# CHAPTER 1

# INTRODUCTION

In today's rapidly changing educational landscape, there is an unprecedented demand for personalized learning experiences and data-driven insights. Our project aims to meet these needs by creating an integrated Student Prediction System. This system not only helps educators forecast student performance, but it also allows for Teaching Evaluation and Feedback Analysis, resulting in a more effective teaching and learning environment. By leveraging the power of predictive analytics, our system enables educators to anticipate student needs and intervene proactively. Using data-driven forecasts, educators can identify students who are at risk of underperforming and implement targeted support strategies to improve their academic performance. This proactive approach allows educators to tailor their interventions to individual student profiles, increasing the effectiveness of their support efforts.

Furthermore, our approach includes Teaching Evaluation in addition to student prediction. Through the examination of several measures like student feedback, instructional effectiveness, and participation in the course, educators can acquire significant understanding regarding the effectiveness of their pedagogical approaches. Equipped with this data, teachers can make well-informed choices to enhance learning opportunities, hone their teaching strategies, and ultimately raise student performance. Furthermore, our approach makes input Analysis easier, allowing teachers to draw insightful conclusions from course evaluations and student input. Teachers can find patterns and trends in qualitative feedback by using text analysis tools. This can provide important context to go along with the quantitative assessment results. This thorough method of feedback analysis enables teachers to pinpoint their areas of strength and growth, resulting in ongoing improvements to the calibre of instruction and student happiness.

In conclusion, our initiative seeks to transform education by utilizing feedback analysis, teaching assessment, and predictive analytics to build a more customized, efficient, and welcoming learning environment. Our goal is to promote student achievement, cultivate a culture of continuous improvement, and eventually change education by providing educators with useful insights and helpful tools.

## 1.1 Objective

Creating a reliable predictive analytics model that can predict student success based on a variety of academic and demographic parameters is one of the main goals. This technique attempts to identify children who might be at risk of underperforming by utilizing machine learning algorithms and historical data. This will allow instructors to intervene early and offer individualized support to these pupils.

The implementation of an extensive framework for assessing the efficacy of instruction is another goal. To evaluate the effect of teaching methods on learning outcomes, this entails examining measures including student feedback, instructional quality, and course participation. Teachers can improve their tactics to better meet the requirements of their students by learning about effective teaching strategies.

Furthermore, the project aims to enable feedback analysis by using text analysis methods. Teachers can better grasp students' perspectives and areas for growth by drawing insightful conclusions from qualitative student feedback and course assessments. The purpose of this goal is to provide teachers with useful knowledge so they can improve their instruction and course options.

In addition, by addressing biases and inequities in instructional strategies and course materials, the project seeks to advance equity and inclusion in education. By utilizing data-driven insights, educators can detect and address possible obstacles to learning, fostering more welcoming and encouraging learning environments for all students.

The project's main objective is to promote innovation in education and institutional reform. Education professionals may continuously improve teaching and learning experiences, which will increase student performance and happiness, by utilizing advanced analytics and technology-driven solutions.

## 1.2 Plan of Action

**Data Collection and Preparation:**

Collect datasets that include student academic records, demographic information, and feedback data. Preprocess the data to remove missing numbers, outliers, and inconsistencies. Examine the datasets to better understand the distribution of features and their relationships. The dataset is collected from the Kaggle (https://www.kaggle.com/datasets/csafrit2/higher-education-students-performance-evaluation) Student_prediction.csv**.** Student feed back is stored in the from of sample_submission.csv(https://www.kaggle.com/datasets/supriodutta2022/multilabeldataset?select=sample_submission.csv)

**Feature Engineering and Selection:**

Identify significant characteristics that may affect student performance, such as age, gender, study hours, GPA, and course identification. Develop new features as needed, such as combining study hours and reading frequency to capture overall study habits. Choose important characteristics for modeling depending on their relevance and association with the target variable (for example, the SUMM score).

**Models for Student Prediction:**

Create machine learning models that predict student performance (SUMM score) using the selected features. Experiment with several algorithms, including Decision Trees, and Random Forests. Fine-tune model hyperparameters with cross-validation and grid search.

**Teaching Evaluation Framework:**

Analyze teaching evaluation data including course engagement, GPA, and study habits. Use visualization techniques to convey evaluation data, such as count plots and correlation analysis. Create a framework to help instructors evaluate teaching performance and find areas for improvement.

**Feedback Analysis:**

Use natural language processing (NLP) to examine qualitative student feedback. Tokenization, stemming, and sentiment analysis are useful methods for extracting relevant insights from text data. Create visualizations and reports that highlight feedback trends and sentiments.

**Integrate and deploy:**

Combine predictive models, instructional assessment tools, and feedback analysis modules to create a cohesive system. Deploy the system at educational institutions or platforms utilized by teachers and students. Provide user training and support to ensure the technology is used effectively.

**Evaluation and iteration:**

Evaluate the integrated system performance in an educational real-world setting. Receive feedback from educators and students to understand strengths and limitations. Iterate on the system and its functionality according to the feedback and changing requirements, update, and enhance the system respectively. The above plan summarily presents the step-by-step plan of applying predictive analytics and teaching evaluation and feedback analysis to enhance education benefits and teaching efficiency.

## 1.3 Literature Survey

**1. Title**: Technology Enhanced Language Learning Research Trends and Practices (A Systematic Review)

**Author:** Nurkhamimi Zainuddin

**Explanation:**

This paper examines the changing environment of technology-enhanced language learning (TELL) in the context of language teaching and learning. The introduction establishes the context by emphasizing the significance that technology has had in language training over the last many decades. The exponential rise of technology has created new opportunities for language educators to explore novel approaches to teaching and learning languages. Previous research has demonstrated the potential of technology to improve language learning outcomes. Scholars such as Carr et al. (2011), Chirimbu and Tafazoli (2013), Patel (2014), and Yeşilel (2016) have emphasized the benefits of TELL, particularly its ability to combine multiple sources of knowledge and educational materials. In today's digital world, where teens and young adults have easy access to cutting-edge technologies, using technology that is relevant to learners' daily experiences is critical for effective language instruction. The research goes into the varied nature of language proficiency, acknowledging that successful communication skills require more than just linguistic ability. Individuals must draw on a variety of psychological resources, including their talents, attitudes, and real-world communication experiences, in order to be competent communicators. This comprehensive view of language competency emphasizes the necessity of using new educational approaches, such as TELL, to develop well-rounded language abilities in learners. Phonetics, grammar, vocabulary, reading, writing, translation, auditory comprehension, and literature appreciation are among the components of language learning investigated in terms of their technology intervention potential. Tseng (2019) contends that technology can have a substantial impact on language acquisition in these domains, providing options for adaptive and efficient vocabulary development, as stated by Asllani and Paçarizi (2021). With the ongoing development of Internet-based technology, an increasing number of students are turning to digital resources to improve their language skills, particularly in English. Against this context, the report explains the objectives and research questions that motivated the thorough review of contemporary trends and practices in TELL within the language education sector. The review focuses on identifying the types and trends of recent studies to gain insights into present practices and guide future research attempts. 17 papers from 2020 to 2022 were carefully selected for examination, with a focus on essential parameters such as aim, methodology, sample characteristics, country of study, and outcomes. A complete analysis and synthesis of these works offer insight on current TELL trends and practices in language teaching and learning situations. The assessment demonstrates a prevalence of quantitative research approaches used to investigate TELL patterns and behaviours, emphasizing empirical evidence and data-driven conclusions. Furthermore, the systematic research reveals differences in the prevalence of TELL across educational levels, with universities having higher adoption rates than primary schools. Considering these findings, the study ends by suggesting future research paths in the field of TELL. It calls for the inclusion of students at all levels of education in TELL activities to get a more comprehensive grasp of its application and efficacy in language teaching and learning. The report also emphasizes the significance of ongoing research to inform evidence-based educational approaches and improve the incorporation of TELL tools and techniques into language schools. In conclusion, this work contributes to the ongoing discussion on TELL by conducting a comprehensive review of recent research in the topic. By explaining current patterns and practices, the article provides vital insights for educators, academics, and policymakers looking to effectively use technology in language teaching. TELL has the potential to transform language instruction by empowering learners to build mastery in a wide range of linguistic skills and abilities.

