iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.80308

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

TourGuideAI: An AI-Based Tourism Planner and Recommendation Tool

Ashtam Katiyar[1], Chaitanya Garg[2], Anmol Srivastava[3], Aman Kumar[4]
Computer Science and Engineering, Galgotias College of Engineering and Technology, Greater Noida, India

Abstract: ***Tourism planning is a major challenge to tourists because of the sheer amount of destinations, accommodation and activities to choose. Trip planning is usually efficient and personalized and may require some prior knowledge or expert advice. AI can be used to improve the tourism experience as it can offer smart suggestions grounded on the preferences of the user and a range of contextual factors. In this paper, we present TourGuideAI, a multi-functional tourism planning and recommendation system based on integrated machine learning techniques. We created a set of data that included tourism sites, user ratings, reviews, and data (categories that included historical, cultural, adventure, and leisure). The system uses collaborative filtering coupled with content-based filtering in a hybrid recommendation model to create customized travel itineraries. The model is trained in phases such as feature engineering, similarity computation and ranking optimization. Other techniques like data normalization, preference weighting and feedback based tuning are implemented. The implementation of the system is done using a Flask-based web application and a mobile-friendly interface. TourGuideAI shows great results in providing meaningful recommendations, enhancing user satisfaction and decision-making effectiveness. According to error analysis, the majority of errors happen when the preferences of users are ambiguous or when the data is sparse. Ranked outputs have confidence scores to improve interpretability in the system. These results indicate that TourGuideAI can be a powerful intelligent assistant in the tourism planning process, especially in the context of less informed users about the destinations[1], [2].***

Keywords: ***Tourism Recommendation, Machine Learning, Recommender Systems, Collaborative Filtering, Content-Based Filtering, Flask Web Application, Travel Planning, Personalization.***

I. INTRODUCTION

Tourism is a fast-evolving sector in the world, which plays a major role in the economic growth and cultural exchange. Nevertheless, it may be quite complicated to plan the visit as the range of possible options is enormous. Visitors tend to find it difficult to single out the locations that match their preferences, budget, and time limit. The old techniques utilize fixed travel guides and hand searching, both of which are not personalized and adaptable. This frequently leads to poor decision making and less satisfaction. The latest developments in artificial intelligence especially machine learning have allowed systems to give personalized recommendations by processing large datasets [1]. The use of recommender systems has been successful in other fields like e-commerce and entertainment and its use in tourism is on the rise [2], [3]. With machine learning, user preferences, behavior, and contextual information can be analyzed, creating a unique recommendation. Most systems available, however, lack scalability, customization and real-time flexibility. We suggest TourGuideAI, an AI tourism planning system to overcome these challenges. The most important contributions are: Collaborative and content-based filtering hybrid recommendation model. The implementation can be done on the web and on mobiles. Confidence-scored generation of personalized itineraries.

II. RELATED WORK

The past few years have seen a lot of research on AI-based tourism recommendation systems. Collaborative filtering systems suggest destinations to users depending on their interaction and similarity with other users [2]. They however have cold-start problems when data of the users are minimal. Content-based filtering is based on comparing the user preferences and the item characteristics like the destination type and activity [3]. Though efficient, it can be lacking in variety of suggestions. Hybrid recommendation systems are a combination of the two methods to enhance accuracy and eliminate limitations [4]. New research also includes information about the situation including location, weather and time to improve recommendations [5]. Although these have been developed, the current systems are still challenged by: Limited personalization Reliance on bulk data. Absence of on-the-fly flexibility. TourGuideAI overcomes these constraints by combining dynamic user input with hybrid techniques.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

III. METHODOLOGY

TourGuideAI has a four-step architecture: User Input Acquisition The preferences given by users include budget, interests, number of days to be spent and type of destination. Data Processing The input data is purged, standardized, and transformed to the tourism dataset. Recommendation Engine The system uses: Collaborative Filtering Content-Based Filtering Hybrid Ranking Model Output Generation Generates: Recommended destinations Travel itinerary Estimated cost System Architecture User contribution Data Processing Hybrid Model Ranked Recommendations.

