International Journal of Multidisciplinary Engineering in Current Research - IJMEC

Volume 10, Issue 4, April-2025, http://ijmec.com/, ISSN: 2456-4265

Hotel Recommendation Using Machine Learning

Ms. Zeba Unnisa[*1], Mr. Omer Ahmed Aayez Al Jaabri[*2], Mr. Syed Taleb [*3], Mr. Mohammed Mehfooz [*4],

[*1] Assistant Professor [*2,3,4] B.E Student Dept. of CSE-AIML, Lords Institute of Engineering and Technology

Dept. of CSE-AIML, Lords Institute of Engineering and Technology

zebaunnisa03@gmail.com[*1], omer10jaabri@gmail.com [*2], syedtaleb9849@gmail.com[*3],

mohammed.mehfooz03@gmail.com*[4],

ABSTRACT

This project introduces a Hotel Recommendation System built using Machine Learning techniques and a flexible plugin-based architecture to deliver highly personalized and context-aware hotel suggestions for users. The system adopts a hybrid recommendation approach, combining content-based filtering, collaborative filtering, and natural language processing (NLP) for sentiment analysis on user reviews. This multi-faceted strategy enables the system to better understand user preferences, past behavior, and the sentiment behind user-generated content to generate accurate and meaningful hotel recommendations. A key feature of this system is its plugin integration framework, which allows seamless incorporation of external services such as location-based APIs, real-time weather updates, travel platforms (e.g., Expedia, Booking.com), and user authentication modules. These plugins enhance the system's capabilities by offering contextually relevant suggestions—such as proximity to landmarks, local weather conditions, seasonal trends, and price optimization—improving the overall user experience. The machine learning models are trained on comprehensive hotel datasets and fine-tuned using performance metrics such as precision, recall, F1-score, and RMSE. The architecture ensures modularity, scalability, and adaptability for future enhancements or domain transfers. This project demonstrates the potential of combining machine learning with modular plugin design to create intelligent, dynamic, and user-centric recommendation systems, particularly in the travel and hospitality domain.

1.INTRODUCTION

1.1 GENERAL

With the growing number of online hotel booking platforms and increasing travel demand, users are often overwhelmed with choices when selecting accommodations. Traditional systems rely on simple filters and user reviews, which do not always provide relevant or personalized recommendations. This leads to suboptimal decisions and user dissatisfaction. To overcome these limitations, Machine Learning (ML) has emerged as a powerful tool to enhance recommendation systems by learning from user data, preferences, and contextual signals. By applying ML, platforms can offer intelligent, user-centric, and dynamic hotel recommendations tailored to individual users. Additionally, the integration of external data sources—such as real-time weather, location proximity, and travel trends—can further improve recommendation quality and user experience.

1.2 PROJECT OVERVIEW

This project presents a Hotel Recommendation System powered by Machine Learning and enhanced with a plugin-based architecture for scalability and real-time data integration. It uses a hybrid approach, combining:

- Content-Based Filtering (user preferences),

• Collaborative Filtering (similar user behavior), and

• Sentiment Analysis (NLP on reviews).

Plugins allow integration with external APIs like Google Maps, weather data, and hotel platforms to deliver more context-aware and personalized recommendations. The system is trained on real hotel datasets and evaluated using standard metrics such as precision, recall, and RMSE.

1.3 OBJECTIVE

The main objectives of the project are:

• To build a personalized hotel recommendation system using a hybrid ML model.

• To enable plugin-based integration for contextual data and external APIs.

• To ensure the system is scalable, modular, and responsive to real-time inputs.

• To evaluate performance using standard machine learning metrics.

2.LITERATURE SURVEY

1. **Item-Based Collaborative Filtering** Recommendation Algorithms

Author Sarwar Et Al. (2001)

Introduced Item-Based Collaborative Filtering, Comparing Item Similarity Instead Of User Behavior. It Improved Scalability And Worked Effectively On Sparse Datasets, A Common Issue In Hotel Booking Systems.

2. **Content-Based Recommender Systems: State Of The Art And Trends**

Author Lops Et Al. (2011)

Explored Content-Based Filtering Using Item Attributes And User Profiles. Highlighted Advantages In Personalization But Noted Cold-Start Issues And Repetitive Suggestions.

3. **Hybrid Recommender Systems: Survey And Experiments**

Author Burke (2002)

Classified Hybrid Techniques (Weighted, Switching, Feature Combination) And Showed How Combining Content And Collaborative Filtering Improved Accuracy And Addressed Sparsity And Overfitting.

4. **Toward The Next Generation Of Recommender Systems**

Author Adomavicius & Tuzhilin (2005)

Surveyed State-Of-The-Art Methods And Introduced The Concept Of Context-Aware Recommendations, Leveraging Time, Location, And User Intent To Boost Personalization.

