International Scientific Journal of Engineering and Management
Volume: 03 Issue: 05 | May – 2024
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata
ISSN: 2583-6129
DOI: 10.55041/ISJEM01618

Aptitude Training System Using Decision Trees

Faizaan Qaisar Anche, Dinesh A, Chillamanda Durga Prasad, Kunal Raj, Dr. Madhura J

[1] *Faizaan Qaisar Anche, Information Science and Engineering, Dayananda Sagar College of Engineering, Bangalore*
[2] *Dinesh A, Information Science and Engineering, Dayananda Sagar College of Engineering, Bangalore*
[3]*Chillamanda Durga Prasad, Information Science and Engineering, Dayananda Sagar College of Engineering, Bangalore*
[4] *Kunal Raj, Information Science and Engineering, Dayananda Sagar College of Engineering, Bangalore*
[5]*Dr.Madhura J, Assistant Professor, Dept. of ISE, Dayananda Sagar College of Engineering, Bangalore*

Abstract — This paper presents the development of an adaptive online aptitude training system utilizing decision trees for students preparing for campus placements. The system aims to assess and enhance students' aptitude and technical skills through personalized learning experiences. Leveraging a vast question bank, the system employs an adaptive algorithm analyzing users' responses, time spent, and performance data from pre-assessments to tailor subsequent assessments. By integrating AI-based question selection, the system ensures relevance and effectiveness in assessing student proficiency. The focus lies on providing targeted exercises to address individual weaknesses, thereby optimizing preparation for campus interviews. This paper underscores the efficacy of decision tree algorithms in delivering personalized aptitude training, contributing to students' success in securing placements.

Keywords: Adaptive Learning, Decision Trees, Aptitude Training, Personalized Education

I. INTRODUCTION

In an increasingly competitive job market, securing lucrative placements through campus interviews requires not only academic excellence but also adeptness in aptitude and technical skills. Recognizing this crucial need, educational institutions and stakeholders have sought innovative approaches to equip students with the necessary competencies for success. In response to this demand, the development of adaptive online aptitude training systems has gained prominence, aiming to provide tailored learning experiences that address individual strengths and weaknesses.

This paper presents the culmination of efforts in developing such an adaptive online aptitude training system, specifically designed to aid students preparing for campus placements. Central to this system's efficacy is the utilization of decision trees, a powerful algorithmic tool capable of analyzing complex datasets and making informed decisions based on predefined criteria. By harnessing the capabilities of decision trees, our system endeavors to enhance students' aptitude and technical skills through personalized learning experiences.

This paper underscores the significance of personalized aptitude training in optimizing students' preparation for campus interviews. Through targeted exercises aimed at addressing individual weaknesses, the system aims to equip students with the requisite skills and confidence to excel in competitive recruitment processes. Furthermore, by highlighting the efficacy of decision tree algorithms in delivering personalized learning experiences, this paper contributes to the broader discourse on leveraging technology for educational enhancement.

II. BACKGROUND AND OBJECTIVES

In the realm of higher education, particularly in technical fields like engineering, the process of campus placements serves as a pivotal juncture for students. Securing lucrative job opportunities through campus interviews requires not only academic proficiency but also adeptness in aptitude and technical skills. However, students often grapple with the challenge of effectively preparing for these assessments amidst the myriad of topics and concepts.

The existing landscape of aptitude training and preparation tools in India reflects a significant gap in personalized and adaptive learning experiences. Traditional methods often fail to address individual weaknesses and preferences, leading to suboptimal learning outcomes and heightened anxiety among students.

The primary objective of our research is to enhance student learning outcomes by providing personalized and adaptive learning experiences tailored to individual needs. Through targeted exercises and adaptive testing, we aim to empower students to identify and address their weaknesses effectively, thereby improving their overall proficiency in aptitude and technical skills.

