A

Mini Project Report

On

PERFORMANCE ANALYSIS ON STUDENT FEEDBACK USING ML ALOGORITHMS

Submitted in partial fulfillment of the requirements for the award of Degree

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING (AI&ML)

by

Md. ARBAAZ AHMED (217R1A66A3)

M. ROHITH REDDY (217R1A6697)

P. HARI PRASAD (217R1A66A6)

Under the Guidance of

SAGGURTHI KIRAN

Assistant Professor CSE-AIML

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (AI&ML)

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, NewDelhi) Recognized Under Section 2(f) & 12(B) of the UGCAct.1956, Kandlakoya (V), Medchal Road, Hyderabad-501401.

2021-2025

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (AI&ML)

CERTIFICATE

This is to certify that the project entitled **"Performance Analysis on Student Feedback Using Machine Learning Algrithms"** being submitted by **Md. Arbaaz Ahmed (217R1A66A3), M. Rohith Reddy (217R1A6697) & P. Hari Prasad (217R1A66A6)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering (AI&ML) to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by them under our guidance and supervision during the year 2024-25.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Saggurthi Kiran
Assistant Professor
INTERNAL GUIDE

Dr. S Rao Chintalapudi
HOD CSE(AI&ML)

EXTERNAL EXAMINER

Submitted for viva voice Examination held on _____

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide **Mr Saggurthi Kiran (**Assistant Professor, CSE(AI&ML)) for her exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **Dr. G. Vinoda Reddy, Dr. K. Mahesh, Dr. Md. Shareef, Mr. A. Ramesh, Dr. V. Malsoru, Mr. M. Balaji** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. S Rao Chintalapudi,** Head, Department of Computer Science and Engineering (AI&ML) for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

MD. ARBAAZ AHMED (217R1A66A3)
M. ROHITH REDDY (217R1A6697)
P. HARI PRASAD (217R1A66A6)

ABSTRACT

Student feedback analysis is time-consuming and laborious work if it is handled manually. This study explores the use of a new deep learning-based method to design a more accurate automated system for analyzing students' feedback (called DTLP: deep learning and teaching process). The DTLP employs convolutional neural networks (CNNs), bidirectional LSTM (Bi LSTM), and attention mechanism. To the best of our knowledge, a deep learning-based method using a unified feature set, which is representative of word embedding, sentiment knowledge, sentiment shifter rules, linguistic and statistical knowledge, has not been thoroughly studied with regard to sentiment analysis of student feedback. The research employs a combination of machine learning and natural language processing on student input data collected from module analysis surveys. Comparative performance evaluations are conducted across various techniques, aiming to identify better-performing methods for various sentiment analysis.

To the best of our knowledge, a deep learning-based method using a unified feature set, which is representative of word embedding, sentiment knowledge, sentiment shifter rules, linguistic and statistical knowledge, has not been thoroughly studied with regard to sentiment analysis of student feedback

In this project we are proposing Machine learning and Natural language Processing (NLP) techniques to improve the accuracy rate of analyzing student performance. DTLP uses multiple strategies to overcome the following drawbacks: contextual polarity; sentence types; words with similar semantic context but opposite sentiment polarity; word coverage limit of an individual lexicon; and word sense variations.

LIST OF FIGURES

TABLE OF CONTENTS

1. INTRODUCTION

1. INTRODUCTION

1.1 PROJECT SCOPE

Data analytics has progressively become crucial to almost any economic development area. The scope of the involves a comprehensive examination of student feedback to evaluate academic performance and identify key trends. Technological advancement has innovated education dramatically. Among multiple ways of contributing to teaching and learning processes, educational institutions use technology as an instrument for gathering information about student experiences and assessing/adjusting their teaching approaches.

Educational institutions should provide a healthy learning environment and must create and manage learning content to support and facilitate a successful teaching and learning process. This project begins with the collection of feedback data from various sources, such as surveys and course evaluations, followed by rigorous preprocessing to clean and normalize the data.

The next steps include extracting relevant features and applying machine learning algorithms, such as sentiment analysis and topic modeling, to analyze and interpret the feedback. The project aims to evaluate the performance of different algorithms through metrics like accuracy and F1-score, and present findings through visualizations that highlight significant trends and insights.

By integrating these insights with existing performance metrics, the project seeks to provide actionable recommendations for improving educational outcomes. Research from the UK's National Student Survey (NSS) highlights the potential of machine learning techniques to process large volumes of qualitative feedback efficiently. By implementing these advanced methods, institutions can identify and address negative aspects of student experiences more promptly, leading to improved satisfaction and educational outcomes

1.2 PROJECT PURPOSE

The purpose of the project "Performance Analysis on Student Feedback Using Machine Learning Algorithms" is to enhance the quality of education by systematically analyzing student feedback. By leveraging advanced machine learning techniques, this project aims to extract actionable insights from qualitative feedback, which is often rich in detail but challenging to process manually.

Implementing this project is crucial for several reasons. Firstly, it enables educational institutions to systematically analyse large volumes of feedback efficiently, extracting valuable insights that would be time-consuming and resource-intensive to process manually. This automated analysis helps identify key areas for improvement in teaching methods and curriculum design, ultimately enhancing the quality of education. Furthermore, understanding student satisfaction and addressing their concerns promptly can significantly improve the overall student experience and retention rates.

Implementing a project focused on performance analysis of student feedback using machine learning algorithms must consider various legal frameworks to ensure compliance and data protection.The Family Educational Rights and Privacy Act (FERPA) in United States is a key law that protects the privacy of student education records.

In addition, for educational institutions, student achievement is critical. The level of satisfaction demonstrates a clear understanding of the university environment, and the services offered to students. Student satisfaction serves as a method to assess the quality of education achieved and the efficiency of the institution operations. Higher education institutions are required to change their programmes, procedures, and structures in accordance with students needs.

1.3 PROJECT FEATURES

Feature extraction and representation techniques for your project include Bag-of-Words (BoW), which represents text a a vector of word frequencies, and TF-IDF (Term Frequency-Inverse Document Frequency), which measures the importance of words in the context of the feedback corpus. When it comes to outlier detection, you might employ multivariate outlier detection techniques such as Mahalanobis Distance or Isolation Forest to identify unusual feedback patterns.

Methods like Recurrent Neural Networks (RNN) is used in analyzing the sequence of words in feedback. The data is gathered and analyzed from some of the datasets like General Text Datasets, Educational and Feedback-Specific Datasets

Training datasets are used to teach machine learning models to recognize patterns and make predictions based on student feedback. They help the model learn from labeled examples, such as sentiment or topic categories, enabling it to generalize to new, unseen feedback. High-quality and diverse training data improve the model's accuracy and robustness. Proper preprocessing and splitting into training, validation, and test sets ensure effective learning and evaluation.

The focal point is on utilizing Opinion Mining approach for categorizing the student’s feedback received all through component estimate survey that is accomplished each semester to understand comments of scholars in regards to several options of coaching and knowledge like module, teaching, assessments, and so forth. The mined and preprocessed datasets be subjected to several supervised opinion mining method like aid Vector gadget (SVM), Naive Bayes (NB), Artificial Neural Networks (NN) applied the usage of Python, the open supply device accessible for opinion mining.

2.SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

SYSTEM ANALYSIS

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, "what must be done to solve the problem?" The system is viewed as a whole and the inputs to the system are identified.

2.1 FEASIBILITY STUDY

The performance analysis of student feedback is conducted to assess the effectiveness of educational initiatives and ensure that proposed changes enhance the learning environment without becoming a burden on the institution. A comprehensive evaluation is essential, beginning with a general overview of the feedback collection process and associated cost estimates for implementing improvements.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

2.2 EXISTING SYSTEM

The Factor Analysis Model (FAM) was proposed to predict the student's performance in Intelligent Tutoring System (ITS) taking into consideration the difficulty level of assessments based on Item Response Theory concept [9] [10]. The difficulty level of tasks can infer measurement of the correlation between the student's performances and assessment questions. To compute the probability of a student solving a task correctly, a set of predictor variables are defined in the FAM including the number of opportunities presented to the student at each task, the duration spent on each step and the difficulty level of each question or latent variable. The results reveal that incorporating the latent variables into the estimates of student performance can s. ignificantly enhance the model [10].

2.2.1 EXIXTING SYSTEMS

Two predictive models were introduced. In the first model, logistic regression was used to predict whether students gained a normal or distinction certificate. In the second predictive model, logistic regression was also used to predict if students achieved certification or not. The results indicated that the number of peer assessment is the most effective feature for acquiring a distinction. The average quiz scores were considered the most reliable predictor for earning a certificate. The accuracy of distinction and normal models were reported with the percentage of 92.6% for the first model and 79.6 % for the second model, respectively[14].

The association between the Virtual Learning Environment (VLE) data and student performance has been investigated at the University of Maryland, Baltimore County (UMBC) [12]. LA used through the implementation of the Check My Activity (CMA) tool. CMA can be defined as an LA tool, which compares students VLE activities with other activities and provides lecturers frequent feedback of students' emotional states. The results showed the students who engage with the course frequently are more likely to earn mark C or higher than those who did not regularly engage problem over a minimal, definitive claim data consisting of diagnosis and procedure codes. Under this setting we tackle the problem of flagging a procedure as legitimate or fraudulent using mixtures of clinical codes along with RNN and RPCA based encodings.