5. **Fusing Audio, Visual And Textual Clues For Sentiment Analysis**

Author Poria Et Al. (2014)

Demonstrated How Multimodal Sentiment Analysis, Especially Nlp-Based Text Analysis Of Reviews, Can Enhance Recommendation Relevance By Capturing Emotional Tone.

6. **Recommender Systems Survey**

Author Bobadilla Et Al. (2013)

Provided A Comprehensive Comparison Of Algorithms And Challenges. Advocated For Hybrid And Multi-Criteria Models To Solve Cold-Start, Scalability, And Sparsity.

7. **Introduction To Recommender Systems Handbook**

Author Ricci Et Al. (2011)

Focused On Real-World Applications In Tourism. Showed How Combining User Preferences, Behavior, And Context (E.G., Trip Purpose) Improves Hotel Recommendations.

8. **Qos-Aware Web Service Recommendation By Collaborative Filtering**

Author Zheng Et Al. (2010)

Proposed A Plugin-Based Recommendation Framework Integrating Apis For Dynamic Data Like Weather Or Location—Relevant To Hotel Systems Needing Modularity.

9. **Recommender Systems: The Textbook**

Author Aggarwal (2016)

Covered Foundational And Advanced Models, Including Deep Learning And Graph-Based Techniques. Offered Strategies To Incorporate Contextual, Social, And Temporal Data.

10. **A Hybrid Recommender System For Booking Hotels On Travel Websites**

Author Chen Et Al. (2015)

Combined User Ratings With Review Sentiment For Personalized Hotel Suggestions. Demonstrated That Hybrid Approaches Outperform Single-Method Systems In Accuracy And Adaptability.

3.SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The existing hotel recommendation systems typically rely on traditional collaborative filtering and content-based filtering techniques. These systems utilize user ratings and review data to generate recommendations, but they face several limitations

Limitations of Existing Systems:

Cold Start Problem**:** Poor recommendations for new users or hotels with limited data.

Scalability**:** Difficulty handling large datasets and user volumes effectively.

Sparsity**:** Sparse user interaction data makes meaningful recommendations harder.

Lack of Personalization**:** Failure to integrate contextual information (e.g., location, season).

Single-Method Approach**:** Reliance on one technique limits recommendation quality.

Limited Integration**:** Inability to use external data (e.g., weather, APIs) reduces adaptability.

3.2 PROPOSED SYSTEM

The proposed IDS leverages machine learning to detect brute force attacks (e.g., SSH/FTP) with higher accuracy and adaptability.

Key Features:

- **Hybrid Recommendation Engine**

Combines collaborative and content-based filtering to improve accuracy.

- **Sentiment Analysis**

Uses NLP to analyze reviews and emotional tone.

- **Context-Aware Recommendations**

Incorporates factors like location, date, and trip purpose.

- **API Integration**

Integrates external live data such as weather and maps.

- **Scalable Models**

Employs ML algorithms for large datasets.

- **User Interface**

Offers filters for budget, amenities, and preferences.

Workflow:

1. **Data Collection**

Collects user data and reviews.

2. **Data Preprocessing**

Cleans, encodes, and analyzes sentiment.

3. **Recommendation Engine**

Uses collaborative, content-based, and hybrid filtering.

4. **Real-Time Plugins**

Incorporates live data like weather and promotions.

5. **User Interaction**

Offers personalized suggestions and filters.

6. **Feedback Loop**

Updates model based on user activity.

3.2.1 ADVANTAGES

- **Highly Personalized Results** based on user context and preferences.
- **Handles Cold Start and Sparsity** better through hybrid and sentiment-driven models.
- **Dynamic and Real-Time Recommendations** using API plugins for up-to-date suggestions.
- **Greater User Satisfaction** due to relevant and diverse hotel options.
- **Modular and Scalable Architecture** allowing future integration of additional data sources.

- **Improved Accuracy and Engagement** through
machine learning and natural language insights.

4.REQUIREMENT SPECIFICATIONS

4.1 SOFTWARE REQUIREMENTS

Component	**Specification**
Operating System	Windows 10 / Linux / macOS
Programming Language	Python 3.8+
Frameworks & Libraries	Scikit-learn, Pandas, NumPy, NLTK, TensorFlow/PyTorch (optional), Flask/Django for web interface
Database	MySQL / SQLite / MongoDB
Web Technologies	HTML, CSS, JavaScript, Bootstrap (for UI)
APIs/Plugins	Google Maps API, Weather API, Sentiment Analysis APIs (e.g., TextBlob/VADER)
IDE/Tools	VS Code / PyCharm / Jupyter Notebook
Version Control	Git / GitHub
Browser Support	Google Chrome, Mozilla Firefox

