International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

AI/ML-Smart Tourist Recommendation System

Sahana Kumari B[1], Rohit Hooli[2], Rohan U S[3], Siddivinayak B[4], Tirupati S K[5]
[1]Assistant Professor, Department of Computer Science and Engineering
[2,3,4,5]Undergraduate Students, Department of Computer Science and Engineering
Sri Dharmasthala Manjunatheshwara Institute of Technology, Ujire, Karnataka, India
Correspondence should be addressed to Sahana Kumari B; kumaribsahana07@gmail.com

***Abstract*—**Tourism recommendation systems help travellers discover destinations according to their individual interests and travel constraints. This paper presents an AI/ML-based smart tourist recommendation system that combines a React/Vite web interface, Python FastAPI backend, SQLite persistence, and a machine learning recommendation component. The system collects user preferences such as interests, budget, travel pace, and accessibility requirements and uses content-based recommendation to rank candidate destinations. TF-IDF feature extraction and cosine similarity are used to estimate the relevance of candidate destinations. The system also integrates destination information, feedback, and itinerary-related functionality. Its modular architecture permits future integration of collaborative filtering, contextual information, semantic embeddings, and larger tourism datasets. The current work emphasizes functional implementation and does not report unsupported numerical recommendation-quality metrics.

***Index Terms*—**Smart Tourism, Recommendation System, Machine Learning, TF-IDF, Cosine Similarity, Content-Based Filtering, Personalized Recommendation

I. Introduction

A. Background

Tourism recommendation has become an important application area of intelligent information systems because travellers increasingly use online services to discover destinations and plan trips. A traveller may need to consider interests, activities, accessibility, budget, travel pace, and available time simultaneously. Intelligent tourism recommender systems aim to reduce this information overload by filtering and ranking alternatives according to user requirements [1], [2].

B. Problem Statement

Conventional tourism search interfaces generally provide broad lists of destinations and require users to manually compare them. Such interfaces do not necessarily model individual preferences in a systematic manner. The proposed system addresses this problem by transforming explicit user preferences into a ranked list of candidate destinations and presenting the results through an integrated web application. Furthermore, the system considers travel interests and destination characteristics to improve recommendation relevance. It reduces the effort required to identify suitable tourist destinations and supports more efficient travel planning. The system also provides a structured way to explore available destinations based on individual requirements and preferences.

C. Objectives

The main objectives are to collect user preferences, represent destination information in a machine-readable form, generate personalized recommendations, apply content-based similarity, incorporate available travel constraints, provide understandable recommendation results, and establish a modular architecture that can support collaborative and context-aware methods.

D. Contributions

The work combines a content-based recommendation method with an end-to-end tourism web application. The prototype integrates preference input, recommendation generation, destination presentation, feedback, and itinerary-related functionality. The architecture separates the recommendation layer from the user interface and persistence layer, allowing future model extensions.

II. Literature Review

General recommender-system literature identifies content-based, collaborative, and hybrid approaches as major recommendation paradigms. Content-based systems recommend items by comparing item characteristics with a representation of user preferences [3]. TF-IDF provides a traditional method for representing textual information through term weighting [4].

Collaborative filtering uses user-item interactions to identify collective patterns. Matrix factorization is an established approach for collaborative recommendation [5]. Tourism recommendation systems additionally need to consider geographical constraints, contextual information, travel preferences, and itinerary requirements [1], [2].

Research in smart tourism has examined recommendation systems, personalization, data management, and context-aware travel recommendation [6], [7]. Personalized itinerary recommendation has also developed toward optimization and learning based approaches [8].

A. Comparative Analysis

Table I compares the major recommendation approaches and their role in the current project.

The current prototype uses a content-based approach as its initial recommendation baseline. Collaborative and hybrid approaches are considered future extensions because they require additional interaction data and experimentation.

TABLE I
COMPARATIVE ANALYSIS OF TOURISM RECOMMENDATION APPROACHES

Approach	Main Input	Key Characteristic	Project Role
Content-based	Item features and user profile	Matches destination features with user preferences	Current baseline
Collaborative	User–item interactions	Learns patterns from user interactions	Future extension
Hybrid	Multiple recommendation signals	Combines content-based and collaborative methods	Future extension

III. System Architecture

A. Architecture Overview

The proposed system follows a client-server architecture. The React/Vite frontend collects user preferences and displays recommendation results. FastAPI provides REST-style backend services and coordinates recommendation requests. The Python recommendation layer performs feature processing and similarity calculations. SQLite provides lightweight persistence for the prototype.

