Implementation of Student feedback classification system using random forest algorithm

[1]Nandani Bharti, [2]Chetna Gayakwad, [3]Trivenee Barskar, [4]Subhangi Deshmukh, [5]Prof. Satish Chadokar, [6]Prof. sonali Rathore

[1,2,3,4]B.Tech scholar, [5,6]Assistant Professor
Department of Computer science & Engineering,
Shri Balaji Institute of Technology & Management
Betul, RGPV University M.P., India.

***Abstract*- The objective of this study is to introduce a model for analysing student feedback to enhance teaching quality in academic institutions and universities. The system integrates machine ,learning algorithms with textual feedback, which provides valuable insights into teaching quality and suggests ways to improve teaching methodology. The textual feedback, comprising comments, opinions, and reviews on teacher performance, is analysed for polarity classification (positive, negative, and neutral). This approach reduces manual effort by automating feedback collection and storage in a accessible database. Among various machine learning techniques, the Random Forest Algorithm (RFA) is employed for text classification, exhibiting high accuracy albeit requiring longer training time for large datasets. The system presents teachers with feedback analysis through graphs and ratings for effective data visualization, facilitating qualitative feedback to enhance student learning.**

Keywords: **Random Forest Algorithm, Training Data, Text Classification, Feature Extraction, Accuracy.**

I.INTRODUCTION

Student input is managed using the Online input System. Students can choose a subject and the corresponding teacher to provide comments regarding the teacher and subject using the online feedback system. A Student feedback system is a system for generating feedback that provides teachers with appropriate feedback regarding the quality of their instruction based on ratings of very poor, poor, average, good, and very good. Under the current method, students have to manually provide feedback. It takes a long time to generate reports under the current system by examining every feedback form. Report production via an online feedback system takes very little time. Through an online feedback system, students can provide feedback for a teacher of a particular subject for a set length of time, sometimes at the end of the month. The current system uses a manual procedure to handle feedback. Under the current system, students can use paper and pen to provide comments on their instructors. Following each student's input, HODs gather the papers and use them to determine the final mark for each lecturer and each subject. The principal then reviews every grade report that the HODs have provided. Thus, it is possible to rate instructors' performance and provide counselling.

II.REVIEW OF LITERATURE

P. Rana et al. [1], proposed a matrix factorization and multi-regression approach-based analyser to predict the student's performance. Initially, it was designed for analysing e-commerce applications. But it can be used to analyse students' performance. It uses a degree planner, which predicts about the students who have very poor performance and may not be able to pass the course. It also forecasts about the future courses by analysing the past performance.

A. Jain et al. [2], proposed a data mining approach based performance analysis tool. It analyses the student's learning and produces the semantic rules that can be used further in analysing the overall performance of the student for that particular course. It uses the decision tree approach for the production of semantic rules. This system uses semantic web and ontology techniques for increasing the quality of study material.

C. Kurniawan et al. [3], developed a method for evaluating the performance of teachers using sentiment analysis and opinion mining. They gathered student feedback and determined the particular teacher's strengths and weaknesses. They assessed the qualitative and quantitative data and offered a teacher's sentiment score for a school and performance by providing them with customized and appropriate academic tools and coaching.

G. Sanuvala et at. [4], to reduce the time and stress associated with reviewing student feedback while teaching. They processed automatically using sentiment analysis to get around it. In order to provide a higher level of pre-processing, they used Support Vector Machine (SVM).

F. Dalipi et al. [5], proposed a sentiment analysis-based automated evaluation system. Real time text feedback is gathered, and using supervised and semi-supervised machine learning approaches, sentiment analysis is done to identify key features.

III.METHODOLOGY

Random Forest Algorithm:

The Random Forest algorithm is a versatile ensemble method designed to address both regression and classification tasks. It achieves this by combining multiple decision trees, thereby reducing individual tree variance while maintaining predictive accuracy. This method employs Bootstrap and Aggregation (bagging), where each decision tree is trained on a random subset of the data. In classification, the final result is determined by majority voting, while in regression, it's the average of all outputs. This process, known as aggregation, ensures robust predictions. To implement Random Forest regression, one must prepare, train, and evaluate the model using test data. If performance is subpar, adjustments can be made to the model or data preprocessing techniques. Random Forest offers a flexible and powerful approach to predictive modelling, well-suited for handling large and complex datasets across various domains.

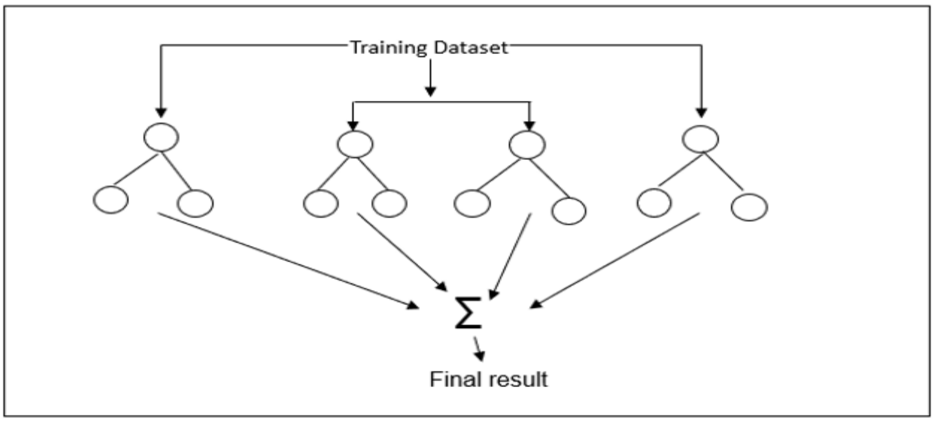


Fig. Random Forest Algorithm

Utilizing the random forest algorithm for student feedback classification involves several key steps:

1. Data Collection:

- Access student feedback data from academic institutions and universities, ensuring compliance with ethical guidelines and obtaining necessary permissions.
- Collect textual feedback, including comments, opinions, and reviews on teacher performance.

2. Data Preprocessing:

- Cleanse the raw feedback data by eliminating irrelevant characters, symbols, and noise.
- Tokenize the text into individual words or phrases.
- Apply stemming and lemmatization to standardize word forms.
- Eliminate stop words and punctuation to refine the text.

3. Feature Extraction:

- Utilize techniques like bag-of-words or TF-IDF to convert textual feedback into numerical feature vectors.
- Explore additional feature engineering methods to capture semantic and syntactic information.
- Incorporate metadata features such as course name, instructor details, and semester information for enhanced classification.

4. Random Forest Model Training:

- Divide the pre-processed dataset into training and testing sets for model evaluation.
- Implement the Random Forest algorithm using appropriate libraries, such as scikit-learn.
- Tune hyperparameters, including the number of trees, maximum depth, and minimum sample split, using cross-validation.
- Train the Random Forest model on the training data to learn the relationship between input features and feedback polarity labels.

5. Evaluation Metrics:

- Assess model performance using standard evaluation metrics like accuracy, precision, recall, and F1-score.
- Visualize classification performance across different polarity classes using confusion matrices.
- Consider additional metrics like ROC-AUC for binary classification tasks.

6. Performance Comparison:

- Compare the Random Forest classifier's performance with alternative machine learning algorithms, such as Support Vector Machines or Naive Bayes.

- Analyze computational efficiency, training time, and classification accuracy to understand the strengths and limitations of the Random Forest approach.

7. System Implementation:

- Develop a user-friendly interface for the student feedback classification system.
- Integrate the trained Random Forest model into the system backend for real-time feedback analysis.
- Ensure scalability and robustness to handle large volumes of feedback data efficiently.

8. Validation and Deployment:

- Validate the implemented system through pilot testing with stakeholders, including educators and administrators.
- Gather feedback and incorporate necessary improvements based on user suggestions.
- Deploy the finalized student feedback classification system in academic institutions and universities to support teaching quality enhancement initiatives.

9. Documentation and Reporting:

- Document the entire methodology, including data collection procedures, preprocessing techniques, model training strategies, and evaluation metrics.

Random forests are well-suited for student feedback analysis due to their ability to handle high-dimensional data effectively, resilience to overfitting, and capability to process both categorical and numerical features efficiently. Additionally, they provide feature importance scores, aiding in identifying influential factors in student feedback analysis.

IV.SYSTEM ARCHITECTURE

DIFFERENT MODULES OF THE SYSTEM

Administrator, Student, Feedback, and Educator are the four entities that make up the Student Feedback System.

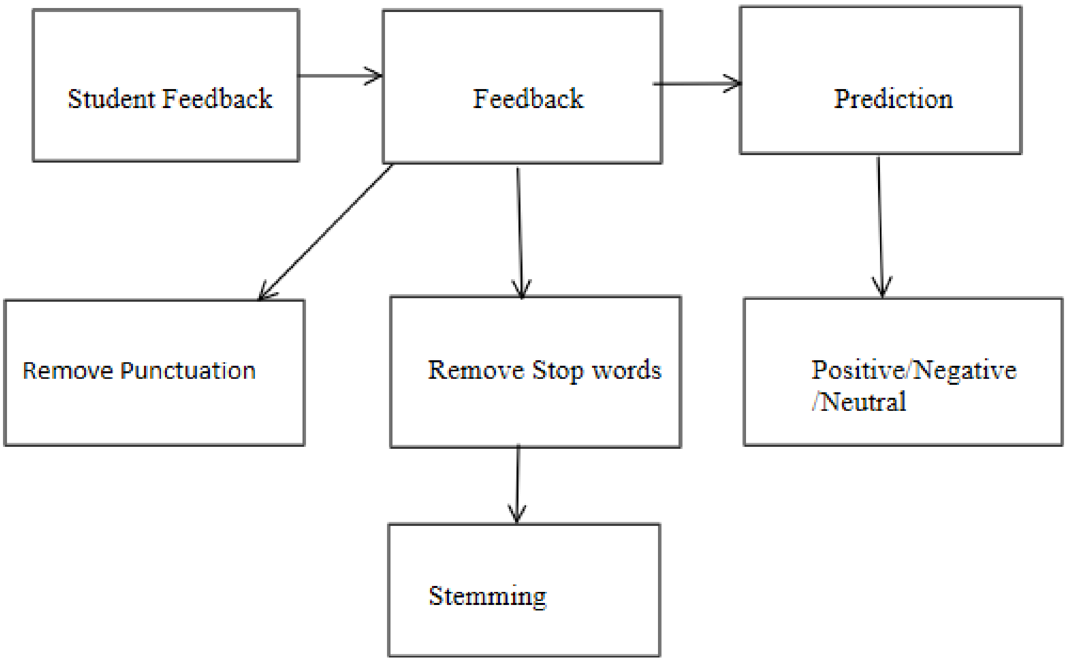


Fig. SYSTEM ARCHITECTURE

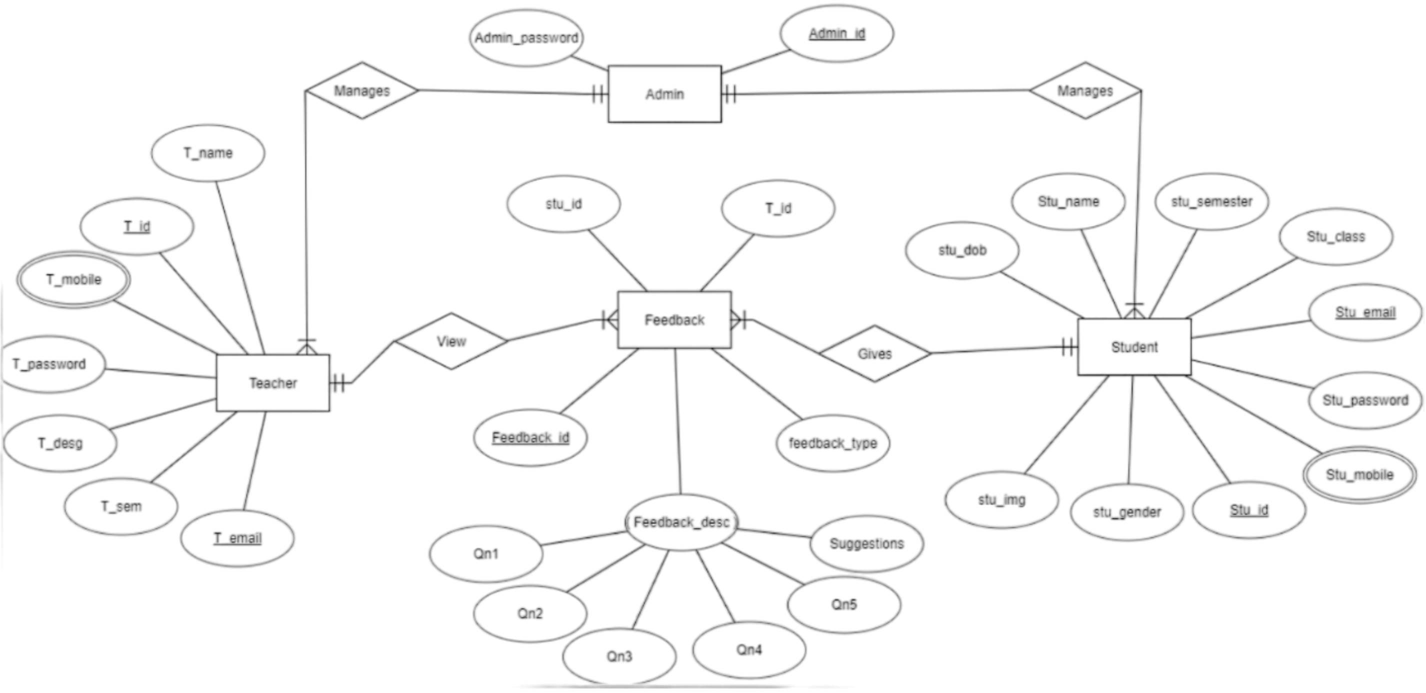
(1) Administrator: The administrator can see and edit comments from students and instructors. In the administrator relation model, the admin's id serves as the primary key. There is a "manages" link between the student and teacher entities. The administrator's cardinality ratio is 1:N with both instructors and pupils. several teachers may be added, and the administrator can review the input from several pupils.

(2) Educator: The input from the students is available to the teacher. The teacher's ID is the main component of the teacher relation schema. "Views" refers to the interaction that takes place between the educator and the feedback sources. The cardinality ratio between the teacher and the feedback object was 1:N. The instructor can read only comments made specifically to them; remarks from multiple pupils are not visible to them. One instructor does not have access to the comments provided by other teachers.

(3) Student: The teacher might receive comments from the students. In the student relation structure, the student's ID acts as the primary key. There is a "gives" link between pupils and feedback. The cardinality ratio between the student and the feedback object is 1:N. One student can send feedback to many professors at once. Students can only give feedback to professors in the same branch and semester.

(4) Feedback: Student comments are stored within the feedback object. The primary keys of the feedback object are feedback_Id, teacher_Id, and Student_Id. The feedback entity offers details on the feedback that a certain student has given to another student. The student and instructor entities are connected by it.

V. ER DIAGRAM



VI. RESULT/OUTPUT

The result of student feedback classification using the random forest algorithm is evaluated through metrics such as accuracy, precision, recall, and F1-score on a test dataset. Visualizing the confusion matrix provides detailed insights into the model's performance. These assessments help gauge the effectiveness of the algorithm in categorizing student feedback accurately. By analysing these metrics, researchers and educators can understand the strengths and weaknesses of the classification model, enabling them to make informed decisions about its implementation and potential improvements.

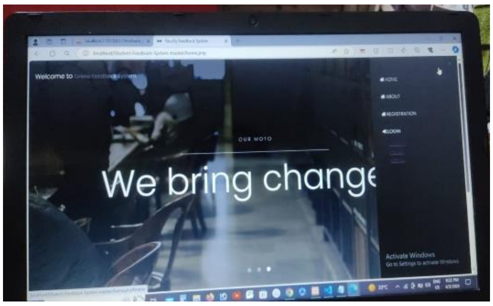

Welcome to Online Feedback System
HOME
ABOUT
REGISTRATION
LOGIN
OUR MOTO
We bring change
Activate Windows
Go to Settings to activate Windows

Fig. home page

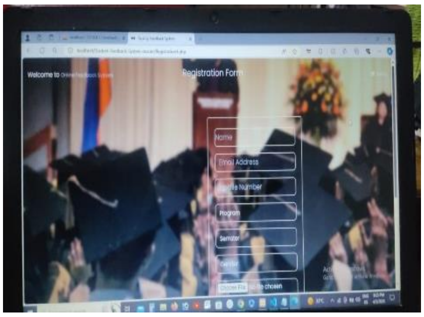


Fig. registration page

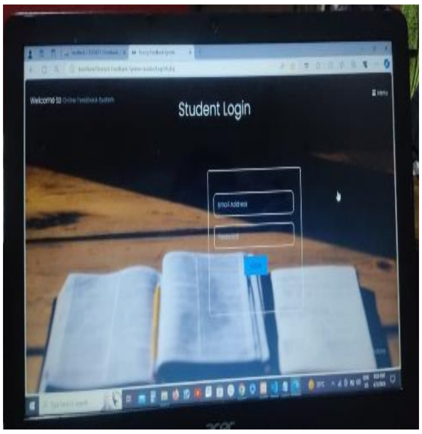


Fig. student login

Fig. Admin Login

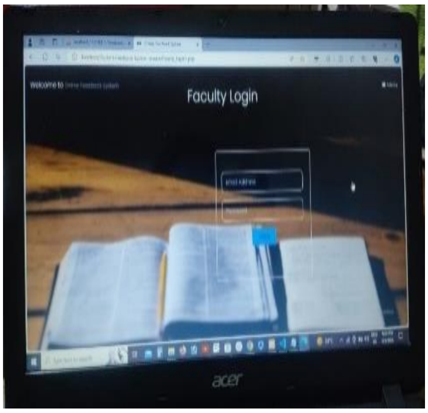


Fig. faculty login

VII. FUTURE SCOPE

The Student Feedback System project's design makes sure that the user's demands will be met in the future. Because of its flexibility, the project can readily adapt to changes without jeopardizing the present system. Due of time restrictions, the design is not perfect. The designs of the screens may yet be improved. A few additional forms can be added in order to properly gather the feedback data.

VIII. CONCLUSION

The successful implementation of this online platform for feedback is the project's achievement. It is proposed that colleges and universities implement a categorization system based on student feedback. The complete program was built using PHP and Bootstrap. When creating the table structures, several constraints were taken into account, and the open-source MySQL platform was used to store the data. procedures were considered, and many of the PHP security procedures were implemented for the project automatically. One kind of this kind of approach is session variables. The application that was created above satisfies the specifications. Enhancements to the project are easily attainable without changing the current programming structure and design.

REFERENCES:

1. Jain, A., and H. Ram Sah. "Student’s Feedback by Emotion and Speech Recognition through Deep Learning." In 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), 2021, pp. 442–447.
2. Rashid, A. "Feature Level Opinion Mining of Educational Student Feedback Data using Sequential Pattern Mining and Association Rule Mining." International Journal of Computer Applications (0975-8887) 81.10 (2020): 31-38.
3. Patel, Rajeev, et al. "Android Based Student Feedback System for Improved Teaching." International Journal of Engineering Research in Computer Science and Engineering (IJERCSE) 6.5 (2019): 100-105.
4. Gore, Bhagyashree. "Sentiment Analysis of Parent Feedback to Gauge their Satisfaction Level." March 2018.
5. Wijanarko, Bambang Dwi, et al. "Questions Classification in Online Discussion Towards Smart Learning Management System." In 2018 International Conference on Information Management and Technology (ICIMTech), Jakarta, Indonesia, 2018.
6. Abu Kassim, Rosni, et al. "Lecturers' Perspective of Student Online Feedback System: A Case Study." In 2017 IEEE 9th International Conference on Engineering Education (ICEED), Kanazawa, Japan, 2017.