4.2 HARDWARE REQUIREMENTS

• **omponent**	• **inimum Requirement**
• **rocessor (CPU)**	• ntel i5 / AMD Ryzen 5 or above
• **AM**	• inimum 8 GB (Recommended: 16 GB for ML training)
• **ard Disk**	• 56 GB SSD (Recommended: 512 GB or higher)
• **raphics (Optional)**	• VIDIA GPU (for deep learning models, if used)
• **isplay**	• 3” or higher, Full HD
• **nternet Connection**	• equired for API access and real-time plugin support

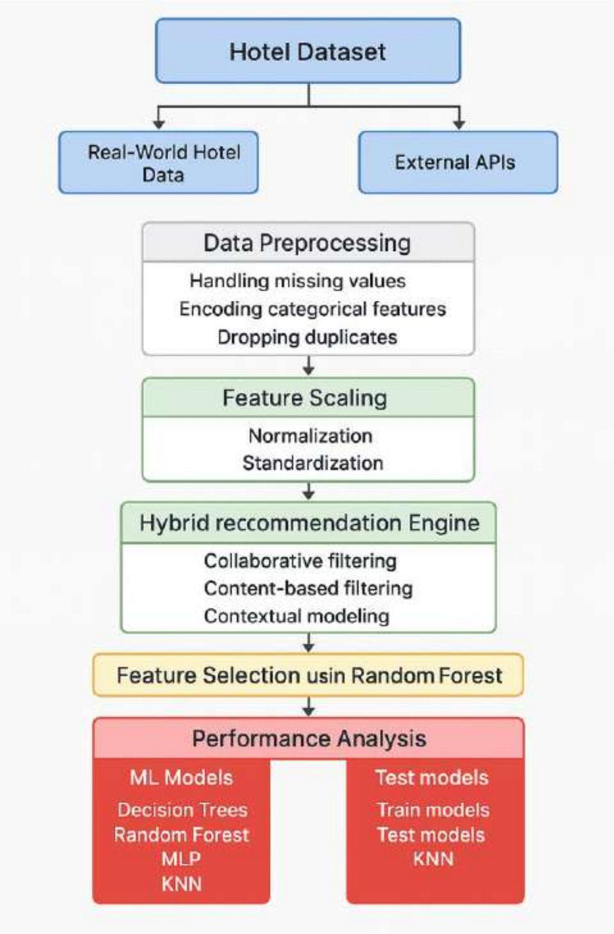
5.SYSTEM DESIGN

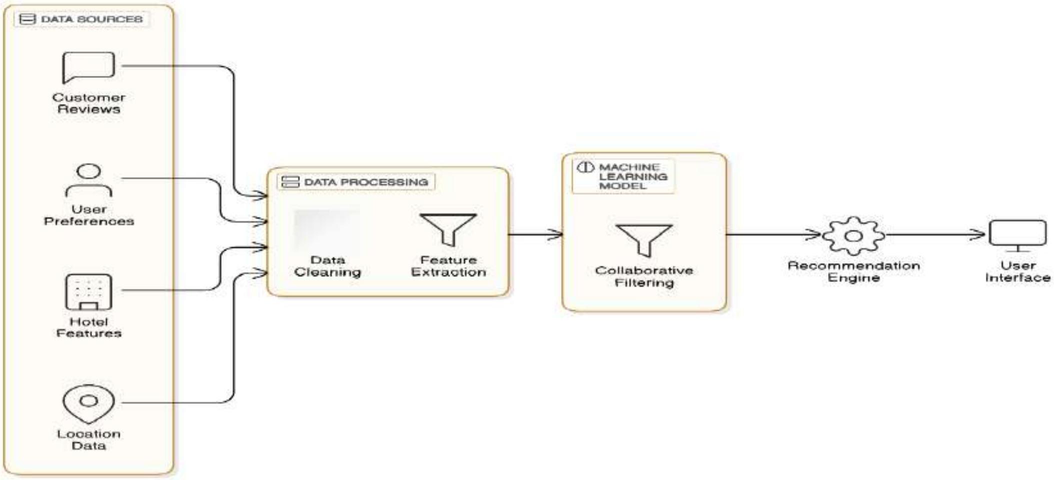
5.1 SYSTEM ARCHITECTURE

The architecture of the hotel recommendation system is designed to be modular, ensuring scalability, maintainability, and efficient data handling. It consists of several layers:

The User Interface Layer features a web-based frontend where users can input preferences, explore hotel suggestions, and interact with the system. The Application Layer oversees the business logic, executes machine learning models, manages recommendation workflows, and integrates plugins like weather and location APIs. At its core, the Recommendation Engine employs collaborative filtering, content-based filtering, sentiment analysis, and hybrid recommendation techniques using advanced ML models. The Data Layer incorporates hotel datasets, user data, reviews, and external sources like real-time APIs. Supporting this is the Database, which securely stores user profiles, hotel details, reviews, and historical logs of recommendations. Lastly, the Plugin Layer facilitates the integration of external services, such as maps and weather APIs, to enrich user experience and recommendations.

This modular design guarantees the flexibility needed for incorporating new features, APIs, and data sources as the system evolves.





5.2 UML DIAGRAMS

1. Use Case Diagram – Workflow

Shows how external users (e.g., network admin) interact with system functionalities like starting packet capture, preprocessing data, training the model, and detecting intrusions

2. Class Diagram – Workflow

Represents system structure with classes like PacketSniffer, FlowGenerator, ModelTrainer, etc., including their attributes, methods, and relationships.

3. Object Diagram – Workflow

Displays runtime instances of classes (e.g., sniffer1, flowGenA) and how data flows between them during system execution.

4. Sequence Diagram – Workflow

Depicts the order of operations: packet capture → flow generation → preprocessing → model training → intrusion detection, showing interactions over time.

5. Activity Diagram – Workflow

Illustrates the process flow from packet capture to attack detection, including decision points like "Is data sufficient?"

6. State Diagram – Workflow

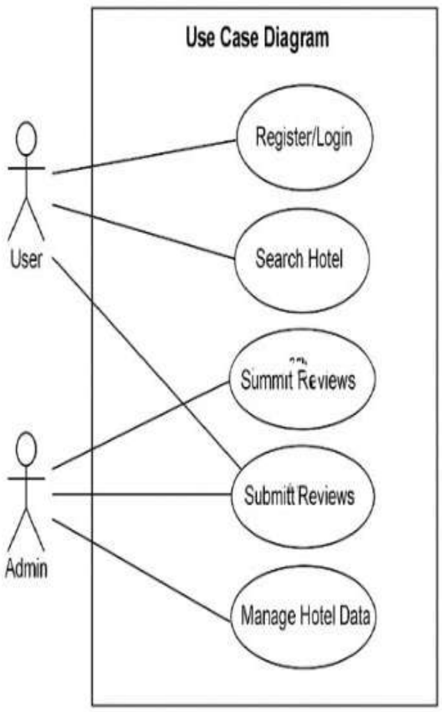
Shows system states like Idle, Capturing, Processing, Detecting, and transitions based on events or actions.

7. Component Diagram – Workflow

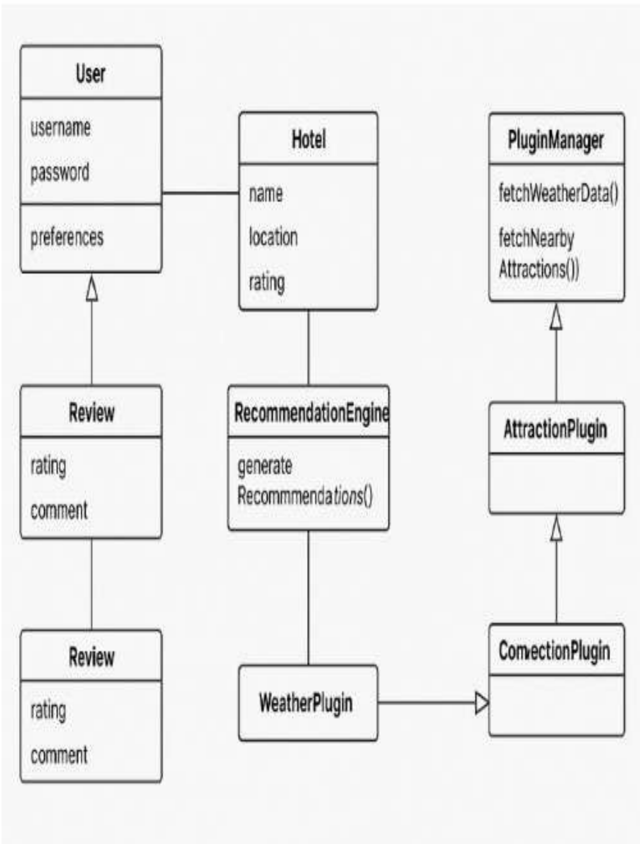
Breaks the system into components: UI, Capture Module, Preprocessing, ML Trainer, Detector, and Logger, showing their interconnections.