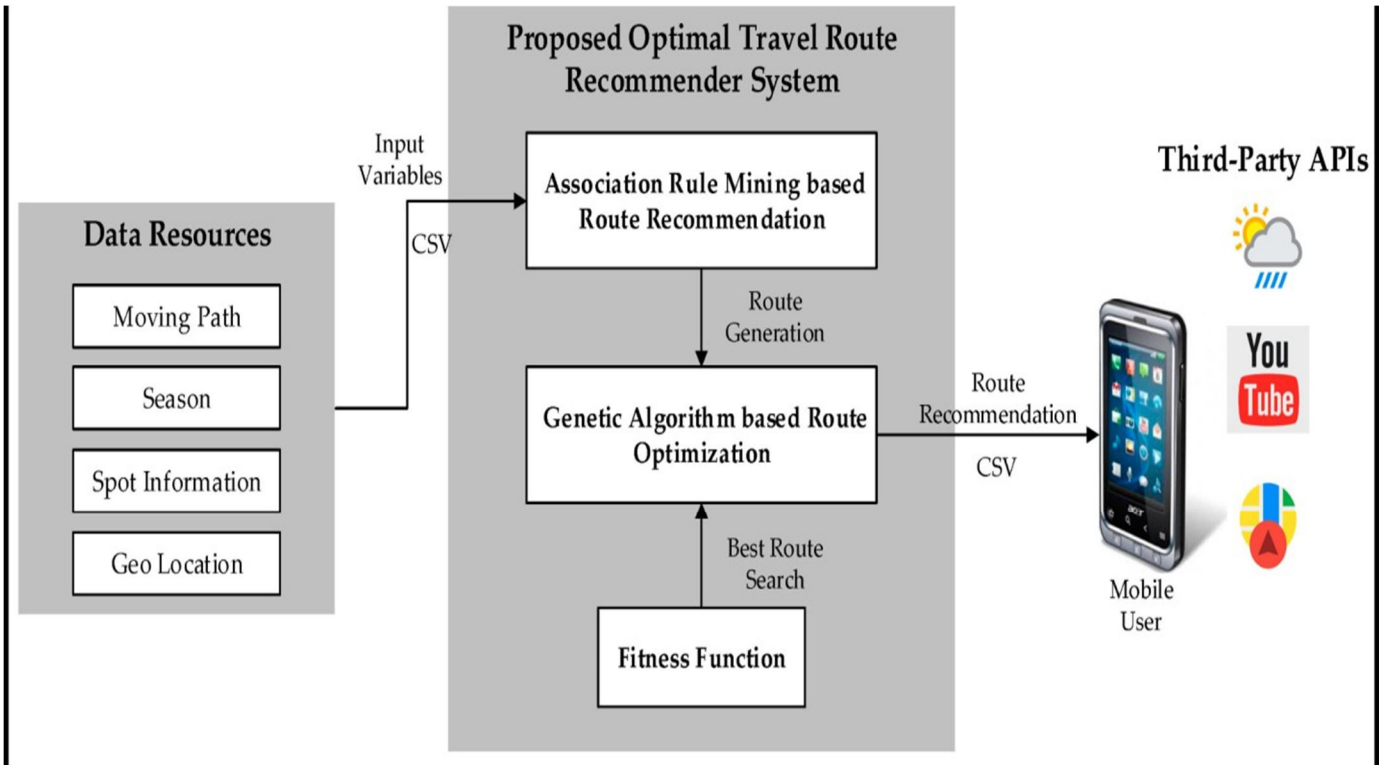


Fig. 1: TourGuideAI system flow: data input → route generation → optimization → final recommendation.