**2. Title:** A Text Mining and Statistical Approach for Assessment of Pedagogical Impact of Students' Evaluation of Teaching and Learning Outcome in Education

**Author:** Kingsley Okoye, And Sandra Dennis Núñez Daruich

**Explanation:**

This study dives into the critical function of technology-enhanced learning (TEL) in higher education institutions (HEIs), focusing on its integration into teaching and learning processes. With TEL becoming more important, educators must select appropriate tools and technology to help both teachers and students, enabling a continuous and effective educational experience. The study makes use of the huge amount of educational data accessible, with a specific emphasis on student evaluations of teaching (SET) collected in a higher education setting. The study's goal is to identify the major aspects impacting student performance and the delivery of TEL-based education, as well as its didactic influence and ramifications, by evaluating a large dataset of 471,968 entries. A mixed methodology is used, integrating qualitative and quantitative methodologies. The study uses the Educational Process and Data Mining (EPDM) model, a text mining method, to extract sentiments and emotional valence stated by students from SET data. This research considers changes based on evaluation type and time (2019–2021). The study uses multivariate analysis of covariance (MANCOVA) and post-hoc testing to investigate the effects of various SET kinds and evaluation periods on students' learning outcomes and views of the teaching-learning processes. The findings shed light on the nuanced dynamics of TEL-based education, emphasizing both its efficacy for continuous education, innovative pedagogies, and teaching transformations, as well as the importance of rigorous research to fully understand its implications for student learning outcomes. Finally, the study provides empirically grounded insights that can be used to refine and improve TEL practices in higher education institutions, establishing an atmosphere favourable to effective digital teaching and learning. The incorporation of technology in education has altered traditional teaching and learning paradigms. In today's higher education scene, Technology-Enhanced Learning (TEL) is critical in supporting dynamic, engaging, and adaptable learning opportunities. This paradigm shift has not only mandated the adoption of new technology, but it has also forced educators to reevaluate their instructional techniques to properly harness these resources. Considering these changes, one of the most significant difficulties for educators is the pedagogical definition of tools and technology that support teaching and learning processes. With so many options available, educators must carefully choose and apply technologies that match with pedagogical aims while also considering student requirements and preferences. Furthermore, there is an increasing emphasis on preserving and sustaining continuous education practices, particularly in the wake of disruptions like the COVID-19 epidemic, which highlighted the significance of adaptable learning models. This study recognizes the potential of educational datasets collected from teaching and learning activities to provide insights that can help improve the educational process. The study's goal is to uncover key elements influencing student performance and the delivery of TEL-based education by examining students' evaluations of teaching (SET) data collected in a higher education context. The dataset, which contains many items (n=471,968), is a valuable source of information for investigating the complexities of the teaching-learning dynamic. Methodologically, the study takes a mixed approach based on the combination of data structure analysis and descriptive decision theory. This approach enables a thorough investigation of the reasoning behind students' assessments of teaching and performance. The Educational Process and Data Mining (EPDM) model, a text mining method, is used in qualitative analysis to extract attitudes and emotional valence stated by students in SET data. This qualitative investigation is supported by quantitative analysis, which uses multivariate analysis of covariance (MANCOVA) and post-hoc testing to investigate the effects of various SET types and evaluation periods on students' learning outcomes and perspectives. The study's findings provide useful insights into the intricate interplay between technology, pedagogy, and student learning outcomes. The study provides practical suggestions for educators aiming to improve teaching and learning methods by identifying the elements that influence student performance and the delivery of TEL-based education. Furthermore, the study emphasizes the necessity of constant review and modification of TEL projects, as well as the need for ongoing research to inspire evidence-based teaching practices. In conclusion, this study makes an important contribution to the field of TEL-based education by conducting a thorough examination of students' evaluations of teaching and performance in a higher education setting. Using educational datasets and a mixed-method approach, the study provides useful insights into the elements that influence student learning outcomes and the delivery of TEL-based education. Moving forward, the report recommends additional research to enrich evidence-based educational approaches and improve the success of TEL projects in higher education.

**3. Title**: Analyzing the Impact of Student Engagement on Learning Outcomes In E-Learning Platforms: A Systematic Review of Literature

**Author:** Lakshani Erandika, Janaka Wijayanayake and Jinendri Prasadika

**Explanation:**

This study offers a thorough examination of the relationship between student involvement and learning outcomes in E-learning platforms, with a specific emphasis on the Sri Lankan setting. As e-learning continues to influence the global education landscape, educators and institutions must grasp the dynamics of student participation and its impact on learning outcomes. The paper opens by detailing the methodological methodology used for the systematic literature review, which followed the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) criteria. Following a rigorous screening process, 54 relevant publications were found and added for evaluation. However, the analysis finds a significant lack in research relevant to Sri Lanka, emphasizing the importance of context-specific investigations in this country. While previous research has primarily focused on European countries, the unique infrastructure and learning practices in Sri Lanka need a more nuanced understanding of the relationship between student involvement and learning results. The literature review revealed a considerable underutilization of the Community of Inquiry (CoI) paradigm in previous studies. The CoI framework is a useful tool for analyzing student participation in e-learning platforms, although its implementation is limited in the literature. Furthermore, the report states that Information Technology (IT) students frequently have more technical skill than students from other fields. While this may not be a significant influencing factor in research on IT students, it does highlight the significance of taking disciplinary distinctions into account when examining student engagement and learning outcomes. The introduction includes basic background information on E-learning, which is defined as the use of information and communication technologies (ICTs) to transform educational practices. The COVID-19 pandemic spurred the development and implementation of E-learning platforms around the world, notably in Sri Lanka, where school and university closures resulted in a rapid shift to online learning. As a result, the availability of E-learning platforms increased substantially, providing students with virtual classrooms, multimedia materials, and collaborative learning spaces. Student engagement has been found as a significant factor impacting learning results in E-learning platforms. Student engagement is defined as the relationship between students and their institutions to improve the learning experience. It includes the time, effort, and resources invested by both sides. Students who are actively engaged are more likely to learn new things, develop crucial skills, and have great learning experiences. However, the report acknowledges that research on student participation in non-formal online learning environments is scarce, emphasizing the need for additional research in this field. Student engagement has been identified as an important factor influencing learning outcomes in e-learning platforms. Student engagement is described as a partnership between students and their institutions that aims to improve the learning experience. It reflects the time, effort, and resources invested by both parties. Students that are actively engaged are more likely to learn new things, improve their abilities, and have positive learning experiences. However, the paper recognizes that research on student participation in non-formal online learning environments is limited, underlining the need for more research in this area. In conclusion, this paper emphasizes the necessity of understanding the relationship between student involvement and learning outcomes in E-learning platforms, particularly in the Sri Lankan setting. Future studies that address research gaps and incorporate a Sri Lankan viewpoint can help to improve E-learning experiences and guide digital education decision-making. Educators and policymakers may improve E-learning practices and promote positive learning outcomes for students in Sri Lanka and elsewhere by conducting ongoing research and inquiry.

**4. Title:** Impact of Student's Evaluation of Teaching: A Text Analysis of The Teachers' Qualities by Gender

**Author:** Kingsley Okoye, Arturo Arrona-Palacios and Claudia Camacho-Zuñiga

**Explanation:**

This article investigates the convergence of modern educational models, data analytics, and information and communication technologies (ICT) for improving the teacher-student experience and educational processes. It emphasizes the significance of user-centric analyses in using text-based data in educational settings to deliver strategic intelligence and improvement ideas. The study emphasizes the extraordinary increase in the availability of text-based data in educational activities, which provides a chance to improve teaching-learning processes using data-driven approaches and technologies. The introduction establishes the strategic necessity of using data in educational environments to drive process innovation and track educational achievements. It recognizes the limitations of standard education curriculum in combining strategic viewpoints and process monitoring, highlighting the need for novel ways that capitalize on the promise of ICT in education. The paper's central proposition is an Educational Process and Data Mining model, which seeks to improve educational processes by using student perspectives and opinions. The EPDM model focuses on examining students' opinions of their teachers, considering the teachers' gender. The study's goal is to find underlying trends and provide beneficial insights for educational development by examining the sentiments and emotional valence conveyed in students' evaluations of teaching. The theoretical basis for the study is based on sentiment analysis, educational innovation, and technological adoption. It emphasizes the necessity of incorporating student feedback into the evaluation of teaching efficacy, as well as the power of data mining techniques to derive actionable insights from textual data. One important component of the study is its use in a higher education setting, where the EPDM model is implemented with SET comments. The study focuses on higher education to meet the context's special needs and difficulties, where student happiness and teacher efficacy are crucial. The study emphasizes the necessity of technology-driven management and the usage of information regarding the teacher-student learning experience. It contends that by using data analytics and ICT, educational institutions may improve the impact of educational process efforts on student satisfaction and learning outcomes. Throughout the study, the importance of technology adoption in education is emphasized, with a focus on ICT's transformative potential in changing traditional teaching-learning models. The EPDM model provides a realistic framework for educators and institutions to use student feedback and sentiment analysis to influence decisions and improve teaching effectiveness. In conclusion, the research adds to the expanding body of literature on educational innovation and technology adoption by presenting a practical methodology for incorporating student input into educational procedures. Using data analytics and ICT, educators and institutions can improve the teacher-student experience and drive continuous improvement in educational outcomes. The study emphasizes the need of using a user-centric approach and relying on data-driven methodologies to meet the changing needs of modern educational models.