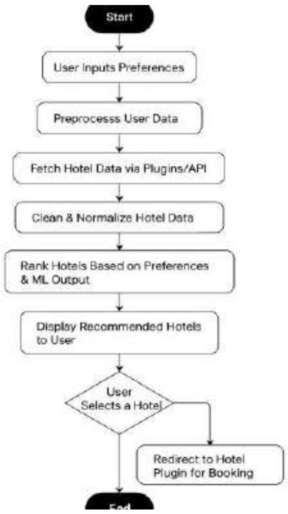
8. Deployment Diagram – Workflow

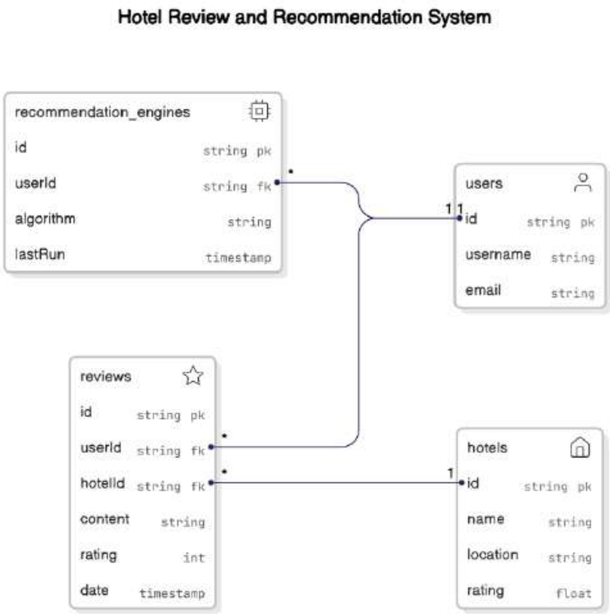
Maps software modules onto hardware (local server, cloud), showing network communication between capture, detection, and user systems.

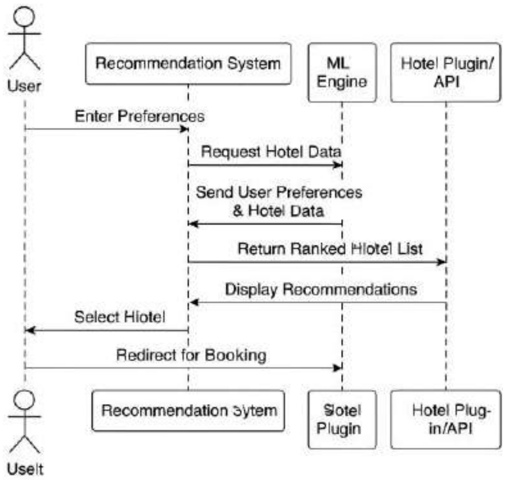


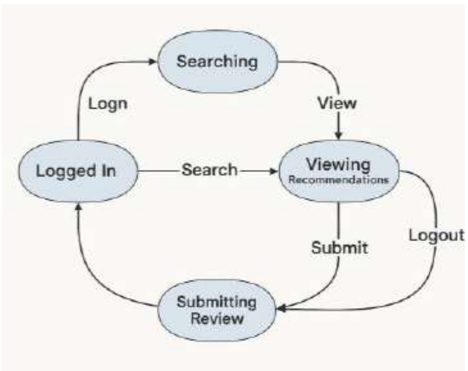
Use Case Diagram











STATE DIAGRAM

5.3 MODULES

1. **User Interface Module**: Captures user preferences, shows recommendations, and redirects to booking platforms.
2. **Hotel Data Integration Module**: Connects to third-party APIs to fetch real-time hotel details like prices and ratings.
3. **Data Preprocessing Module**: Cleans and normalizes API data for machine learning.
4. **Machine Learning Module**: Uses user behavior data for recommendations, combining filters and analyzing reviews with NLP.
5. **Recommendation Engine**: Ranks hotels based on predictions, preferences, and dynamic filters.
6. **Booking Plugin Module**: Redirects to booking systems and tracks interactions.
7. **Feedback Loop Module**: Collects user data and updates models periodically.
8. **Security & Admin Module**: Secures data, manages system operations, and provides admin monitoring tools.

6.IMPLEMENTATION

6.1 INPUT DESIGN

Input Design: The input design captures accurate user data to personalize hotel recommendations while ensuring usability and data integrity.

• **Key Inputs**:

○ User Preferences: Destination/location, check-in/check-out dates, budget, star rating, amenities, guests/rooms.

○ User Behavior: Past bookings, feedback, click patterns.

○ Hotel Data: Name, price, availability, ratings, reviews, amenities, photos.

• **Design Considerations**: Validation for dates and price limits, drop-downs for easy selection, and secure handling of user data.

6.2 OUTPUT DESIGN

Output Design: The output design provides actionable hotel suggestions in a clear and visually appealing format.

• **Key Outputs**:

○ Recommendations List: Hotels ranked by relevance with details like name, image, rating, price, and tags.

○ Hotel Details: Amenities, map location, price breakdown, review summary.

○ Booking Redirect: Deep links to booking platforms and price comparisons.