However, we notice that with the re-encryption key, the proxy can transform all ciphertext of Alice no matter which keyword the ciphertext have. In this case, without Alice's delegation, Bob can still read all the message of Alice, this can be make serious security risks to the e-healthcare system. To address this issue, Weng *et al*. introduced the concept of conditional proxy re-encryption, where the re-encryption key is linked with a condition so that the delegate can only decrypt ciphertext which satisfying the special condition.

The next steps include extracting relevant features and applying machine learning algorithms, such as sentiment analysis and topic modeling, to analyze and interpret the feedback. The project aims to evaluate the performance of different algorithms through metrics like accuracy and F1-score, and present findings through visualizations that highlight significant trends and insights. By integrating these insights with existing performance metrics, the project seeks to provide actionable recommendations for improving educational outcomes. Research from the UK's National Student Survey (NSS) highlights the potential of machine learning techniques to process large volumes of qualitative feedback efficiently. By implementing these advanced methods, institutions can identify and address negative aspects of student experiences more promptly.

2.2.2 LIMITATIONS OF EXISTING SYSTEM

Following are the disadvantages of existing system:

- Making false diagnoses to justify procedures that are not medically necessary.
- Performing medically unnecessary procedures to claim insurance payments.
- Billing for each step of a procedure as if it is a separate procedure, also called “unbundling”.

2.2 PROPOSED SYSTEM

The OULAD dataset was captured from the Open University Learning Analytics Dataset (OULAD)repository. The open university in the UK delivers the online course in various topic for undergraduate and postgraduate students in the periodbetween 2013-2014. The main composite table called “studentInfo” is linked to all tables.

The "studentInfo" table includes information relevant to students’ demographic characteristics [15]. The information related to students performance are collected in “Assessments” and Student Assessment tables. The table “Assessments” contains information about the number, weight and the type of assessments required for each module. In general, each module involves a set of assessments, followed by the final exam. The assessments are Tutor Marked Assessment (TMA), Computer Marked Assessment (CMA). The final average grade is computed with the sum of all assessments (50%) and final exams (50%).comprehensive understanding of profile authenticity. Notably, our proposed system achieves an impressive accuracy of 82 percent, surpassing the performance of existing systems. This accomplishment underscores the effectiveness of the Support Vector Machine classifier and Naive Bayes.

2.3.1 PROPOSED APPROACH

To conduct a performance analysis of student feedback using machine learning algorithms, the first step is to define the objective, focusing on deriving insights about course effectiveness and student satisfaction. Data collection involves gathering feedback from surveys and evaluations, capturing both numerical ratings and textual comments, along with demographic information. Preprocessing is essential; this includes cleaning the data by removing duplicates and handling missing values, as well as processing text through tokenization and stop-word removal. Exploratory Data Analysis (EDA) helps visualize feedback distributions and identify correlations.

Selecting appropriate machine learning models depends on the data type: classification algorithms like Logistic Regression or Random Forest can predict categories, while regression models can forecast numerical ratings. Training the model involves splitting the dataset, using cross-validation for robustness, and tuning hyperparameters. Performance metrics such as accuracy, precision, and Mean Absolute Error help evaluate the model's effectiveness. The analysis should summarize findings, highlighting key factors influencing satisfaction, and provide actionable recommendations for improving teaching methods based on the results. Finally, suggestions for future work could include real-time feedback analysis or integration with learning management systems, utilizing tools like Python and visualization libraries for effective implementation.

To conduct a comprehensive performance analysis of student feedback using machine learning algorithms, the process begins by clearly defining the objective: to extract valuable insights about course effectiveness and student satisfaction. Data collection involves gathering feedback from various sources, such as end-of-course evaluations, anonymous online surveys, and mid-semester forms, capturing both quantitative ratings and qualitative comments. Preprocessing the data is essential, which includes cleaning it to remove duplicates and irrelevant entries, as well as processing textual feedback through techniques like tokenization, stop-word removal, and lemmatization. Exploratory Data Analysis (EDA) aids in visualizing data distributions and identifying trends, while sentiment analysis can categorize text responses into positive, negative, or neutral sentiments. When selecting machine learning models, classification algorithms like Random Forest and support vector machines are useful for predicting discrete outcomes, while regression models such as Linear Regression can forecast numerical ratings. Training the models involves splitting the dataset, employing cross-validation for robustness, and tuning hyperparameters to optimize performance. Evaluating the models with appropriate metrics—accuracy and F1-score for classification, and Mean Absolute Error for regression—helps assess their effectiveness. The results should be interpreted to draw actionable insights, highlighting key factors influencing student satisfaction and suggesting specific improvements. Future work could explore real-time feedback mechanisms and integration with learning management systems, utilizing tools like Python, Pandas, and machine learning frameworks such as TensorFlow or PyTorch for implementation. This structured approach provides a solid foundation for analyzing student feedback and enhancing educational outcomes.

After model selection—where choices range from classification techniques like Random Forest and Neural Networks to regression methods such as Support Vector Regression—it's essential to ensure that models are trained on diverse data splits and subjected to rigorous validation processes. This step is crucial for minimizing overfitting and improving model generalizability. Evaluating model performance requires a nuanced understanding of metrics; for instance, while accuracy provides a broad overview, precision and recall are particularly important for assessing how well models identify specific areas of concern in student feedback.

Interpretation of the results should be approached with a focus on actionable outcomes, drawing connections between feedback trends and potential curriculum or instructional adjustments. For instance, if analysis reveals consistent dissatisfaction with certain course elements, targeted improvements can be suggested. Additionally, incorporating advanced machine learning techniques, such as ensemble methods or unsupervised learning for clustering student sentiments, can unveil deeper insights. Future directions might include integrating real-time feedback systems that allow for ongoing adjustments and improvements, enhancing student engagement and learning experiences. Utilizing tools like Jupyter Notebooks for interactive data analysis and visualization, along with libraries like TensorFlow for building sophisticated models, will significantly streamline the project and yield impactful results in educational settings.

2.3.2 ADVANTAGES OF THE PROPOSED SYSTEM

The proposed system implemented using the machine learning techniques, the proposed system is processing in the following way.

Data-Driven Insights

Machine learning algorithms can analyze large volumes of student feedback data to identify patterns and trends that might not be apparent through manual analysis. This enables educational institutions to make informed decisions based on empirical evidence.

Enhanced Accuracy

ML models can improve the accuracy of sentiment analysis and feedback categorization. By using techniques like natural language processing, the system can effectively classify and interpret nuanced student sentiments, leading to more precise.

Real-Time Feedback Processing

A machine learning approach allows for the real-time analysis of student feedback, enabling institutions to respond quickly to emerging issues. This can help improve student experiences proactively rather than reactively.

Scalability

As student populations grow, machine learning systems can scale to handle increasing amounts of feedback data without significant changes to the underlying processes, making them highly efficient.

Customized Learning Experiences

By analyzing feedback at a granular level, institutions can tailor their courses and teaching methods to better meet the needs of their students, thereby enhancing the overall learning experience.

Improved Resource Allocation

Insights from the analysis can guide administrators in allocating resources more effectively, such as focusing support services on areas where students express the most concern.

2.4 HARDWARE & SOFTWARE REQUIREMENTS

4.1 HARDWARE REQUIREMENTS:

Hardware interfaces specify the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

- Processor - Pentium –IV or higher
- RAM - 4 GB (min)
- Hard Disk - 20 GB
- Key Board - Standard Windows Keyboard
- Mouse - Two or Three Button Mouse