**5. Title**: A Literature review of the research on Student Evaluation of Teaching in Higher Education

**Author:** Luying Zhao, and Pei Xu

**Explanation:**

This article goes into the diverse world of student evaluation of teaching (SET), a popular method for evaluating teaching quality and teacher performance in both Chinese and foreign colleges. SET has become a contentious issue in the field of teaching assessment, drawing significant attention from researchers around the world. The primary goal of this work is to critically examine existing research findings on SET from Chinese and foreign academic circles. By combining these findings, the paper hopes to provide useful insights for the development of index systems and the practical implementation of course evaluations, particularly in the context of ideological and political theory classes for Chinese college students. The introduction establishes the framework by defining SET as an activity in which students assess teachers' instructional effectiveness and quality. The beginnings of SET can be traced back to the early twentieth century in the United States, when Purdue University developed the first standardized student evaluation scale in 1915. Since then, SET has grown into an important component of the educational system, gradually finding its way into Chinese higher education following initiatives launched by the Ministry of Education in 2001. The introduction defines the purpose of the article, which includes a rigorous examination of theoretical advances in both Chinese and foreign academic circles addressing SET. The methodology section explains the methods used for the literature review, which includes documentary analysis and logical analysis. The authors favor recent publications from respected journals and authors, taking into account the citation rates of selected material. Furthermore, the research uses logical analysis to structure the evaluation, concentrating on essential factors related with SET such as reliability and validity, indicator systems, application challenges, and relevant countermeasures. The paper digs into the finer points of dependability and validity in the context of SET. Reliability refers to the stability and consistency of evaluation outcomes, whereas validity asks whether SET achieves its intended aims and consequences. Despite some doubts about the dependability of SET due to methodological limitations, empirical research has generally validated its efficacy. Scholars have used several statistical methods to estimate reliability, yielding inconsistent results. However, a common belief is that SET tools used in universities are reasonable, dependable, and efficient. On the validity front, proponents believe that student evaluations are scientifically founded since kids have the cognitive and judgment skills required to assess educational efficacy. However, the legitimacy of SET remains a disputed issue, with differing views on its efficacy in accurately measuring instructional quality. In addition to investigating the reliability and validity of SET, this research digs into the larger landscape of factors influencing teaching evaluation in higher education. It recognizes that, while SET is a useful instrument for getting feedback on teaching effectiveness, it is only one component of comprehending the complex dynamics of the teaching-learning process. As a result, the paper intends to shed light on several aspects of SET, such as its design, implementation issues, and enhancement ideas. One important component of the work is the development of indicator systems for SET. This includes determining the important measures and criteria for evaluating instructional quality and teacher performance. Scholars have proposed a variety of indicator systems, ranging from quantitative metrics like student satisfaction ratings to qualitative evaluations of teaching methods and classroom dynamics. By reviewing current literature, the research provides insights into the creation and refining of indicator systems customized to the specific setting of ideological and political theory courses in Chinese institutions. Furthermore, the research dives into the practical problems connected with implementing SET in higher education settings. Biases in student assessments, differences in the interpretation of evaluation results, and faculty resistance are all potential problems. By critically exploring these issues, the paper hopes to offer solutions for addressing them and increasing the utility of SET as a tool for continual improvement in teaching quality. Furthermore, the research investigates the role of SET in promoting accountability and openness in higher education institutions. By including students in the evaluation process, SET encourages better stakeholder participation and accountability among faculty members. However, the research recognizes the importance of balancing, as overreliance on SET data may unintentionally weaken faculty autonomy and impede innovative teaching techniques. In conclusion, this study presents a thorough discussion of the theoretical foundations, practical issues, and challenges related with SET in higher education. By summarizing previous research findings and providing critical insights, this work adds to ongoing efforts to improve teaching evaluation processes and increase quality assurance in educational institutions. The paper's interdisciplinary approach and sophisticated analysis make it an invaluable resource for educators, administrators, and politicians working to promote excellence in teaching and learning

## 1.4 Existing System

Academic databases such as Google Scholar, PubMed, and IEEE Xplore can be utilized to identify existing literature on this topic, as well as publications of higher learning focusing on education, data mining, statistics, and educational technology. In your search queries, include terms like "text mining," "statistics," "pedagogical impact," "student evaluation," "teaching effectiveness," and "learning outcomes" to uncover relevant publications, research papers, and conference proceedings. In today's system, evaluations of teaching efficacy are typically relied on traditional approaches like surveys or online questionnaires issued to students at the end of the course. These evaluations are typically composed of Likert-scale questions or open-ended prompts in which students provide feedback on various aspects of the course, such as the instructor's teaching style, course content, assessment methods, and overall learning environment. Once collected, feedback data is typically analyzed manually or using basic statistical methods. Manual analysis entails reading through the responses and categorizing them according to common themes or sentiments. Alternatively, basic statistical techniques such as calculating mean scores or percentages of positive responses can be used to assess teaching effectiveness based on aggregated feedback. While traditional methods offer some insights into teaching effectiveness, they have several limitations. Manual analysis is time-consuming and subjective, which can lead to interpretation biases. Basic statistical techniques may lack the sophistication required to detect subtle patterns or relationships in feedback data.

**For Example**: Indiana University Bloomington school of Education

This is a real-time example of a university using software to predict learning outcomes in education. Here we may explore the overview of this map and analyze the patterns and structure across (MAPSAT Legacy site). Many well-meaning individuals desire to enhance education. So do we. We believe that education, both K-12 and higher education, has the potential to be significantly more effective, efficient, and rewarding than it is today. Educators who have taught for a long have witnessed various publicly publicized innovations come and go. Site-based management, constructivist classrooms, technological integration, school restructuring, and, yes, even systemic transformation are some of the terms used to describe the advances. Educators have accurately recognized that not much has changed based on what they can see.

• "Change what?"
• "Change how?"
• "How do you know the change is working?"

To know how to alter something, we must first determine what needs to change. We must determine whether the adjustment achieves the purpose and does not have any negative, unforeseen consequences. Change for the sake of change is not productive. The "how" is unimportant it do not know what must change. Consider an ancient bridge that is crumbling; it is structurally weak and impedes traffic flow. If the bridge is not repaired, it will collapse, sending vehicles into the river. When engineers construct a new bridge, they follow appropriate scientific theories. Until now, there has been no reliable technique to forecast whether new educational system designs will perform better than existing ones. We have had no reliable method for characterizing the components of any educational system or assessing the consequences of change throughout the system. New designs and curriculum have been patched together, much like repairing rust spots on an old car with body filler and paint, installing new seat covers, or replacing tires. The overall organization is unaltered. 3 Many studies have focused on the transition process. We feel it is equally vital to focus on the outcomes of change—that is, how well the new system is expected to work and how well it really works. We require both approaches—process and outcomes—as they are complementary. The change process could be successful, but the resulting new system may not provide the expected results. The new system may be effective, but the transition may leave staff, families, instructors, and children dissatisfied and fatigued. For optimal results, both methods and outputs must be satisfactory. We are focusing specifically on the topic of projecting educational system results. Predictions must be founded on scientific theory, consequences, and supporting facts. If the forecasts are not based on scientific theory, how can we justify devoting significant time and money only to produce something that is no better—or possibly worse—than what we now have? It's no surprise that educational practitioners frequently doubt, reject, and sabotage the work of educational reformers. The stakes are high. Mistakes can have terrible effects, especially when overhauling a whole educational system. Understanding systemic change is not an easy task. Educators will have to master new thinking habits. Hart (1993) observed that most individual belief systems do not include dynamic cycles. Cognitive maps of belief structures are typically linear, with few if any feedback loops. Hart noticed that exceptions occurred among those in occupations that trained them to think in dynamic cycles (for example, ecologists and systems engineers). Similarly, Senge (2006) has shed light on business organizations by identifying archetypal patterns of dynamic cycles. These patterns are difficult to convey or understand using static print and illustrations. To address this understanding gap, Senge and his colleagues created role-playing activities and computer simulations to help businesspeople understand these patterns of dynamic relationships—some of which contradict individual intuitions about how systems like businesses grow and change. For these reasons, we believe that educators will benefit greatly from being able to employ computer software to develop new educational systems.

1. 'Get Ready, SET, Go!' Change Model
2. The theory model choices set known as Axiomatic Theories of Intentional Systems (ATIS).
3. Computer software: Analysis of Patterns in Time and Configuration (APT&C). 4. Computer software for predicting educational system outcomes (PESO). SimEd Technologies is a collaborative tool that uses computer technology to define educational institutions, forecast system changes, and document the effects of change.

We will discuss the 'Get Ready, Set, go!' approach for predicting educational system outcomes and guiding the transformation process. This inquiry-based change paradigm will use appropriate theory and computer applications that are currently being developed.

### 1.4.1 Disadvantages of Existing System

**Subjectivity:**

Traditional evaluation methods like surveys and questionnaires rely significantly on students' subjective responses. This subjectivity can lead to biased outcomes since students' evaluations of teaching efficacy may differ depending on their unique preferences, experiences, and biases.

**Limited Scope:**

Traditional evaluation methods frequently focus on surface-level features of teaching, such as teacher manner or course material, while ignoring deeper pedagogical aspects, such as instructional tactics, learning outcomes, and student involvement. As a result, these approaches may not provide a complete picture of instructional success.

**Inadequate Feedback:**

Traditional assessments use Likert-scale questions and open-ended prompts, which may not successfully capture nuanced feedback or constructive criticism. Students may fail to communicate their ideas or provide significant suggestions for improvement, resulting in ambiguous or ineffective feedback for teachers.

**Low Response Rates:**

Traditional evaluations are frequently voluntary, resulting in low response rates and potential response bias. Students who opt to participate may not be representative of the whole student body, skewing the evaluation results.

**Limited Use of Technology:**

Traditional evaluation techniques sometimes rely on paper-based surveys or online questionnaires, which may lack the interactive features and data analytics capabilities of modern educational technology. This limited use of technology impedes the gathering, processing, and interpretation of evaluation data.

**Time-consuming Analysis:**

Instructors may find it time-consuming and labour-intensive to manually analyze evaluation data, such as reading through and categorizing comments. This may lead to delayed feedback and less opportunity for timely course improvement.

**Lack of Actionable Insights:**

While traditional evaluation methods can provide descriptive feedback on teaching effectiveness, they may not provide actionable insights or recommendations for instructional development. Without clear suggestions for improvement, teachers may struggle to make meaningful changes to their teaching techniques.

**Comparison Difficulties:**

Traditional assessment methods might make it difficult to compare teaching performance among instructors or courses due to differences in evaluation criteria, answer formats, and student populations. This lack of consistency reduces the usefulness of evaluation data in generating meaningful comparisons and informed conclusions.

## 1.5 Proposed System

In the context of Impact of student evaluation and learning outcomes in education system implements strategic changes aimed at enhancing teaching approaches and improving student outcomes. The suggested solution improves the present framework of impact of student learning outcomes incorporating sophisticated technologies and data analytics, allowing educators to gain actionable insights and make evidence-based decisions.

**Predictive Analysis for Student Performance**

Building project's data analysis capabilities, the suggested system combines predictive analytics models to better estimate student success measures. Using historical data and machine learning algorithms, the system predicts academic achievement levels, identifies at-risk kids, and proposes targeted interventions to increase retention and success rates. It predicts the SUMM the SUMM values are lies in between 0 to 7 (0: Fail, 1: DD, 2: DC, 3: CC, 4: CB, 5: BB, 6: BA, 7: AA) this can be predicted by using various parameters like study hours, based on the attendance, listening the class, impact of student, course_id, student_id and preparing exam. The preparing exam can be considered as whether the student is preparing for exam from hole academic year or preparing nearest to the exam timing.

**Teaching Evaluation Module**

The proposed teaching assessment module enhances impact of student evaluation of learning outcomes in education by allowing teachers to evaluate their teaching efficacy using data-driven analytics. By assessing characteristics such as student performance patterns and levels of classroom engagement, instructors can acquire useful insights into their instructional methods and adapt teaching strategies to improve learning outcomes.

**Advanced Text Analysis for Student Feedback.**

The proposed solution builds on your project's qualitative analysis capabilities by incorporating advanced text analysis algorithms to extract actionable insights from student feedback and essays. Natural language processing and sentiment analysis algorithms recognize repeating themes, feelings, and concerns expressed by pupils, allowing educators to address root causes and tailor training.

**Strong data security measures**

To protect student data within impact of student evaluation and learning outcomes in education, the proposed system employs strong encryption protocols, authentication systems, and role-based access controls. The solution protects sensitive information from unwanted access and cyber threats while remaining compliant with data protection rules.

**Smooth Integration and User Experience.**

The suggested system prioritizes seamless integration with student learning outcomes existing infrastructure to reduce disruptions and improve user experience. Intuitive interfaces, interactive visualizations, and collaborative tools enable educators to efficiently employ data in their teaching techniques, increasing user uptake and engagement.

**Scalability and adaptability**

The suggested system is designed with scalability and adaptability in mind, allowing for future expansion and developing educational demands within the framework of impact of student evaluation and learning outcomes in education. Modular architecture and adaptable frameworks allow the system to add new features and adapt to changing needs, assuring long-term viability and sustainability.

## 1.5.1 Advantages of Proposed System

**Enhanced Student Performance Monitoring:**

The system uses predictive analytics to give educators advanced tools for monitoring student performance more effectively. This allows for early identification of problematic students and tailored interventions to boost academic attainment and retention rates.

**Personalized Learning Experiences:**

By combining predictive analytics and text analysis, the system allows instructors to tailor learning experiences for students based on their unique requirements, preferences, and learning styles. The method encourages increased engagement and academic performance by modifying training to students' unique requirements.

**Improved Teaching Effectiveness:**

The teaching assessment module allows educators to objectively analyze their educational approaches and make data-driven changes to improve teaching effectiveness. Analyzing classroom engagement levels, student performance patterns, and teaching approaches allows teachers to discover areas for improvement and use evidence-based teaching strategies.

**Actionable Insights from Qualitative Data:**

The solution uses powerful text analysis to derive significant insights from qualitative data sources such as student feedback and writings. This enables educators to obtain a better understanding of students' experiences, attitudes, and concerns, allowing for more informed decision-making and focused interventions.

**Enhanced Collaboration and Professional Development:**

The system encourages instructors to collaborate and share knowledge via intuitive dashboards, interactive visualizations, and collaborative tools. The system encourages professional growth and ongoing improvement in teaching techniques by offering a forum for exchanging best practices, collaborating on instructional tactics, and learning from one another's experiences.

**Data Security and Privacy Protections:**

The system protects student data with sophisticated encryption methods, authentication systems, and role-based access controls. By conforming to industry-leading security standards and compliance laws, the system protects sensitive data from unauthorized access and cyber threats, creating trust and confidence in stakeholders.

**Scalability and adaptability:**

The system is designed to accommodate future growth and growing educational needs. Its modular architecture and adaptable frameworks allow for seamless integration of new features and support for changing needs, assuring reliability and sustainability.

**Evidence-Based Decision-Making:**

By providing educators with actionable insights obtained from data analytics, the system encourages evidence-based decision-making in teaching and learning. This allows educators to make more informed decisions, execute targeted interventions, and assess the impact of their teaching practices on student outcomes, promoting accountability and continuous development.

**Educator and Administrator Empowerment:**

The system's user-friendly interface and intuitive features enable educators and administrators to effectively employ technology in their daily work. By providing educators with timely and relevant data, tools, and resources, the system allows them to focus on what is most important: student learning and achievement.

**Enhanced Student Engagement and happiness:**

The system improves student engagement and happiness by personalizing learning experiences, proactively addressing student needs, and fostering a collaborative and innovative culture. This increases student motivation, persistence, and academic achievement, creating a positive learning atmosphere favourable to growth and success.

### 1.5.2 Limitations

**Limited Data Sources:**

While your project may now rely on accessible data sources, future iterations may investigate new routes for data acquisition. This could include collaborating with educational institutions to gain access to their internal databases, including data from online learning platforms, or implementing emerging technologies such as IoT devices in classrooms. By broadening the data sources, the system can record a greater range of student behaviours, interactions, and learning outcomes, resulting in more accurate predictions and insights.

**Algorithm Optimization:**

In addition to fine-tuning existing algorithms, future research and development initiatives could investigate incorporating sophisticated machine learning approaches such as deep learning and reinforcement learning. These approaches offer the potential to extract more complex patterns and relationships from data, increasing the system's forecast accuracy and adaptability. Furthermore, enhancing algorithms for parallel processing and distributed computing architectures can boost scalability and speed, allowing the system to handle greater datasets and support a rising user base.

**User Interface Complexity:**

Simplifying the user interface to make it more intuitive and usable is critical for increasing customer happiness and adoption. Future updates of the system may include user experience research, usability testing, and iterative design changes to discover pain areas and enhance the interface. Adding tools such as configurable dashboards, interactive visualizations, and guided lessons can help users use the system more effectively and make informed decisions based on the information presented.

**Limited Feedback Mechanisms:**

Effective feedback mechanisms are required for continual improvement and user engagement. Future system updates could include features like in-app surveys, user forums, and automatic feedback prompts to seek feedback from educators, administrators, and students. Furthermore, using analytics tools to track user interactions and behaviour inside the system can provide useful insights into usage trends and areas for improvement.

# CHAPTER 2

# SYSTEM ANALYSIS

## 2.1 Software Requirements Specification

**Purpose**

The goal of this project is to establish the requirements for developing the Student Prediction System. This approach seeks to predict student performance and provide insights for teaching evaluation using demographics, study habits, and academic background.

**Scope.**

The Student Prediction System will include modules for student prediction, teacher evaluation, and text analysis. It will use machine learning algorithms to forecast and analyze data, giving users actionable insights to help them make decisions in educational settings.

**Data collecting**

Involves gathering student input from a variety of sources, such as course evaluations, surveys, and written remarks.

**Data preprocessing**

Involves cleaning, normalizing, and integrating student feedback data for analysis.

**Text mining and sentiment analysis**

Include analyzing textual feedback to identify significant topics, attitudes, and trends.

**Statistical analysis:**

Using quantitative analysis to detect correlations and patterns in student feedback data.

**Visualization and reporting:**

The presentation of analysis results via visualizations, dashboards, and reports.

**Implementation with Anaconda**

Using Anaconda's tools and libraries, such as Python, Jupyter Notebooks, and data analysis packages (for example, pandas, scikit-learn, matplotlib, word-cloud).

**Text mining**

Is the technique of collecting meaningful information from textual material by analyzing sentiment, themes, and patterns.

**Sentiment analysis**

The process of categorizing textual material as good, negative, or neutral depending on its expressed sentiment.

**Statistical analysis**

Is the process of examining quantitative data to find correlations, trends, and relationships.

**Pandas:**

Pandas is Python's the authority data manipulation package, and it provides significant capabilities for managing structured data. It makes it easier to load, clean, filter, and convert data. Pandas is used in this project to preprocess a dataset that includes student demographics, academic records, and teaching evaluations. It allows the project to do critical EDA activities like descriptive statistics, data visualization, and correlation analysis, which provide insights into the relationships between variables.

The command to install pandas

>>pip install pandas

**Scikit-learn (sklearn):**

Scikit-learn, known for its diverse set of machine learning techniques and tools, provides the project with predictive modeling and evaluation capabilities. Regression models in Scikit-learn are used to forecast student success based on demographic information, study habits, and academic history. Classification algorithms evaluate teaching efficacy by examining metrics and student comments. Scikit-learn also makes feature selection, model evaluation, and hyperparameter tuning easier, ensuring that predictive models are robust and accurate.

>>pip install sklearn

**NLTK (Natural Language Toolkit):**

NLTK is a robust package for natural language processing (NLP) activities, including text tokenization, stemming, lemmatization, and part-of-speech tagging. In this project, NLTK preprocesses textual data, such as student essays and feedback, to extract useful insights. By breaking down text into its essential components and transforming it to numerical notation, NLTK allows the project to successfully incorporate textual data into predictive models.

>>pip install nltk

**Gensim**

Gensim addresses topic modeling, document embedding, and other sophisticated NLP approaches. In the project, Gensim supplements NLTK by adding text analysis capabilities. It simplifies tasks like document similarity analysis, subject extraction, and word embedding. Gensim reveals hidden semantic structures inside text by expressing it in high-dimensional vector spaces. This allows the project to extract nuanced insights and patterns from student feedback and essays.

>>pip install gensim

**Matplotlib**

Matplotlib is a flexible Python charting toolkit that is commonly used to create static, interactive, and animated displays. Matplotlib is a significant data visualization tool in this project, providing a wealth of charting functions and customization choices for successfully displaying findings. Matplotlib is used to depict several features of the dataset, such as student performance score distributions, relationships between variables, and temporal trends. The project can use Matplotlib to create histograms, bar plots, line plots, scatter plots, and other visual representations of the data, which are intuitive and helpful to stakeholders. Furthermore, Matplotlib allows the project to define plot aesthetics such as colors, markers, line styles, labels, and annotations, ensuring that visualizations are both visually appealing and useful. By introducing Matplotlib into the project workflow, the team can effectively convey findings, trends, and patterns to stakeholders, allowing for more data-driven decision-making and a better understanding of educational outcomes.

>>pip install matplotlib

## 2.2 SYSTEM REQUIREMENTS

### 2.2.1 User Requirements

- PC Or Laptop with x86-64 (64 bit) Compatible Processors
- 1.6 GHz Processor is recommended
- Windows 7/8/9/11

### 2.2.2 Software Requirements

| | | |
|---|---|---|
| **Operating System** | **:** | Windows XP or latest version. |
| **Platform** | **:** | Python Technology. |
| **Python Version** | **:** | Python 3.6 |
| **Framework** | **:** | Flask |
| **Front End** | **:** | HTML, CSS, JavaScript |
| **Back End** | **:** | python |

### 2.2.3 Hardware Requirements

| | | |
|---|---|---|
| **System** | : | Pentium IV 2.4 GHz or higher processor. |
| **Hard Disk** | : | Minimum 40 GB of storage space. |
| **Monitor** | : | 15-inch VGA Color monitor. |
| **Mouse** | : | Logitech Mouse or compatible. |
| **RAM** | : | Minimum 40 GB of RAM. |
| **Keyboard** | : | Standard Keyboard. |

## 2.3 Functional and Nonfunctional Requirements

### 2.3.1 Functional Requirements

Function requirements specify the specific functions, features, and capabilities that the system must possess to fulfil its intended purpose. Below are the function requirements for a system aimed at evaluating the impact of student evaluation and learning outcomes in education:

**Data Collection and Integration**:

The system should allow for the collection and integration of various types of data, including student evaluation data, learning outcomes data, demographic information, and course-related metrics. It should support the import of data from external sources such as student information systems (SIS), learning management systems (LMS), and assessment platforms.

**student_prediction.csv**

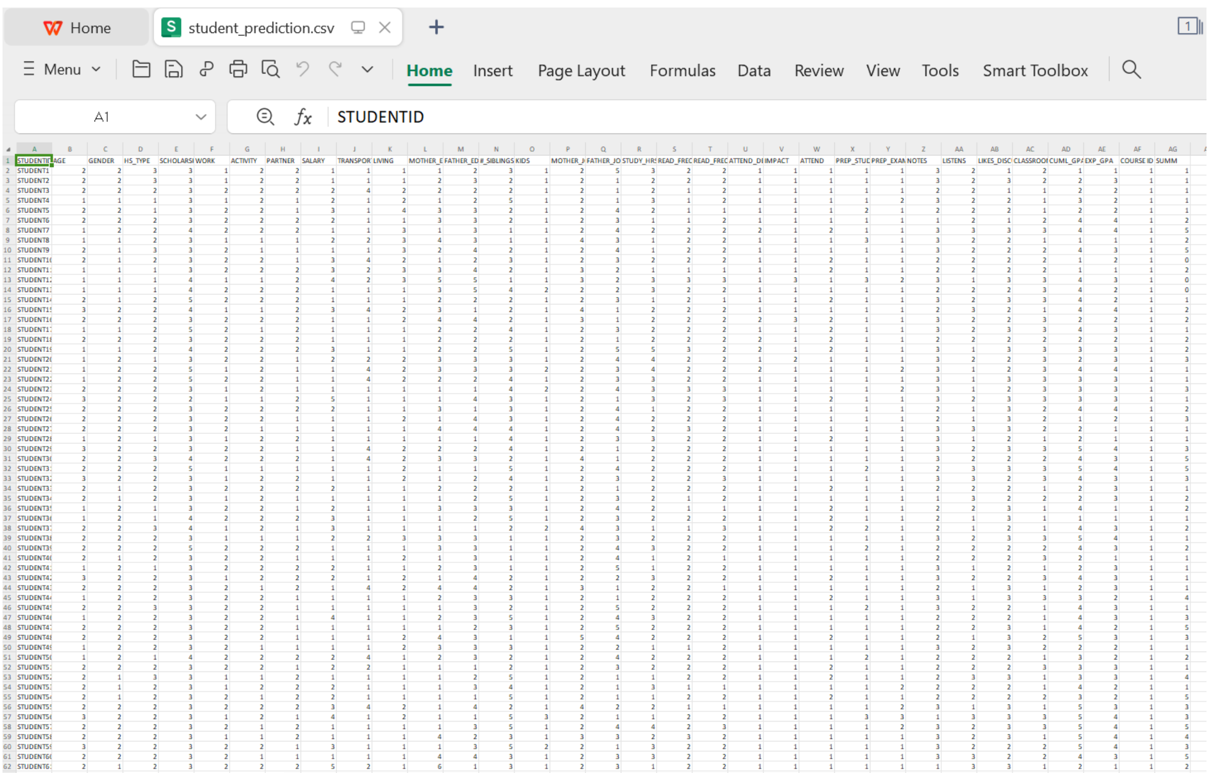



**Figure 2.1**: student_prediction.csv dataset

1-STUDENTID (1 to 145)
2- AGE (1: 18-21, 2: 22-25, 3: above 26)
3- GENDER (1: female, 2: male)
4-HS_TYPE (1: private, 2: state, 3: other)

5-SCHOLARSHIP (1: None, 2: 25%, 3: 50%, 4: 75%, 5: Full)
6-WORK (1: Yes, 2: No)
7-ACTIVITY (1: Yes, 2: No)
8-PARTNER (1: Yes, 2: No)
9-SALARY (1: USD 135-200, 2: USD 201-270, 3: USD 271-340, 4: USD 341-410, 5: above 410)
10-TRANSPORT (1: Bus, 2: Private car/taxi, 3: bicycle, 4: Other)
11-LIVING (1: rental, 2: dormitory, 3: with family, 4: Other)
12- MOTHER_EDU (1: primary school, 2: secondary school, 3: high school, 4: university, 5: MSc., 6: Ph.D.)
13- FATHER_EDU (1: primary school, 2: secondary school, 3: high school, 4: university, 5: MSc., 6: Ph.D.)
14- #_SIBLINGS (1: 1, 2: 2, 3: 3, 4: 4, 5: 5 or above)
15- KIDS (1: married, 2: divorced, 3: died - one of them or both)
16-MOTHER_JOB (1: retired, 2: housewife, 3: government officer, 4: private sector employee, 5: self-employment, 6: other)
17- FATHER_JOB (1: retired, 2: government officer, 3: private sector employee, 4: self-employment, 5: other)
18-STUDY_HRS (1: None, 2: <5 hours, 3: 6-10 hours, 4: 11-20 hours, 5: more than 20 hours)
19- READ_FREQ (non-scientific books/journals): (1: None, 2: Sometimes, 3: Often)
20- READ_FREQ_SCI (scientific books/journals): (1: None, 2: Sometimes, 3: Often)
21- ATTEND_DEPT (1: Yes, 2: No)
22- IMPACT (1: positive, 2: negative, 3: neutral)
23- ATTEND (1: always, 2: sometimes, 3: never)
24- PREP_STUDY (1: alone, 2: with friends, 3: not applicable)
25- PREP_EXAM (1: closest date to the exam, 2: regularly during the semester, 3: never)
26- NOTES (1: never, 2: sometimes, 3: always)
27- LISTENS (1: never, 2: sometimes, 3: always)
28-LIKES_DISCUSS (1: never, 2: sometimes, 3: always)
29- CLASSROOM (1: not useful, 2: useful, 3: not applicable)
30- CUML_GPA (1: <2.00, 2: 2.00-2.49, 3: 2.50-2.99, 4: 3.00-3.49, 5: above 3.49)
31- EXP_GPA (1: <2.00, 2: 2.00-2.49, 3: 2.50-2.99, 4: 3.00-3.49, 5: above 3.49)
32-COURSE ID (1 to 9)
33- SUMM (0: Fail, 1: DD, 2: DC, 3: CC, 4: CB, 5: BB, 6: BA, 7: AA

**Data Analysis and Interpretation**:

The system should provide tools for analyzing and interpreting student evaluation data and learning outcomes. It should support statistical analysis, data visualization, and trend identification to uncover patterns and insights in the data. Advanced analytics techniques such as regression analysis, correlation analysis, and predictive modeling should be available to identify relationships between variables and predict future outcomes.

**Curriculum Alignment and Assessment**:

The system should facilitate the alignment of curriculum objectives with assessment methods and learning outcomes. Educators should be able to map learning objectives to specific assessment tasks and track the attainment of learning outcomes over time. The system should support formative and summative assessment practices, allowing educators to evaluate student progress and mastery of course objectives.

**Feedback and Reporting**:

The system should enable the generation of comprehensive reports and dashboards to communicate assessment results and learning outcomes to stakeholders. Reports should include summary statistics, trends, and visualizations to facilitate data-driven decision-making and program evaluation. Feedback mechanisms should be available to provide students with actionable insights and recommendations for improvement based on assessment results.

**Student_submission.csv**

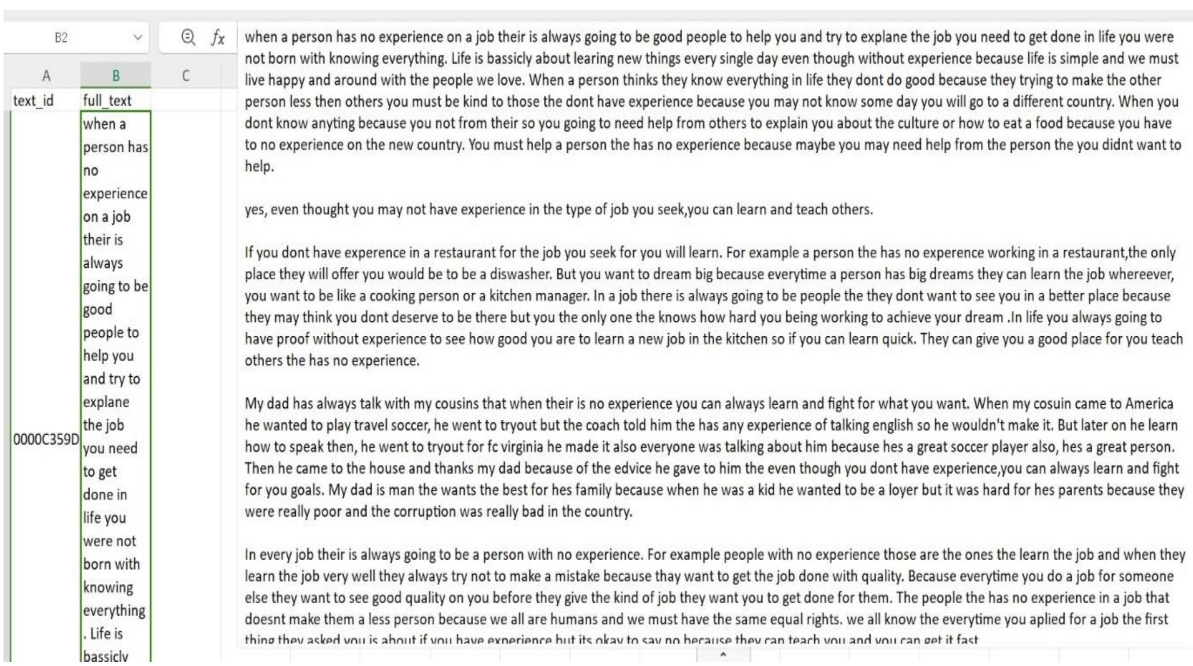



**Figure 2.2**: student_submission.csv dataset

| text_id | concerned | resentful | uncertain | dissatisfied | confident | contented |
|---|---|---|---|---|---|---|
| 0000C359D | 3.0943634 | 2.8380463 | 3.0481667 | 2.9718794 | 2.8290546 | 2.9254607 |
| 000BAD50D | 3.0452755 | 2.8925458 | 3.2076716 | 3.034207 | 3.1281083 | 3.4701545 |
| 00367BB25 | 3.1425845 | 3.1864544 | 3.6063074 | 3.2928004 | 3.7125855 | 3.6055372 |

**Figure 2.3**: Submission.csv dataset

| text_id | full_text | concerned | resentful | uncertain | dissatisfied | confident | contented |
|---|---|---|---|---|---|---|---|
| 0016926B07 | I think that students would benefit from learning at home,because they wont have to change and get up early in the morning to shower and do there hair. taking only classes helps them because at there house they'll be pay more attention. they will be comfortabl | 3.5 | 3.5 | 3 | 3 | 4 | 3 |
| | When a problem is a change you have to let it do the best on you no | | | | | | |

**Figure 2.4**: Train.csv dataset

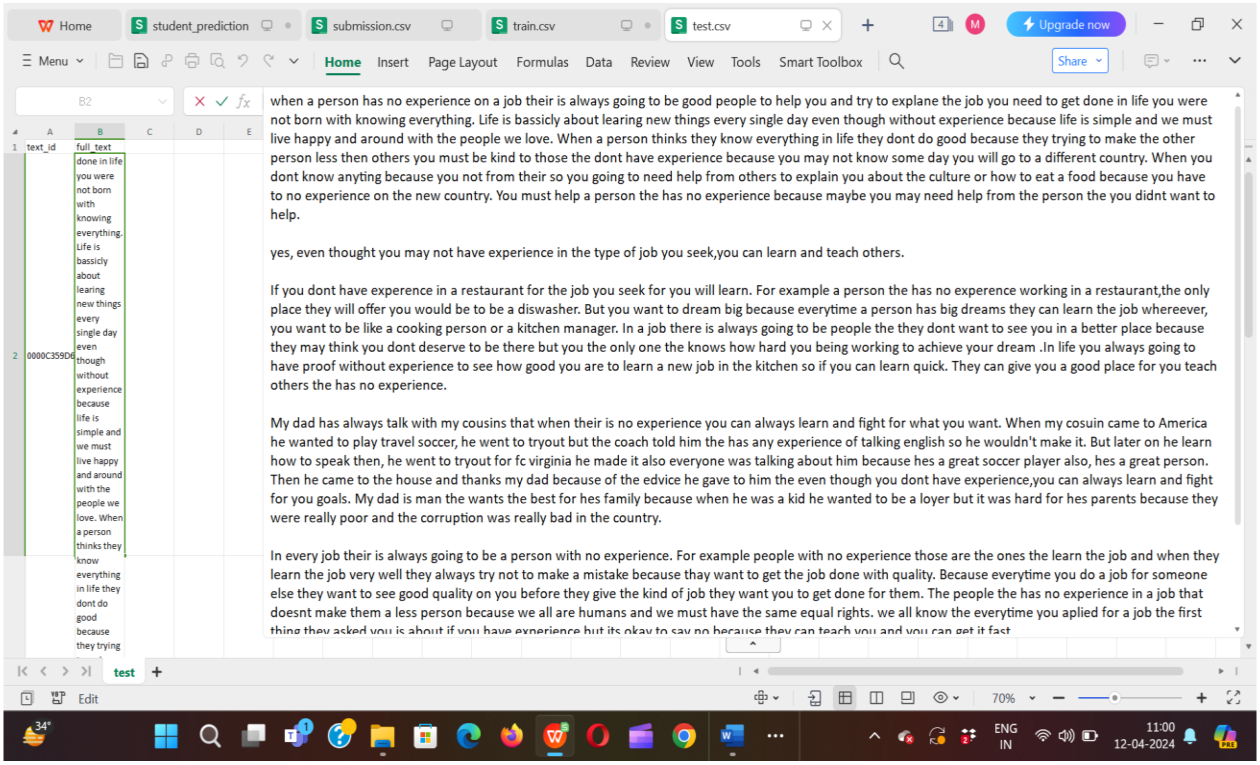



**Figure 2.5**: Test.csv data

**Student Engagement and Support**:

The system should promote student engagement and involvement in the assessment process. Students should have access to their assessment data, learning progress, and personalized feedback to support their academic development. Support resources, such as tutorials, study guides, and academic advising services, should be accessible through the system to assist students in addressing areas of weakness and achieving their academic goals.

**Administrative Tools and Workflow Management:**

Administrators should have access to tools for managing assessment workflows, including data collection, analysis, and reporting. The system should support role-based access control, allowing administrators to assign permissions and responsibilities to users based on their roles and responsibilities. Workflow automation features should streamline administrative tasks and facilitate collaboration among stakeholders involved in the assessment process.

**Integration and Interoperability**:

The system should integrate seamlessly with existing educational technologies and systems, including student information systems, learning management systems, and assessment platforms. It should support data exchange protocols and standards to facilitate interoperability and data sharing with external systems and databases. APIs and web services should be available to enable integration with third-party applications and services for enhanced functionality and interoperability.

These function requirements outline the essential capabilities and features that a system for evaluating the impact of student evaluation and learning outcomes in education should possess. By meeting these requirements, the system can effectively support assessment practices, improve educational outcomes, and enhance the overall quality of teaching and learning experiences.

## 2.3.2 Non-Functional Requirements

Non-functional requirements describe the qualities or attributes of the system, such as performance, security, usability, and scalability. Here are the non-functional requirements for a system evaluating the impact of student evaluation and learning outcomes in education:

**Performance:**

The system should respond promptly to user interactions, with minimal latency for data retrieval, analysis, and reporting. It should handle concurrent user access and large volumes of data without compromising performance or responsiveness.

**Scalability:**

The system should scale seamlessly to accommodate increases in data volume, user base, and system complexity over time. It should support horizontal scaling by adding additional hardware resources or employing distributed processing techniques as needed.

**Reliability:**

The system should operate reliably under normal conditions and maintain availability in the face of potential failures or disruptions. It should incorporate mechanisms for fault tolerance and error handling to minimize downtime and ensure continuous operation.

**Security:**

The system should adhere to industry best practices for data security and privacy protection, ensuring the confidentiality, integrity, and availability of sensitive information. User authentication and authorization mechanisms should be implemented to control access to system resources and data based on user roles and permissions. Data encryption should be employed for data transmission and storage to prevent unauthorized access or data breaches.

**Usability:**

The system should have an intuitive and user-friendly interface that is accessible to users with varying levels of technical expertise. Navigation should be straightforward, and features should be logically organized to facilitate user interaction and task completion. Help resources, tooltips, and documentation should be available to assist users in understanding system functionality and performing tasks effectively.