Furthermore, our objective extends to facilitating effective campus placement preparation by equipping students with the requisite aptitude and technical knowledge. Through these objectives, we aspire to level the playing field for all students, ensuring that each individual has an equal opportunity to succeed in the competitive landscape of campus placements and technical interviews.

III. TECHNOLOGIES USED

Decision Tree Regressor

A decision tree regressor is a machine learning algorithm used for solving regression problems. It works by

ISSN: 2583-6129
DOI: 10.55041/ISJEM01618

recursively partitioning the input space into regions and fitting a simple model (usually a constant value) to each partition. At each step, the algorithm selects the feature and the split point that best separates the data according to a chosen criterion, often based on reducing variance or minimizing mean squared error. This process continues until a stopping criterion is met, such as reaching a maximum depth or minimum number of samples in a node. Decision tree regressors are interpretable, easy to visualize, and can capture complex nonlinear relationships in the data, making them widely used in various domains. However, they are prone to overfitting, especially with deep trees, and may not generalize well to unseen data without appropriate regularization techniques**.**

Suitability for Question Difficulty Estimation

The Decision Tree Regressor offers a suitable approach for estimating the difficulty of the next question in educational settings for several reasons. Firstly, its inherent interpretability enables clear understanding of the rationale behind predictions, fostering transparency and explainability, which are vital in educational contexts. Secondly, its ability to capture complex non-linear relationships allows for flexible modeling of factors influencing question difficulty, accommodating the intricate interplay of various variables. Lastly, its ease of implementation and understanding make it accessible even to users with limited machine learning experience, facilitating its practical application in educational systems. Overall, the Decision Tree Regressor presents a compelling option for estimating question difficulty due to its interpretability, non-linearity, and ease of implementation.

Challenges of Decision Tree Regressor

The Decision Tree Regressor model encounters several challenges that may impact its performance and applicability in certain contexts. Firstly, it is prone to overfitting, particularly when the tree becomes deep or when training data is limited, which can result in poor generalization to unseen data. Additionally, the model exhibits instability, being sensitive to variations in training data and thereby yielding high variance in predictions. To address this, ensemble methods such as Random Forests are often employed. Moreover, Decision Tree Regressors face difficulties with continuous variables, as they may require numerous splits to effectively partition the data, leading to overly complex trees. Lastly, compared to more intricate models like neural networks, Decision Trees have limited expressiveness, potentially struggling to capture intricate patterns within the data. Despite these challenges, the Decision Tree Regressor remains a valuable tool in certain contexts, provided its limitations are carefully considered and mitigated as necessary.

IV. SYSTEM ARCHITECTURE

Fig-1 shows the system architecture for training and integrating a Decision Tree Regressor model into a user interface. It begins with the collection and preprocessing of relevant data on factors influencing question difficulty. This data is then utilized to train the Decision Tree Regressor model, employing techniques like cross-validation to ensure robust performance. Following model training, evaluation and tuning phases assess its performance, with hyperparameters adjusted as needed for optimization. Once trained and evaluated, the model is integrated into the user interface through an API or service layer, facilitating seamless interaction with users.

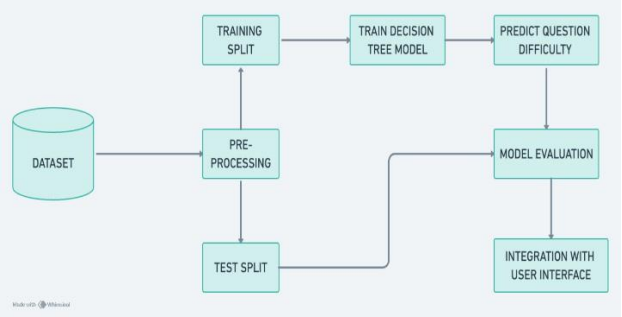

Fig-1: System Architecture for Model Training

V. IMPLEMENTATION

The system implementation involves the development of several key modules to effectively train and utilize a Decision Tree Regressor model for estimating question difficulty and assessing student proficiency levels.

Module for Training and Saving the Model

This module focuses on training and saving the model. Its primary goal is to train the Decision Tree Regressor using a provided dataset, and then persisting the trained model to disk for future use. This module takes as input a training dataset containing information on student ability, current question difficulty, answer status, and the anticipated difficulty of the next question. Upon completion, it outputs the trained model, saved in a specified file format, ensuring its availability for subsequent use.

Module for Estimating Question Difficulty

This module is dedicated to estimating question difficulty based on various factors such as student ability, current question difficulty, and answer status. Leveraging the trained Decision Tree Regressor model, this module predicts the difficulty level of the next question. It accepts inputs comprising data on student ability, current question difficulty, and answer status, delivering as output the predicted difficulty level of the subsequent question.

Module for Deploying the Model

To facilitate interaction with the trained model, a Flask-based API module serves as the interface. Its primary objective is to define endpoints that receive input data for

International Scientific Journal of Engineering and Management
Volume: 03 Issue: 05 | May – 2024
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata
ISSN: 2583-6129
DOI: 10.55041/ISJEM01618

estimating the difficulty of the next question and return the predicted difficulty level as a response. This API enables seamless integration into web applications or other systems, enhancing accessibility and usability.

Module for Question Selection

The question selection module focuses on dynamically selecting questions based on user proficiency levels and learning objectives. It utilizes the trained Decision Tree Regressor model to predict the difficulty level of potential questions, considering factors such as student ability and previous performance. Additionally, this module incorporates strategies for ensuring question diversity and relevance, optimizing the learning experience for users. By continuously adapting question selection based on user feedback and performance data, this module facilitates effective learning progression and engagement.

Module for User Interface

The user interface component is responsible for providing an intuitive and user-friendly interface for interacting with the adaptive aptitude training system. It encompasses web pages, forms, and interactive elements that enable users to access assessment features, view question recommendations, and track their learning progress. The user interface component prioritizes usability, accessibility, and responsiveness to accommodate a diverse user base.

Pseudocode

Algorithm 1 Decision Tree Regression Model

{Import required libraries}
Require: Pandas as pd
Require: `train_test_split` from `sklearn.model_selection`
Require: `DecisionTreeRegressor` from `sklearn.tree`
Require: `mean_squared_error` from `sklearn.metrics`
Require: `joblib` for model persistence

{Load data}
Step 1: Load the Dataset from the CSV File:
Read the CSV file into a DataFrame using pandas.

{Split data}
Step 2: Split the Dataset into Independent (X) and Dependent (y) Variables:
Define X as the features ('student_ability', 'curr_question_diff', 'ans_status').
Define y as the target variable ('next_question_diff').

{Split into train and test sets}
Step 3: Split the Dataset into Training and Testing Sets:
Use `train_test_split` to divide X and y into training and testing sets, with a test size of 20% and a random state of 42.

{Model creation and training}
Step 4: Create and Train the Decision Tree Model:
Initialize a `DecisionTreeRegressor` with a random state of 42.
Fit the model to the training data (X_{train}, y_{train}).

{Model prediction}
Step 5: Predict on the Testing Set:
Use the trained model to predict the target variable (y_{pred}) for the testing features (X_{test}).

{Model evaluation}
Step 6: Evaluate the Model:
Calculate the mean squared error (mse) between the actual and predicted values.

{Model persistence}
Step 7: Save the Trained Model to a File:
Use `joblib.dump` to save the trained model to a file named 'decision_tree_model.pkl'.

Fig-2: Pseudocode for Training, Testing and Saving the Model.

Fig-2 shows pseudocode for a python program that utilizes the scikit-learn library to implement a Decision Tree Regressor model for estimating the difficulty of the next question. Initially, the program loads a training dataset from a CSV file containing information on student ability, current question difficulty, answer status, and the anticipated difficulty of the next question. It then splits this dataset into independent features and the target variable. Subsequently, the dataset is divided into training and testing sets using the train_test_split function, with 80% of the data allocated for training and 20% for testing. A DecisionTreeRegressor model is instantiated and trained on the training data, followed by predictions on the testing set to evaluate model performance using mean squared error. Finally, the trained model is saved to disk using joblib for future use. This program streamlines the process of training and persisting a Decision Tree Regressor model for estimating question difficulty, offering a valuable tool for educational assessment and recommendation systems.

Algorithm 2
Step 1: Import required libraries:
- Flask for creating the web application
- request and jsonify from Flask to handle HTTP requests and responses
- joblib for loading the trained model

Step 2: Create a Flask application instance:
- Initialize the Flask application.

Step 3: Load the trained model:
- Use joblib.load to load the trained Decision Tree Regressor model from the file 'decisiontreemodel.pkl'.

Step 4: Define a route to handle prediction requests:
- Use the @app.route decorator to specify the URL endpoint ('/predict') and HTTP method ('POST') for handling prediction requests.
- Define a function predict() to process incoming prediction requests.
- Extract features from the JSON request data ('student ability', 'curr question diff', 'ans status').
- Make predictions using the loaded model based on the extracted features.
- Return the prediction as a JSON response containing the predicted 'next question diff'.

Step 5: Run the Flask application:
- Use the conditional statement `if __name__ == 'main'` to ensure the application is only executed if it's run directly (not imported as a module).
- Start the Flask application with `app.run()`.

Fig-3: Pseudocode for Deploying the Model

Fig-3 shows pseudocode for a python script that utilizes the Flask framework to deploy a RESTful API for predicting the difficulty level of the next question in an educational setting. Upon execution, the script loads the pre-trained Decision Tree Regressor model stored in a file named 'decision_tree_model.pkl' using the joblib library. It then defines a single route ('/predict') to handle HTTP POST requests containing JSON data. The predict() function extracts relevant features from the incoming JSON request, including student ability, current question difficulty, and answer status. It uses the loaded model to predict the difficulty level of the next question based on the extracted features and returns the prediction as a JSON response. Finally, the script runs the Flask application, allowing it to listen for incoming requests on the specified host and port. This Flask API streamlines the process of integrating the trained model into web applications or other systems, providing a seamless interface for estimating question difficulty in educational contexts.

ISSN: 2583-6129
DOI: 10.55041/ISJEM01618

VI. DISCUSSIONS

The development of an adaptive online aptitude training system using decision trees represents a significant advancement in the realm of personalized learning experiences for students preparing for campus placements. Through the utilization of decision tree algorithms, the system dynamically adapts to individual user responses, providing tailored assessments and targeted exercises to enhance student learning outcomes.

The implications of implementing an adaptive aptitude training system are far-reaching and multifaceted. Firstly, by offering personalized learning experiences, the system addresses the diverse needs and learning styles of students, thereby fostering a more inclusive educational environment. This approach not only enhances student engagement but also promotes deeper learning and retention of concepts.

Moreover, the integration of decision tree algorithms enables the system to make informed decisions about question selection, optimizing the learning trajectory for each user. This adaptive nature ensures that students are appropriately challenged, leading to continuous improvement in their aptitude and technical skills.

Despite its numerous benefits, the adaptive aptitude training system also has certain limitations that warrant consideration. Firstly, the effectiveness of the system relies heavily on the quality and diversity of the question bank. Limited or outdated questions may restrict the system's ability to provide relevant and challenging assessments, thereby undermining its efficacy.

Looking ahead, there are several avenues for further research and development in the field of adaptive aptitude training systems. Firstly, exploring the integration of advanced machine learning techniques, such as deep learning algorithms, could enhance the system's predictive capabilities and adaptability. By leveraging more sophisticated models, the system can better capture the nuances of student learning and provide more personalized recommendations.

Furthermore, there is potential for expanding the scope of the system beyond aptitude and technical skills to encompass broader career readiness competencies such as communication, teamwork, and problem-solving. By offering a comprehensive suite of training modules, the system can better prepare students for the multifaceted demands of the modern workforce.

VII. FUTURE RESEARCH AND DIRECTION

Future research in the realm of adaptive online aptitude training systems can explore several promising avenues for innovation and improvement. Firstly, investigating the integration of advanced machine learning techniques, such as deep learning algorithms, could enhance the system's adaptability and predictive capabilities, leading to more personalized learning experiences. Additionally, exploring the potential of incorporating real-time performance analytics and feedback mechanisms can provide valuable insights into students' learning progress and areas for improvement. This could involve the development of interactive dashboards or visualization tools to present data in a meaningful and actionable manner. These research should focus on advancing the sophistication and effectiveness of adaptive aptitude training systems to better serve the diverse needs of students preparing for campus placements.

VIII. CONCLUSION

In conclusion, the development of an adaptive online aptitude training system using decision trees offers a promising solution to the challenges faced by students preparing for campus placements. Through personalized learning experiences and targeted exercises, the system enhances student learning outcomes and fosters a more inclusive educational environment.

By leveraging decision tree algorithms, the system dynamically adapts to individual user responses, providing tailored assessments and selecting questions that match the user's proficiency level. This adaptive approach ensures that students are appropriately challenged, thereby promoting deeper learning and retention of concepts. By facilitating effective campus placement preparation, the system increases students' chances of securing desirable job opportunities and reduces the stress associated with the recruitment process.

While the system has its limitations, such as reliance on the quality and diversity of the question bank and the potential for algorithmic biases, its implications for personalized learning and career readiness are profound. With further research and development, adaptive aptitude training systems hold the potential to revolutionize the educational landscape and empower students to achieve their full potential in the competitive job market.

In summary, the adaptive online aptitude training system represents a significant step forward in personalized learning experiences for students preparing for campus placements. By harnessing the power of decision tree algorithms, the system equips students with the skills and confidence they need to succeed in the modern workforce.

AKNOWLEDGEMENT

We would like to express our cordial thanks to Dr. Madhura J, Information Science and Engineering Department, Bangalore for providing moral support, encouragement and advanced research facilities.

REFERENCES

[1] B. Keskin and M. Gunay, "A Survey On Computerized Adaptive Testing," 2021 Innovations in Intelligent Systems and Applications Conference (ASYU), Elazig, Turkey, 2021, pp. 1-6, doi: 10.1109/ASYU52992.2021.9598952.

[2] A. Aksoy, J. W. Ledet and M. Gunay, "Design Considerations of a Flexible Computer Based Assessment System," 2019 4th International Conference on Computer Science and Engineering (UBMK), Samsun, Turkey, 2019, pp. 1-6, doi: 10.1109/UBMK.2019.8907044.

[3] S. Phuvipadawat, W. Gulyanon, P. Aimmanee and T. Theeramunkong, "A comparability approach to item reduction in Computerized Adaptive Testing," 2008 4th IEEE International Conference on Management of Innovation and Technology,

ISSN: 2583-6129
DOI: 10.55041/ISJEM01618

Bangkok, Thailand, 2008, pp. 1456-1460, doi: 10.1109/ICMIT.2008.4654586.

[4] Angel Syang and Nell B. Dale. 1993. “Computerized adaptive testing in computer science: assessing student programming abilities.” SIGCSE Bull. 25, 1 (March 1993), 53–56. https://doi.org/10.1145/169073.169109

[5] Jean Piton-Gonçalves and Sandra Maria Aluísio. 2012. “An architecture for multidimensional computer adaptive test with educational purposes.” In Proceedings of the 18th Brazilian symposium on Multimedia and the web (WebMedia '12). Association for Computing Machinery, New York, NY, USA, 17–24. doi.org/10.1145/2382636.2382644

[6] R. Kavitha, A. Vijaya and D. Saraswathi, "Intelligent item assigning for classified learners in ITS using Item Response Theory and Point Biserial Correlation," 2012 International Conference on Computer Communication and Informatics, Coimbatore, India, 2012, pp. 1-5, doi: 10.1109/ICCCI.2012.6158813.