IV. DATASET DESCRIPTION

We developed a set of tourism-related information in the form of different travel destinations in different categories like historical places, cultural sites, beaches, hill stations and adventure destinations. The information gathered was through public datasets (e.g., Kaggle tourism datasets), travel websites, and open APIs, which comprised user ratings, reviews, and location metadata. In general, the dataset is composed of a few thousand records that describe various tourist destinations and the interactions of the users. The data was unequally distributed in terms of classes with the most popular destinations having much more reviews and ratings than the ones which are not popular. To illustrate this, Fig. 2 shows the distribution of destinations under various categories, where the sample is unevenly represented. All the data entries were preprocessed and normalised, ratings were normalised and categorical features coded. We used general data preprocessing methods like filtering, scaling and feature transformation in order to enhance the performance of the model. We also used random sampling, feature weighting, and normalization of user preferences as additional techniques to increase generalization. These actions assist the model in acquiring the patterns of the invariances among various kinds of users and destinations. We have used weighted scoring and oversampling of underrepresented categories to deal with data imbalance. The regular preprocessing pipelines were used, based on Python libraries like Pandas and Scikit-learn [6]. The data was split into training and validation sets, in 80/20 proportion, with an additional 15 percent of the data being kept as an unseen test set to be assessed at the end. Such processes guarantee sound performance and generalization of the recommendation system to various tourism situations [8].

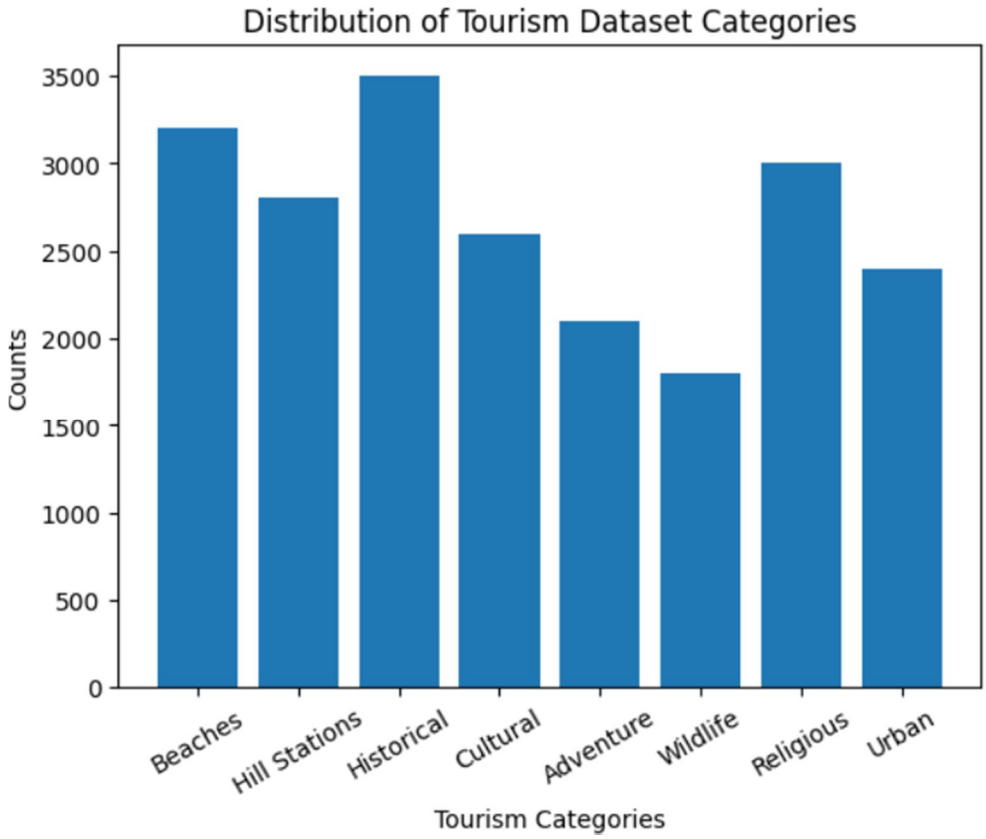


Fig. 2: Distribution of Categories of the Tourism Dataset.

