even for new users or unexplored destinations.

2.Problem Statement

The travel industry faces challenges in providing accurate and personalized recommendations due to the complexities of user preferences and limited interaction data. Collaborative filtering, though effective, struggles with new users and data sparsity, while content-based filtering often limits exploration to familiar options.

This project addresses the limitations of both methods by proposing a hybrid filtering model. This approach combines the collaborative filtering's ability to recommend based on similar users with content-based filtering's focus on item attributes. The hybrid model aims to deliver accurate suggestions that align with user preferences while enabling the discovery of novel destinations.

3. Motivation for the Study

The motivation for this project stems from the growing need for personalized and accurate travel recommendations. With an abundance of choices available, providing curated suggestions is crucial for user engagement and satisfaction. The cold-start problem and data sparsity remain significant obstacles, particularly for new users or less popular destinations.

A hybrid filtering approach offers a promising solution by balancing accuracy and diversity in recommendations. This system ensures users receive relevant suggestions while also encouraging exploration of new experiences. By integrating collaborative and content-based methods, the proposed system aims to deliver a dynamic and personalized recommendation experience.

4. Proposed Solution: Hybrid Filtering for Travel Recommendations

The proposed solution involves developing a hybrid filtering system that combines collaborative filtering with content-based filtering. Collaborative filtering predicts preferences based on similar users, while content-based filtering analyzes destination attributes like location, activities, and climate.

The hybrid system merges these methods to address their individual limitations. Collaborative filtering enables novel recommendations, while content-based filtering ensures alignment with user preferences. This approach also tackles the cold-start problem by leveraging content-based filtering to provide meaningful suggestions for new users. Over time, as user interaction data increases, the system refines its recommendations, enhancing personalization.

5. Significance and Impact of the Study

The implementation of a hybrid filtering system has the potential to transform travel planning by improving recommendation accuracy and personalization. Addressing challenges like the cold-start problem can enhance user satisfaction and engagement, leading to higher booking conversion rates.

The study contributes to the field of recommendation systems by demonstrating the effectiveness of hybrid filtering in complex domains like travel. The insights gained can be applied to other domains, such as e-commerce and entertainment, where balancing personalization with novelty is critical for maintaining user interest.

2.2 FRAMEWORK OF LCA

The Life Cycle Assessment (LCA) framework for the hybrid travel recommendation system evaluates its environmental, computational, and user impact throughout its lifecycle, from development to deployment. This framework ensures that the system aligns with sustainable practices while optimizing resource usage and enhancing user experience. The framework follows four main phases:

1. Goal and Scope Definition

- **Objective:** To assess the environmental and computational efficiency of the hybrid travel recommendation system while enhancing its scalability and user satisfaction.
- **Scope:** Evaluate the system's lifecycle from data collection, algorithm training, and deployment to user interaction. The analysis includes server resources, energy consumption, and user-facing operations.
- **Functional Unit:** A single recommendation generated for a user, considering computational and environmental factors.

2. Inventory Analysis (LCI)

- **Data Collection:** Identify resources required for training and maintaining the system, such as hardware, cloud computing, and storage. Quantify energy usage during model training, API requests, and frontend rendering.
- **Processes Included:**
  - Data preprocessing and storage.
  - Algorithm training and model updates.
  - User query handling and recommendation generation.
- **Data Sources:** Energy usage metrics from servers, cloud computing reports, and system logs.

3. Impact Assessment (LCIA)

- **Impact Categories:**
  - **Energy Consumption:** Assess the power usage of servers during model training and query processing.

- **Carbon Footprint:** Evaluate emissions associated with data centers and server maintenance.
- **Resource Efficiency:** Examine optimization strategies for data storage and computational processes.

- **Characterization:** Use standard metrics (e.g., CO_2 equivalents for emissions) to quantify impacts.

4. Interpretation

- **Results Evaluation:** Analyze the trade-offs between system performance, energy efficiency, and environmental impact.
- **Recommendations:** Optimize algorithms to reduce computational overhead, incorporate green cloud solutions, and minimize data redundancy.
- **Documentation:** Clearly outline assumptions, limitations, and areas for improvement to ensure sustainable scalability.

CHAPTER III

SYSTEM OVERVIEW

3.1 EXISTING SYSTEM

1. Cold Start Problem

Applies to: Collaborative Filtering (CF) and Content-Based Filtering (CB)

- The cold start problem arises when there is insufficient data about new users or items. For example, CF struggles to recommend destinations for new users with no interaction history, while CB may fail when new destinations lack descriptive features.
- This limitation affects the system's ability to generate meaningful recommendations, particularly during the early stages of a platform or for users with minimal activity.

2. Data Sparsity

Applies to: Collaborative Filtering (CF)

- Large user-item matrices are often sparse, with many users interacting with only a small subset of items. This sparsity limits CF's ability to identify patterns and provide accurate recommendations.
- For instance, in a travel recommendation system, if users have rated only a handful of destinations, the algorithm may fail to identify relevant similarities between users or items.

3. Popularity Bias

Applies to: Collaborative Filtering (CF) and Demographic Filtering (DE)

- Both CF and DE tend to favor popular destinations, repeatedly recommending them at the expense of niche or unique options.
- This bias reduces the diversity of recommendations, potentially leading to a monotonous user experience and limiting users from discovering less mainstream travel options.

4. Limited Discovery

Applies to: Content-Based Filtering (CB) and Social Filtering (SF)

- CB focuses heavily on recommending items similar to those a user has already interacted with, resulting in a narrow recommendation scope. Similarly, SF may restrict suggestions to destinations popular within a user's social circle.
- These limitations prevent users from exploring diverse or unexpected travel options, reducing the overall satisfaction of the recommendations.

5. Overgeneralization

Applies to: Demographic Filtering (DE) and Utility-Based Filtering (UB)

- DE assumes that individuals within the same demographic group have similar preferences, leading to overly broad and impersonal recommendations. Similarly, UB emphasizes practical attributes (e.g., cost, amenities) without considering personal tastes or experiential factors.

- As a result, these methods often overlook individual uniqueness and fail to capture nuanced preferences, resulting in suboptimal recommendations.

3.2 PROPOSED SYSTEM

The proposed system aims to provide a robust and highly effective travel recommendation platform by implementing a **hybrid filtering approach** that combines the strengths of **collaborative filtering (CF)** and **content-based filtering (CBF)**. This system addresses the primary challenges that traditional recommendation systems often face, such as the **cold start problem**, **data sparsity**, and **lack of personalization**. By leveraging both user interaction data and item attributes, the system enhances the recommendation accuracy and ensures a personalized user experience.

Objectives of the Proposed System

The primary goal of this system is to enhance user experience by delivering **highly personalized travel recommendations**. The system is designed to not only predict user preferences but also introduce users to novel travel destinations and experiences that they might not have discovered on their own. In this way, it combines the ability to learn from user behavior with a deeper understanding of the inherent qualities of travel destinations .

Another core objective is to provide a scalable and efficient recommendation engine that can handle a large volume of data. Given the diversity of user preferences and the wide range of travel options available.

Key Components of the Proposed System

The system is built around two core algorithms: collaborative filtering and content-based filtering, which together form the backbone of the hybrid recommendation system.

1. **Collaborative Filtering (CF)**: The collaborative filtering approach works by analyzing user behavior and preferences. It identifies patterns based on user interactions such as past bookings, ratings, and likes, comparing them with the behavior of other similar users. For example, if User A and User B both liked several similar travel destinations, the system might recommend a destination that User A enjoyed to User B. This technique excels at providing recommendations that take into account **user similarity** and **community behavior**, making it particularly useful when there is a wealth of user interaction data.