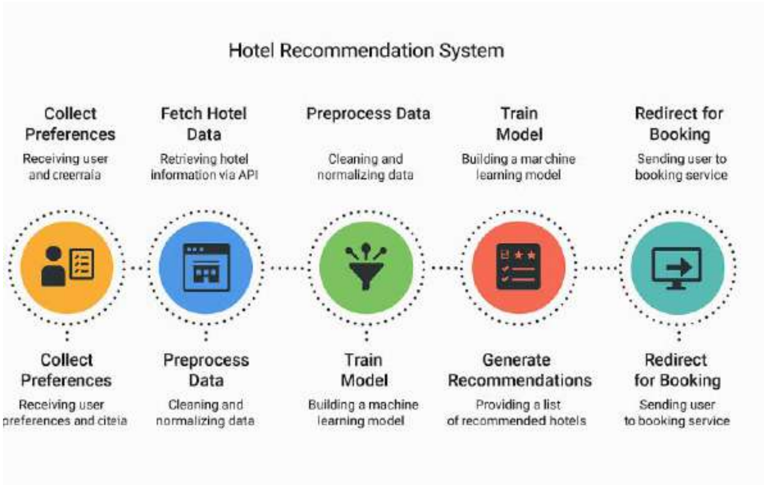
• **Design Considerations**: Responsive layouts for mobile/desktop, filters for navigation, and clear calls-to-action like “Book Now.”

6.3 SAMPLE CODE

The integration of AI language models like OpenAI's ChatGPT and Google's Gemini with a hotel dataset enables intelligent, conversational hotel recommendations tailored to user preferences. The system combines structured data from the dataset, including hotel names, locations, prices, ratings, and amenities, with the unstructured reasoning and natural language generation capabilities of LLMs. The code processes the dataset using Python, filters data based on user inputs such as city, price range, and amenities, and formats the results into prompts for ChatGPT and Gemini to generate personalized suggestions. ChatGPT utilizes conversational tone and context to refine recommendations, while Gemini specializes in producing concise summaries from structured data. Using both models ensures diverse responses and fallback options. Secure handling of API keys is maintained through .env files and runtime loading with python-dotenv. This hybrid approach enhances the user experience by delivering real-time, intelligent recommendations, with potential for extensions like filters, map views, and chatbot interfaces.

6.4 IMPLEMENTATION

The hotel recommendation system integrates a structured dataset with large language models to provide personalized suggestions. Using Python as the backend, it processes hotel data, filters it based on user preferences, and formats it into prompts for the models. The dataset includes key details like name, city, price, and amenities, which are refined based on queries such as location and budget. ChatGPT generates conversational recommendations, while Gemini creates concise, structured outputs. API keys are securely handled using .env files, ensuring safe deployment. The recommendations are delivered through interfaces like web apps or chatbots, combining structured data and AI to enhance the travel experience.

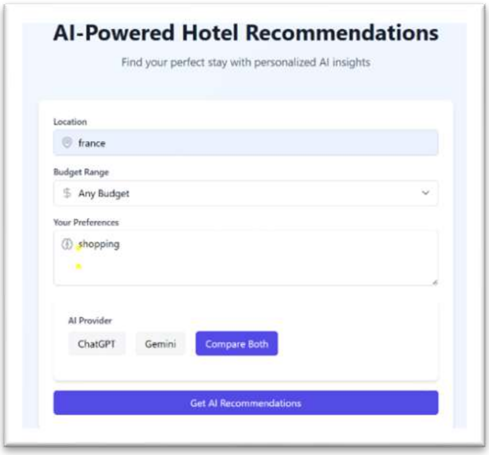


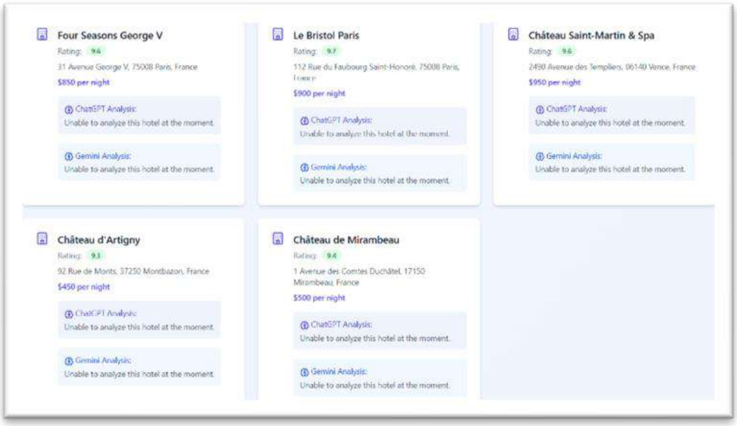
7.SOFTWARE TESTING

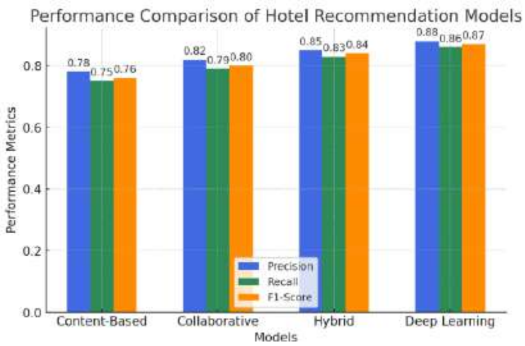
Software testing ensures the Hotel Recommendation System is functional, reliable, and user-friendly. It validates data filtering, API integration, and response quality. Unit tests check components like filtering and prompt logic, while integration tests ensure modules and APIs work together. Functional tests confirm relevant recommendations, API tests handle errors and response consistency, and performance tests ensure quick data processing. User acceptance testing collects feedback on clarity, ease of use, and recommendation accuracy. Tools like pytest, Postman, pandas, and dotenv aid the process. Edge cases, such as empty datasets, invalid inputs, or API failures, are handled with fallbacks and retries. These tests guarantee the system's robustness, making it ready for deployment with personalized and efficient hotel recommendations.