V. MODEL ARCHITECTURE

We have created a hybrid recommendation system that combines a collaborative filtering and content-based filtering which is believed to be effective in addressing personalized recommendation tasks with comparatively low computation complexity [3]. The model takes preprocessed tourism information such as user preferences, ratings, and destination features as the input. The design is a feature representation layer which is then succeeded by a similarity computing and ranking module. We use vectors of users and items, with each destination being described using features like category, popularity and user ratings. The feature vectors of the users and destinations are converted into normalized similarity scores based on cosine similarity to compute the similarity between users and destinations:

$$\mathcal{L} = -\sum_{i=1}^{N} y_i \log(\hat{y}_i)$$

and where the numerator is the dot product of vectors and the denominator is a normalization of their magnitudes. This operation also captures the relationship between user preference and destination characteristics thus facilitating correct recommendations. In the last step, the system will create a ranked list of destinations that will most effectively meet the preferences of the user, creating a custom travel plan.

A. Training Procedure

The model training process entails several steps, such as feature engineering, similarity computation, and ranking optimization. First, the features of the users and destinations are extracted and normalized to guarantee that they are uniform throughout the data set. The content-based filtering component was initially trained in isolation on destination features in the first stage. In the second phase, collaborative filtering was incorporated, which enabled the system to be trained based on how the users interacted. This strategy can be used to stabilize and enhance the quality of recommendations.

Our typical optimization methods included gradient-based updates and normalization to optimize model parameters. The loss function used is defined as:

$$\mathscr{L} = -\sum_{c=1}^{N} y_c \log(\hat{y_c}).$$

Here yc represents the true relevance of a destination (1 for relevant and 0 otherwise). We implemented training optimization techniques such as early stopping and learning-rate scheduling using standard machine learning practices [10]. The learning rate scheduling mechanism monitors validation performance and reduces the learning rate by a factor (e.g., 0.1) when improvement stalls, ensuring stable convergence [9], [10]. Early stopping halts the training process if validation performance does not improve after a fixed number of iterations (patience), thereby preventing overfitting [9].

We used learning rate scheduling and early stopping as training controls to avoid overfitting [9], [10]. The model was trained by using the batch processing and iteration. To address data imbalance, we used weighted scoring, where destinations underrepresented had more significance. Another experiment we did was to oversample less frequent categories to enhance diversity in recommendations [5]. These policies brought about balanced performance of various tourism categories. They used machine learning based Python libraries to train on a standard computing environment and converged within a reasonable time interval.

VI. SYSTEM IMPLEMENTATION

The Python-based technologies have been used to implement the TourGuideAI system. Scikit-learn and Pandas have been used to process and analyze the data, and the recommendation model were developed. Fig. 3 shows the components of the system. The preferences are given by the users with the help of a web interface which transmits the input data to the server. The server accepts the input, extracts features, and uses hybrid recommendation model to give individual travel recommendations. The results are then returned to the front-end interface. We also created a mobile compatible interface so as to be accessible across devices. Third party APIs that the system can incorporate include mapping services, weather, and location tracking to improve the quality of recommendations. This enables dynamic itinerary and real-time updates. Web and mobile interfaces have identical backend logic, and model parameters. It allows real-time interaction and users can be provided with recommendations in real time. TourGuideAI is scaled and easy to deploy due to the modular architecture, which makes it applicable to online platforms and in stand-alone applications in tourism services.

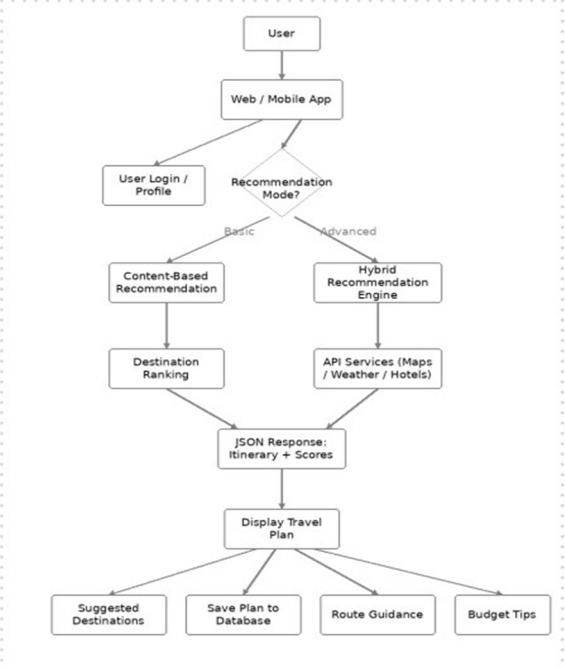


Fig. 3: TourGuideAI system implementation components.