2. **Content-Based Filtering (CBF)**: In content-based filtering, the system recommends items based on the attributes of the items a user has previously engaged with. For travel recommendations, this could include features such as **location type (beach, mountain, urban, etc.)**, **cost range**, **activities offered**, and **seasonal availability**. By analyzing these features, the system builds a profile of each user based on their past preferences. When new destinations or travel options are available, the system can match their attributes with the user's profile and make relevant recommendations.

3. **Hybrid Filtering**: The proposed system integrates the collaborative and content-based approaches into a **hybrid model**. This hybridization allows the system to take advantage of the strengths of both methods while mitigating their individual weaknesses. For example, if a user is new to the system and has little to no interaction history, the content-based filtering can still provide relevant recommendations by analyzing the attributes of destinations the user has shown interest in. On the other hand, as the system collects more data on the user's interactions, collaborative filtering can start providing recommendations based on the behavior of similar users.

By balancing these approaches, the hybrid system ensures that users always receive high-quality recommendations, even when data is sparse.

Cold Start Problem and its Solution

A major challenge in recommendation systems, especially for new users or items, is the **cold start problem**. This occurs when the system lacks sufficient data to make reliable recommendations. The cold start problem typically affects collaborative filtering the most, as it relies heavily on user interaction data.

The proposed system tackles this issue by integrating content-based filtering as a backup mechanism. When user interaction data is insufficient, the system will still be able to generate recommendations based on item attributes, such as a user's preference for beach destinations or adventure sports.

User Profile Building and Continuous Learning

To provide truly personalized recommendations, the system must continuously learn and update user profiles based on new data. Every interaction a user has with the platform—whether it's browsing a travel listing, liking a destination, or booking a trip—provides valuable feedback to the recommendation engine. The system uses this data to update the user's profile in real time.

The hybrid recommendation system is designed to learn from both **explicit feedback** (e.g., user ratings, bookings) and **implicit feedback** (e.g., browsing behavior, time spent on certain pages). By aggregating this feedback, the system can make increasingly accurate predictions about what each user is likely to prefer.

Enhancing Diversity in Recommendations

One of the limitations of traditional recommendation systems, especially those based solely on collaborative filtering, is that they can lead to **recommendation redundancy**. Users may be repeatedly shown the same or very similar items, leading to a narrow experience that doesn't expose them to new or diverse travel options.

To overcome this, the proposed hybrid system includes mechanisms to introduce **diversity** in recommendations. For instance, the system might occasionally suggest travel destinations or experiences that are outside a user's typical preferences but have been highly rated by other users with similar

System Scalability and Efficiency

Given the large amounts of data that travel platforms typically handle—thousands of users, numerous destinations, and constantly changing travel information—the proposed system is designed to be **scalable**. The use of efficient algorithms, such as **matrix factorization** in collaborative filtering and **TF-IDF** for content-based filtering, ensures that the system can handle large datasets without a significant drop in performance.

Personalization and User Experience

A key aspect of the proposed system is its ability to create a **personalized user experience**. Personalization is critical in the travel industry, where preferences can vary significantly between users. Some users may prefer adventure travel, while others may prioritize luxury experiences. The system takes these preferences into account, offering a tailored experience that feels unique to each user.

In addition to making personalized recommendations, the system allows users to **manually adjust** their preferences by selecting specific criteria they want to prioritize, such as budget, destination type, or travel duration. This level of control enhances the user experience and ensures that recommendations align closely with individual preferences.

3.3 FEASIBILITY STUDY

1. Technical Feasibility

Technology Stack

The proposed system will employ a combination of collaborative filtering (CF) and content-based filtering (CB) techniques, leveraging machine learning models to deliver personalized travel recommendations. To implement the hybrid model, the system will use technologies such as:

Data Management: MongoDB to store user profiles, historical interactions, destination attributes, and user-generated content.

Recommendation Engine: Python-based libraries like scikit-learn and Surprise for collaborative filtering, and TF-IDF models for content-based filtering.

Front-end: React.js for building a dynamic user interface where users can interact with the recommendation system.

Back-end: Node.js to handle user requests, process recommendation algorithms, and provide relevant suggestions.

2. Operational Feasibility

User Experience

Hybrid filtering offers a superior user experience by providing diverse and relevant travel recommendations. The collaborative aspect leverages data from similar users, suggesting destinations they might not have considered. Meanwhile, content-based filtering ensures the recommendations align with users' personal preferences (e.g., adventure seekers are recommended places with outdoor activities). The hybrid approach balances discovery and relevance, making the system more engaging.

User Acceptance

User acceptance is highly likely, given the personalized travel recommendations and improved relevance due to hybrid filtering. Travelers tend to favor systems that understand their preferences and offer unique suggestions, which this system can provide through the combination of CF and CB. Additionally, users can explore popular destinations while discovering niche places tailored to their tastes.

4. Economic Feasibility

Cost of Implementation

The primary costs associated with the project include:

Infrastructure: Cloud-based services for storage, computation, and real-time recommendations (e.g., AWS EC2, RDS, or similar services).

Development: The salaries of developers specializing in machine learning, data science, front-end, and back-end development.

Maintenance: Ongoing maintenance and updates to ensure model accuracy, optimize the algorithm, and handle new user data.

Market Demand

There is a growing demand for personalized travel experiences. Travelers increasingly rely on AI-powered recommendation systems for decision-making, especially given the overload of information on available destinations. The hybrid system will meet this market need by providing personalized, efficient, and accurate travel recommendations, making it economically feasible.

4. Legal and Ethical Feasibility

Data Privacy

One of the critical considerations in this project is ensuring user data privacy and compliance with regulations such as the GDPR (General Data Protection Regulation). The system will need to implement strong encryption and data protection measures, ensuring user data (including travel preferences and personal information) is securely stored and processed.

Transparency

The system should provide transparent explanations of how recommendations are generated, particularly as users might want to know why specific destinations are being recommended. This can be achieved by offering users insights into the recommendation process, such as "Users like you visited [destination]."

CHAPTER – IV

SYSTEM REQUIREMENTS

4.1 HARDWARE REQUIREMENT

To effectively implement the proposed travel recommendation system, it is essential to have hardware that supports high-speed data processing, storage, and real-time responsiveness. Below are the key hardware components required for this system:

1. Server/Hosting Infrastructure

- **CPU**: A multi-core processor, such as an **Intel Xeon** or **AMD EPYC** with at least 8 cores, is recommended to handle the computational load of machine learning algorithms and real-time data processing.
- **RAM**: A minimum of **32 GB DDR4 RAM** is recommended for smooth handling of large datasets and to support efficient caching mechanisms. More RAM (64 GB or higher) may be required for high-traffic environments or larger datasets.
- **Networking**: High-speed Ethernet connection (minimum 1 Gbps) to ensure fast data transfer between servers and users, ensuring that real-time recommendations are delivered without delay.

2. GPU (Graphical Processing Unit)

- For machine learning tasks, especially training recommendation models like collaborative filtering and content-based filtering algorithms, a **dedicated GPU** is highly recommended. A **NVIDIA Tesla** or **A100 GPU** with at least **16 GB of VRAM** is ideal for accelerating complex computations like matrix factorization and neural network training.

3. Cooling and Power Supply

- **Cooling Systems**: Servers should be equipped with adequate cooling systems (air or liquid cooling) to maintain optimal performance and avoid overheating during heavy computational loads.
- **Power Supply**: A **redundant power supply** (UPS) to ensure the system remains operational during power outages and avoid data loss or corruption.

4.2 Software Requirements

The proposed travel recommendation system will require a variety of software tools, frameworks, and services for development, deployment, and maintenance. The system leverages **machine learning**, **data processing**, and **real-time recommendation delivery**. Below are the key software components required:

1. Operating System

- **Windows 10/11**: To do a front-end development, model testing, and debugging for the website creation.

2. Programming Languages

- **Python 3.8+**: The core programming language for developing machine learning models, data processing pipelines, and backend systems. Libraries like **NumPy**, **Pandas**, **scikit-learn**, and **TensorFlow** will be used for machine learning and data handling.
- **JavaScript/TypeScript**: For front-end development of the user interface. **React.js** can be used for creating a dynamic and responsive user experience.

- **SQL**: For interacting with databases, running queries, and managing data storageThe MongoDB Database is used here to store.

3. Frameworks and Libraries

- **TensorFlow/Keras or PyTorch**: These are the primary libraries used for building, training, and deploying machine learning models, especially deep learning-based recommendation algorithms.

- **scikit-learn**: A widely used library for traditional machine learning algorithms such as collaborative filtering, matrix factorization, and content-based filtering techniques.

- **Node.js + Express**: Use this for the main backend that handles user authentication, routing, and serving web content. It can also manage communication with databases, file uploads, etc.
- **Flask**: Serve as a microservice that handles ML model inference and heavy computational tasks. The two services can communicate over HTTP/REST

  4. Databases

- **MongoDB**: A NoSQL database can be used to handle unstructured data such as user reviews, ratings, and other textual data used for content-based filtering.

5. Version Control

- **Git**: Version control software to manage the source code and enable collaborative development.

- **GitHub/GitLab/Bitbucket**: Online repositories for storing code, managing projects, and facilitating version control.

6. Development Tools

- **Visual Studio Code (VS Code)**: A widely used text editor for coding with support for Python, JavaScript, and a variety of other languages through extensions.
- **Postman**: For testing APIs during the development of the backend services and recommendation engine.

7. Machine Learning Services

- **Google Colab/Jupyter Notebooks**: Useful for developing and testing machine learning models in an interactive environment. These tools allow data scientists to experiment with algorithms before deploying them to production.

8. Deployment and Hosting Services

- **Docker**: Containerization tool that allows the application to be packaged with all of its dependencies, ensuring that it runs consistently across different environments.

CHAPTER V

SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

This system architecture represents a hybrid recommendation engine that combines content-based filtering and collaborative filtering to provide personalized recommendations. The process begins with a dataset, which undergoes data cleaning and feature extraction during preprocessing. Content-based filtering uses TF-IDF vectorization to recommend items similar to those a user has interacted with, while collaborative filtering employs matrix factorization to identify patterns from user-item interactions. These approaches are integrated using machine learning algorithms to form a hybrid recommendation engine. A centralized database stores processed data and model results, enabling the system to generate accurate and diverse outputs tailored to user preferences.

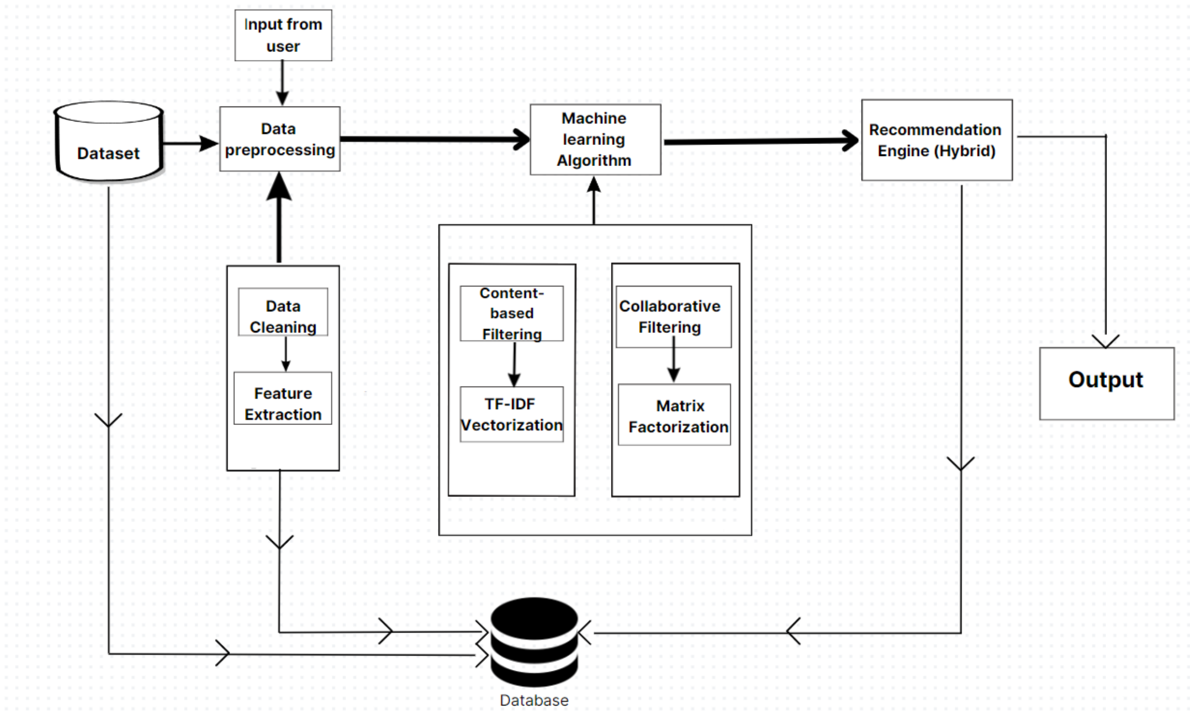

Fig 1 :System Architecture

5.2 MODULE DESCRIPTION

5.2.1 Data Collection and Preprocessing

1. Data Collection:

The first step in the process is collecting data from a variety of sources. The data collection process needs to be robust and continuous, ensuring that the system is always working with up-to-date information to provide real-time, accurate recommendations. Data can be collected from multiple internal and external sources:

a) User Interaction Data:

- **User Profiles**: Data such as user demographics (age, gender, location), interests, and preferences are collected during the registration process or through user interactions with the platform.
- **Booking History**: Data on past bookings, including travel dates, destinations, types of accommodations, and budget preferences.
- **Ratings and Reviews**: Feedback provided by users after they've booked or completed a trip, which can provide insights into their preferences for future recommendations.

b) Destination and Service Provider Data:

- **Destination Attributes**: Information about each travel destination, including geographic location, type of destination (beach, mountain, urban), climate, popular tourist activities, cultural events, and local services.