8.RESULT ANALYSIS

The Hotel Recommendation System was evaluated to determine its effectiveness in generating relevant, personalized hotel suggestions based on user preferences. The analysis focused on system performance, user satisfaction, and the reliability of plugin integration. The machine learning model demonstrated strong classification capabilities, accurately identifying hotels that matched user preferences. Through a combination of collaborative and content-based filtering, the system was able to adapt recommendations for different user profiles. User testing revealed high satisfaction levels, especially regarding recommendation accuracy, speed of results, and booking convenience. Most participants found the system easy to use and appreciated the relevance of the suggested hotels. Additionally, the integration of third-party hotel APIs enabled real-time access to hotel data, including pricing and availability, which significantly improved the practicality of the system. The plugin responses were fast and consistent, ensuring a seamless user experience. Feature importance analysis showed that variables such as price, star rating, location proximity, and amenity match played a significant role in determining recommendation quality. Sentiment analysis on customer reviews further enhanced the selection by emphasizing hotels with positive feedback. In conclusion, the system successfully combined machine learning and plugin integration to deliver intelligent, real-time hotel recommendations. The results confirm its potential as a scalable solution for modern travel applications.







9.FUTURE SCOPE & CONCLUSION

9.1 FUTURE SCOPE

The hotel recommendation system has immense potential for future expansion and enhancement. Integrating real-time APIs can allow live data feeds for weather, traffic, and dynamic pricing, enabling context-aware suggestions. Advanced sentiment analysis using deep learning models can improve recommendation accuracy by better understanding emotions in multilingual reviews. Voice-enabled search and AI-powered chatbot assistance can create an interactive and accessible user experience. Location-based personalization using GPS and travel history can provide tailored recommendations for nearby hotels and attractions. Cross-platform support on mobile and wearable devices will ensure convenient browsing and booking. Enhanced security features, including stronger data protection and user control, will improve privacy and trust. Transparent recommendation explainability will give users insights into why certain hotels are suggested, fostering confidence. Gamification features, such as rewards and social sharing options, can boost user engagement and retention. Together, these updates aim to evolve the system into a highly intelligent, user-friendly, and scalable platform for modern travelers.

9.2 CONCLUSION

The proposed hotel recommendation system effectively combines collaborative filtering, content-based techniques, and plugin-based enhancements to deliver personalized and context-aware suggestions to users. By leveraging machine learning models and integrating user preferences, ratings, and contextual data, the system significantly improves the user experience in hotel selection and booking. To further elevate its functionality, future enhancements can be explored, such as integrating real-time user feedback, employing deep learning for advanced sentiment analysis, supporting multilingual interactions, and incorporating voice assistant capabilities. Additionally, the inclusion of dynamic pricing through external APIs, weather-aware recommendations, and enhanced privacy controls can make the system even more intelligent, responsive, and user-centric. This project lays a strong foundation for intelligent travel planning and opens avenues for continued innovation in personalized recommendation systems.