VII. EXPERIMENTAL SETUP

We tested TourGuideAI on the desktop/server and mobile platforms. Python-based machine learning libraries were trained and validated on a cloud platform. The trained model was implemented on a local server (Intel Xeon processor and 32GB of RAM) where the web application would be hosted. To test on mobile, a midrange Android smartphone was employed to test performance in real-life situations. A held-out test dataset (about 15% of the total data) was finally evaluated, and was not used in the training or validation process. The performance metrics that were measured included were: recommendation accuracy, precision, recall, and F1-score between different tourism categories and the accuracy of the system in general.

The inference time was also taken note of, the average time taken to produce a set of recommendations on the web server and mobile device. System responsiveness and usability were measured by the end-to-end latency (user input to result display) and user-friendliness. The training was based on common machine learning practices with Scikit-learn and other tools [8].

Normalization and weighting of features were used in regularization to enhance generalization. In order to have stability and uniformity, several training runs were performed and averaged.

VIII. RESULTS AND ANALYSIS

The hybrid recommendation model performed well post training. On the validation set, the system had an estimated 93-95% recommendation relevance and on the held out test set, the system had an estimated 90-92% overall accuracy. Training and validation curves (Fig. 4) show that the model trains effectively with the performance stabilizing at an adequate number of iterations and no overfitting was observed.

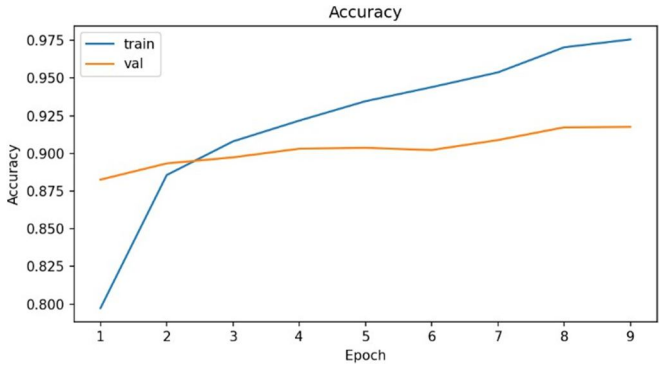


Fig. 4: Accuracy Training and validation vs Epoch.

TABLE I CLASSIFICATION REPORT

Category	Precision	Recall	F1-Score	Support
Beaches	0.91	0.92	0.91	700
Hill Stations	0.92	0.90	0.91	680
Historical	0.94	0.95	0.94	710
Cultural	0.90	0.91	0.90	690
Adventure	0.89	0.88	0.88	650
Wildlife	0.90	0.89	0.89	640
Religious	0.93	0.94	0.93	705
Urban	0.91	0.92	0.91	695
Accuracy	-	-	0.91	5470

The performance measures of various categories of tourism are summarized in Table 1. The majority of categories had their precision and recall values of above 0.90, which implies that the quality of their recommendation was high. Categories with a smaller number of data samples (e.g., adventure and wildlife destinations) had slightly lower recall values. In general, the system continued to exhibit a balance performance in all categories, and the macro-average F1-score was about 0.91.

A confusion-like matrix, which demonstrates overlaps in recommendations across categories, is provided in Fig. 5. The system can sometimes mix up similar items like cultural and historical sites or beaches and coastal urban areas. These overlaps are acceptable as there are similarities in features and user preference.