[7] H. Hirose, "An Insight to Estimated Item Response Matrix in Item Response Theory," in IEEE Access, vol. 11, pp. 82239-82247, 2023, doi: 10.1109/ACCESS.2023.3300375.

[8] Y. L. P. Vega, G. M. F. Nieto, S. M. Baldiris and J. C. G. Guevara Bolaños, "Application of item response theory (IRT) for the generation of adaptive assessments in an introductory course on object-oriented programming," 2012 Frontiers in Education Conference Proceedings, Seattle, WA, USA, 2012, pp. 1-4, doi: 10.1109/FIE.2012.6462377.

[9] Yan Xu, Quanlong Li, Huirong Dong, and Yuanlong Chen. 2020. A Learning Ability Evaluation Method Based On Item Response Theory and Machine Learning Method. In Proceedings of the 2020 3rd International Conference on Big Data and Education (ICBDE '20). Association for Computing Machinery, New York, NY, USA, 55–60. doi: 10.1145/3396452.3396455

[10] S. Tomić, V. Paunović and I. Bosnić, "Computer-based question and exam evaluation in summative knowledge assessment," 2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO), Opatija, Croatia, 2020, pp. 1520-1525, doi: 10.23919/MIPRO48935.2020.9245326.

[11] I. Hatzilygeroudis, C. Koutsojannis, C. Papavlasopoulos and J. Prentzas, "Knowledge-Based Adaptive Assessment in a Web-Based Intelligent Educational System," Sixth IEEE International Conference on Advanced Learning Technologies (ICALT'06), Kerkrade, Netherlands, 2006, pp. 651-655, doi: 10.1109/ICALT.2006.1652526.

[12] Luca Benedetto, Andrea Cappelli, Roberto Turrin, and Paolo Cremonesi. 2020. R2DE: a NLP approach to estimating IRT parameters of newly generated questions. In Proceedings of the Tenth International Conference on Learning Analytics & Knowledge (LAK '20). Association for Computing Machinery, New York, NY, USA, 412–421. doi: 10.1145/3375462.3375517

[13] Denysde Medina Sotolongo, Natalia Martínez Sánchez, and Zoila Zenaida García Valdivia. 2007. “Using artificial intelligent techniques to build adaptative tutoring systems”. In Proceedings of the 2007 Euro American conference on Telematics and information systems (EATIS '07). Association for Computing Machinery, New York, NY, USA, Article 27, 1–7. https://doi.org/10.1145/1352694.1352722

[14] Y. Park, "Personalized Practice Exam Recommendation for Helping Students Prepare for Course Assessment," 2022 International Conference on Advanced Learning Technologies (ICALT), Bucharest, Romania, 2022, pp. 390-391, doi: 10.1109/ICALT55010.2022.00120.

[15] S. Narayanan, V. S. Kommuri, S. N. Subramanian and K. Bijlani, "Question bank calibration using unsupervised learning of assessment performance metrics," 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, 2017, pp. 19-25, doi: 10.1109/ICACCI.2017.8125810.

[16] S. Narayanan, F. M. Saj, K. Bijlani and S. P. Rajan, "Automatic Assessment Item Bank Calibration for Learning Gap Identification," 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, India, 2018, pp. 1429-1435, doi: 10.1109/ICACCI.2018.8554481.

[17] Dena F. Mujtaba and Nihar R. Mahapatra. 2021. “Multi-objective optimization of item selection in computerized adaptive testing. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '21)”. Association for Computing Machinery, New York, NY, USA, 1018–1026. https://doi.org/10.1145/3449639.3459334

[18] T. Sakumura and H. Hirose, "Bias Reduction of Abilities for Adaptive Online IRT Testing Systems," *2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*, Kumamoto, Japan, 2016, pp. 450-455, doi: 10.1109/IIAI-AAI.2016.122.