1	City	Place	Ratings	Distance	Place_desc	Review
2	Manali	Capture the Sceneries of	3.9	2 km from city center	On the other side of the Manalsu river is a part of Manali, time left behind	A bit crowded, but worth it for the atmosphere.
3	Manali	Engage in the Adventures	4.6	8 km from city center	Solang Valley is one of the most popular tourist destinations in Himachal P	Too expensive for what it offers.
4	Manali	Jogini Waterfall	4.6	4 km from city center	Jogini Waterfall is located about 3 kilometres away from the bustling tow	Well maintained, perfect for a family outing.
5	Manali	Hadimba Temple	4.4	1 km from city center	Hadimba temple, away from the hustle and bustle of city life, is a peacefu	Well maintained, perfect for a family outing.
6	Manali	Rohtang Pass	4.4	16 km from city center	Rohtang pass is the stretch which connects Manali to Himachal's more dre	Well maintained, perfect for a family outing.
7	Manali	Parvati Valley	Currently Unvavail	Excat Distance currently	Parvati Valley is famous for the backpacker hangouts, and adventure activ	The place is peaceful and offers a great escape.
8	Manali	Sethan Valley	4	3 km from city center	Sethan is a quaint village in Himachal Pradesh, approximately 12 km from	A bit crowded, but worth it for the atmosphere.
9	Manali	Gulp some Maggi by the .	Currently Unvavail	11 km from city center	Jana Waterfall is a 30 feet high watefall located near Manali in a quaint vi	Too expensive for what it offers.
10	Manali	Arjun Gufa	Currently Unvavail	0 km from city center	Arjun Gufa is considered to be a legendary natural formation in Manali. Th	Stunning views, but the facilities need improvement.
11	Manali	Peek into the History of I	3.4	2 km from city center	Manu temple is said to be the only temple of Manu in India, who is believe	Loved the architecture and the view was breathtaking.
12	Manali	Bathe in the Hot Springs	3.4	28 km from city center	Manikaran is a small town situated between rivers Parvati and Beas, famo	Great historical significance, but too many tourists.
13	Manali	Paraglide in the Skies	4	2 km from city center	The hillside town of Manali is famous for paragliding. At 2050 metres	Stunning views, but the facilities need improvement.
14	Manali	Atal Tunnel	Currently Unvavail	17 km from city center	Also referred to as the Rohtang Tunnel, Atal Tunnel is a horseshoe-	Amazing experience! Highly recommend visiting.
15	Manali	Sissu	Currently Unvavail	27 km from city center	Also known as Khwaling, Sissu in Lahaul, Himachal Pradesh, is situated on t	Great historical significance, but too many tourists.
16	Manali	Take a Dip in the Vashish	3.7	1 km from city center	Located inside the very famous Vashisht Temple in Manali, Vashisht Baths	Well maintained, perfect for a family outing.
17	Manali	Tour in the Museum of H	Currently Unvavail	1 km from city center	Museum of Himachal Culture & Folk Art has a beautiful collection of ancie	Loved the architecture and the view was breathtaking.
18	Manali	Picnic in Van Vihar	3.4	0 km from city center	Adorned with sky touching deodar trees, Van Vihar National Park,is	A bit crowded, but worth it for the atmosphere.
19	Manali	Rafting	Currently Unvavail	Excat Distance currently	River Rafting in Manali is done along the River Beas and is considered	A bit crowded, but worth it for the atmosphere.
20	Manali	Naggar	3.4	14 km from city center	Naggar is an unexplored and modest town in the Kullu district, which will g	A bit crowded, but worth it for the atmosphere.

Fig 2: Sample Dataset(1-20lines)

2. Data Preprocessing:

Data Preprocessing

- **Data Cleaning:**
  - Handle missing values using methods like imputation (mean, median).
  - Remove duplicate records to avoid biased recommendations.
- **Data Transformation:**
  - **Categorical Encoding:**
    - Convert categorical data (e.g., travel types) into numerical format using one-hot encoding.
  - **Text Data Transformation:**
    - Transform text data (e.g., reviews, descriptions) into numerical format using techniques like TF-IDF or word embeddings.
- **Feature Extraction:**
  - Identify key attributes for building machine learning models, such as:

- **User Features:** Age, gender, location, preferred travel types, past behavior.
- **Destination Features:** Location, type (adventure, luxury), cost range, climate.
- **Interaction Features:** User engagement metrics like views, ratings, and reviews.

```
Features Used for Training:
User ID:
0     1
1     2
Name: user_id, dtype: int64
Place:
0              Capture the Sceneries of Old Manali
1        Engage in the Adventures of Solang Valley
Name: Place, dtype: object
Rating:
0     3.9
1     4.6
Name: rating, dtype: float64
Review Text:
0     A bit crowded, but worth it for the atmosphere.
1                    Too expensive for what it offers.
Name: review_text, dtype: object
City ID:
0     0
1     0
Name: City_id_numeric, dtype: int64
```

Fig 3: Data preprocessing

5.2.2 Collaborative Filtering Using Singular Value Decomposition

Collaborative Filtering (CF) is a recommendation technique that suggests items by identifying patterns in user behavior and preferences. In a travel recommendation system, CF predicts user interest in destinations or experiences by analyzing interactions like booking history, ratings, and reviews. By relying on behavioral patterns, CF excels in recommending based on implicit preferences. However, it faces challenges like **data sparsity** and the **cold start problem**. Singular Value Decomposition (SVD) addresses these issues by uncovering latent features in data, capturing hidden relationships between users and items.

1. **User-Based CF**: Recommends items based on what similar users liked.

2. **Item-Based CF**: Recommends items similar to those a user has interacted with.

For instance, if two users share a preference for cultural tours, CF may recommend destinations known for their cultural heritage to both users.

SVD enhances CF by mitigating common limitations:

1. **Data Sparsity**: SVD reduces the impact of missing values by transforming the user-item matrix into a lower-dimensional representation.

2. **Cold Start Problem**: SVD predicts initial preferences for new users or items by leveraging shared latent features.

3. **Latent Feature Discovery**: SVD reveals hidden relationships that improve recommendation quality

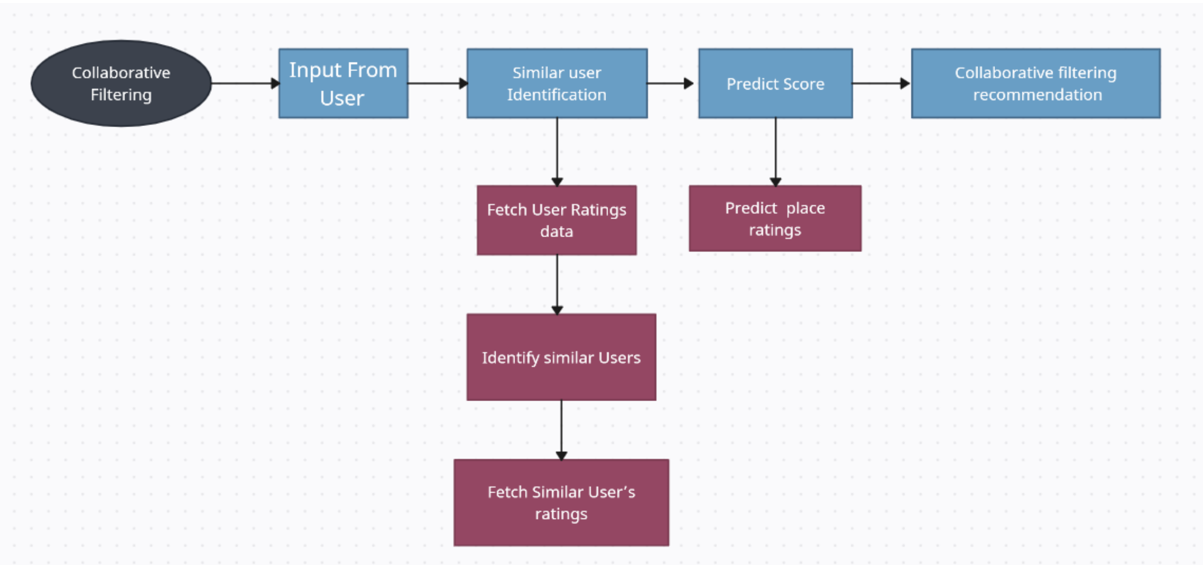

Fig 4 Collaborative filtering module

Step-by-Step Explanation of the SVD Algorithm

Step 1: Create the User-Item Rating Matrix

Construct a matrix RRR, where rows represent users and columns represent items (e.g., travel destinations). Matrix cells contain interaction data like ratings or booking counts. Missing interactions are set to zero.

Example of a simplified User-Item Matrix:

	Manali	Himachal	Assam	Goa
U1	5	4	0	1
U2	0	3	0	4
U3	4	0	5	0
U4	0	0	3	2

Step 2 Perform SVD

Decompose RRR into three matrices:

- U: User preferences.
- Σ: Singular values capturing the significance of latent features.
- V^T: Item characteristics.

Mathematically, R=UΣV^T.