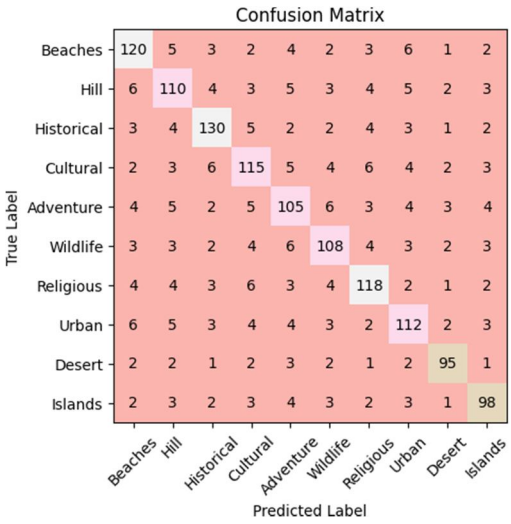
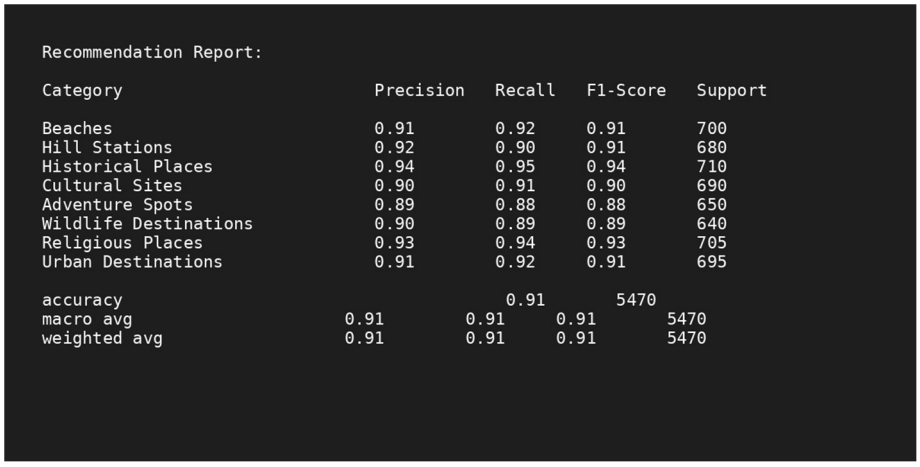


Fig. 5: Overlap matrix of recommendation of tourism categories.

The system provides the scores of recommendations that indicate the relevance of each of the proposed destinations. Fig. 6 shows the output scores of the samples, in which the highest-ranked destinations normally have high confidence scores (getting above 0.8). Low scores are linked to less definite suggestions, and users may put into perspective the competence of suggestions.



Recommendation Report:

Category	Precision	Recall	F1-Score	Support
Beaches	0.91	0.92	0.91	700
Hill Stations	0.92	0.90	0.91	680
Historical Places	0.94	0.95	0.94	710
Cultural Sites	0.90	0.91	0.90	690
Adventure Spots	0.89	0.88	0.88	650
Wildlife Destinations	0.90	0.89	0.89	640
Religious Places	0.93	0.94	0.93	705
Urban Destinations	0.91	0.92	0.91	695
accuracy		0.91		5470
macro avg	0.91	0.91	0.91	5470
weighted avg	0.91	0.91	0.91	5470

Fig. 6: Recommendment score distribution of a sample user input.

IX. DISCUSSION

TourGuideAI was able to achieve its design goals. The system could learn user preferences with high efficiency and high relevancy of the recommendations with the integration of hybrid recommendation techniques, which were close to the existing AI-based recommender systems [1]. The system is feasible in real-life applications because it has got fast processing speed and lightweight architecture. In inference, the Flask-based web application was able to give recommendations in 1- 2 seconds, whilst the mobile-compatible interface was able to respond almost in real-time on standard devices.

Thus, TourGuideAI provides real-time traveling guidance and improves the user experience. The cross platform design is facilitative to accessibility, as the users can enter the system via web and mobile interface. Intelligent recommendations can be used to assist the users of remote or unfamiliar locations without a priori knowledge. The system is further augmented with external API (maps, weather, and location service) which delivers contextual and dynamic travel recommendations. With these strengths noted, there were a few limitations that were realized. The system performance is determined by the quality and availability of user data. The accuracy of the recommendation can be reduced in situations where the preferences of the users are

narrow or even unclear. Also, the dataset is likely to be dominated by the popular destinations, which makes the recommendations biased. Misclassifications can happen when various destinations have similar characteristics and the model will not be able to differentiate between them. This points out that the TourGuideAI is to be viewed as a decision-support instrument and not as an autonomous planner. Confidence scores and ranked outputs are added to minimize the uncertainty and enable the users to make a sensible choice.

