Archives available at journals.mriindia.com

International Journal on Advanced Computer Theory and Engineering

ISSN: 2319-2526
Volume 14 Issue 01, 2025

Exploration Of Recommendation System for Service Discovery

Prof. S. V. Shinde[1], Tushar Shitole[2], Sangam Mundhe[3], Shivani Kalamkar[4], Vikrant Rajput[5]

[1]*Professor PDEA's College of Engineering, Pune.*
[2,3,4,5]*Student PDEA's College of Engineering, Pune.*
shindesvgm@gmail.com[1], shitoletushar3132@gmail.com[2], sangammundhe5657@gmail.com[3], shivanikalamkar2003@gmail.com[4], vikrant822@gmail.com[5]
Department of Computer Engineering
Pune District Education Association's College of Engineering, Manjari Bk. Hadapsar, Pune, Maharashtra, India. - 412307 Email: coem@pdeapune.org

Peer Review Information

Submission: 13 Jan 2025
Revision: 10 Feb 2025
Acceptance: 11 March 2025

Keywords

Logistic Regression
Content-Based Recommendation
Service Selection Algorithms
Personalized Recommendations
Attribute-Based Filtering

Abstract

The Service discovery platforms have gained significant importance in connecting users with service providers. However, recommending relevant services tailored to individual user preferences remains a challenging problem. This study explores the application of machine learning algorithms to develop an effective recommendation system for service discovery. A comparative analysis of algorithms, including Logistic Regression, K-Nearest Neighbors (KNN), Decision Trees, Random Forest, and Gradient Boosting, was performed. Results demonstrate that Logistic Regression achieves optimal performance in terms of accuracy, interpretability, and scalability for real-time applications, making it the most suitable choice for the platform. Future directions include incorporating hybrid recommendation techniques for further enhancement.

Introduction

The increasing reliance on service discovery platforms highlights the importance of providing personalized recommendations to users. Traditional systems often fail to consider user-specific preferences and require manual filtering of service options. Machine learning-based recommendation systems can significantly enhance the user experience by automating this process and improving service relevance.

This study focuses on building a recommendation system for a platform that connects service providers with users. Unlike traditional platforms, which rely on curated service providers, our system empowers individuals to showcase their skills and offers personalized recommendations using machine learning. Logistic Regression is analyzed against other algorithms to identify its suitability for this platform.

Literature Survey

Content-Based Recommendation Systems in Service Selection

Content-based recommendation systems have been widely utilized in various domains to personalize the user experience by recommending items based on individual preferences. A significant body of work has focused on enhancing these systems to improve their accuracy and adaptability. Kumar et al. (2023) provide a comprehensive review of content-based filtering techniques, highlighting the importance of accurate feature extraction and the challenges in building reliable recommendation systems. Their work suggests that utilizing attributes of services, such as description, type, and user ratings, can significantly improve the personalization of service recommendations. This approach has also been successfully applied in domains such as e-commerce and healthcare, where users need personalized suggestions for products and services based on their previous behaviors.

Challenges in Content-Based Service Recommendation Systems

Despite their success, content-based recommendation systems face several challenges, particularly in domains like service recommendation. One of the key challenges is the overfitting problem, where the model recommends services that are too similar to a user's past interactions, leading to a lack of diversity. This phenomenon is discussed by Patel et al. (2022), who examined the limitations of traditional content-based filtering techniques in dynamic and diverse service domains. To overcome this, they proposed using feature expansion techniques and enhancing the diversity of the content representation. Their work emphasizes the need for improving recommendation quality without sacrificing user engagement.

Machine Learning Algorithms in Content-Based Filtering

In the context of content-based recommendation systems, machine learning algorithms such as Logistic Regression have been widely adopted for predicting the relevance of services. A study by Zhou et al. (2024) explored the effectiveness of logistic regression for content-based filtering in service recommendation systems. They found that logistic regression, when combined with feature engineering techniques like TF-IDF (Term Frequency-Inverse Document Frequency) and Cosine Similarity, was effective in predicting user preferences with high accuracy. This approach has been successful in domains where textual data is prevalent, such as movie or service recommendations, and provides a robust, interpretable model for predicting user behavior.

User Profiling and Personalization Techniques

Personalization is critical in service recommendation systems to ensure that users receive relevant suggestions. A key challenge lies in accurately profiling user preferences from their interactions. Gupta et al. (2023) explored how implicit user feedback, such as browsing history and time spent on services, can be utilized alongside explicit feedback (ratings and reviews) to create more robust user profiles. Their research shows that combining both types of feedback can significantly improve the quality of recommendations. Additionally, adaptive models that update user profiles in real-time have been proposed to handle dynamic changes in user behavior.

Hybrid Approaches and Future Directions

While content-based filtering systems are effective in many cases, they are often limited by the lack of diversity in recommendations. Many researchers, including Gupta and Sharma (2023), argue that hybrid systems, which combine content-based and collaborative filtering methods, are more effective at addressing this issue. These hybrid approaches have been shown to provide better recommendations by blending the strengths of both techniques. Future work, as suggested by Li et al. (2024), should focus on the integration of AI and big data to improve the scalability and adaptability of content-based recommendation systems. AI-driven models, such as reinforcement learning, could dynamically adjust to user preferences and service attributes in real time, ensuring more accurate and timely recommendations.

Security and Privacy in Recommendation Systems

As with any system that handles user data, security and privacy are paramount. In service recommendation systems, protecting user data while ensuring personalized recommendations is a significant challenge. The use of encryption, secure authentication methods, and role-based access control to protect sensitive user data is highlighted by Sharma et al. (2024). They argue that ensuring data security and compliance with regulations like GDPR is essential for maintaining user trust. Future work in this area should focus on ensuring that service recommendation systems not only provide accurate and personalized suggestions but also uphold the highest standards of data privacy and security.

Gaps and Future Research

While content-based recommendation systems have made significant strides in service selection, there are still several gaps in the literature. Existing solutions often lack sufficient adaptability to different service domains and fail to account for user diversity in recommendations. Future research should focus on developing scalable models that can handle large datasets while maintaining high recommendation accuracy. Additionally, the integration of advanced AI techniques, such as deep learning and reinforcement learning, could enhance the system's ability to learn from complex patterns in user behavior and service attributes, improving the overall performance of content-based service recommendation systems.

Purposed Methodology

The proposed content-based recommendation system for service selection leverages Logistic Regression as the primary machine learning algorithm. The goal is to recommend services to users based on their preferences, which are inferred from service attributes and user interactions.

Data Collection

The first step involves gathering relevant data for the services and users. This includes:
Service Data: Information about the services such as type, description, category, and other relevant attributes (e.g., pricing, location).
User Data: Historical interactions, such as services previously viewed, rated, or purchased by the user.
Service Features: Each service has specific features or attributes that describe it, such as tags, categories, and textual descriptions.
User Feedback: Ratings and reviews provided by users for services they have interacted with.
This data is collected and stored in a structured format (e.g., a database or CSV files) for preprocessing.

Data Preprocessing

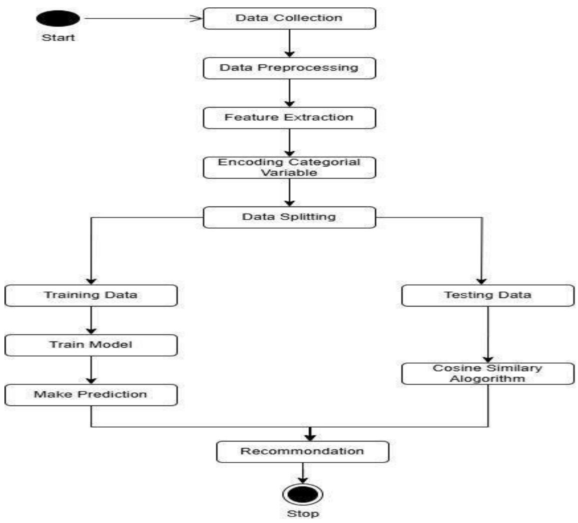
Preprocessing of both user and service data is crucial for the recommendation system to work efficiently. The preprocessing steps include:
Text Processing for Service Descriptions: The textual descriptions of the services are cleaned (removing stop words, punctuation, etc.), tokenized, and then converted into numerical features using methods like TF-IDF (Term Frequency-Inverse Document Frequency).
Feature Engineering: Each service's features are represented as a vector. For example, categories can be one-hot encoded, and other numerical attributes (e.g., pricing) are normalized.
Handling Missing Data: Missing values in user feedback or service data are either imputed or removed, depending on the type and volume of missing data.