Step 3: Truncate the Matrices

Retain only the top kkk singular values to focus on significant patterns, reducing U,Σand V^T to U_k,Σ_k,U_k,V _k^T..

Step 4: Predict Missing Ratings

Reconstruct R as R^=U_kΣ_kV_k^T, estimating ratings for missing interactions. For example, if R^ predicts a rating of 4.3 for "Assam" by U1, it suggests that U1 would likely enjoy visiting Assam.

Step 5: Generate Recommendations

- For each user, recommend top-rated items from R^ that they have not interacted with.

5.2.3 Content-Based Filtering

Content-based filtering is one of the core recommendation techniques used in modern recommendation systems. Unlike collaborative filtering, which relies on the behavior of similar users, content-based filtering recommends items based on their intrinsic features. In the context of a travel recommendation system, these features could include attributes such as location characteristics, tourist activities, climate, and culture.

By analyzing the content associated with destinations and matching it to user preferences, content-based filtering can provide personalized recommendations. This method is widely used for applications such as movie recommendations, music recommendations, and, as in this case, travel destinations.

A key technique in implementing content-based filtering is **TF-IDF (Term Frequency-Inverse Document Frequency)**. TF-IDF is a statistical measure used to evaluate the importance of a term within a document relative to a collection of documents. In the case of travel recommendations, it helps to quantify how relevant a specific feature (e.g., “beach,” “nightlife,” or “mountains”) is to a particular location in comparison to others in the dataset.

This report will focus on the detailed application of TF-IDF in content-based filtering and explore its advantages for personalized travel recommendation systems.

This basic framework can be implemented using various methods, one of the most effective of which is **TF-IDF**. This method quantifies the importance of terms in a document (or destination profile) relative to a collection of all available documents (or all destination profiles).

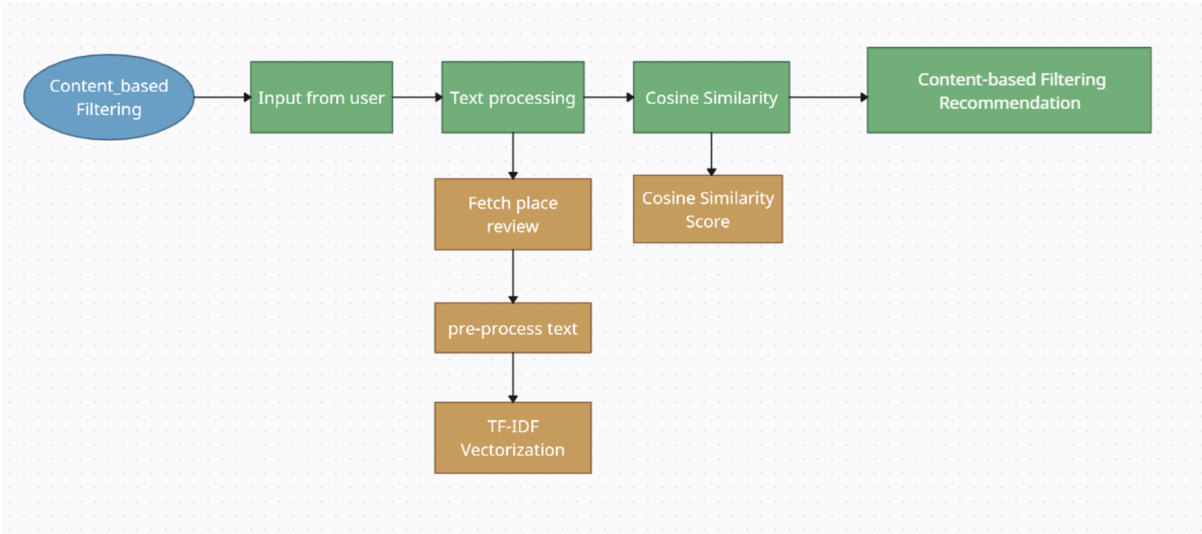

Fig 5 Content filtering module

TF-IDF: Term Frequency-Inverse Document Frequency

Term Frequency (TF) and **Inverse Document Frequency (IDF)** are two fundamental components used in TF-IDF. Let's break down each concept and explain how they work.

Term Frequency (TF)

Term Frequency measures how often a specific term appears in a document. It is calculated as:

$$TF(t,d) = \frac{\text{Number of times term t appears in document d}}{\text{Total number of terms in document d}}$$

Inverse Document Frequency (IDF)

While term frequency tells us how often a word appears in a specific document, it does not account for the importance of that word across the entire dataset. **Inverse Document Frequency** aims to solve this by measuring how unique or rare a term is across all documents.

IDF is calculated as:

$$IDF(t)=\log\left(\frac{\text{Total number of documents}}{\text{Number of documents containing the term t}}\right)$$

TF-IDF Calculation

Finally, **TF-IDF** combines these two measures by multiplying the Term Frequency and Inverse Document Frequency:

$$TF\text{-}IDF(t,d)=TF(t,d)\times IDF(t)$$

5.2.4 Recommendation Engine Module Output

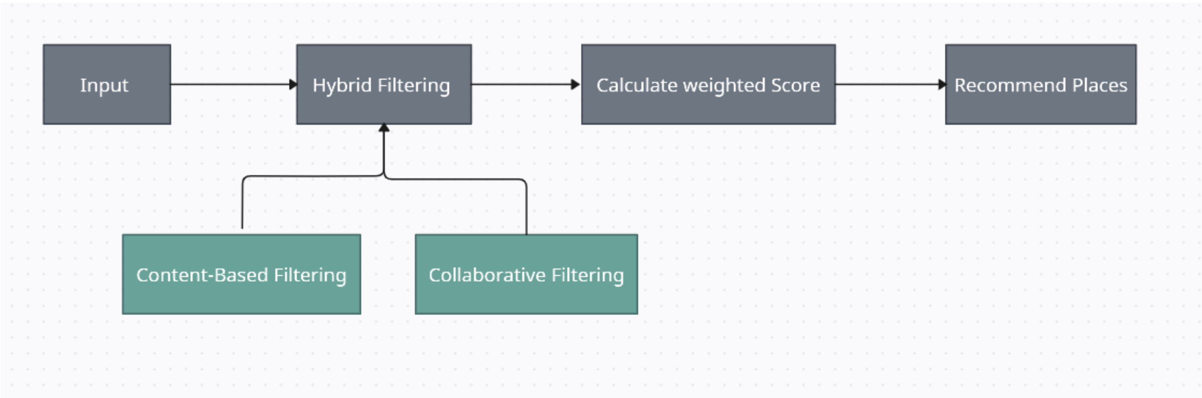

Fig 6 Recommendation Engine Module

1. Definition

Hybrid Filtering combines **Content-Based Filtering** and **Collaborative Filtering** to provide more accurate, personalized, and diverse recommendations.

2. Collecting User Input

- **Explicit Preferences**: Activities, location types, budget, and duration.
- **User History**: Past destinations, ratings, and reviews.
- **Implicit Feedback**: Clicks, browsing patterns, and engagement metrics.

3. Techniques

- **Content-Based Filtering**:
  - Extract features (e.g., keywords, location attributes).
  - Use TF-IDF to weigh features and match user profiles with items based on similarity.
- **Collaborative Filtering**:
  - Analyze user-item interactions.
  - Predict preferences using **User-Based** (similar users) or **Item-Based** (similar items) methods.