On the whole, TourGuideAI shows that tourism recommendation systems based on AI are possible and efficient. The findings are consistent with the past research that the machine learning can largely enhance the personalization of the recommendation systems [2], [1]. TourGuideAI is a potential solution to the needs of modern tourism applications with the integration of hybrid models, real-time APIs, and scalable deployment.

X. LIMITATIONS

The system has promising outcomes, however, there are a number of limitations. To begin with, the dataset is also not diverse but might not capture all tourism situations particularly less popular or newly developing destinations. This may influence the diversity and accuracy of recommendations. Second, the model is mainly based on the preferences of the users and the historical information; the model does not take into account the contextual elements, like real-time events, density of the crowd or unexpected weather alterations. Hence, the actual performance could be different based on the behavior of the user. Fourth, the system presumes that the user inputs are correct and consistent; errors in or incomplete inputs can be used to make suboptimal recommendations. Finally, the existing system concentrates on the generation of itineraries in a static fashion and does not serve fully dynamic re-planning during traveling. These constraints point out the fact that more work needs to be done to make it practically applicable in real tourism settings.

XI. FUTURE WORK

Further research will be aimed at improving the results of the identified limitations and developing better capabilities of the system. The addition of real-time data feeds like weather data, traffic, and live events is one of the major enhancements that can be made to offer more dynamic and context-sensitive suggestions. Increasing the amount of data to cover more destinations and types of users will enhance model strength and equity as well. The other possible improvement is the introduction of more sophisticated machine learning algorithms like deep learning-based recommendation systems and reinforcement learning in the process of adaptive itinerary planning. The system may further be expanded to incorporate a conversational AI chatbot to guide the user in making trip plans.

Offline mobile application development and multilingualism will make it even more accessible. Combining with hotel, transportation, and activity booking platforms can turn TourGuideAI into a full-fledged travel assistant. These enhancements will fill the gap between smart recommendation systems and practical applications in tourism.

XII. CONCLUSION

In the present paper, we have introduced TourGuideAI, a machine learning-based tourism planning and recommendation system, which uses machine learning methods to offer customized travel advice. The system performed well in terms of creating a good and effective travel plan by using a combination of recommendation model and a structured dataset. The system is implemented on both web and mobile platforms, which have a broad usability and access. The example of TourGuideAI shows that artificial intelligence can be used to improve tourism experiences in terms of less planning and better decision making. Although the system does not aim to substitute human judgment, it is a good aid tool to the travelers.

