



# IJRASET

International Journal For Research in  
Applied Science and Engineering Technology



# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 13    **Issue:** V    **Month of publication:** May 2025

**DOI:** <https://doi.org/10.22214/ijraset.2025.70873>

[www.ijraset.com](http://www.ijraset.com)

Call:  08813907089

E-mail ID: [ijraset@gmail.com](mailto:ijraset@gmail.com)

# Smart City Exploration: A Reviews-Based Place Recommendation System

Ayush Meshram<sup>1</sup>, Sushant Langhi<sup>2</sup>, Kartik Kirve<sup>3</sup>, Jatin Yadav<sup>4</sup>, P.T.Kohok<sup>5</sup>

<sup>1, 2, 3, 4</sup>Student, <sup>5</sup>Assistant Professor, Department of Computer Engineering, Pune Institute of Computer Technology, Pune, India

**Abstract:** *The expansion of cities has created a demand for advanced systems that can aid residents and tourists in navigating urban areas. This paper introduces a recommendation system that aims to improve the city exploration experience by utilizing a review-based approach. By analyzing user-generated reviews, sentiment analysis, and location/place data, the system suggests points of interest, enhancing decision-making for users [3] [4]. The system uses machine learning methods, such as collaborative filtering and natural language processing, to analyze information. By considering both location-specific details and the sentiment conveyed in reviews, the approach guarantees that recommendations are suited to the specific needs and preferences of the user.*

**Index Terms:** Recommendation System, Sentiment Analysis, Machine Learning, Collaborative Filtering, Natural Language Processing, Personalized Recommendations, User Reviews.

## I. INTRODUCTION

### A. Cities and Urbanization

In the era of digital travel planning, users increasingly rely on platforms like TripAdvisor, Google Maps, and Lonely Planet to explore places and build itineraries. While these platforms offer general suggestions and popular lists of attractions, they fall short in delivering truly personalized, context-aware recommendations. Our project addresses that gap by building a smart place recommendation system that empowers users to discover destinations tailored to their specific needs. Users can explore cities or regions based on their chosen place types (e.g., nature, historical, restaurants), apply a diverse set of 20–30 filters (such as wheelchair access, parking availability, restroom facilities, pet-friendliness), and select locations of interest. What sets this system apart is the integration of sentiment analysis on real user reviews, enabling the platform to rank and recommend locations not just based on popularity, but on how positively people feel about them.

### B. Impact of Recommendation Systems for City Exploration

Recommendation systems are widely implemented across various domains such as e-commerce, entertainment, and travel, helping users navigate vast information landscapes with ease [1]. These systems can play a crucial role in simplifying decision-making by curating relevant points of interest based on individual needs. By analyzing user inputs, historical trends, and content such as reviews or ratings, these systems can enhance user satisfaction and engagement. Our recommendation system uses different methods, including collaborative filtering, content-based filtering, and hybrid approaches, to analyze data and offer personalized recommendations [5]. Using user-generated content, such as place metadata, reviews and ratings, our system can provide highly personalized and relevant recommendations [6].

As users share their opinions on the suggested places, a continuous feedback loop guarantees that the system becomes more precise and in sync with the user's preferences, resulting in a more intuitive and enjoyable experience for city exploration. Despite the abundance of apps and services, finding your way around a city can still be overwhelming, particularly when taking into account the wide range of user preferences and the expansive nature of urban spaces. While traditional maps and static guides provide basic navigation, they fail to consider individual interests, making them less valuable for those seeking a personalized experience. For visitors, tailored suggestions can assist them in maximizing their limited time in the city, while for locals, the system can propose activities or services that improve their everyday life.

## II. METHODOLOGY

The methodology follows a systematic approach to process and analyze raw data, detect fake reviews, and generate accurate recommendations. The entire workflow involves several stages, starting from data collection and cleaning to implementation of the final recommendation system.

### A. Data Collection

The dataset used in this study was compiled through a combination of the Google Maps API and web scraping techniques, and was stored [14]. It includes key information such as placenames, business categories, user reviews, ratings, and other associated metadata.

Source: Google Maps (via Outscraper API)

Collected Attributes:

- Basic details: name, type, subtypes, rating, user\_ratings\_total
- Location info: borough, address, coordinates
- Reviews: reviews, review\_score, review\_link, review\_time
- Metadata: Nested fields like about (e.g., parking, wheelchair access, toilet availability)

### B. Data Cleaning and Filtering

#### 1) Initial Cleaning:

- Dropped redundant columns (e.g., address, coords, description) that were either noisy or not useful for model training.
- Removed rows with missing or null values in essential fields such as reviews, subtypes, and rating.
- Converted all text data to lowercase to ensure consistency in processing.

Toolkits Used: pandas, numpy

#### 2) Filtering for Review-Rich Entries:

To ensure sentiment analysis was meaningful, we filtered out places with fewer than 100 reviews. This threshold was empirically chosen to maintain a balance between data quality and sample size.

- Rationale: A higher number of reviews increases confidence in aggregate sentiment polarity and user satisfaction.
- Impact: Reduced dataset size while improving the reliability of sentiment-based recommendations.

#### 3) Subtypes Text Preprocessing:

The subtypes column (e.g., "vegetarian restaurant", "historic site") was critical for understanding semantic content.

Steps:

- Tokenized and lowercased the text.
- Removed punctuation and numeric values.
- Eliminated stop words using NLTK.
- *Optional:* Performed stemming/lemmatization for improved vector representation (experimented with both).

Example:

- *Raw:* ["Vegetarian Restaurant", "Family-Friendly Spot"]
- *Cleaned:* ["vegetarian", "restaurant", "family", "friendly", "spot"]

#### 4) Word Embedding Generation (Word2Vec):

Trained a Word2Vec model using the cleaned subtypes tokens to generate 50-dimensional word vectors.

For each place, the average of its token embeddings was computed to obtain a single vector representation.

Library Used: `gensim.models.Word2Vec`

Model Parameters:

`Word2Vec(sentences=tokenized_data, vector_size=50, window=5, min_count=1, sg=1)`

These embeddings served as numeric features for the Random Forest classifier.

### C. Text Preprocessing and Word2Vec Embeddings

The textual data, primarily composed of user reviews, undergoes several preprocessing steps to prepare it for machine learning tasks. First, all text is converted to lowercase to ensure uniformity, and non-alphabetic characters such as punctuation marks and numbers are removed. Tokenization is then applied, breaking each review into individual words. Common stop words (e.g., "and", "the", "is") are eliminated as they do not provide meaningful insights for semantic understanding. Following this, stemming or lemmatization may be employed to reduce words to their base forms, enhancing consistency across the corpus. Once the text is cleaned, the Word2Vec model is trained on the preprocessed data. Word2Vec generates dense vector embeddings for each word by analyzing its surrounding context within reviews [16]. This approach captures the semantic relationships between words and transforms unstructured textual data into a meaningful numerical format, enabling effective input for subsequent stages such as place type classification and sentiment analysis.

#### D. Category Classification Model

1) *Label Encoding of Place Types*: Mapped string-based type labels (e.g., "restaurant", "attraction") to numeric values using LabelEncoder. This transformed the output variable into a format suitable for machine learning algorithms.

2) *Model Training using Random Forest*:

- Split data into training and test sets using an 80:20 ratio.
- Trained a Random Forest Classifier using Word2Vec vectors as input features.
- Classifier Used: `sklearn.ensemble.RandomForestClassifier`
- Key Hyperparameters:

`n_estimators=100, max_depth=None, random_state=42`

3) *Evaluation of Classifier*: Measured accuracy, precision, and recall on the test dataset. Sample test predictions were manually inspected to validate semantic consistency.

Expected Accuracy: 80–90% (varies by place diversity and embedding quality)

#### E. Fake Review Removal and Sentiment Analysis

1) *Fake and Redundant Review Detection*: Applied heuristic rules to clean the review dataset:

- Heuristic 1: Excessive Sentiment—Reviews containing overly positive language and repetitive keywords (e.g., "best", "awesome") were flagged as possibly inauthentic.
- Heuristic 2: Duplication—Reviews with identical titles and content were considered redundant and removed.
- Heuristic 3: Outliers—Reviews with unusually short (<5 words) or excessively long (>300 words) text were flagged as unnatural.

Libraries Used: `TextBlob`, `re`, and `nltk`

2) *Sentiment Polarity Scoring*: Used `TextBlob`, a lexicon-based sentiment analyzer, to calculate polarity for each review [3].

Polarity Range:  $[-1, 1]$  — Negative (below 0), Neutral (0),

Positive (above 0)

Each review was labeled as:

- positive if `polarity > 0.1`
- negative if `polarity < -0.1`
- neutral otherwise

This generated a new field `sentiment_score`, which was later used to rank places based on user perception.

#### F. Feature Extraction from Nested JSON (about)

1) *Flattening and Parsing JSON Fields*: The `about` field (e.g., `{'wheelchair_accessible': True, 'parking': False}`) was parsed using `json.loads`. Individual boolean columns were created for each key, such as:

- `has_parking`
- `has_toilets`
- `wheelchair_accessible`

2) *Handling Missing Values*: Any missing or None entries in the `about` field were set to `False`, interpreting the absence of information as unavailability of that feature.

Purpose: Enabled boolean filtering for personalized recommendations (e.g., show only places with wheelchair access).

#### G. System Design

The final recommendation system is deployed as a Flask-based web application. User authentication features, including Google OAuth, are integrated to allow secure access to the system. The recommendation system leverages the processed data to suggest relevant places to users based on their preferences. The application also integrates Leaflet.js for interactive map visualization, allowing users to view recommendations on a map and filter them according to different criteria.

### III. IMPLEMENTATION

#### A. Data Collection

Objective: Collect data from various sources to build a comprehensive dataset for analysis.

#### TechnologiesUsed:

- Python:Forscriptinganddatamanipulation.
- Pandas:Fordatahandlingandanalysis.
- ExcelFiles:Asprimarydatasource.

Process: Data is collected from multiple Excel files: `outscraper_first.xlsx`, `data_pos.xlsx`, and `data_neg.xlsx`. The dataset includes information about places, reviews, and additional metadata for analysis.

#### B. Data Preprocessing

Objective: Clean and preprocess the dataset to make it suitable for analysis and modeling.

#### TechnologiesUsed:

- Pandas:For datamanipulationandcleaning.
- NumPy:For numericaloperations.

#### Process:

- LoadData: Load the dataset from Excel files.
- CheckforNullValues: Identify and handle columns with null values.
- Remove Unnecessary Columns: Drop columns that are not relevant for the analysis.
- FilterRows: Remove rows with null values in critical columns.
- SaveCleanedData: Save the cleaned dataset to a new Excel file (`data500.xlsx`).

#### C. Sentiment Analysis

Objective: Perform sentiment analysis on reviews to classify them as positive, negative, or neutral.

#### TechnologiesUsed:

- TextBlob:For sentimentanalysis.
- Pandas:For datamanipulation.

#### Process:

- LoadData: Load the dataset containing reviews.
- Sentiment Analysis: Use TextBlob to calculate the sentiment score of each review [4].
- Classify Reviews: Classify reviews based on their sentiment score.
- SaveProcessedData: Save the processed dataset with sentiment scores to new Excel files (`pos_processed_reviews.xlsx`, `neg_processed_reviews.xlsx`).

#### D. Fake Review Detection

Objective: Detect and remove fake reviews based on various indicators.

#### TechnologiesUsed:

- Pandas:For datamanipulation.
- TextBlob:For sentimentanalysis.

#### Process:

- LoadData: Load the processed review datasets.
- Detect Fake Reviews: Detect reviews with excessive sentiment, duplicate reviews, and reviews with suspicious lengths.
- CombineIndicators: Combine all fake review indicators to flag fake reviews.
- Remove Fake Reviews: Remove flagged fake reviews from the dataset.
- SaveCleanedData: Save the cleaned dataset without fake reviews to new Excel files (`cleaned_pos_reviews.xlsx`, `cleaned_neg_reviews.xlsx`).

#### E. Data Expansion and Transformation

Objective: Expand and transform JSON data into a tabular format.

#### TechnologiesUsed:

- Pandas:For datamanipulation.

- JSON:ForhandlingJSONdata.

Process:

- LoadData: Load the dataset containing JSON data (e.g., about column).
- ExpandJSONData: Convert the JSON data into a dictionary and flatten it.
- Merge with Place ID: Merge the expanded data with the place\_id column.
- SaveTransformedData: Save the transformed data to a new Excel file (about\_expanded.xlsx).

#### F. Data Storage

Objective: Insert the cleaned and transformed data into a MySQL database.

Technologies Used:

- MySQL: For database storage.
- Python(mysql-connector): For database connectivity.
- Pandas: For data manipulation.

Process:

- Connect to MySQL: Establish a connection to the MySQL database.
- Create Table: Create a table dynamically based on the DataFrame structure.
- Insert Data: Insert the DataFrame data into the MySQL table, ensuring proper handling of NaN values.

#### G. Recommendation Engine

Objective: Develop a recommendation system to suggest places based on user preferences.

Technologies Used:

- Python: For scripting and data manipulation.
- Pandas: For data handling.
- Scikit-Learn: For machine learning algorithms.
- Gensim(Word2Vec): For text vectorization.

Process:

- Text Preprocessing: Convert text data into numerical vectors using Word2Vec [16] [17].
- Train Classifier: Train a Random Forest classifier on the training data [?].
- Evaluate Model: Evaluate the model's accuracy on the test data.
- Recommendation Logic: Use the trained model to recommend places based on user preferences and filters.

#### H. Web Application

Objective: Develop a Flask web application for user interaction and recommendations.

Technologies Used:

- Flask: For web application development.
- HTML/CSS/JavaScript: For frontend development.
- Bootstrap: For responsive design.
- Leaflet: For map visualization.
- Google OAuth: For user authentication.

Process:

- Initialize Flask App: Set up the Flask application and define routes.
- User Authentication: Implement signup, sign-in, and Google OAuth functionalities.
- Recommendation System: Integrate the recommendation engine to suggest places based on user preferences [2].
- Map Integration: Integrate Leaflet.js for map visualization of recommended places.
- Frontend Development: Use Bootstrap for the frontend, implement JavaScript for dynamic interactions and AJAX requests.

#### IV. RESULTS AND DISCUSSION

##### A. Data Collection and Preprocessing

###### 1) Accuracy:

- **Data Sources:** The system collects data from reliable sources such as the Google Places API and pre-collected Excel files, ensuring that the dataset is comprehensive and up-to-date.
- **Data Cleaning:** Preprocessing steps, including handling null values, removing unnecessary columns, and filtering rows, ensure that the dataset is clean and relevant for analysis.
- **Sentiment Analysis:** Using TextBlob for sentiment analysis allows for accurate classification of reviews as positive, negative, or neutral, which is crucial for understanding user preferences and feedback.
- **Fake Review Detection:** Detecting and removing fake reviews based on various indicators ensures that the dataset remains reliable and free from biased or manipulated data.

###### 2) Effectiveness:

- **Comprehensive Data:** The combination of data from the Google Places API and pre-collected Excel files provides a rich dataset for analysis.
- **Clean Data:** The preprocessing steps ensure that the data is clean, which in turn improves the accuracy of the recommendation engine.
- **Sentiment Analysis:** Accurate sentiment analysis helps in understanding user preferences and feedback, ensuring relevant recommendations.
- **Fake Review Detection:** Removing fake reviews ensures that the recommendations are based on genuine and reliable data.

##### B. Recommendation Engine

###### 1) Accuracy:

- **Text Vectorization:** Using Word2Vec for text vectorization ensures that text data is accurately converted into numerical vectors, which is essential for machine learning algorithms.
- **Machine Learning Model:** The Random Forest classifier is trained on the preprocessed data to classify places by type, with evaluation metrics such as accuracy score used to assess its performance.
- **Collaborative Filtering:** Collaborative filtering techniques ensure personalized recommendations based on user preferences and past interactions.

###### 2) Effectiveness:

- **Personalized Recommendations:** The recommendation engine generates personalized suggestions based on user preferences and interactions.
- **Accurate Classification:** The Random Forest classifier accurately classifies places based on their type, improving the relevance of the recommendations.
- **Collaborative Filtering:** The use of collaborative filtering enhances the recommendations by considering the preferences of similar users [13].

##### C. User Feedback Analysis

###### 1) Accuracy:

- **Sentiment Analysis:** TextBlob is used to classify user feedback as positive, negative, or neutral, ensuring that feedback is accurately analyzed.
- **Feedback Integration:** Integrating user feedback into the recommendation engine helps improve future recommendations.

###### 2) Effectiveness:

- **Continuous Improvement:** The system continuously refines its recommendations based on user feedback, ensuring that suggestions remain relevant and accurate over time.
- **User Satisfaction:** Accurate sentiment analysis of user feedback aids in understanding user satisfaction and areas for improvement, enhancing the system's overall effectiveness.

**D. Web Application**

**1) Accuracy:**

- **User Authentication:** The implementation of secure user signup, sign-in, and Google OAuth ensures safe data storage and access.
- **Map Integration:** Integrating Leaflet for map visualization ensures accurate display of recommended places on the map.

**2) Effectiveness:**

- **User-Friendly Interface:** The Flask web application provides a simple and intuitive interface for interaction, improving the user experience.
- **Dynamic Interactions:** JavaScript and AJAX enable dynamic interactions, making the web application responsive and efficient.
- **Map Visualization:** Leaflet provides interactive map visualizations that help users explore recommended places effectively.

**E. Comparison with Existing Systems**

**1) Existing Systems:**

- Popular recommendation systems such as Google Maps, Yelp, and TripAdvisor use advanced algorithms and large datasets to provide recommendations.
- Many systems rely on collaborative filtering and sentiment analysis to improve recommendation accuracy [7] [9].

**2) Implemented System:**

- **Comprehensive Data Collection:** Data from the Google Places API and Excel files ensures the dataset is comprehensive and up-to-date.
- **Accurate Data Preprocessing:** The preprocessing steps, including sentiment analysis and fake review detection, ensure the data is reliable.
- **Effective Recommendation Engine:** Machine learning and collaborative filtering ensure personalized and relevant recommendations.
- **User Feedback Analysis:** Continuous integration of user feedback enhances the system's effectiveness.
- **User-Friendly Web Application:** The Flask web application enhances user experience through an intuitive interface and dynamic interactions.

**F. Screen Shots**

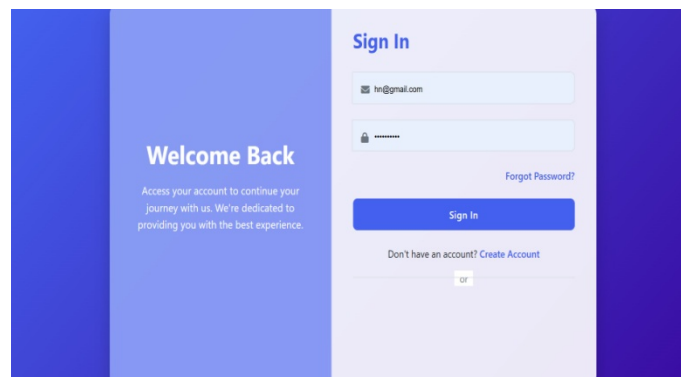


Figure 1: Login webpage

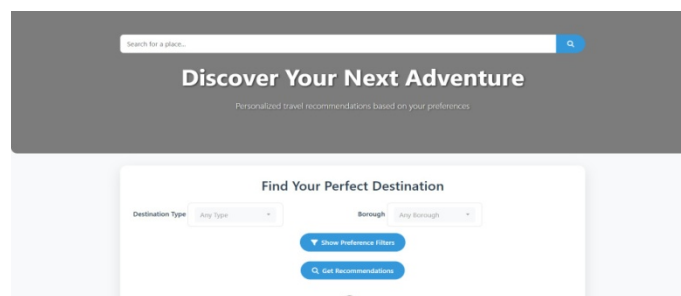


Figure 2: Dashboard view

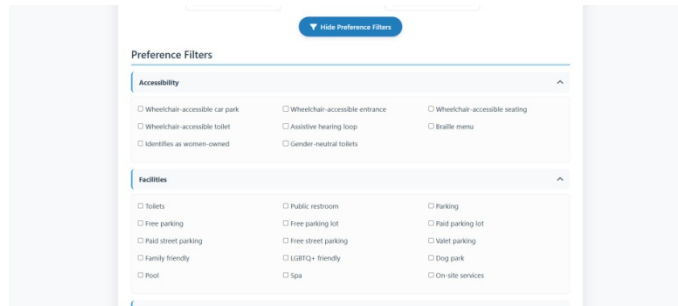


Figure3:Filterssection

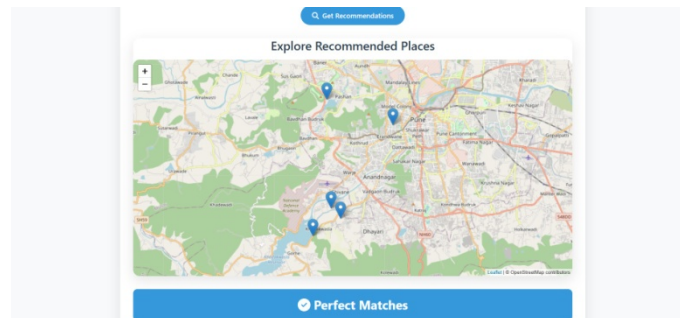


Figure4:LocationDisplayonmap

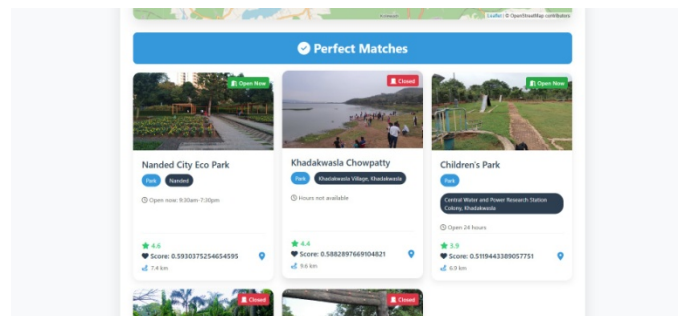


Figure5:ActualRecommendation

## V. CONCLUSION

In this implementation, we developed a comprehensive travel recommendation system that leverages advanced data preprocessing, machine learning algorithms, and user feedback analysis to provide personalized and relevant recommendations. The system architecture includes data collection from reliable sources, accurate data preprocessing, an effective recommendation engine, and a user-friendly web application.

### A. Summary of Implementation

#### 1) Data Collection and Preprocessing::

- Data Sources: Data was collected from the Google Places API and pre-collected Excel files, ensuring a comprehensive and up-to-date dataset.
- Preprocessing Steps: The dataset underwent rigorous preprocessing, including handling null values, removing unnecessary columns, sentiment analysis using TextBlob, and fake review detection.
- Text Vectorization: Text data was converted into numerical vectors using Word2Vec, making it suitable for machine learning algorithms.

2) *RecommendationEngine::*

- Machine Learning Model: A Random Forest classifier was trained to classify places based on their type, ensuring accurate classification.
- Collaborative Filtering: Collaborative filtering techniques were employed to provide personalized recommendations based on user preferences and past interactions.
- User Feedback Analysis: User feedback was continuously integrated into the recommendation engine to improve the accuracy and relevance of future recommendations.

3) *WebApplication::*

- User Authentication: User signup, sign-in, and Google OAuth functionalities were implemented to ensure secure user data storage and access.
- Map Integration: Leaflet was integrated for interactive map visualization of recommended places.
- User-Friendly Interface: The Flask web application provided a responsive and user-friendly interface for interaction and visualization, enhancing the overall user experience.

B. *Improvements Over Existing Systems*

- 1) *Comprehensive Data Collection::* The system collects data from multiple reliable sources, ensuring a rich and up-to-date dataset. Existing systems often rely on a single data source, which may limit the comprehensiveness of the dataset.
- 2) *Accurate Data Preprocessing::* The preprocessing steps, including sentiment analysis and fake review detection, ensure that the dataset is clean and reliable. Existing systems may not employ such rigorous preprocessing steps, leading to potential biases or inaccuracies in the dataset.
- 3) *Effective Recommendation Engine::* The recommendation engine uses a combination of machine learning algorithms and collaborative filtering techniques to provide personalized and relevant recommendations. Existing systems may rely solely on collaborative filtering, which may not capture the nuances of user preferences as effectively [7].
- 4) *Continuous User Feedback Analysis::* The system continuously improves its recommendations based on user feedback, ensuring that the recommendations remain relevant and accurate over time. Existing systems may not integrate user feedback as effectively, leading to potential stagnation in recommendation quality.

C. *Future Work and Enhancements*

- 1) *Advanced Machine Learning Techniques:* Future work could explore advanced machine learning techniques such as deep learning and reinforcement learning to enhance the accuracy and relevance of recommendations [10].
- 2) *Real-Time Data Updates:* Integrating real-time data updates from the Google Places API to ensure that the recommendations are always up-to-date. Dynamic dataset and recommendation updates could be implemented.
- 3) *Enhanced User Feedback Analysis:* Implement more advanced techniques for user feedback analysis, such as NLP, to better understand user preferences and feedback [11][12].
- 4) *Scalability and Performance:* Optimize the system for scalability and performance to handle a large number of users and recommendations efficiently. Implement load balancing and caching mechanisms to improve responsiveness and efficiency.
- 5) *User Interface Enhancements:* Enhance the user interface to provide a more intuitive and engaging user experience. Implement additional visualization tools and interactive features to help users explore recommendations more effectively.

## REFERENCES

- [1] Pu, Zihao & Du, Hongyu & Yu, Sizhe & Feng, Duanyu. (2020). Improved Tourism Recommendation System. 121-126.10.1145/3383972.3384074.
- [2] Kawai, Yukiko & Zhang, Jianwei & Kawasaki, Hiroshi. (2009). Tour recommendation system based on web information and GIS. 990 - 993.10.1109/ICME.2009.5202663.
- [3] Osman, Nurul. (2020). Contextual Sentiment Based Recommender System to Provide Recommendation in the Electronic Products Domain. International Journal of Machine Learning and Computing. 9.10.18178/ijmlc.2019.9.4.821.
- [4] Gong, Y. (2024). Research on recommendation algorithm based on user sentiment analysis. Proceedings of the 4th International Conference on Signal Processing and Machine Learning, 84-91.
- [5] Zhang, H.-R., Min, F., He, X., & Xu, Y.-Y. (2015). A hybrid recommender system based on user-recommender interaction. Mathematical Problems in Engineering, 2015, Article ID 145636, 11 pages.



- [6] Phan,Lan&Huynh,Hung&Huynh,Hiep.(2018).ImplicativeRating-BasedHybridRecommendationSystems.InternationalJour-nal of Machine Learning and Computing. 8. 223-228. 10.18178/i-jmlc.2018.8.3.691.
- [7] S. M. Al-Ghuribi and S. A. Mohd Noah, "Multi-Criteria Review-BasedRecommenderSystem–TheStateoftheArt,"inIEEEAccess,vol.7,pp.169446-169468,2019,doi:10.1109/ACCESS.2019.2954861.
- [8] Ko,Hyeoung&Lee,Suyeon&Park,Yoonseo&Choi,Anna.(2022).ASurvey of Recommendation Systems: Recommendation Models, Tech-niques, and Application Fields. Electronics. 11. 141. 10.3390/electron-ics11010141.
- [9] Burke, Robin. (2002). Hybrid Recommender Systems: Survey andExperiments. User Modeling and User-Adapted Interaction. 12.10.1023/A:1021240730564.
- [10] F.S,eker,"EvolutionofMachineLearninginTourism:AComprehensiveReviewofSeminalResearch",JournalofArtificialIntelligenceandDataScience, vol. 3, no. 2, pp. 54–79, 2023.
- [11] Ma, S. (2024). Enhancing Tourists' Satisfaction: Leveraging ArtificialIntelligence in the Tourism Sector. Pacific International Journal, 7(3),89–98.
- [12] G.Linden,B.SmithandJ.York,"Amazon.comrecommendations:item-to-item collaborative filtering," in IEEE Internet Computing, vol. 7, no.1, pp. 76-80, Jan.-Feb. 2003.
- [13] Zheng, Yu & Xie, Xing & Ma, Wei-Ying. (2010). GeoLife: A Collabo-rative Social Networking Service among User, Location and Trajectory.IEEE Data Eng. Bull.. 33. 32-39.
- [14] Onwuegbuzie, Anthony J., Nancy L. Leech, and Kathleen MT Collins."Qualitative analysis techniques for the review of the literature." Qual-itative Report 17 (2012): 56.
- [15] Church,KennethWard."Word2Vec."NaturalLanguageEngineer-ing 23.1 (2017): 155-162.
- [16] Ma, Long, and Yanqing Zhang. "Using Word2Vec to process big textdata." 2015 IEEE International Conference on Big Data (Big Data).IEEE, 2015.
- [17] Tan, Kian Long, et al. "RoBERTa-LSTM: a hybrid model for sentimentanalysiswithtransformerandrecurrentneuralnetwork."IEEEAccess10(2022): 21517-21525.



10.22214/IJRASET



45.98



IMPACT FACTOR:  
7.129



IMPACT FACTOR:  
7.429



# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24\*7 Support on Whatsapp)