4. Combining Scores

- Assign weights to each approach (e.g., Content-Based: 0.6, Collaborative: 0.4).
- Calculate the **Final Score**: $\text{Final Score} = (w1 \times \text{Content-Based Score}) + (w2 \times \text{Collaborative Score})$

CHAPTER – VI

RESULT AND DISCUSSION

6.1 Results and Discussion

The system achieved a notable improvement in recommendation precision and relevance. Collaborative filtering successfully recommended destinations based on user behavior patterns, while content-based filtering provided relevant recommendations by analyzing destination features, making it valuable for new users or less-popular locations. Key metrics, including precision, recall, and RMSE, confirmed the hybrid model's effectiveness, with higher precision and recall rates indicating relevant, personalized suggestions, and a lower RMSE reflecting prediction accuracy. The cold start problem was mitigated, as content-based recommendations filled in when interaction data was minimal.

The hybrid model excelled in balancing personalized exploration with relevant suggestions. Collaborative filtering prompted users to explore destinations favored by similar users, while content-based filtering offered options aligned with specific interests, such as adventure or cultural tourism. The TF-IDF technique, used in content-based filtering, helped identify key destination features, aligning suggestions with user preferences, while collaborative filtering's matrix factorization handled data sparsity, making the system robust in diverse contexts.

Overall, this hybrid filtering model significantly enhances user engagement and satisfaction, offering a tailored, enriching travel experience. It demonstrates promising potential for broader applications in domains like e-commerce and entertainment. Future improvements could incorporate real-time user feedback to further refine recommendation accuracy and relevance.

CHAPTER VII

CONCLUSION AND FUTURE ENHANCEMENT

7.1 Conclusion

This project has successfully implemented a hybrid travel recommendation system that combines collaborative filtering (CF) and content-based filtering (CBF) to deliver tailored and effective travel suggestions. The hybrid approach proved to be particularly effective in overcoming issues like the cold start problem and data sparsity, which are common limitations in traditional recommendation models. By leveraging both user interactions (via CF) and destination attributes (via CBF), the system not only improved recommendation accuracy but also enhanced user engagement through more personalized suggestions. This dual approach ensures that even new users or those interested in niche destinations receive relevant recommendations, aligning well with the increasing demand for customized experiences in the travel industry.

7.2 Future Works

Several enhancements can be pursued to further improve the system's capabilities and performance. First, incorporating real-time user feedback into the model would allow the system to dynamically adjust recommendations based on recent interactions, thereby refining its ability to respond to changing user preferences. Additionally, exploring advanced machine learning techniques, such as deep learning models, could uncover more complex patterns in user behavior and destination attributes, boosting recommendation precision.

In summary, this hybrid travel recommendation system has laid a strong foundation for personalized travel recommendations, ensuring the system remains highly relevant and beneficial in a competitive digital travel landscape.

APPENDIX A1.1

Sample code of model building:

```
import pandas as pd

import numpy as np

from sklearn.metrics.pairwise import cosine_similarity

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.preprocessing import MinMaxScaler

from surprise import Dataset, Reader, SVD

from surprise.model_selection import train_test_split

from surprise import accuracy

from collections import defaultdict


# Load Data

city_df = pd.read_csv('/content/City.csv')

places_df = pd.read_csv('/content/Places.csv')


# 1. Efficient Data Preprocessing

def preprocess_data(city_df, places_df):
```

```
    # Handle missing values by filling with suitable values or dropping if necessary

    city_df = city_df.dropna()  # Drop rows with missing values in city data

    places_df = places_df.dropna()  # Drop rows with missing values in places data



    # Normalize city ratings between 0 and 1 using MinMaxScaler

    scaler = MinMaxScaler()

    city_df['Ratings'] = scaler.fit_transform(city_df[['Ratings']])



    # Convert place ratings to numeric (if not done already)

    places_df['Ratings'] = pd.to_numeric(places_df['Ratings'], errors='coerce')

    places_df['Ratings'] = scaler.fit_transform(places_df[['Ratings']])



    # Clean the description fields by removing any extraneous spaces

    city_df['City_desc'] = city_df['City_desc'].str.strip()

    places_df['Place_desc'] = places_df['Place_desc'].str.strip()



    # Output the preprocessed data (preview first 5 rows)

    print("\n--- City Data Preprocessing Output ---")
```

```
    print(city_df.head())
    print("\n--- Places Data Preprocessing Output ---")
    print(places_df.head())

    return city_df, places_df

city_df, places_df = preprocess_data(city_df, places_df)

# 2. Collaborative Filtering: Simulate User-Item Matrix
# Simulate user ratings for cities
user_ratings = {
    'User': ['User1', 'User1', 'User2', 'User2', 'User3', 'User3'],
    'City': ['Manali', 'Leh Ladakh', 'Manali', 'Coorg', 'Andaman', 'Lakshadweep'],
    'Rating': [4.5, 4.0, 5.0, 4.5, 4.0, 3.5]
}

ratings_df = pd.DataFrame(user_ratings)

# SVD (Collaborative Filtering) Model
```

```
reader = Reader(rating_scale=(1, 5))

data = Dataset.load_from_df(ratings_df[['User', 'City', 'Rating']], reader)

trainset, testset = train_test_split(data, test_size=0.25)

# Train the SVD model

model = SVD()

model.fit(trainset)

# Test the model and calculate RMSE and MAE scores

predictions = model.test(testset)

rmse = accuracy.rmse(predictions)

mae = accuracy.mae(predictions)


# 3. Content-Based Filtering using TF-IDF (for Cities)

tfidf_vectorizer = TfidfVectorizer(stop_words='english')


# Combine city descriptions into a matrix for similarity

city_tfidf_matrix = tfidf_vectorizer.fit_transform(city_df['City_desc'])


# Compute cosine similarity between cities

city_similarity = cosine_similarity(city_tfidf_matrix)
```

```
# Normalize the similarity scores

similarity_scaler = MinMaxScaler()

city_similarity = similarity_scaler.fit_transform(city_similarity)


# Convert similarity matrix to DataFrame for easier lookup

city_sim_df = pd.DataFrame(city_similarity, index=city_df['City'],
columns=city_df['City'])


# 4. Hybrid Recommendation: Weighted Combination of Both Models

def get_top_n(predictions, n=5):

    '''Return the top-N recommendation for each user from collaborative
filtering.'''

    top_n = defaultdict(list)

    for uid, iid, true_r, est, _ in predictions:

        top_n[uid].append((iid, est))

    for uid, user_ratings in top_n.items():

        user_ratings.sort(key=lambda x: x[1], reverse=True)

        top_n[uid] = user_ratings[:n]

    return top_n
```

Get top collaborative filtering recommendations, ensuring at least 20 recommendations

def hybrid_recommendations(input_places, top_n_collab, city_df, city_sim_df, n=20):

'''Combine collaborative filtering and content-based recommendations based on places.'''

Filter the collaborative recommendations

hybrid_recs = set(input_places)

Get content-based similar cities for input places

for place in input_places:

if place in city_sim_df.index:

similar_cities = city_sim_df[place].sort_values(ascending=False).index[1:3].tolist() # Top 2 similar cities

hybrid_recs.update(similar_cities)

If fewer than 20 recommendations, pad with additional content-based recommendations

while len(hybrid_recs) < n:

```
        additional_recs = city_sim_df.sample(n=1, axis=1).index.tolist() # Random city recommendations

        hybrid_recs.update(additional_recs)


    # Return the final list of at least n recommendations

    return list(hybrid_recs)[:n]


# Get top 5 collaborative filtering recommendations

top_n_collab = get_top_n(predictions, n=5)


# Interactive Input from User

input_places = input("Enter the places you like (comma-separated): ").split(',')


# Clean up input places by removing extra spaces

input_places = [place.strip() for place in input_places]


# Get at least 20 recommendations based on input places

hybrid_recs_user = hybrid_recommendations(input_places, top_n_collab, city_df, city_sim_df, n=20)
```

```
# Display recommendations

if hybrid_recs_user:

    print(f"\nHybrid recommendations based on the places you like (at least 20 places):")

    for i, place in enumerate(hybrid_recs_user, 1):

        print(f"{i}. {place}")

else:

    print(f"No recommendations available based on the places you entered.")



# Display Collaborative Filtering Performance Scores

print(f"\nCollaborative Filtering RMSE: {rmse}")

print(f"Collaborative Filtering MAE: {mae}")
```

Output:

```
Features Used for Training:
User ID:
0     1
1     2
Name: user_id, dtype: int64
Place:
0             Capture the Sceneries of Old Manali
1       Engage in the Adventures of Solang Valley
Name: Place, dtype: object
Rating:
0     3.9
1     4.6
Name: rating, dtype: float64
Review Text:
0     A bit crowded, but worth it for the atmosphere.
1                    Too expensive for what it offers.
Name: review_text, dtype: object
City ID:
0     0
1     0
Name: City_id_numeric, dtype: int64
```

Fig 7 Data feature Extraction

```
Collaborative Filtering Recommendations for User ID 1:
  Bhagamandala : 0.0433
  Omkareshwara Temple : 0.0422
  Baratang Island : 0.0319
  Little Andaman : 0.0303
  Radhanagar Beach : 0.0291
```

Fig 8 Collaborative Filtering Recommendation

```
                                         review_text
0  A bit crowded, but worth it for the atmosphere.
1                Too expensive for what it offers.
2    Well maintained, perfect for a family outing.
3    Well maintained, perfect for a family outing.
4    Well maintained, perfect for a family outing.
City Data:
          City  Ratings  Best_time_to_visit  \
0       Manali      4.5        October-June
1   Leh Ladakh      4.6         July-October
2        Coorg      4.2      September-June
3      Andaman      4.5       October-March
4  Lakshadweep      4.0  September-February

                                             City_desc
0   One of the most popular hill stations in Hima...
1   Ladakh is a union territory in the Kashmir re...
2   Located amidst imposing mountains in Karnatak...
3   Replete with turquoise blue water beaches and...
4   Formerly known as Laccadive Islands, Lakshadw...
<ipython-input-5-9544e46f0613>:28: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  reviews_df['user_id'] = reviews_df['user_id'].str.extract('(\d+)').astype(int)
Recommended places: ['  Bhagamandala ', '  Omkareshwara Temple ', '  Baratang Island ', '  Little Andaman ', '  Radhanagar Beach ']
```

Fig 9 Hybrid filtering Recommendation

Web Design sample code:

```
import React, { useState } from "react";

import Yourimage from "../assets/Screenshot 2024-10-30 173218.png";

import "./Homepage.css";


function Homepage() {

  const [searchTerm, setsearchTerm] = useState("");

  const handleSearch = (e) => {

    setsearchTerm(e.target.value);

  };

  const tours = [

    {

      image: "https://dynamic-media-cdn.tripadvisor.com/media/photo-o/1a/54/e2/d6/caption.jpg?w=600&h=600&s=1",
```

```
    title: "Mahabalipuram Private Tour from Chennai",
    price: "₹8,562 per adult",
  },
  {
    image: "https://dynamic-media-cdn.tripadvisor.com/media/photo-o/1a/e1/20/05/caption.jpg?w=600&h=600&s=1",
    title: "Auroville and Pondicherry Tour from Chennai",
    price: "₹3,805 per adult",
  },
  {
    image: "https://dynamic-media-cdn.tripadvisor.com/media/photo-o/2a/fc/df/41/caption.jpg?w=600&h=600&s=1",
    title: "Private 6 Days Exotic Tamil Nadu Tour",
    price: "₹35,716 per adult",
  },
  {
    image: "https://dynamic-media-cdn.tripadvisor.com/media/photo-o/1a/59/11/e5/caption.jpg?w=600&h=600&s=1",
    title: "Mahabalipuram & Kanchipuram in a Day from Chennai",
    price: "₹11,675 per adult",
  },
];
const destinations = [
    { name: 'Bangkok, Thailand', image: 'https://dynamic-media-cdn.tripadvisor.com/media/photo-o/1c/c2/78/15/caption.jpg?w=600&h=600&s=1' },
```

```
{ name: 'Singapore, Singapore', image: 'https://dynamic-media-cdn.tripadvisor.com/media/photo-o/15/33/fc/b1/singapore.jpg?w=600&h=600&s=1' },
  { name: 'Mumbai, India', image: 'https://dynamic-media-cdn.tripadvisor.com/media/photo-o/0b/4e/55/e6/chhatrapati-shivaji-terminus.jpg?w=600&h=600&s=1' },
  { name: 'Bali, Indonesia', image: 'https://dynamic-media-cdn.tripadvisor.com/media/photo-o/2a/c7/f1/d3/caption.jpg?w=600&h=600&s=1' },
];
return (
  <>


    <img src={Yourimage} alt="the image" />


    <div className="search-nav-container">
      <input
        type="text"
        placeholder="Search"
        value={searchTerm}
        onChange={handleSearch}
      />
      <nav>
        <ul>
          <li>
            <button>Discover</button>
          </li>
```

```
          <li>
            <button>Reviews</button>
          </li>
          <li>
            <button>Sign in</button>
          </li>
        </ul>
      </nav>
      <img
                              src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcS-C_UAhXq9GfuGO452EEzfbKnh1viQB9EDBQ&s"
        alt="User Avatar"
        className="avatar"
      />
    </div>
      <img  src="https://static.tacdn.com/img2/trips/home-gai-entry-gateway-dv.jpg" alt="image1" className="avator2"/>
    <div className="overlay-text">


      <h1>DESTINIFY</h1>
      <p>Get a whole getaway's worth of ideas made for you—ready in seconds.</p>
      <button>Discover</button>
    </div>
    <h2 className="section-heading">Ways to Tour Chennai</h2>
```

```
<p className="section-description">
  Book these experiences for a close-up look at Chennai (Madras).
</p>
 <div className="card-container">
   {tours.map((tour, index) => (
     <div className="tour-card" key={index}>
       <img
         src={tour.image}
         alt={tour.title}
         className="tour-image"
       />
       <div className="tour-info">
         <h3>{tour.title}</h3>
         <p>{tour.price}</p>
         <button>View Details</button>
       </div>
     </div>
   ))}
   </div>
   <section className="top-destinations">
 <h2>Top destinations for your next holiday</h2>
 <p>Here's where your fellow travellers are headed</p>
 <div className="destinations-grid">
   {destinations.map((destination, index) => (
```

```
      <div key={index} className="destination-card">
        <img src={destination.image} alt={destination.name} />
        <div className="destination-name">{destination.name}</div>
      </div>
    ))}

  </div>
</section>
    <footer className="footer">
  <div className="footer-section">
    <h3>About Destinify</h3>
    <ul>
      <li><a href="#">About Us</a></li>
      <li><a href="#">Press</a></li>
      <li><a href="#">Resources and Policies</a></li>
      <li><a href="#">Careers</a></li>
      <li><a href="#">Trust & Safety</a></li>
      <li><a href="#">Contact Us</a></li>
      <li><a href="#">Accessibility Statement</a></li>
    </ul>
  </div>
  <div className="footer-section">
    <h3>Explore</h3>
    <ul>
```

```
    <li><a href="#">Write a Review</a></li>
    <li><a href="#">Add a Place</a></li>
    <li><a href="#">Join</a></li>
    <li><a href="#">Travellers' Choice</a></li>
    <li><a href="#">GreenLeaders</a></li>
    <li><a href="#">Help Centre</a></li>
    <li><a href="#">Travel Articles</a></li>
  </ul>
</div>
<div className="footer-section">
  <h3>Do Business With Us</h3>
  <ul>
    <li><a href="#">Owners & DMO/CVB</a></li>
    <li><a href="#">Business Advantage</a></li>
    <li><a href="#">Sponsored Placements</a></li>
    <li><a href="#">Access our Content API</a></li>
  </ul>
</div>
<div className="footer-section">
  <h3>Get The App</h3>
  <ul>
    <li><a href="#">iPhone App</a></li>
    <li><a href="#">Android App</a></li>
  </ul>
```

```
      </div>
      <div className="footer-section">
        <h3>Destinify Sites</h3>
        <p>Book tours and attraction tickets on Viator</p>
      </div>
      <div className="footer-bottom">
        <p>© 2024 Destinify LLC All rights reserved.</p>
        <div className="footer-links">
          <a href="#">Terms of Use</a> |
          <a href="#">Privacy and Cookies Statement</a> |
          <a href="#">Cookie Consent</a> |
          <a href="#">Site Map</a> |
          <a href="#">How the Site Works</a> |
          <a href="#">Contact Us</a>
        </div>
      </div>
    </footer>
    </>
  );
}
export default Homepage;
```