This separation allows the frontend, backend, recommendation model, and persistence layer to be developed and tested independently. It also provides a foundation for replacing or extending individual components without redesigning the complete application.

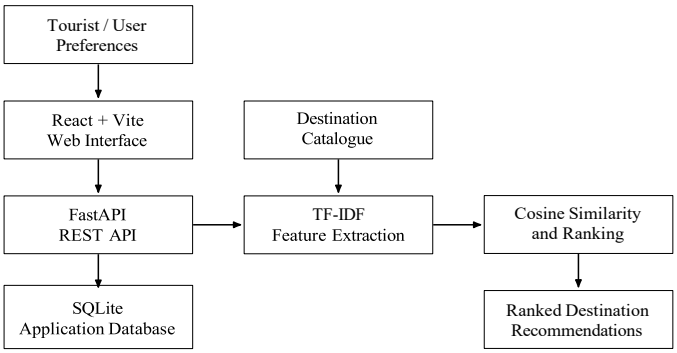


Fig. 1. System architecture of the proposed smart tourist recommendation system.

B. Major Modules

The proposed system consists of the User Preference Module, Destination Data Module, Recommendation Module, Recommendation Ranking Module, Feedback Module, Itinerary Module, and Web Interface Module.

The User Preference Module collects traveller requirements, including interests, budget, travel pace, and accessibility. The Destination Data Module maintains destination information such as location, categories, descriptions, and travel attributes. The Recommendation Module performs feature extraction and similarity calculations. The Ranking Module orders candidate destinations according to their similarity scores and available preference constraints.

The Feedback Module supports user interaction, while the Itinerary Module provides itinerary-related functionality. The Web Interface Module allows users to submit preferences and view the resulting recommendations.

C. Data Flow

The data flow begins when a user submits travel preferences through the web interface. The backend validates the request and constructs a representation of the user's preferences. Candidate destinations are transformed into the same feature space using TF-IDF feature extraction.

Cosine similarity is then calculated between the user preference representation and destination feature vectors. The candidate destinations are ranked according to their similarity scores and applicable preference constraints. The resulting recommendations are returned to the frontend for display.

Finally, users can inspect destination information, provide feedback, and proceed to itinerary-related functionality.

IV. Methodology

A. Data Preparation

Destination records contain available attributes such as name, location, category or interest tags, descriptive text, rating-related information where available, budget, travel style, and accessibility. The development catalogue contains a small labelled set of sample destinations for prototype validation. It is not considered a statistically representative dataset for the complete tourism domain.

B. TF-IDF Feature Extraction

TF-IDF is used to transform textual information into numerical feature vectors. For a term t in document d:

$$TFIDF(t, d) = TF(t, d) \times IDF(t) \quad (1)$$

where $TF(t, d)$ represents the frequency of a term in a document and $IDF(t)$ represents its inverse document frequency. The user's interests and destination descriptions are represented in a common feature space, allowing their textual characteristics to be compared.

C. Cosine Similarity

Cosine similarity measures the similarity between the user vector U and destination vector D_i:

$$S_i = \frac{U \cdot D_i}{|U||D_i|} \quad (2)$$

where S_i represents the content similarity between the user preference representation and destination D_i. A higher similarity indicates greater correspondence between their represented characteristics.

D. Preference-Aware Ranking

The system can incorporate additional preference-related constraints alongside content similarity:

$$Score_i = \alpha S_i + \beta B_i + \gamma P_i + \delta A_i \quad (3)$$

where S_i represents content similarity, B_i represents budget compatibility, P_i represents travel-pace compatibility, and A_i

represents accessibility compatibility. The coefficients $\alpha, \beta, \gamma, \delta$ should be determined from the actual implementation or experiments rather than assumed without evidence.

E. Recommendation Workflow

The recommendation process consists of the following stages: 1) the traveller enters preferences through the web interface; 2) the FastAPI backend validates the submitted information; 3) user preferences are converted into a feature representation; 4) destination records are converted into corresponding vectors; 5) cosine similarity is calculated for candidate destinations; 6) available preference constraints are incorporated into ranking; 7) ranked recommendations are returned to the frontend; and 8) the user can inspect destinations and continue to feedback or itinerary functionality.

V. Implementation

The frontend is implemented using React and Vite to provide an interactive web interface. Python FastAPI is used to provide REST-style backend services and coordinate application requests. SQLite provides lightweight persistence for the prototype.

Python libraries including pandas, NumPy, and scikit-learn support data processing and machine learning operations. Communication between the frontend and backend follows an HTTP/JSON model. User preferences are submitted to backend services, processed by the recommendation component, and returned as structured recommendation results.