4.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software

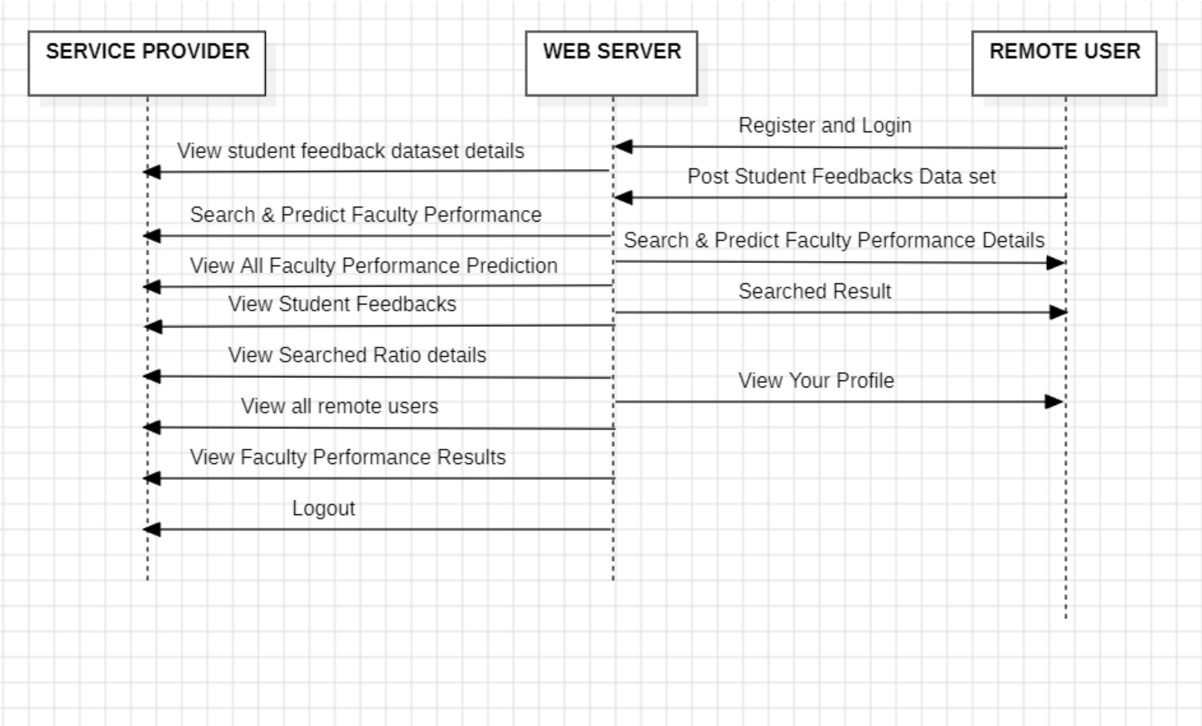
- OPERATING SYSTEM : Windows 7 Ultimate
- CODE LANGUAGE : Python
- FRONT-END : Python
- BACK-END : Django-ORM
- DESIGNING : HTML, CSS, JavaScript
- DATABASE : MySQL
- WEB SERVER : XAMPP Server

3. ARCHITECTURE

3.ARCHITECTURE

3.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed for classification, starting from input to final prediction.



3.1: Project Architecture of for Performance Analysis on Student Feedback using Machine learning Algorithm

DESCRIPTION

Two sets of experiments are conducted in this case study. In the first experiment, the dynamic behavioral features are considered to predict student performance, while the static behavioral attributes are employed in the second experiment. The problems are formulated as classification and regression. The regression setting is considered when we aim to predict students' assessments grades, whereas classification setting is utilised when we seek to predict final student performance in the entire course.

The performance analysis of student feedback using machine learning algorithms comprises several key components that work together to provide comprehensive insights. First, data collection is crucial, where feedback is gathered from various sources such as surveys, course evaluations, and online platforms, ensuring a rich dataset that includes both quantitative ratings and qualitative comments. Next, data preprocessing involves cleaning and preparing the data, which includes removing duplicates, handling missing values, and applying text processing techniques like tokenization and sentiment analysis to enhance the quality of the dataset. Following this, exploratory data analysis (EDA) is performed to visualize trends and patterns within the feedback, helping to identify key areas of concern and student sentiments. The model selection phase involves choosing appropriate machine learning algorithms tailored to the specific goals of the analysis, whether for classification or regression tasks. This is complemented by model training and validation, where the data is split into training and testing sets, and techniques like cross-validation are employed to ensure the model's robustness.

Once the model is trained, the performance evaluation phase utilizes metrics such as accuracy, precision, recall, and F1-score for classification, or Mean Absolute Error for regression, providing a quantitative measure of the model's effectiveness. Finally, the interpretation of results is essential, as it translates the model's outputs into actionable insights for educators and administrators. This structured component view ensures that the analysis not only identifies areas for improvement but also facilitates data-driven decision-making to enhance the overall student experience.

As discussed previously, the student should participate in five CMA assessments and six TMA assessments, in addition to the final exam. The assessments should be handed in within a specific period. Due to the TMA assessment weighing 45% of the final result, while the CMA assessment weighs only 5%, our temporal analysis is based on the submission date of the TMA. In first set experiments, student performance is predicated in a timely manner, as can be seen, in Figure 1 the course is subsequently the into six-time intervals, corresponding with assessment submission dates. The student behavioral records are distributed according to the assessment date.

In the second set of experiments, we evaluated the trajectories student performance by aggregate the student's behavioral activities across the six-time slices into a single time slice. The behavioral features, demographic features and temporal features are used as input variables. We did not account for past assessments grade, and final exam mark as target class is computed based on these features. The dataset contains 4004 records where the proportion of "fail", "withdrawn" and "pass" classes are 28%t,40% and 32% respectively.

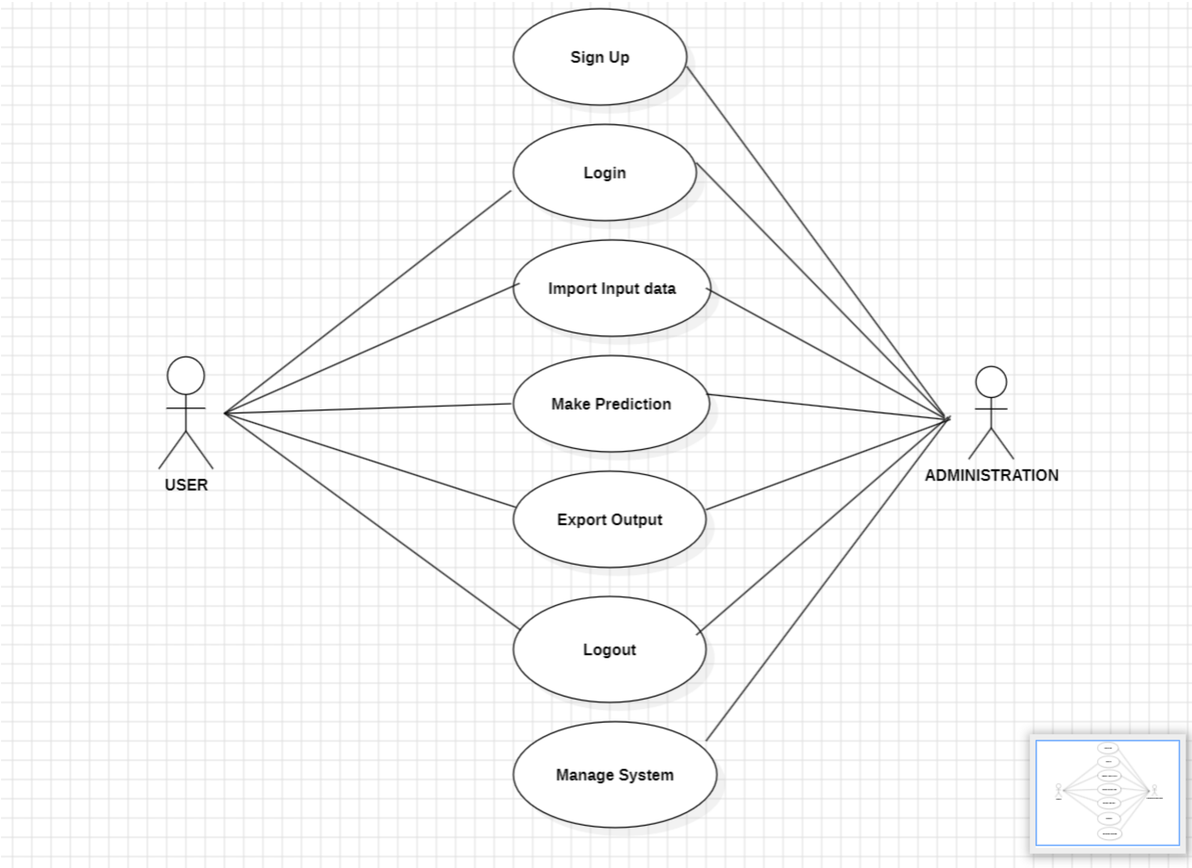
The final component, results interpretation, is where the insights derived from the model are translated into actionable recommendations for educators and administrators. This may involve identifying specific areas for improvement in courses or teaching methods, leading to targeted interventions aimed at enhancing the overall student experience.

Together, these components create a comprehensive system that not only identifies issues within student feedback but also facilitates continuous improvement through data-driven decision-making. This holistic approach ensures that educational institutions can adapt to student needs effectively, fostering an environment of engagement and satisfaction.

3.2 USE CASE DIAGRAM

In the use case diagram, we have basically one actor who is the user in the trained model.

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures.



3.2: Use Case Diagram for Performance Analysis on Student Feedback using Machine learning Algorithm

DESCRIPTION

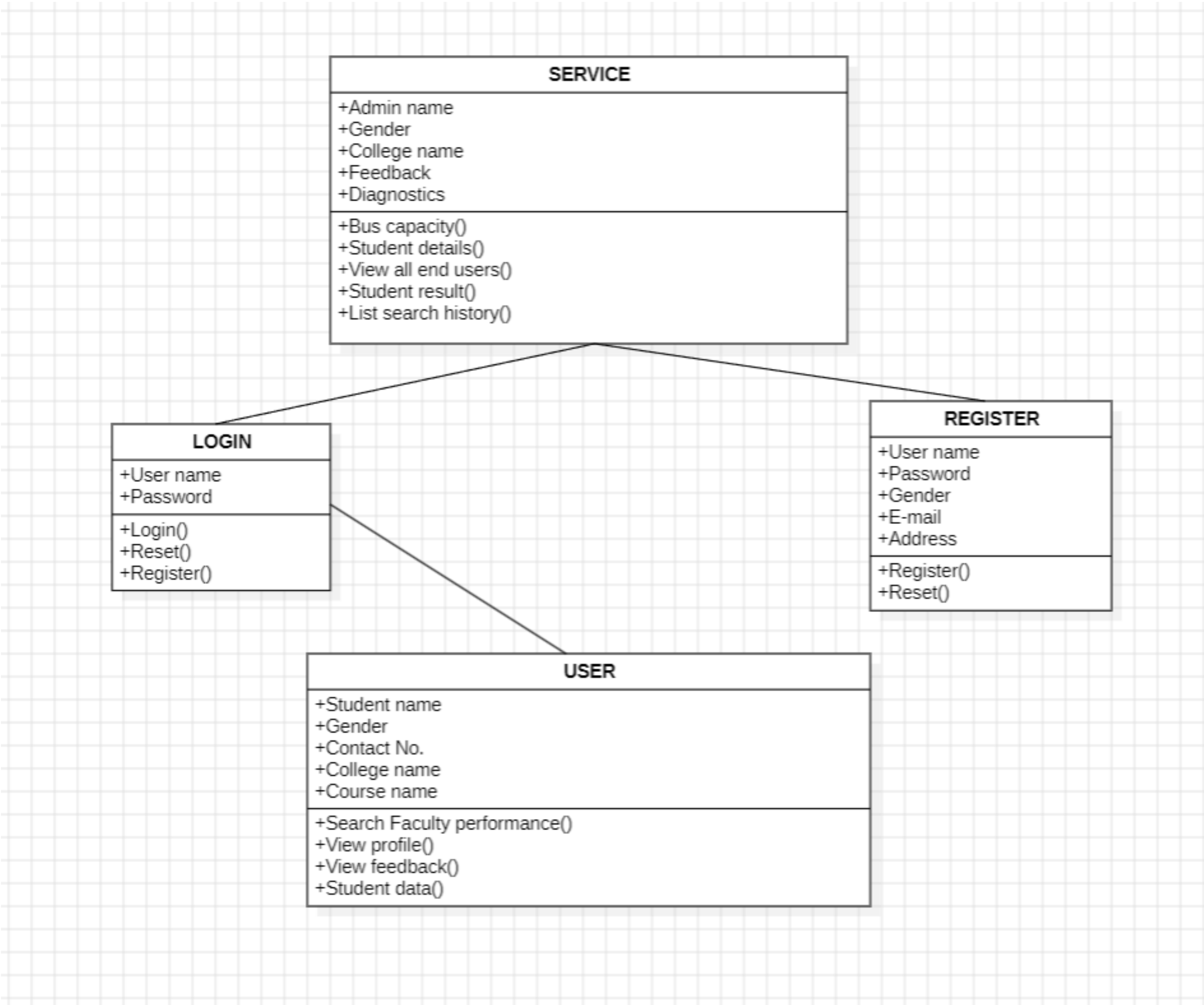
The use case diagram encapsulates the functionalities available to the service provider and remote user entities. For the service provider, a set of methods is defined to facilitate access to train and test user profile datasets, view accuracy results in a bar chart, examine detailed accuracy outcomes, register and log in to their account, predict fraudulent user identification statuses, view their own profile, explore all predicted user identification results, find and view specific prediction ratio results, inspect their own profile identity ratio results, and download predicted datasets. These methods collectively empower the service provider to interact comprehensively with the system, from data access and analysis to user management.

On the other hand, the remote user class is equipped with methods for registering and logging into their account, viewing their own profile, predicting profile identification statuses, and obtaining an overview of all registered remote users. The simplicity of the remote user's functionality aligns with the limited scope of operations expected from this entity.
Within the use case diagram, a clear relationship is established between the service provider and the remote user.

This association allows the service provider to view all registered remote users, fostering a collaborative environment within the system. This structural depiction offers a concise overview of the system's architecture, highlighting the essential functionalities available to both service providers and remote users, and the relationships that facilitate a seamless interaction between these entities.

3.2 CLASS DIAGRAM

Class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.



3.3: class diagram for performance analysis on student feedback using machine learning algorithms

DESCRIPTION

The class diagram for the project outlines the core components and their interactions within the system. At the center of the diagram is the Student class, which includes attributes such as studentID, name, and email to uniquely identify and store basic information about each student. Associated with the students is the Feedback class, which contains feedbackID, studentID, courseID, and content attributes to capture the feedback details related to specific courses.

The Course class, comprising attributes like courseID, courseName, and instructor, structurally represents the academic offerings and their corresponding instructors. The pivotal FeedbackAnalyzer class integrates machine learning algorithms to process and derive insights from the feedback data. Its methods are instrumental in converting unstructured feedback into quantifiable performance metrics..

In the result, the class diagram demonstrates an effective design for analyzing student feedback through machine learning. It facilitates the integration of student data, feedback, and course information, enabling detailed performance insights. The system's architecture supports informed decision-making and enhances educational outcomes, ultimately driving progress.

Once the model is selected, the process moves to model training and validation, which involves splitting the dataset into training and testing subsets. Techniques such as cross-validation help ensure the model's robustness and prevent overfitting. After training, the model’s performance is evaluated using metrics tailored to the specific task at hand, such as accuracy, precision, recall, and F1-score for classification tasks or Mean Absolute Error for regression.

3.4 SEQUENCE DIAGRAM

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the logical view of the system under development.

SEQUENCE DIAGRAM FOR DSAS A SECURE DATA SHARING AND AUTHORIZED SEARCHABLE FRAMEWORD FOR e-HEALTHCARE SYSTEM

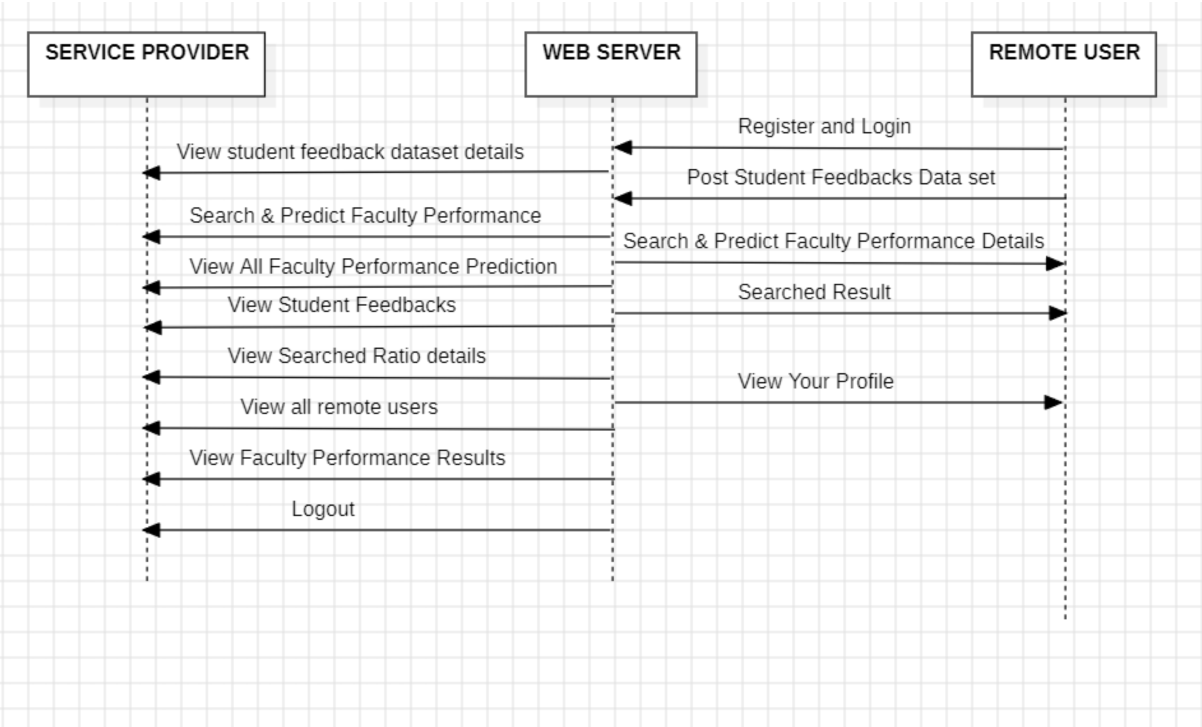


Figure 3.4 Sequence Diagram for Performance Analysis on Student Feedback using Machine learning Algorithm

DESCRIPTION

The sequence diagram delineates the interaction and flow of messages between entities in the system, specifically focusing on the service provider, web server for building fraudulent health insurance claim identification, remote user, and the web server for identifying fraudulent claims. Let's elaborate on this sequence in two paragraphs.

The sequence initiates with the Service Provider sending a request to the Web Server for Building Fraudulent health insurance claim identification, signalling the commencement of the fraudulent health insurance claim identification process. This entails tokenization, stop-word removal, stemming, and lemmatization, all of which are pivotal for refining the dataset for subsequent analysis. Once the data is prepared, the web server employs a Support Vector Machine (SVM) algorithm to train the system on the authentic profiles, enhancing its ability to distinguish between genuine and fraudulent claims. This training process is encapsulated in a sequence of messages exchanged between the Service Provider and the Web Server for Building Fraudulent health insurance claim identification, illustrating a collaborative effort in the system's construction.

Subsequently, the sequence transitions to the identification phase, where a Remote User initiates a request to the Web Server for Identifying Fraudulent claims. This prompts the web server to retrieve the relevant datasets and pre-process the user's input profile using the same NLP techniques employed during the training phase.The result of this prediction is relayed back to the Remote User, providing them with valuable insights into the legitimacy of the profile in question.

4. IMPLEMENTATION

4.1 SUPPORT VECTOR MACHINE

In classification tasks a discriminant machine learning technique aims at finding, based on an independent and identically distributed (iid) training dataset, a discriminant function that can correctly predict labels for newly acquired instances. Less powerful than generative approaches, which are mostly used when prediction involves outlier detection, discriminant approaches require fewer computational resources and less training data, especially for a multidimensional feature space and when only posterior probabilities are needed. From a geometric perspective, learning a classifier is equivalent to finding the equation for a multidimensional surface that best separates the different classes in the feature space.

SVM is a discriminant technique, and, because it solves the convex optimization problem analytically, it always returns the same optimal hyperplane parameter—in contrast to genetic algorithms (GAs) or perceptrons, both of which are widely used for classification in machine learning. For perceptrons, solutions are highly dependent on the initialization and termination criteria. The aim of GAs and perceptrons is only to minimize error during training, which will translate into several hyperplanes' meeting this requirement.

4.2 NAÏVE BAYES

The Naïve Bayes approach is a supervised learning method which is based on a simplistic hypothesis: it assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature.
Yet, despite this, it appears robust and efficient. Its performance is comparable to other supervised learning techniques. Various reasons have been advanced in the literature. The naive bayes classifier is a linear classifier, as well as linear discriminant analysis, logistic regression or linear SVM (support vector machine). The difference lies on the method of estimating the parameters of the classifier (the learning bias).

While the Naive Bayes classifier is widely used in the research world, it is not widespread among practitioners which want to obtain usable results. On the one hand, the researchers found especially it is very easy to program and implement it, its parameters are easy to estimate, learning is very fast even on very large databases.

4.3 K-NEAREST NEIGHBORS (KNN)

- Simple and a very powerful classification algorithm
- Classifies based on a similarity measure
- Non-parametric
- Lazy learning
- Does not "learn" until the test example is given
- Whenever we have a new data to classify, we find its K-nearest neighbors from the training data

Example:

- Feature space means, space with categorization variables (non-metric variables)
- Learning based on instances, and thus also works lazily because instance close to the input vector for test or prediction may take time to occur in the training dataset
- Training dataset consists of k-closest examples in feature space.

4.4 DECISION TREE CLASSIFIERS

Decision tree classifiers are used successfully in many diverse areas. Their most important feature is the capability of capturing descriptive decision-making knowledge from the supplied data. Decision tree can be generated from training sets. The procedure for such generation based on the set of objects (S), each belonging to one of the classes C1, C2, …, Ck is as follows:

Step 1: If all the objects in S belong to the same class, for example Ci, the decision tree for S consists of a leaf labeled with this class

Step 2: Otherwise, let T be some test with possible outcomes O1, O2,…, On. Each object in S has one outcome for T so the test partitions S into subsets S1, S2,… Sn where each object in Si has outcome Oi for T.

Gradient boosting is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees.

4.5 LOGISTIC REGRESSION CLASSIFIERS

Logistic regression analysis studies the association between a categorical dependent variable and a set of independent (explanatory) variables. The name logistic regression is used when the dependent variable has only two values, such as 0 and 1 or Yes and No. The name multinomial logistic regression is usually reserved for the case when the dependent variable has three or more unique values, such as Married, Single, Divorced, or Widowed. Although the type of data used for the dependent variable is different from that of multiple regression, the practical use of the procedure is similar.

This program computes binary logistic regression and multinomial logistic regression on both numeric and categorical independent variables. It reports on the regression equation as well as the goodness of fit, odds ratios, confidence limits, likelihood, and deviance. It performs a comprehensive residual analysis including diagnostic residual reports and plots. It can perform an independent variable subset selection search, looking for the best regression model with the fewest independent variables.

4.6 RANDOM FOREST

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned. Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance
The first algorithm for random decision forests was created in 1995 by Tin Kam Ho using the random subspace method, which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg.

An extension of the algorithm was developed by Leo Breiman and Adele Cutler, who registered "Random Forests" as a trademark in 2006 (as of 2019, owned by Minitab, Inc.).The extension combines Breiman's "bagging" idea and random selection of features, introduced first by Ho [1] and later independently by Amit and Geman [13] in order to construct a collection of decision trees with controlled variance. Random forests are frequently used as "blackbox" models in businesses, as they generate reasonable predictions across a wide range of data while requiring little configuration.

4.7 SAMPLE CODE

```
from django.db.models import  Count, Avg
from django.shortcuts import render, redirect
from django.db.models import Count
from django.db.models import Q
import datetime



# Create your views here.
from Remote_User.models import
ClientRegister_Model,review_Model,student_performance_model,recommend_Model,perfor
mance_ratio_model,search_ratio_model



def serviceproviderlogin(request):
    if request.method  == "POST":
        admin = request.POST.get('username')
        password = request.POST.get('password')
        if admin == "SProvider" and password =="SProvider":
            performance_ratio_model.objects.all().delete()
            search_ratio_model.objects.all().delete()
            return redirect('View_Remote_Users')

    return render(request,'SProvider/serviceproviderlogin.html')
```

```
def viewtreandingquestions(request,chart_type):
    dd = {}
    pos,neu,neg =0,0,0
    poss=None
    topic =
student_performance_model.objects.values('ratings').annotate(dcount=Count('ratings')).order
_by('-dcount')
    for t in topic:
        topics=t['ratings']

pos_count=student_performance_model.objects.filter(topics=topics).values('names').annotate
(topiccount=Count('ratings'))
        poss=pos_count
        for pp in pos_count:
            senti= pp['names']
            if senti == 'positive':
                pos= pp['topiccount']
            elif senti == 'negative':
                neg = pp['topiccount']
            elif senti == 'nutral':
                neu = pp['topiccount']
        dd[topics]=[pos,neg,neu]
    return
render(request,'SProvider/viewtreandingquestions.html',{'object':topic,'dd':dd,'chart_type':cha
rt_type})

def Search_Student_Performance(request): # Search

   if request.method == "POST":
        kword = request.POST.get('keyword')
        print(kword)
        obj =
student_performance_model.objects.all().filter(Q(Enrollment_No_contains=kword) |
Q(names_contains=kword))
```

```
        obj1 = student_performance_model.objects.get(Q(Enrollment_No_contains=kword) | Q(names_contains=kword))
        Diagnostic_Assessments_Grade = obj1.Diagnostic_Assessments_Grade
        Formative_Assessments_Grade = obj1.Formative_Assessments_Grade
        Interim_Assessments_Grade=obj1.Interim_Assessments_Grade
        Summative_Assessments_Grade=obj1.Summative_Assessments_Grade

grade = ((Diagnostic_Assessments_Grade+Formative_Assessments_Grade+Interim_Assessments_Grade+Summative_Assessments_Grade)/28)*100

        if grade != 0:
            search_ratio_model.objects.create(names=kword, ratio=grade)
        return render(request, 'SProvider/Search_Student_Performance.html', {'objs': obj,'ratio': grade})
    return render(request, 'SProvider/Search_Student_Performance.html')
def View_All_StudentPerformance_Prediction_Details(request):
    sname = ''
    Eno = ''
    gender = ''
    cname = ''
    dname = ''
    collegename =''
    obj1 = student_performance_model.objects.values('names',
                                         'Enrollment_No',
                                         'Gender',
                                         'Course_Name',
                                         'Degree_Name',
                                         'College_Name',
                                         'Diagnostic_Assessments_Grade',
                                         'Formative_Assessments_Grade',
                                         'Interim_Assessments_Grade',
                                         'Summative_Assessments_Grade'
                                         )
```

```
    performance_ratio_model.objects.all().delete()
    for t in obj1:
        sname = t['names']
        Eno = t['Enrollment_No']
        gender = t['Gender']
        cname = t['Course_Name']
        dname = t['Degree_Name']
        collegename = t['College_Name']
        Diagnostic_Assessments_Grade = t['Diagnostic_Assessments_Grade']
        Formative_Assessments_Grade = t['Formative_Assessments_Grade']
        Interim_Assessments_Grade = t['Interim_Assessments_Grade']
        Summative_Assessments_Grade = t['Summative_Assessments_Grade']
        performance = ((Diagnostic_Assessments_Grade + Formative_Assessments_Grade +
Interim_Assessments_Grade + Summative_Assessments_Grade) / 28) * 100

performance_ratio_model.objects.create(names=sname,ENo=Eno,Gender=gender,Course_Na
me=cname,Degree_Name=dname,College_Name=collegename,perfromance=performance)

    obj = performance_ratio_model.objects.all()

    return render(request, 'SProvider/View_All_StudentPerformance_Prediction_Details.html',
{'objs': obj})

def View_Remote_Users(request):
    obj=ClientRegister_Model.objects.all()
    return render(request,'SProvider/View_Remote_Users.html',{'objects':obj})

def ViewTrendings(request):
    topic                                                                  =
student_performance_model.objects.values('topics').annotate(dcount=Count('topics')).order_b
y('-dcount')
    return  render(request,'SProvider/ViewTrendings.html',{'objects':topic})
```

```
def negativechart(request,chart_type):
    dd = {}
    pos, neu, neg = 0, 0, 0
    poss = None
    topic                                                                       =
student_performance_model.objects.values('ratings').annotate(dcount=Count('ratings')).order
_by('-dcount')
    for t in topic:
        topics = t['ratings']
        pos_count                                                               =
student_performance_model.objects.filter(topics=topics).values('names').annotate(topiccount
=Count('ratings'))
        poss = pos_count
        for pp in pos_count:
            senti = pp['names']
            if senti == 'positive':
                pos = pp['topiccount']
            elif senti == 'negative':
                neg = pp['topiccount']
            elif senti == 'nutral':
                neu = pp['topiccount']
        dd[topics] = [pos, neg, neu]
    return
render(request,'SProvider/negativechart.html',{'object':topic,'dd':dd,'chart_type':chart_type})

def charts(request,chart_type):
    chart1 = search_ratio_model.objects.values('names').annotate(dcount=Avg('ratio'))
    return render(request,"SProvider/charts.html", {'form':chart1, 'chart_type':chart_type})

def charts1(request,chart_type):
    chart1                                                                      =
performance_ratio_model.objects.values('names').annotate(dcount=Avg('perfromance'))
    return render(request,"SProvider/charts1.html", {'form':chart1, 'chart_type':chart_type})
```

```
def View_Student_Performance_Details(request):
    obj =student_performance_model.objects.all()
    return render(request, 'SProvider/View_Student_Performance_Details.html', {'list_objects': obj})

def likeschart(request,like_chart):
    charts =performance_ratio_model.objects.values('names').annotate(dcount=Avg('perfromance'))
    return render(request,"SProvider/likeschart.html", {'form':charts, 'like_chart':like_chart})

def View_Students_Assessments_Grades(request):
    obj = student_performance_model.objects.all()
    return render(request, 'SProvider/View_Students_Assessments_Grades.html', {'list_objects': obj})

def View_Search_Ratio(request):
    obj = search_ratio_model.objects.all()
    return render(request, 'SProvider/View_Search_Ratio.html', {'list_objects': obj})
```

4.8 DATASET DESCRIPTION

Our dataset, sourced from the Kaggle website, encompasses a diverse range of attributes aimed at discerning the authenticity of user profiles, comprising of 2676 instances. include identifiers such as ID, name, and screen name, facilitating unique identification and tracking of individual profiles. Additionally, categorical attributes like default profile, which indicates whether a user has customized their profile layout, and location, denoting the geographical information provided by the user, contribute to understanding user behavior and characteristics. These attributes serve as valuable features for discerning genuine profiles from fake ones, offering insights into user engagement and interaction patterns.

Moreover, quantitative attributes like favorites count, statuses count, followers count, friends count, and favorites count provide quantitative measures of user activity and engagement on the platform. These metrics offer valuable signals regarding the level of activity, popularity, and interaction of a user within the platform's ecosystem. Additionally, attributes such as profile image URL, profile banner URL, and profile background image URL provide insights into the visual elements associated with user profiles, which may influence user perception and credibility. By leveraging these diverse attributes, our dataset offers a comprehensive view of user profiles, enabling robust analysis and classification of genuine and fraudulent claims.

4.9 PERFORMANCE METRICS

ACCURACY

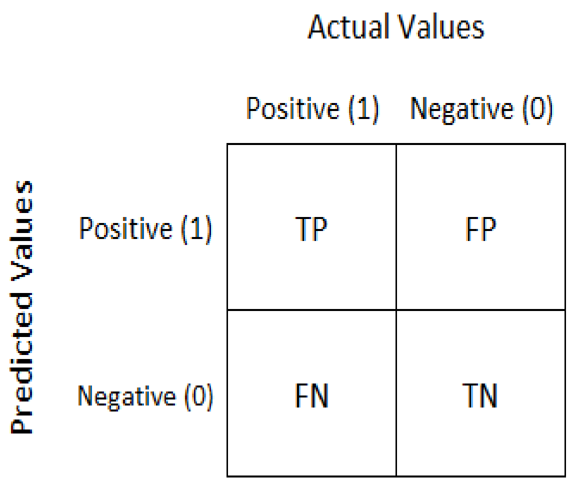
Accuracy is a crucial performance metric used in fraudulent health insurance claim identification systems to assess the overall correctness of classifications. It represents the proportion of correctly identified insurance claims, whether genuine or fake, among all users in the dataset. In the context of fraudulent health insurance claim identification, accuracy measures the system's ability to correctly classify users without distinction between false positives (genuine profiles misclassified as fake) and false negatives (fraudulent claims misclassified as genuine). A high accuracy score indicates that the system effectively distinguishes between genuine and fraudulent claims, providing confidence in its reliability.

Accuracy= (Number of correctly identified fraudulent claims/ Total number of profiles) ×100%

CONFUSION MATRIX

The confusion matrix is a fundamental tool in evaluating the performance of classification models in fraudulent health insurance claim identification. It provides a tabular representation that summarizes the performance of a classification algorithm by comparing predicted class labels with actual class labels. In the context of fraudulent health insurance claim identification, the confusion matrix comprises two classes: genuine profiles (often labeled as 0) and fraudulent claims (labeled as 1). The matrix consists of four quadrants: true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN).

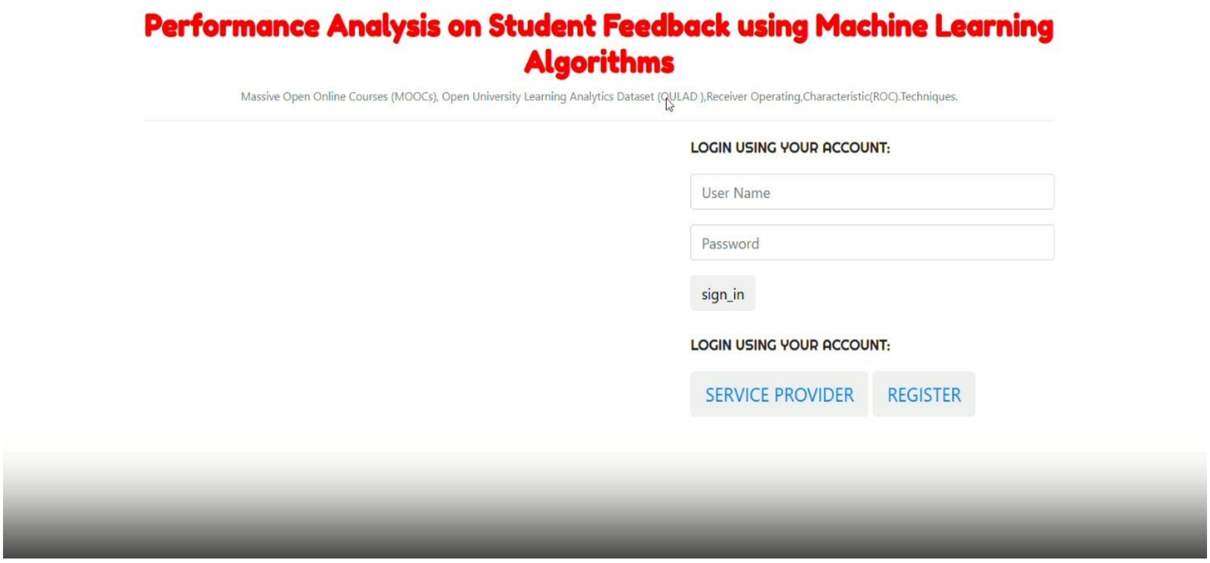
True positives (TP) represent the number of fraudulent claims correctly identified by the model. These are instances where the model correctly predicts a profile as fake when it is indeed fake. False positives (FP) indicate the number of genuine profiles incorrectly classified as fake by the model. These occur when the model mistakenly predicts a genuine profile as fake. True negatives (TN) represent the number of genuine profiles correctly identified as genuine by the model. These instances occur when the model correctly predicts a genuine profile as genuine. Lastly, false negatives (FN) denote the number of fraudulent claims incorrectly classified as genuine by the model.