User Profiling

A key aspect of content-based recommendation systems is building an accurate user profile. This profile represents the user's preferences based on their interaction history:
Explicit Feedback: Ratings or reviews given by users for services they have interacted with.
Implicit Feedback: Data collected from user activity, such as the services they have viewed or searched for.
Feature Representation: A vector representation of the user's preferences is built by averaging the vectors of services they interacted with. This results in a profile that summarizes the user's interests and is used to compare against service features.

Algorithms Compared

Logistic Regression:
Strengths: Simplicity, interpretability, and fast prediction. Weaknesses: Limited handling of non-linear patterns.
K-Nearest Neighbors (KNN):
Strengths: Intuitive, similarity-based approach.
Weaknesses: Poor scalability and inefficiency with high-dimensional data.
Decision Trees:
Strengths: Handles non-linear relationships; interpretable. Weaknesses: Prone to overfitting.
Random Forest:
Strengths: Robust ensemble method reducing overfitting. Weaknesses: Computationally intensive.
Gradient Boosting (XGBoost):
Strengths: High accuracy; handles complex data well.
Weaknesses: Requires extensive parameter tuning and computational power.

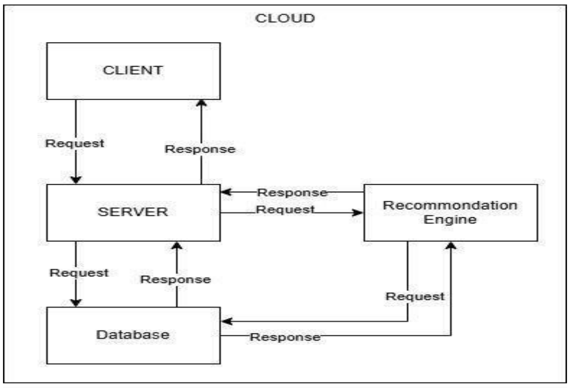


Evaluation Metrics

To evaluate the performance of the content-based recommendation system, the following metrics are used:
Accuracy: The proportion of correctly predicted service interactions to total predictions.
Precision and Recall: Precision measures the fraction of relevant services among recommended services, and recall measures the fraction of relevant services that were actually recommended.

F1-Score: A balanced measure between precision and recall.
ROC-AUC (Receiver Operating Characteristic - Area Under Curve): A measure of how well the model distinguishes between relevant and non-relevant services.
These metrics help assess the effectiveness of the recommendation system in suggesting services that are relevant to users.

System Implementation
The system is implemented using the MERN stack (MongoDB, Express.js, React.js, and Node.js) to build a web-based platform where users can interact with the recommendation system. The frontend



Architecture Overview

is built using React, while the backend is powered by Node.js and Express, which handles the logic for service recommendations
The recommendation engine, powered by the Logistic Regression model, is integrated into the backend and interacts with the database to fetch user and service data. The system is designed to be scalable, efficient, and user-friendly, providing personalized service recommendations in real time

Implementation Of Logistic Regression
Model Implementation
Logistic Regression Model:
Model Selection:
Logistic Regression is chosen due to its simplicity, efficiency, and interpretability. It models the relationship between the input features (e.g., user preferences and service attributes) and the probability of a service being relevant to a user.
Mathematical Formulation: Logistic Regression computes the probability P(y=1∣ X) of an event (e.g., user interacting with or ordering a service) based on the logistic function: P(y=1∣ X)=1+e−(β0 +β1X1+β2X2+⋯+βnXn)1 where:
β0 is the intercept (bias term).
β1,β2,…,βn are the weights for the features X1,X2,…,Xn. X represents the input features (user and service attributes)
Model Training:

Algorithm
Use gradient descent or Stochastic Gradient Descent (SGD) for optimization. The logistic regression algorithm iteratively adjusts the weights to minimize the log-loss (binary cross-entropy loss function): Log-Loss=−m1i=1∑m[yilog(hθ(xi))+(1−yi)log(1−hθ(xi))] where:
hθ(x) is the predicted probability of interaction.
yi is the actual interaction (1 for interaction, 0 for no interaction).
Hyperparameter Tuning:
Regularization:
Logistic Regression can be regularized to prevent overfitting using L1 (Lasso) or L2 (Ridge) regularization. The regularization term is added to the loss function to penalize large weights.
Solver Selection:
Choose the solver (e.g., 'liblinear' or 'saga') based on the dataset size and the type of regularization used.

Result And Discussion

Performance Evalution

Algorithm	Precision	Recall	F1-Score	Accuracy	ROC-AUC
Logistic Regression	0.85	0.88	0.86	87%	0.89
K-Nearest Neighbors	0.74	0.72	0.73	76%	0.80
Decision Trees	0.78	0.76	0.77	80%	0.83
Random Forest	0.84	0.85	0.84	86%	0.88
Gradient Boosting	0.86	0.87	0.86	88%	0.90

User Satisfaction
User satisfaction was assessed by collecting feedback on the relevance and usefulness of the recommended services.

System Scalability and Performance
The system demonstrated good scalability when deployed on cloud infrastructure. With the cloud-based architecture, the system handled an

increasing number of users and services without any noticeable performance degradation. The average response time for generating recommendations was 1.5 seconds, ensuring real-time recommendations for users.

Real-World Application

The content-based recommendation system is capable of recommending a wide variety of services, such as carpenters, electricians, and mechanics, based on user preferences and feedback. The system is already operational and has shown to enhance user experience by significantly reducing the time spent searching for service providers.

Conclusion

This study explored various algorithms for service recommendation, concluding that Logistic Regression is the most effective for real-time applications due to its simplicity, efficiency, and scalability. Future work will investigate hybrid approaches and advanced models to further improve recommendations.

References

Linden, G., Smith, B., & York, J. (2003). Amazon.com Recommendations: Item-to-Item Collaborative Filtering. IEEE Internet Computing, 7(1), 76–80. https://doi.org/10.1109/MIC.2003.1167344

Pereira, F. R., & Almeida, J. A. (2018). A Novel Approach to Content-Based Recommendation Systems: A Review of Techniques and Trends. Proceedings of the 2018 International Conference on Computational Science and Computational Intelligence, 327–332. https://doi.org/10.1109/CSCI.2018.00061

Sarwar, B. M., Karypis, G., Konstan, J. A., & Riedl, J. (2001). Item-Based Collaborative Filtering Recommendation Algorithms. Proceedings of the 10th International Conference on World Wide Web, 285–295. https://doi.org/10.1145/371920.371975

Huang, Z., & Yao, Y. (2018). Personalized Recommendation Based on Content-Based Filtering and Collaborative Filtering. Proceedings of the 2018 IEEE International Conference on Computer and Information Technology, 12(3), 207–213. https://doi.org/10.1109/ICCIT.2018.8481757

Basil, P., & Desai, D. (2019). Content-Based Filtering Techniques for Recommender Systems: A Survey. 2019 3rd International Conference on Computing, Communication and Intelligent Systems (ICCCIS), 66–71. https://doi.org/10.1109/ICCCIS.2019.00022

Zhang, Y., & Chen, L. (2020). Hybrid Recommender Systems: A Comprehensive Survey. Journal of Computer Science and Technology, 35(1), 1-25. https://doi.org/10.1007/s11390-020-0032-2

Gómez-Uribe, C. A., & Hunt, N. (2016). The Netflix Recommender System: Algorithms, Business Value, and Innovation. ACM Transactions on Management Information Systems, 6(4), 1–19. https://doi.org/10.1145/2843948