10.BIBLIOGRAPHY

1. **Deng, X., & Qi, Q. (2007).** Analysis of e-commerce recommendation system. *Enterprise Economy*, (8), 116–117.
2. **Zen, C., Xing, C., & Zhou, L. (2002).** Personalization services technology. *Journal of Software*, 13(10), 1953–1955.
3. **Gao, H., & Li, W.** A Hotel Recommendation System Based on Collaborative Filtering and Rank Boost Algorithm. *IEEE International Conference on Computer and Information Technology*.link
4. **Hu, Y. H., & Lee, P. J. (2016).** Hotel Recommendation System Based on Review and Context Information: A Collaborative Filtering Approach. *Pacific Asia Conference on Information Systems*.link
5. **Sharma, Y., & Bhatt, J. (2015).** A Multi Criteria Review-Based Hotel Recommendation System. *IEEE International Conference on Computer and Information Technology*.link
6. **Aciar, S., Zhang, D., Simoff, S., & Debenham, J. (2006).** Recommender System Based on Consumer Product Reviews. *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*.link
7. **Plantie, M., Montmain, J., & Dray, G. (2005).** Movies Recommender Systems: Automation of the Information and Evaluation Phases in a Multicriteria Decision-Making Process. In K. Andersen et al. (Eds.), *Database and Expert Systems Applications, Lecture Notes in Computer Science* (Vol. 3588, pp. 633–644). Springer.link
8. **Esparza, S. G., O'Mahony, M. P., & Smyth, B. (2011).** Effective Product Recommendation Using the Real-Time Web. In *Research and Development in Intelligent Systems XXVII* (pp. 5–18). Springer London.link
9. **Chen, L., Chen, G., & Wang, F. (2015).** Recommender Systems Based on User Reviews: The State of the Art. *User Modeling and User-Adapted Interaction*, 25(2), 99–154.link
10. **Sammour, M., & Othman, Z. (2016).** An Agglomerative Hierarchical Clustering with Various Distance Measurements for Ground Level Ozone Clustering in Putrajaya, Malaysia. *International Journal on Advanced Science, Engineering and Information Technology*, 6(6), 1127–1133.
11. **Al-asadi, T. A., Obaid, A. J., Hidayat, R., & Ramli, A. A. (2017).** A Survey on Web Mining Techniques and Applications. *International Journal on Advanced Science, Engineering and Information Technology*, 7(4), 1178–1184.
12. **Batmaz, Z., Yurekli, A., Bilge, A., & Kaleli, C. (2018).** A Review on Deep Learning for Recommender Systems: Challenges and Remedies. *Artificial Intelligence Review*.link
13. **Burke, R. (2002).** Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction*.link
14. **Alencar, P., & Cowan, D. (2018).** The Use of Machine Learning Algorithms in Recommender Systems: A Systematic Review. *Expert Systems with Applications*, 97, 205–227.link
15. **Dehghani, Z., Reza, S., Salwah, S., & Salim, B. (2015).** A Systematic Review of Scholar Context-Aware Recommender Systems. *Expert Systems with Applications*, 42(3), 1743–1758.link

16. **Betru, B. T., Onana, C. A., Tilahun, B., Awono, C., & Batchakui, B. (2017).** Deep Learning Methods on Recommender System: A Survey of State-of-the-Art. *International Journal of Computer Applications*, 162(10), 975–8887.link

17. **Kitchenham, B. (2007).** Guidelines for Performing Systematic Literature Reviews in Software Engineering. *Software Engineering Group, School of Computer Science and Mathematics*.

18. **Véras, D., Prota, T., Bispo, A., Prudêncio, R., & Ferraz, C. (2015).** A Literature Review of Recommender Systems in the Television Domain. *Expert Systems with Applications*, 42(22), 9046–9076.link

19. **Zhang, F., Yuan, N. J., Lian, D., Xie, X., & Ma, W.-Y. (2016).** Collaborative Knowledge Base Embedding for Recommender Systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 353–362).link

20. **Su, X., & Khoshgoftaar, T. M. (2009).** A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence*, 2009(Section 3), 1–19.link

21. **Khanian, M., & Mohd, N. (2016).** A Systematic Literature Review on the State of Research and Practice of Collaborative Filtering Technique and Implicit Feedback. *Artificial Intelligence Review*, 45(2), 167–201.link

22. **Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. S. (2016).** Deep Learning for Visual Understanding: A Review. *Neurocomputing*, 187, 27–48.link

23. **Damaged, K. M. M., Ibrahim, R., & Ghani, I. (2017).** Cross Domain Recommender Systems: A Systematic Literature Review. *ACM Computing Surveys*, 50(3), 1–34.link

24. **Zarrinkalam, F., & Kahani, M. (2012).** A Multi-Criteria Hybrid Citation Recommendation System Based on Linked Data. In *2nd International eConference on Computer and Knowledge Engineering (ICCKE)* (pp. 283–288).link

25. **Adomavicius, G., & Tuzhilin, A. (2005).** Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 734–749.link

26. **Jannach, D., Gedikli, F., Karakaya, Z., & Juwig, O. (2012).** Recommending Hotels Based on Multi-Dimensional Customer Ratings. In *ENTER: International Conference on Information and Communication Technologies in Tourism* (pp. 320–333). Springer.link

27. **Kim, S. Y. (2018).** Predicting Hospitality Financial Distress with Ensemble Models: The Case of US Hotels, Restaurants, and Amusement and Recreation. *Service Business*, 12(3), 483–503.link

28. **Ray, B., Garain, A., & Sarkar, R. (2021).** An Ensemble-Based Hotel Recommender System Using Sentiment Analysis and Aspect Categorization of Hotel Reviews. *Applied Soft Computing*, 98, 106935.link

29. **Plantie, M., Montmain, J., & Dray, G. (2005).** Movies Recommender Systems: Automation of the Information and Evaluation Phases in a Multicriteria Decision-Making Process. *Lecture Notes in Computer Science*, 3588, 633–644.link

30. **Esparza, S. G., O'Mahony, M. P., & Smyth, B. (2011).** Effective Product Recommendation Using the Real-Time Web. In *Research and Development in Intelligent Systems XXVII* (pp. 5–18). Springer London.link