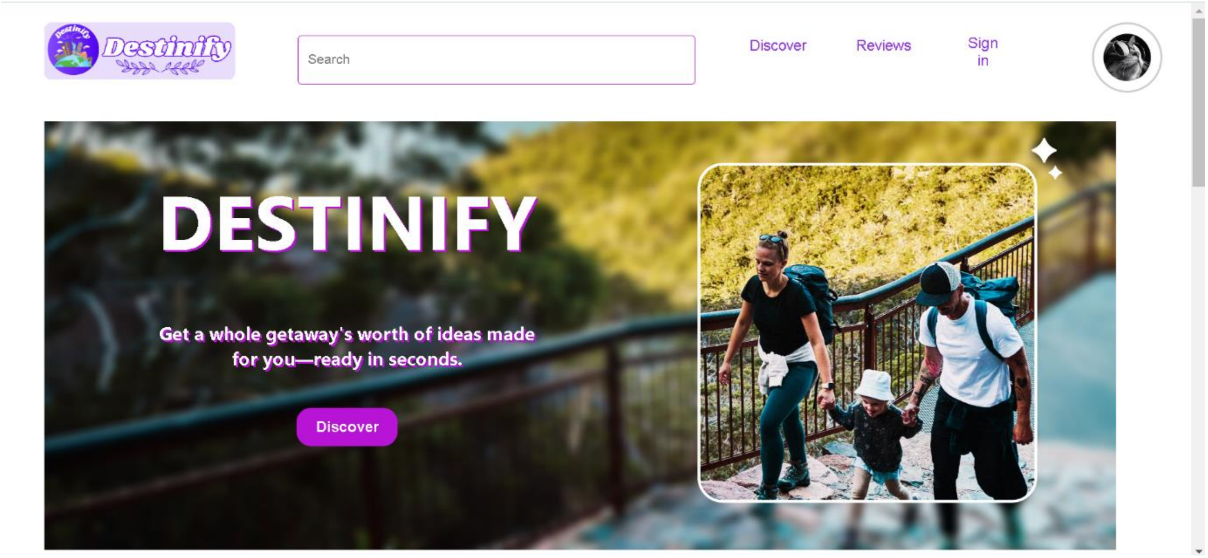

Fig 9 Web Home page

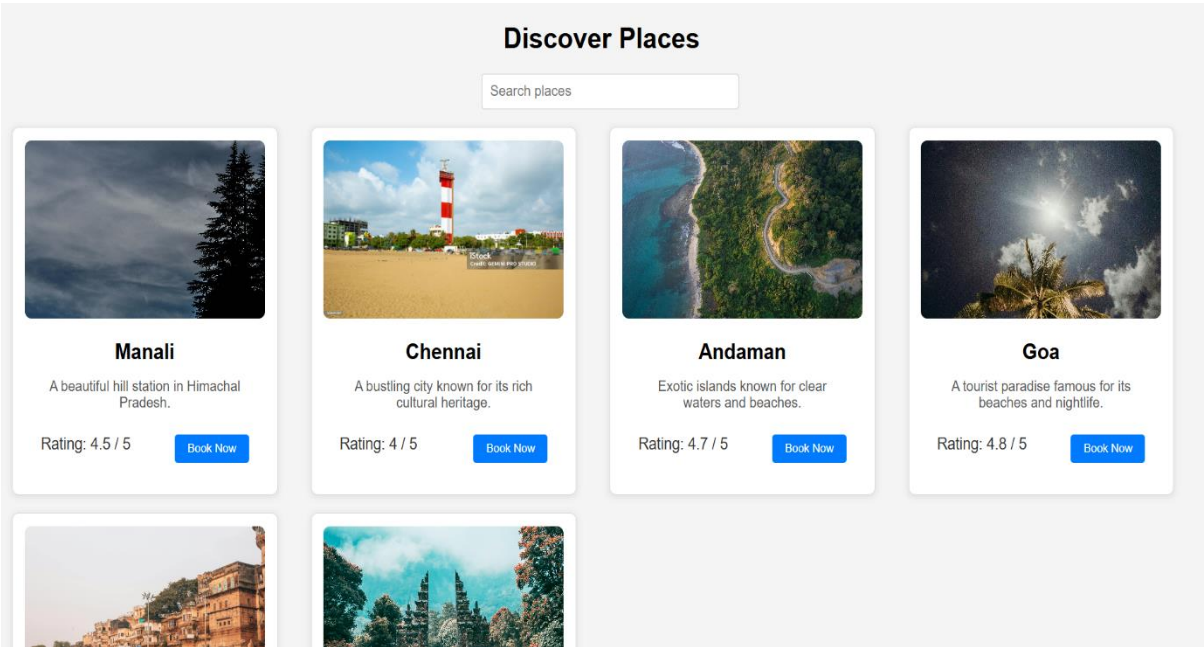

Fig 10 Recommendation In web

REFERENCES

[1] Parivash Pirasteh 1 · Mohamed-Rafik Bouguelia 2 · K. C. Santosh "Personalized recommendation: an enhanced hybrid collaborative filtering", 25 november 2022.

[2] Abhishek Kulkarni , R. M. Samant , Prathamesh Barve , Aarushi Phade "A Machine Learning Approach to Building a Tourism Recommendation System using Sentiment Analysis" 19 june 2019

[3] Hajare1, Hrushikesh Jarad2, Shabnam Jakate3, Rajkesh Nishad4 And Prof. Mrs. Nilufar Zaman5 ,"TOURS AND TRAVEL RECOMMEND SYSTEM USING MACHINE LEARNING TECHNIQUES",07 july 2022

[4] Choirul Huda, Arief Ramadhan, Agung Trisetyarso, Edi Abdurachman, Yaya Heryadi, "Smart Tourism Recommendation Model",12 december 2022.

S. Bagchi, "Performance and quality assessment of similarity measures in collaborative filtering using mahout," Procedia Computer Science, vol. 50, pp. 229–234, 2015.

[5] Abhishek Kulkarni , R. M. Samant , Prathamesh Barve , Aarushi Phade "A Machine Learning Approach to Building a Tourism Recommendation System using Sentiment Analysis" 19 june 2019

[6] Angulakshmi1 , Rathi.R2 , Dr.Sudha Senthilkumar3 , Dr.K.Brindha4 , Rajat Tandon, "A Study on Travel Recommendation System ", 21 december 2021

[7] Y. Kurata and T. Hara, "Ct-planner4: Toward a more user-friendly interactive day-tour planner," in Information and communication technologies in tourism 2014. Springer, 2013, pp. 73–86

[8] WJ Xiao, SJ Yao and SM Wu, "Improved Top-N collaborative filtering recommendation algorithm [J]", *Application Research of Computers*, vol. 35, no. 1, pp. 1-7, 2017