The modular implementation separates user interaction, backend services, recommendation processing, and application data. This structure allows the current content-based recommendation component to be extended with collaborative filtering, semantic embeddings, and context-aware recommendation methods. The system also incorporates input validation and exception handling to improve reliability during user interactions. The recommendation results are presented through the frontend in a clear and user-friendly format. The use of lightweight technologies simplifies deployment, testing, and maintenance. Furthermore, the modular design supports future enhancements without requiring major changes to the existing architecture.

VI. Results and Discussion

The implemented prototype demonstrates the functional workflow from preference collection to recommendation generation and destination presentation. The system also supports destination information, feedback, and itinerary-related functionality.

The content-based method provides a practical baseline because it can operate without requiring a large historical user-interaction dataset. However, recommendation quality depends on the coverage and quality of the destination catalogue. The content-based method also has limited ability to learn collective traveller behaviour compared with collaborative approaches.

The current development catalogue is limited and therefore should not be used to claim general recommendation accuracy. Formal evaluation requires a verified evaluation dataset and controlled experiments. Suitable measures include Precision@K, Recall@K, NDCG@K, response latency, and user satisfaction. No numerical values are reported here because they are not established by the current prototype evidence.

VII. Future Enhancements

Future development can extend the system with collaborative filtering to use historical user-item interactions. A hybrid recommendation model can combine content-based and collaborative signals.

Semantic embeddings can improve the representation of destination descriptions and user interests. Context-aware recommendation can incorporate location, travel time, season, and availability. Real-time tourism information can further improve the usefulness of recommendations.

The itinerary component can be extended using route optimization and multi-day planning techniques. Explainable recommendation can provide users with understandable reasons for suggested destinations. A larger verified tourism dataset and systematic evaluation are required before reporting quantitative recommendation performance. Production deployment would additionally require appropriate privacy, security, scalability, and monitoring mechanisms. Additionally, a feedback mechanism can be introduced to refine recommendations based on user ratings and preferences. Mobile application support can also be developed to provide personalized travel recommendations while users explore destinations.

VIII. Conclusion

This paper presented an AI/ML-based smart tourist recommendation system that combines a web application with a machine learning recommendation component. The proposed architecture uses React/Vite, FastAPI, SQLite, and Python-based recommendation processing.

The current recommendation baseline uses TF-IDF feature extraction and cosine similarity to match user preferences with destination information. Additional factors such as budget, travel pace, and accessibility can be incorporated into the ranking process.

The modular architecture provides a foundation for future development including collaborative filtering, hybrid recommendation, semantic embeddings, contextual information, itinerary optimization, and systematic evaluation. The current prototype demonstrates functional integration, while larger verified datasets and controlled experiments are required for quantitative validation.

References

[1] J. Borra`s, A. Moreno, and A. Valls, “Intelligent tourism recommender systems: A survey,” *Expert Systems with Applications*, vol. 41, no. 16, pp. 7370–7389, 2014. doi: 10.1016/j.eswa.2014.06.007.

[2] S. Renjith, A. Sreekumar, and J. Jathavedan, “An extensive study on the evolution of context-aware personalized travel recommender systems,” *Information Processing & Management*, vol. 57, no. 1, Art. no. 102078, 2020. doi: 10.1016/j.ipm.2019.102078.

[3] M. J. Pazzani and D. Billsus, “Content-Based Recommendation Systems,” in *The Adaptive Web*, vol. 4321, Springer, 2007, pp. 325–341. doi: 10.1007/978-3-540-72079-9_10.

[4] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988. doi: 10.1016/0304-4573(88)90021-0.

[5] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009. doi: 10.1109/MC.2009.263.

[6] R. A. Hamid et al., "How smart is e-tourism? A systematic review of smart tourism recommendation system applying data management," *Computer Science Review*, vol. 39, Art. no. 100337, 2021. doi: 10.1016/j.cosrev.2020.100337.

[7] K. Selwon and J. Szyman´ski, "Advancements and challenges in recommendation systems for the tourism industry: A comprehensive literature review," *Computer Science Review*, vol. 62, Art. no. 101024, 2026. doi: 10.1016/j.cosrev.2026.101024.

[8] S. Halder, K. H. Lim, J. Chan, and X. Zhang, "A survey on personalized itinerary recommendation: From optimisation to deep learning," *Applied Soft Computing*, vol. 152, Art. no. 111200, 2024. doi: 10.1016/j.asoc.2023.111200.