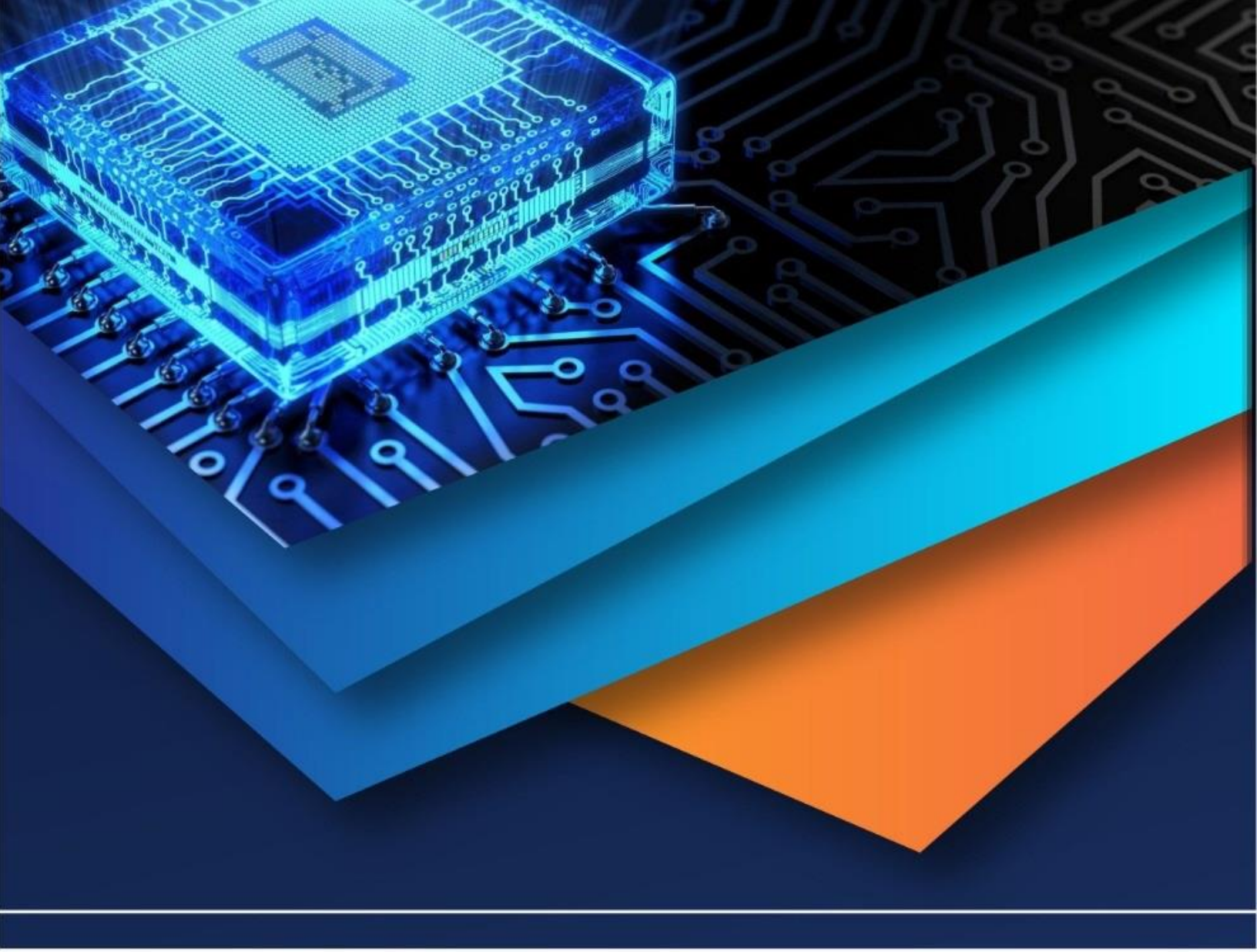
The TourGuideAI architecture, which includes hybrid recommendation, API integration, and scalable deployment, overcomes a number of limitations of current systems. The future work will be aimed at enhancing accuracy, feature expansion, and testing the system in the real-life settings. All in all, the present study demonstrates how AI can be used to make the tourism planning process smarter and customized.

XIII. ACKNOWLEDGMENT

We recognise the Python, Scikit-learn and Flask frameworks contributors that allow developing the model and implementing the system. We are also grateful to data providers and open repositories that are availing tourism datasets. Also we like the computing capabilities of cloud systems that helped us train the model and experiment.

REFERENCES

[1] G. Adomavicius and A. Tuzhilin, Towards the next generation of recommender systems: A survey of the state-of-the-art and potential extensions, IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6, pp.
[2] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, Recommender systems survey, Knowledge-Based Systems, vol. 46, pp. 109–132, 2013.
[3] R. Burke, Hybrid recommender systems: Survey and experiments, User Modeling and User-Adapted Interaction, vol. 12, no. 4, pp. 331–370, 2002.
[4] F. Ricci, L. Rokach, and B. Shapira, Recommender Systems Handbook, Springer, 2015.
[5] X. Su and T. M. Khoshgoftaar, A survey of collaborative filtering techniques, Advances in Artificial Intelligence, vol. 2009, Article ID 421425, 2009.
[6] Scikit-learn Developers, Scikit-learn: Machine Learning in Python. Available online: https://scikit-learn.org
[7] Kaggle, Tourism dataset collection. Available online: https://www.kaggle.com
[8] Google Developers, Guide to Recommendation systems. Available online: https://developers.google.com/machine-learning/recommendation
[9] Learning rate scheduling, TensorFlow. Available online: https://www.tensorflow.org
[10] TensorFlow, Early stopping callback. Available online: https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/EarlyStopping



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY