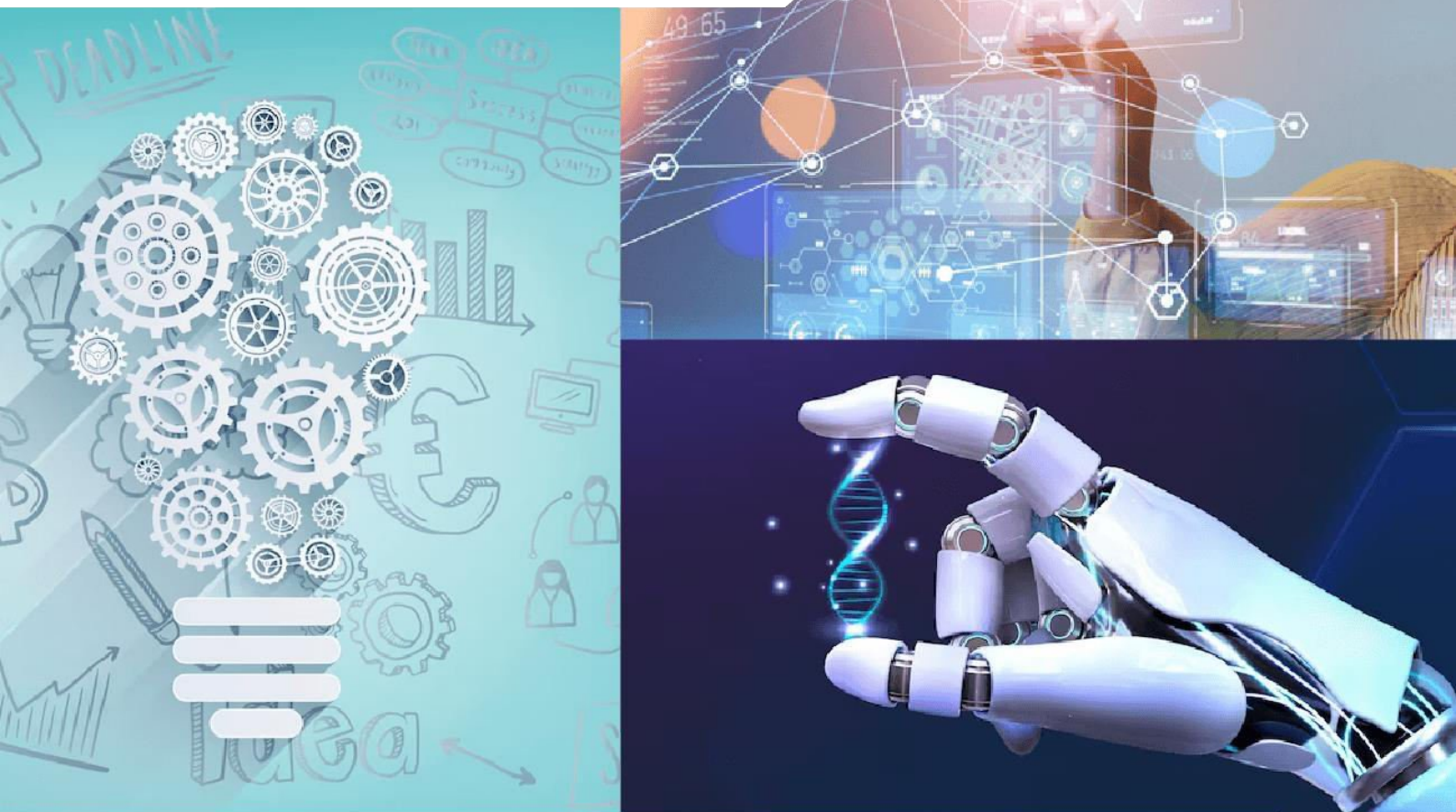




International Journal of Advanced Research in Education and Technology (IJARETY)

Volume 13, Issue 2, March - April 2026

Impact Factor: 8.152



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



Optimizing Mobile App Recommendation Using Crowd sourced Educational Data

M.Pavan kumar Reddy¹, Darji Sneha², Gadige Akshitha³, Gannoju Pooja⁴

Assistant Professor, Department of CSE, Guru Nanak Institute of Technology, Hyderabad, Telangana, India ¹

Students, Department of CSE, Guru Nanak Institute of Technology, Hyderabad, Telangana, India ^{2,3,4}

ABSTRACT: The increasing availability of educational mobile applications has made it difficult for students to identify relevant and effective learning tools. This paper proposes a personalized mobile app recommendation system using crowdsourced educational data to improve recommendation quality. The system analyzes user behavior, app usage patterns, and textual data using Natural Language Processing (NLP) techniques for feature extraction. A Random Forest classifier is used to model user preferences and generate recommendations for different academic levels. Experimental results show improved accuracy, scalability, and effectiveness in addressing challenges such as data sparsity and the cold-start problem.

KEYWORDS: Personalized Recommendation System, Educational Mobile Applications

I. INTRODUCTION

The widespread adoption of smartphones and mobile applications has significantly transformed modern education by enabling students to access learning resources anytime and anywhere. Educational apps support various activities such as subject learning, skill development, and academic planning. However, the rapid increase in the number of available applications makes it difficult for students to identify the most relevant tools suited to their learning needs. Traditional recommendation systems often rely on static metrics such as ratings and downloads, which do not accurately reflect user preferences or learning effectiveness.

To address these limitations, this paper proposes a personalized mobile app recommendation system using crowdsourced educational data. The system analyzes user behavior, app usage patterns, and textual information such as app descriptions and feedback.

II. LITERATURE SURVEY

A study by Aslif Ali Laghari (2025) focuses on improving mobile app recommendation systems by using crowdsourced educational data along with machine learning and deep learning techniques. The research shows that collecting data from users, such as reviews, feedback, and app usage patterns, helps in better understanding user preferences and behavior. By applying machine learning methods like collaborative filtering and content-based filtering, the system can identify similarities among users and recommend suitable applications. In addition, deep learning models are used to capture complex patterns in large datasets, which improves the accuracy and personalization of recommendations. The system follows key stages such as data collection, preprocessing, feature extraction, model training, and recommendation generation. A feedback mechanism is also included to continuously update and improve the system based on user interactions.

III. EXISTING SYSTEM

Existing mobile app recommendation systems primarily rely on techniques such as collaborative filtering, content-based filtering, and deep learning models like Gated Recurrent Units (GRU) to generate recommendations based on historical user data. These systems analyze app ratings, download counts, and user interaction patterns to predict user preferences, with GRU models effectively capturing sequential and time-dependent behavior.

Existing System Disadvantages

- Depends on static data
- Cold-start problem
- Data sparsity issues

Proposed System

The proposed system is a personalized mobile app recommendation system that uses crowdsourced educational data to suggest suitable apps for students. It analyzes user behavior, app usage, and app descriptions to understand individual preferences. Natural Language Processing (NLP) is used to process text data, and a Random Forest algorithm is applied to generate accurate recommendations.

Proposed System Advantages:

- Flexibility
- Improves learning and user experience
- Adapts to user needs over time

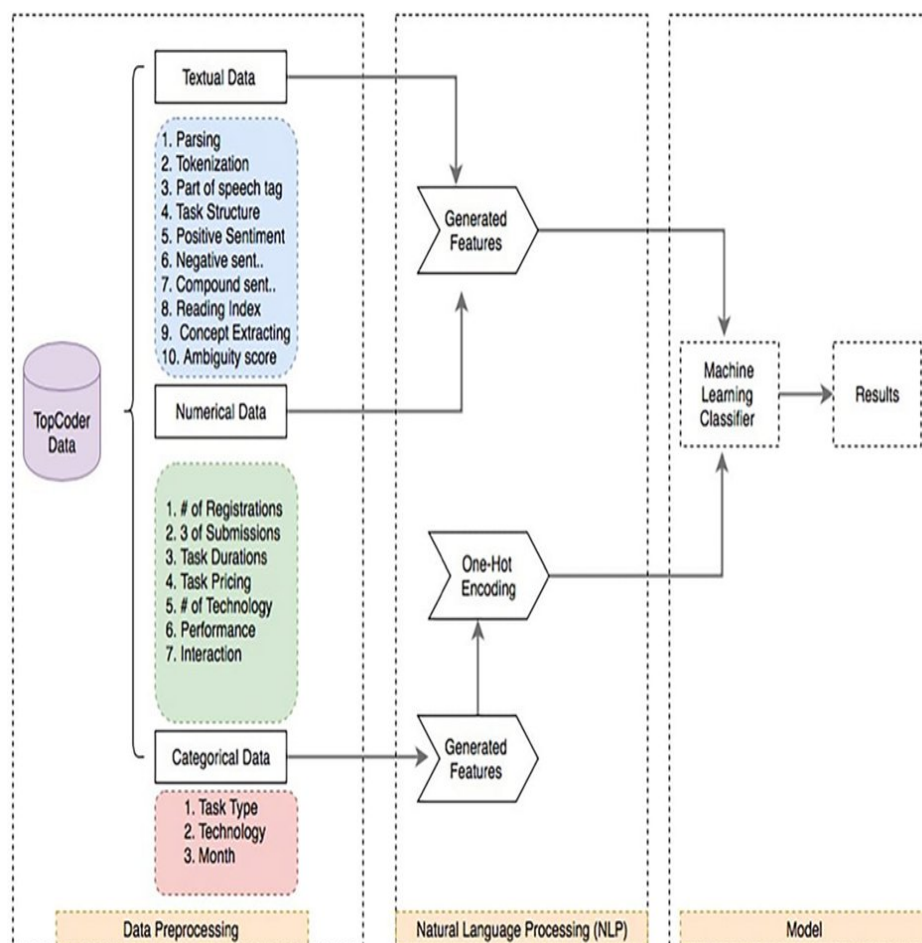
IV. SYSTEM ARCHITECTURE

Fig 1.1 System Architecture

The system architecture consists of several stages starting with data collection, where user app usage, feedback, and app details are gathered as crowdsourced data. This data is then preprocessed using Natural Language Processing (NLP) techniques such as cleaning and tokenization to convert it into useful features

Methodology**Modules Name:**

- Data Collection
- Dataset
- Data Preparation
- Model Selection

- Analyze and Prediction
- Accuracy on Test Set
- Saving the Trained Model

1) Data Collection Module: In this module, data is collected from various sources such as user interactions, app usage logs, feedback, and surveys. It may include details like how frequently apps are used, user preferences, and ratings.

2) Data Set Module: The collected data is organized into a structured dataset in this module. It includes features such as app name, category, description, user ratings, and usage patterns.

3) Data Preparation Module: In this module, the dataset is cleaned and preprocessed to remove errors, duplicates, and irrelevant data. Natural Language Processing (NLP) techniques like text cleaning, tokenization, and stop-word removal are applied to handle textual data.

4) Model Selection : This module focuses on selecting a suitable machine learning algorithm, such as the Random Forest classifier. The model is trained using the prepared dataset to learn patterns between user behavior and app features.

5) Analyze And Prediction Module: In this module, the trained model is used to analyze new user data and generate app recommendations. It compares user preferences with learned patterns and predicts the most suitable apps.

6. Accuracy on Test Set: Here, the model is tested using a separate test dataset to evaluate its performance. Metrics such as accuracy, precision, and recall are used to measure how well the model predicts recommendations.

7.Saving The Trained Model: In this final module, the trained model is saved using tools like pickle so it can be reused later. This avoids the need to retrain the model every time and allows faster predictions.

V. IMPLEMENTATION

The proposed system is implemented using Python, leveraging libraries such as NumPy, Pandas, and Scikit-learn for efficient data processing and machine learning operations. Initially, the crowdsourced educational dataset is collected and subjected to preprocessing using Natural Language Processing (NLP) techniques, including text cleaning, tokenization, and stop-word removal, to transform unstructured textual data into meaningful representations. Subsequently, feature extraction and vectorization techniques are applied to convert the processed data into numerical form suitable for model training. The dataset is then partitioned into training and testing subsets to ensure reliable performance evaluation. A Random Forest classifier is employed as the core predictive model due to its robustness, scalability, and ability to handle high-dimensional data. The model is trained to learn the relationship between user behavior, app characteristics, and academic levels, enabling accurate recommendation generation. Performance is evaluated using standard metrics such as accuracy, ensuring the reliability of the system. Finally, the trained model is serialized and stored for deployment, allowing real-time personalized app recommendations and seamless integration into mobile or web-based educational platforms.

Algorithm Used

Existing Algorithm

The existing system utilizes the Gated Recurrent Unit (GRU), a type of Recurrent Neural Network (RNN) designed to handle sequential data and capture temporal dependencies in user behavior. GRU employs gating mechanisms, namely the update gate and reset gate, to control the flow of information and retain relevant past data while discarding unnecessary information. This enables the model to effectively analyze time-dependent app usage patterns and generate recommendations based on user interaction history.

Proposed Algorithm

Random Forest Classifier :

The proposed system employs the Random Forest Classifier, an ensemble learning algorithm that constructs multiple decision trees and combines their outputs to improve prediction accuracy and robustness. It processes both behavioral and textual features extracted using Natural Language Processing (NLP) techniques. Each decision tree independently predicts suitable app category for a user, and the final recommendation is determined through majority voting. Random Forest is chosen for its ability to handle high-dimensional data, reduce overfitting, and provide better interpretability through feature importance analysis. This makes it highly suitable for generating accurate, scalable, and personalized app recommendations in educational environments.

VI. EXPERIMENTAL RESULTS

Home Page:



Fig 2: Home Page

The Home Page is the main interface of the application where users first land after opening the system. It provides an overview of the app recommendation system and its purpose. The page usually displays options like viewing recommended educational apps, exploring categories (UG, PG, Graduate), and basic navigation menus.

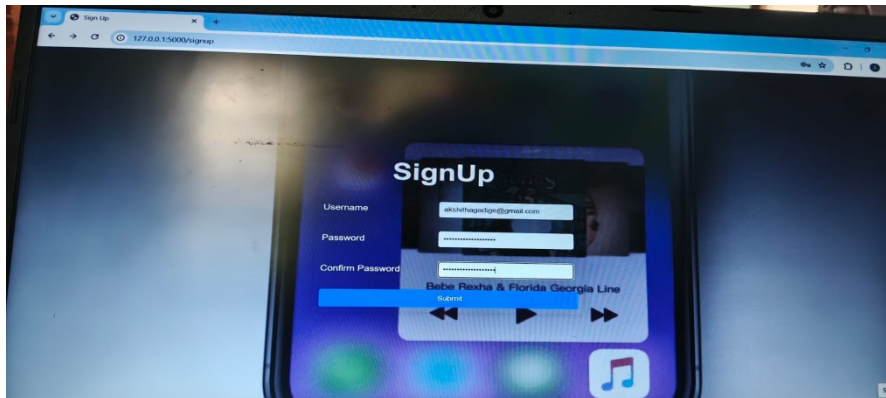


Fig 2: Regesister page:

The registration page is used for new users to create an account. Users enter details like name, email, and password. The system saves this information. After registering, the user can log in and use the system.

Login Page:

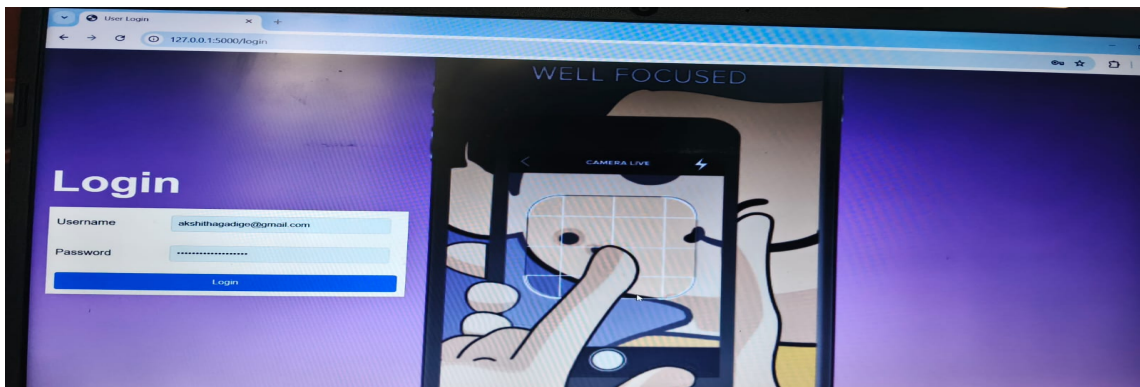


Fig 4: Login Page

The login page is used to access the system. Users enter their email and password. The system checks the details. If correct, the user is logged in and can use the application.

VII. CONCLUSION

This project presents a personalized educational app recommendation system designed to enhance the learning experience of university students across Undergraduate, Postgraduate, and Graduate levels. By leveraging app usage data and textual app descriptions, the system applies Natural Language Processing techniques to extract meaningful features that capture students' preferences and academic needs. The Random Forest Classifier serves as the core predictive model due to its robustness, interpretability, and ability to handle high-dimensional data. Experimental evaluation demonstrates that the model achieves high accuracy, precision, and recall, producing reliable and relevant recommendations

VIII. FUTURE ENHANCEMENT

The system can be enhanced using advanced algorithms, real-time updates, multi-language support, larger datasets, improved security, and continuous user feedback for better recommendations.

REFERENCES

- [1].M. Zhai, X. Wang, and X. Zhao, "The importance of online customer reviews characteristics on remanufactured product sales: Evidence from the mobile phonemarketonAmazon.com,"J.RetailingConsum.Services, vol. 77, Mar. 2024, Art. no. 103677.
- [2] C. C. Aggarwal, "An introduction to recommender systems," in Rec ommender Systems: The Textbook. Cham, Switzerland: Springer, 2015, pp. 1–28.
- [3] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item to-item collaborative filtering," IEEE Internet Comput., vol. 7, no. 1, pp. 76–80, Jan. 2003.
- [4]N.TintarevandJ.Masthoff,"Beyondexplainingingleitemrecommendations,"in Recommender Systems Handbook. Cham,Switzerland: Springer, 2022, pp. 711–756.
- [5] C. Musto, M. de Gemmis, P. Lops, F. Narducci, and G. Semeraro, "Semantics and content-based recommendations," in Recommender Sys tems Handbook. Cham, Switzerland: Springer, 2022, pp. 251–298.
- [6] P. Winoto and T. Y. Tang, "The role of user mood in movie recommenda tions," Expert Syst. Appl., vol. 37, no. 8, pp. 6086–6092, Aug. 2010.
- [7] P. Li and A. Tuzhilin, "A dynamical system framework for explor ing consumer trajectories in recommender system," SSRN, May 2024, pp. 1–55.
- [8] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering tech niques," Adv. Artif. Intell., vol. 2009, no. 1, Oct. 2009, Art. no. 421425.
- [9] M. Asad, S. Shaukat, E. Javanmardi, J. Nakazato, and M. Tsukada, "A comprehensive survey on privacy-preserving techniques in feder ated recommendation systems," Appl. Sci., vol. 13, no. 10, p. 6201, May 2023.
- [10] M. Singh, "Scalability and sparsity issues in recommender datasets: Asurvey," Knowl. Inf. Syst., vol. 62, no. 1, pp. 1–43, Jan. 2020.

International Journal of Advanced Research in Education and Technology

ISSN: 2394-2975

Impact Factor: 8.152