4.1: Confusion matrix

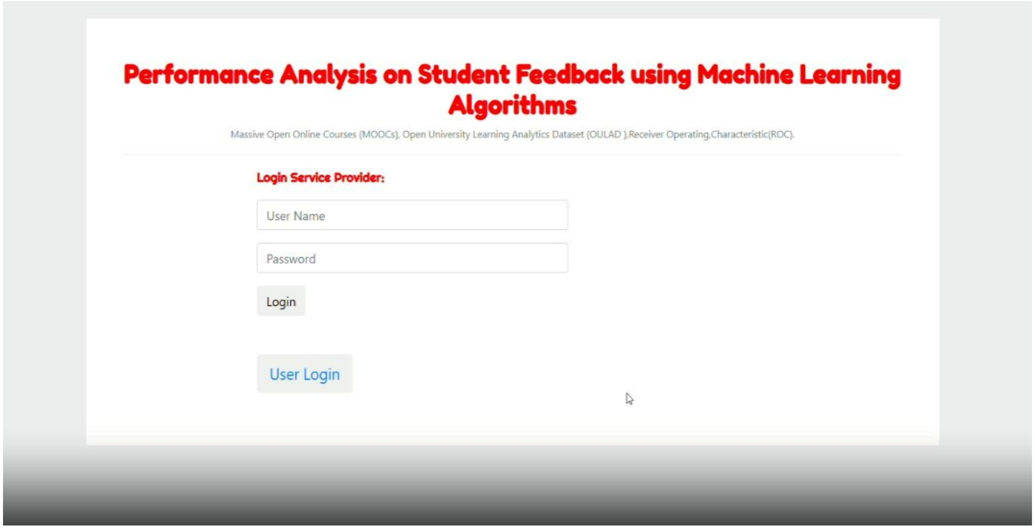
5. SCREENSHOTS

SCREENSHOTS



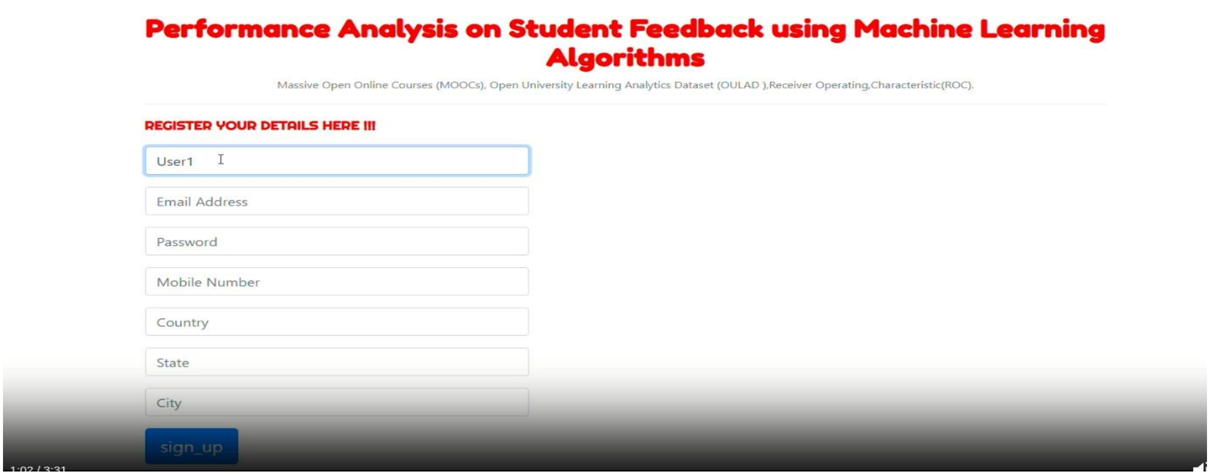
5.1: Login page

The project's homepage interface serves as the gateway for users, offering a seamless login experience. Users input their credentials in designated fields, ensuring secure access to the platform. With a focus on user-friendly design and robust security measures, the interface sets the stage for a positive user interaction.



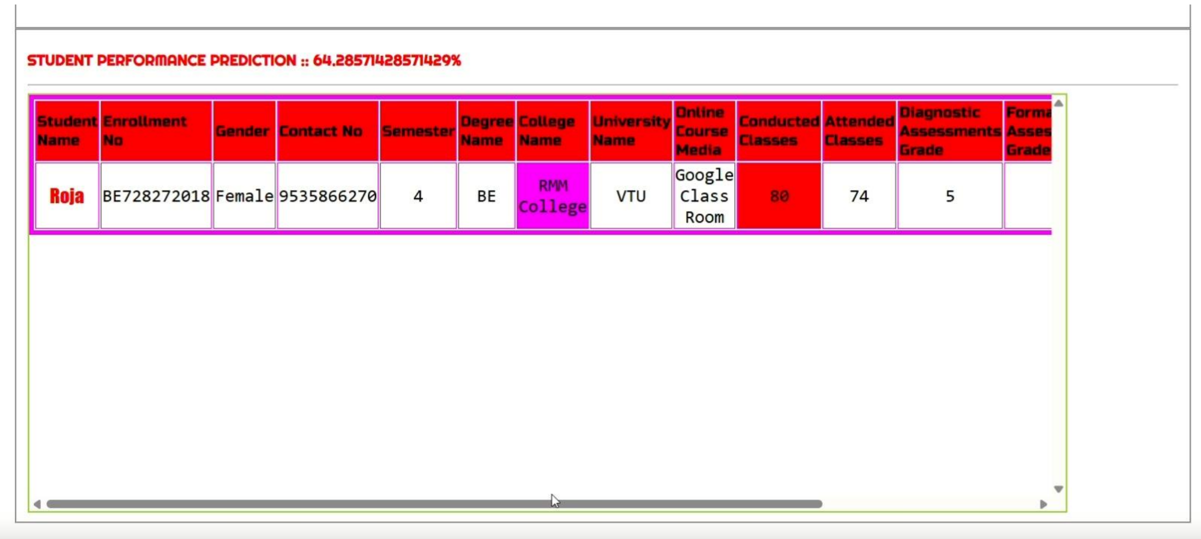
5.2: Service Provider login page

The Alice login page facilitates secure access for providers using their credentials. Users enter their login details in the designated fields, ensuring a streamlined and authenticated experience. With a focus on security and user-friendly design, the interface enhances the service provider's login process.



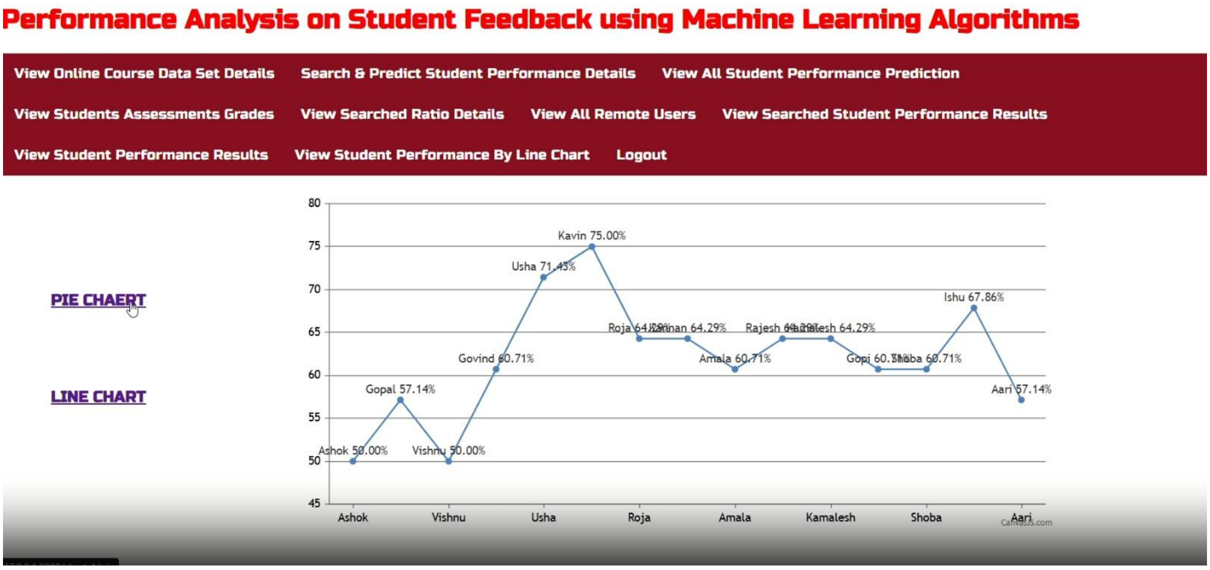
5.3: User login page

The Bob login page allows new users to sign up by providing necessary details. Users input their information in the designated fields, ensuring a straightforward and secure registration process. With an emphasis on simplicity and data protection, the interface enhances the user's experience during registration.



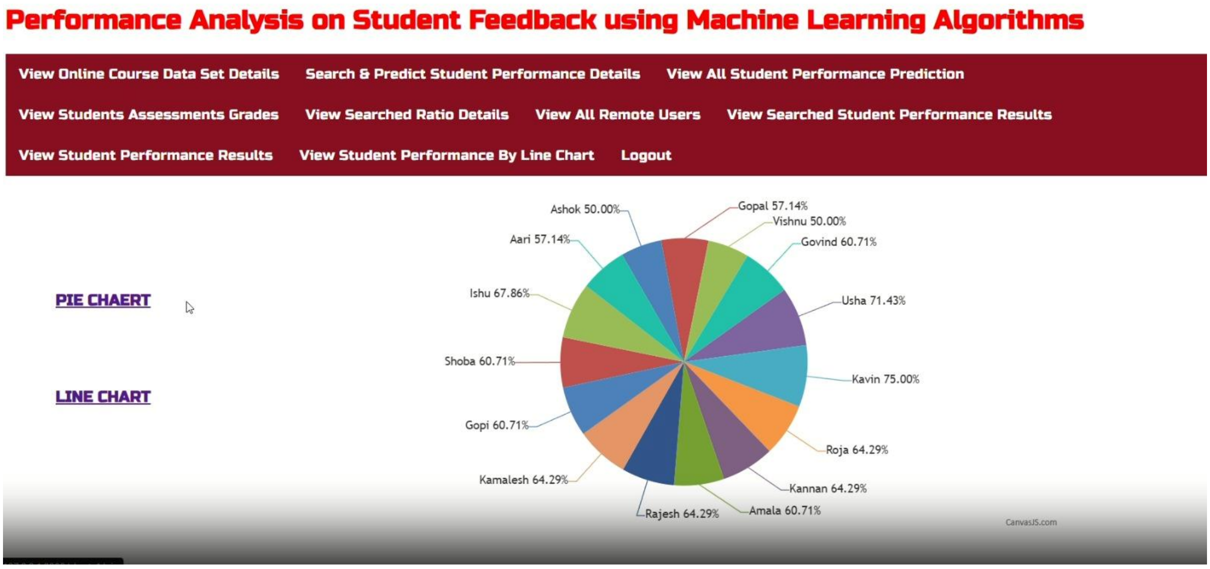
5.4: Predicting faculty performance page

The dataset in this e-healthcare system consists of personal health records (PHRs) collected from patients through various sensors and wearable devices. Each record in the dataset contains multiple columns, each representing a specific aspect of the patient's health information. These columns are designed to capture a comprehensive profile of the patient's medical condition and relevant health metrics.



5.5: Analysis of faculty performance

In our e-healthcare system, authorized users search patient records securely and efficiently by generating a trapdoor with their private key and the relevant keyword. This trapdoor is sent to the cloud server, which searches encrypted indices for matching records without revealing keywords or plaintext data. The cloud server retrieves and, if necessary, re-encrypts the records for authorized users. This method ensures patient data confidentiality while enabling healthcare professionals to access critical information for effective diagnosis and treatment.



5.6: Student Dataset results

In our e-healthcare system, once patient search results are retrieved, authorized users decrypt the data to access the needed information. The decryption process involves the user's private key and the received encrypted records. This ensures that only authorized personnel can view the sensitive patient health records (PHRs). By maintaining robust encryption and decryption protocols, the system guarantees data confidentiality and security, allowing healthcare professionals to securely obtain and utilize critical patient information for diagnosis and treatment.

6. TESTING

6. TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product.

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive.

Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields.

6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

- Functional testing is centered on the following items:
- Valid Input : identified classes of valid input must be accepted.
- Invalid Input : identified classes of invalid input must be rejected.
- Functions : identified functions must be exercised.
- Output : identified classes of application output.
- Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

6.3 TEST CASES

Test Case ID	**Test Case Name**	**Input**	**Expected output**	**Actual Output**	**Test Case Pass/Fail**
1	User credentials	Username:Govind Password : Govind@123	It should move to user home page	It moves to the user home page	Pass
2	Check Username	Username: XYZ (Which is invalid)	It shows the error The username is not available	It shows the error The username is not available	Pass
3	Creating an account	Username: hello (if username is already taken)	Gives the error Username already exists	Gives the error that username already exists	Pass
4	registration	Mail ID (Already exists)	Shows the message Account exists with the given Mail ID. Try login	Shows the message Account exists with the given Mail ID. Try login	pass
5	Registration details	Invalid Phone number (more than 10 numbers)	Gives the message “Invalid Details”	Gives the message “Invalid Details”	Pass

6.1:Test Case

7.CONCLUSION

7. CONCLUSION & FUTURE SCOPE

7.1 CONCLUSION

The advancement in technologies has innovated the education field dramatically. Deep learning-based approaches have attracted lots of attention recently in many fields including education. In this work, DTLP was proposed to classify student's comments into positive and negative sentence. DTLP takes advantage of the CNN, BiLSTM, and attention mechanism, where coarse-grained local features are produced by CNN, BiLSTM takes into account the sequential processing, attention mechanism highlights discriminative and effective features.

The final student performance predictive model revealed that student engagement with digital material has a significant impact on their success in the entire course. The findings' results also demonstrate that long-term students' performance achieves better accuracy than students' assessments grades prediction model, due to the exclusion of temporal features in regression analysis. The date of student deregistration from the course is a valuable predictor that is significantly correlated with student performance.

7.2 FUTURE SCOPE

In future work, a new feature set as an additional independent variable to improve the classification model will be considered. Another avenue for future work: to investigate and develop a method using various deep learning-based methods. Furthermore, we would like to consider "student our system. Moreover, we aim to apply idea presented in paper (Camastra et al. 2020) to the "Feedback Summarization Results module to enhance graph-based knowledge representation.

Future research direction involves the use of temporal features for predicting students' assessments grades model. With temporal feature time series analysis will be untaken, might be more advanced machine leering will be utilized. Analyzing student feedback using machine learning algorithms is a powerful approach to gaining insights into educational effectiveness and student satisfaction.

BIBLIOGRAPHY

BIBLIOGRAPHY

REFERENCES

[1] K. F. Hew and W. S. Cheung, “Students’ andminstructors’ use of massive open online courses (MOOCs): Motivations and challenges,” Educ. Res. Rev., vol. 12, pp. 45–58, 2014.

[2] H. B. Shapiro, C. H. Lee, N. E. Wyman Roth, K. Li, M. Çetinkaya-Rundel, and D. A. Canelas, “Understanding the massive open online course (MOOC) student experience: An examination of attitudes, motivations, and barriers,” Comput. Educ., vol. 110, pp. 35–50, 2017.

[3] J. Renz, F. Schwerer, and C. Meinel, “openSAP: Evaluating x MOOC Usage and Challenges for Scalable and Open Enterprise Education.,” *Int. J.* AdvCorp. Learn., vol. 9, no. 2, pp. 34–39, 2016.

[4] S. Li, Q. Tang, and Y. Zhang, “A Case Study on Learning Difficulties and Corresponding Supports for Learning in cMOOCs| Une étude de cas sur les difficultés d’apprentissage et le soutien correspondant pour l’apprentissage dans les cMOOC,” *Can.* J. Learn.

Technol. *Rev. Can*. l’apprentissage la Technol., vol. 42, no. 2, 2016.

[5] S. Zutshi, S. O’Hare, and A. Rodafinos, “Experiences in MOOCs: The Perspective of Students,” Am. J. Distance Educ., vol. 27, no. 4, pp. 218–227, 2013.

[6] Z. Wang, T. Anderson, L. Chen, and E. Barbera, “Interaction pattern analysis in cMOOCs based on the connectivist interaction and engagement framework,”

Br. J. Educ. Technol., vol. 48, no. 2, pp. 683–699,2017.

[7] W. Xing and D. Du, “Dropout Prediction in MOOCs: Using Deep Learning for Personalized Intervention,” J. Educ. Comput. Res. p.0735633118757015., 2018.

[8] M. J. Gallego Arrufat, V. Gamiz Sanchez, and E. Gutierrez Santiuste, “Trends in Assessment in Massive Open Online Courses,” *Educ. Xx1*, vol. 18, no. 2, pp. 77–96, 2015.

[9] B. J. Hao Cen, Kenneth Koedinger and Carnegie, “Learning Factors Analysis – A General Method for Cognitive Model Evaluation and Improvement,” in In International Conference on Intelligent Tutoring Systems, 2006, vol. 8, pp. 164–175.

[10] K. R. Koedinger, E. A. Mclaughlin, and J. Stamper, “Automated Student Model Improvement,” in Educational Data Mining, proceedings of the 5th International Conference on, 2012, pp. 17–24.

[11] J. Sinclair and S. Kalvala, “Student engagement in massive open online courses,

” Int. J. Learn. Technol., vol. 11, no. 3, pp. 218–237, 2016.

[12] J. Mullan, “Learning Analytics in Higher Education,” London, 2016.

[13] P. and K. Al-Shabandar, R., Hussain, A.J., Liatsis, “Detecting At-Risk Students With Early Interventions Using Machine Learning Techniques,” IEEE Access,

vol. 7, pp. 149464–149478, 2019.

[14] S. Jiang, A. E. Williams, K. Schenke, M. Warschauer, and D. O. Dowd, “Predicting MOOC Performancewith Week 1 Behavior,” in Proceedings of the 7th

International Conference on Educational Data Mining (EDM), 2014, pp. 273–275.

[15] L. Analytics and C. Exchange, “OU Analyse : Analysing at - risk students at The Open University,” in in Conference, 5th International Learning Analytics and Knowledge (LAK) (ed.), 2015, no. October 2014.

GITHUB LINK

https://github.com/Arbaaz018/Performance-Analysis-on-Student-Feedback-usingMLalgorithms/tree/main/students_performance_prediction