**Compatibility:**

The system should be compatible with a range of web browsers, operating systems, and devices to ensure accessibility for users across different platforms. It should support integration with external systems and technologies commonly used in educational environments, such as learning management systems and student information systems.

**Maintainability:**

The system should be modular and well-structured, with clear separation of concerns to facilitate code maintenance, updates, and enhancements. Documentation should be comprehensive and up-to-date, covering system architecture, design decisions, and technical specifications. Codebase should adhere to coding standards and best practices to promote readability, maintainability, and extensibility.

**Availability:**

The system should be available for use during scheduled operational hours, with minimal planned downtime for maintenance and updates. Backup and recovery mechanisms should be in place to ensure data integrity and continuity of service in the event of system failures or disasters.

**Performance:**

The system should be capable of handling high loads and concurrent user interactions without degradation in performance. Response times for critical operations, such as data analysis and report generation, should be optimized to ensure efficient use of system resources.

**Accessibility:**

The system should comply with accessibility standards, such as the Web Content Accessibility Guidelines (WCAG), to ensure equal access and usability for users with disabilities. Features such as keyboard navigation, screen reader compatibility, and alternative text for multimedia content should be implemented to support accessibility.

These non-functional requirements define the operational characteristics and qualities of the system, ensuring that it meets the performance, security, usability, and scalability needs of users and stakeholders.

## 2.4 Feasibility Study

### 2.4.1 Economical Feasibility

The economic feasibility of the project involves assessing the costs and benefits of its development and implementation. Given the resources required, including software development, hardware infrastructure, personnel, and ongoing maintenance, the project is economically viable. While there are some initial costs associated with acquiring the necessary technologies and expertise, the potential benefits outweigh these. The project's ability to improve educational outcomes, increase student engagement, and optimize resource allocation creates opportunities for significant cost savings and long-term. By carefully analyzing economic factors and projecting potential financial gains, the project is economically viable and has the potential to deliver significant value relative to its costs.

### 2.4.2 Technical Feasibility

From the perspective of technology, the project is highly feasible, relying on well-established technologies and frameworks to develop and implement the system. Python, Flask, Pandas, and machine learning libraries provide solid foundations for developing project functionalities. Furthermore, the availability of skilled developers and adequate resources for acquiring required hardware and software improves technical feasibility. The system's modular architecture promotes flexibility and scalability, allowing for the seamless integration of new features while also accommodating future enhancements. Overall, the project's technical feasibility is clear, as the necessary tools, expertise, and infrastructure are readily available to support its development and execution.

### 2.4.3 Operational feasibility:

Operational feasibility examines a project's compatibility with existing processes, workflows, and organizational structures. This project demonstrates operational feasibility through user-friendly interfaces, modular design, and the possibility of seamless integration with existing educational environments. The system's interfaces are intuitive and simple to use, reducing the need for extensive training and increasing user acceptance. Furthermore, the modular architecture allows for flexible customization to meet varying institutional needs and preferences. Stakeholder readiness and support for the system add to its operational feasibility, as users are more likely to adopt a solution that improves their workflow and productivity. Overall, the project has strong operational feasibility, with the potential to seamlessly integrate into existing educational practices while providing tangible benefits to users.

# CHAPTER 3

# SYSTEM DESIGN

## 3.1 Design Models (UML, DFD and ERD)

### Sequence diagram

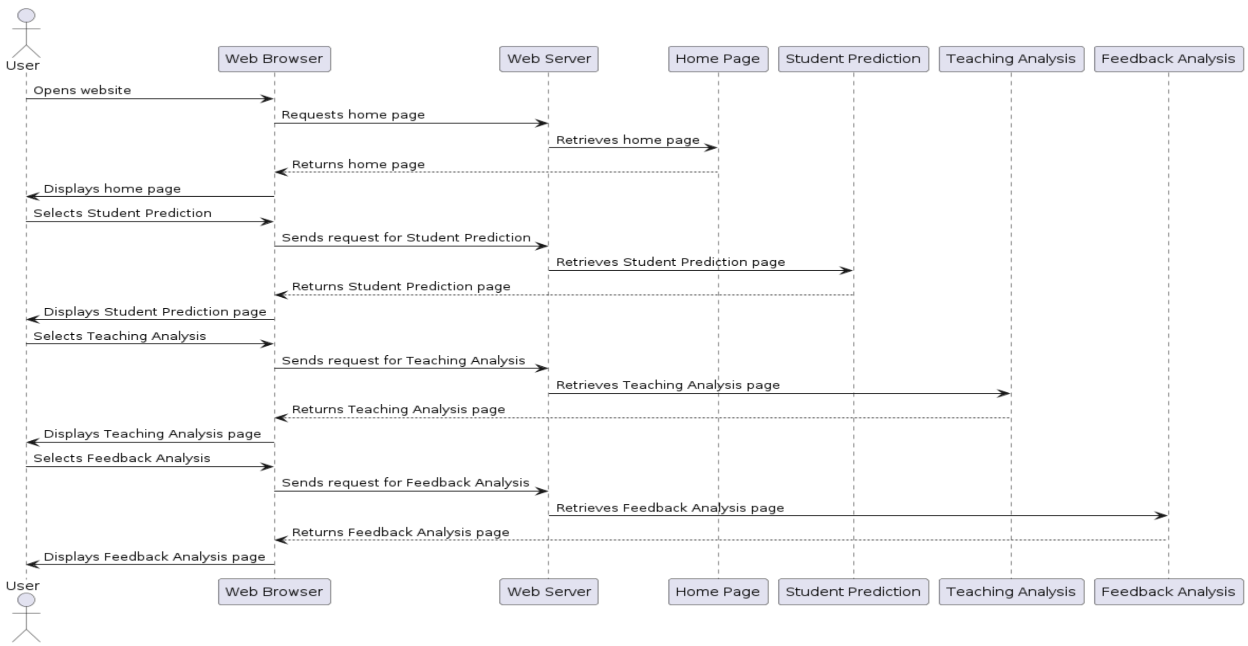



**Figure 3.1:** Sequence diagram

The image represents a sequence diagram of a person interacting with a website. The user loads the webpage into a web browser, which sends a request to the web server. The web server then delivers the home page to the web browser, which renders it for the user. The user can see three buttons that is student Prediction button, teaching analysis, and feedback analysis the user can interact with any of this webpage. If user next the Student Prediction button, which initiates a request to the web server. The web server then delivers the Student Prediction page to the web browser, which renders it for the user. The user then picks the Teaching Analysis button, which initiates a request to the web server. The web server then delivers the Teaching Analysis page to the web browser, which renders it for the user. The user then picks the Feedback Analysis button, which initiates a request to the web server. The web server then delivers the Feedback Analysis page to the web browser, which renders it for the user.

## Use case diagram

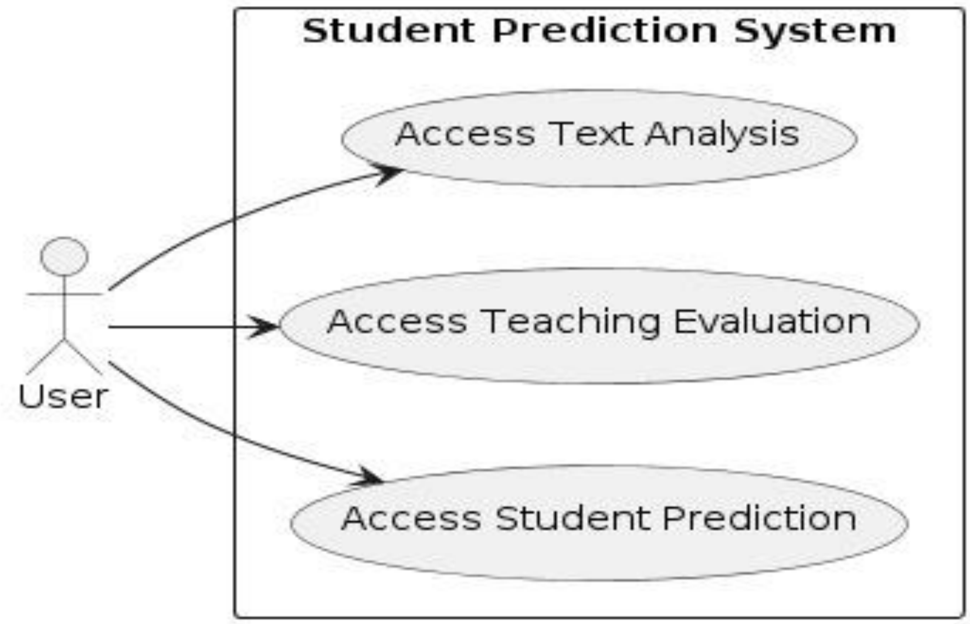



**Figure 3.2:** Use case diagram

According to the graphic, the student prediction system appears to provide insights and predictions about students through three primary components:

**Access Text Analysis**: This component is expected to analyze textual data about students, such as written assignments, discussion postings, or other text-based inputs. This textual data analysis could aid in identifying trends, themes, or indicators that can be used to influence overall student predictions.

**Access Teaching Evaluation:** This component assesses the teaching procedures and methods used. By analyzing teaching quality, effectiveness, and impact, the system may be able to identify areas where modifications could improve student outcomes.

**Access Student Prediction:** This is the system's final output, in which the insights and data acquired from text analysis and teaching evaluation are used to provide predictions regarding individual student performance, progress, or possible obstacles. This prediction component is the result of the system's efforts to offer the user (most likely an instructor or administrator) with useful information to aid decision-making and interventions.

The overall structure of this student prediction system indicates a thorough strategy to examining many data sources, evaluating teaching practices, and, ultimately, predicting student success. This type of technology could help educators, administrators, and educational institutions identify at-risk pupils, better allocate resources, and conduct focused interventions to increase student achievement.

**Data Processing Diagram**

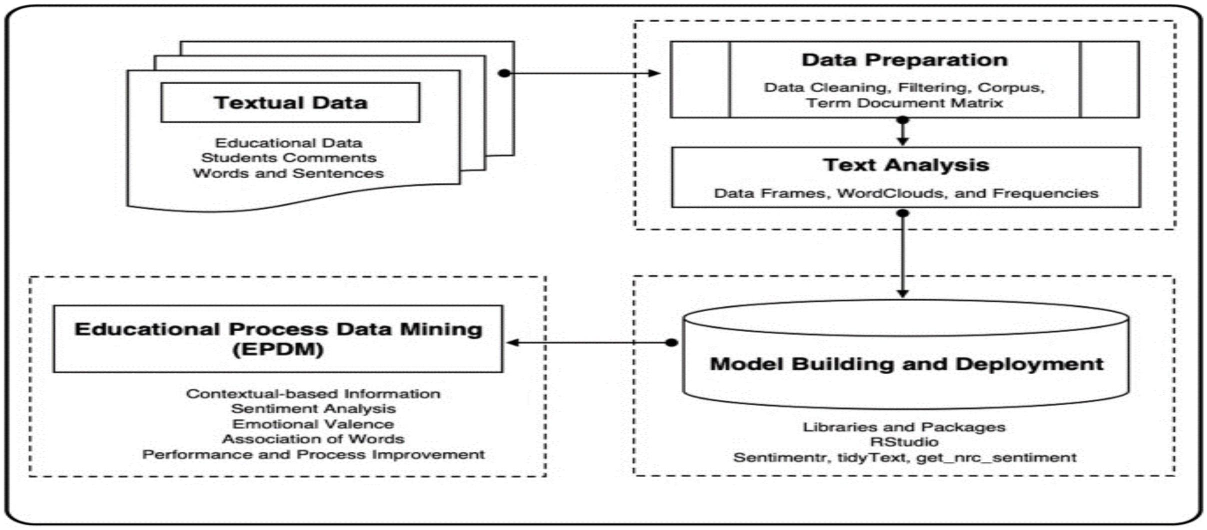



**Figure 3.3:** Data Processing Diagram

The image has provided depicts a flowchart for a process called Educational Process Data Mining (EPDM)

**Textual Data**: This is the raw input for the process. It consists of educational data, which could include students' comments, words, and sentences. This data is typically unstructured and needs to be processed to extract useful information.

**Data Preparation**: This step involves cleaning and organizing the raw textual data. It may include removing irrelevant information (data cleaning), selecting the relevant parts of the data (filtering), organizing the data into a structured format (corpus), and creating a term document matrix, which is a mathematical matrix that describes the frequency of terms that occur in a collection of documents.

**Text Analysis**: In this phase, the prepared data undergoes analysis. This can involve creating data frames (tabular structures in R or Python that organize data into rows and columns), generating word clouds (visual representations of text data where the size of each word indicates its frequency or importance), and calculating frequencies of specific words or phrases.

**Model Building and Deployment**: This step is about creating a predictive model or algorithm that can analyze the text data and deploying it for practical use. It involves using libraries and packages from programming environments like RStudio, and functions like sentiment, tidytext, get_nrc_sentiment which are used for sentiment analysis and other text analysis tasks.

**Educational Process Data Mining (EPDM)**

This is the overarching process that encompasses all the previous steps. It focuses on extracting contextual-based information, analyzing sentiment and emotional valence (the intrinsic attractiveness/goodness or averseness/badness of an event, object, or situation), understanding the association of words, and using all this information for performance and process improvement within an educational setting. The flowchart outlines a systematic approach to handling and analyzing textual data in education, which can be used to gain insights into student feedback, improve educational content, and enhance teaching methods.

## Activity Diagram:

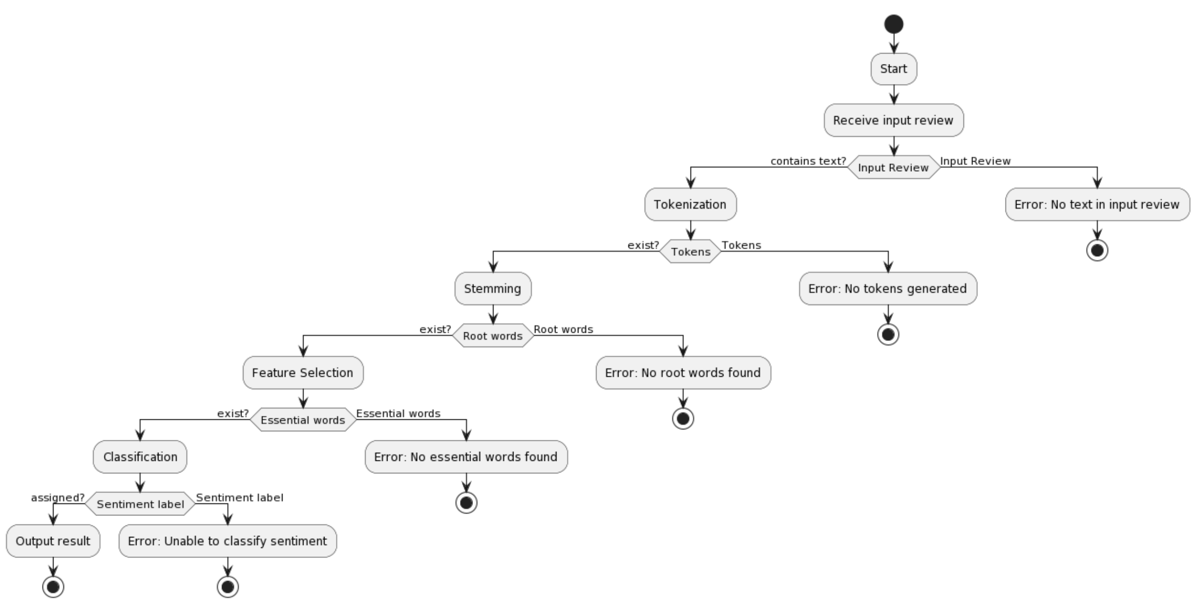



**Figure 3.4:** Activity Diagram

**Start**: The process begins with the "Start" symbol, indicating the starting point of the activity.

**Receive input review**: The system receives an input review from the user. This review contains text that needs to be analyzed for sentiment.

**Check if input review contains text**: This decision point checks if the input review actually contains text. If there is no text in the input review, an error message is displayed, indicating that there is no text to analyze.

**Tokenization**: If the input review contains text, the text is tokenized, meaning it is split into individual words or tokens. Tokenization is a fundamental step in natural language processing.

**Check if tokens exist**: This decision point checks if tokens were successfully generated from the input text. If no tokens are generated, it means there was an issue with the tokenization process, and an error message is displayed.

**Stemming**: Assuming tokens are generated successfully, the tokens are then stemmed. Stemming involves reducing each word to its root form. For example, "running" and "runs" would both be stemmed to "run".

**Check if root words exist**: This decision point verifies if root words were successfully generated through stemming. If there are no root words, it means there was an issue with the stemming process, and an error message is displayed.

**Feature Selection**: After stemming, essential words or features are selected from the root words. Feature selection involves identifying key words that are relevant for sentiment analysis.

**Check if essential words exist**: This decision point checks if essential words were successfully selected. If there are no essential words, it indicates an issue with the feature selection process, and an error message is displayed.

**Classification**: The selected essential words are then used to classify the sentiment of the input review. This step involves determining whether the sentiment is positive or negative.

**Check if sentiment label is assigned**: This decision point verifies if a sentiment label was successfully assigned to the input review. If there is no assigned sentiment label, it means there was an issue with the classification process, and an error message is displayed.

**Output result**: If a sentiment label is successfully assigned, the result (positive or negative sentiment) is outputted

**Stop:** The process ends here, indicated by the "stop" symbol.

## 3.2 Functional Design

**Home Page**

The main page welcomes users and provides an overview of the system's capabilities. It has navigation links to many modules and features, including student prediction, teaching analysis, and feedback analysis. Additionally, the home page may include system-related announcements, news, or updates.

**Student Prediction Module**

The Student Prediction Module enables users to forecast student performance based on a variety of input parameters, including previous academic records, attendance, study habits, and socioeconomic status. Users can enter student information into a user-friendly interface or upload datasets for batch processing. Machine learning algorithms examine input data to create predictive models. The module provides prediction results as well as insights into the elements that influence student performance.

**Teaching Analysis Module**

The Teaching Analysis Module allows users to assess teaching effectiveness by analyzing teacher performance, course outcomes, and student comments. It combines information from course evaluations, student questionnaires, and academic success measurements. Users can view instructional assessment indicators including teacher ratings, course completion rates, and student engagement levels. The module also allows for comparison analysis across several instructors or courses.

**Feedback Analysis Module**

The comments Analysis Module uses textual comments from students or teachers to extract significant information. It uses natural language processing (NLP) techniques for sentiment analysis, topic modeling, and keyword extraction. Users can investigate sentiment patterns, find repeating themes, and extract actionable feedback from unstructured text data. Word clouds, sentiment heatmaps, and topic clusters are examples of visualizations that can assist users comprehend feedback patterns and sentiment distribution.

**Data Visualization and Reporting**

The Data Visualization and Reporting module uses interactive visuals and configurable reports to effectively communicate insights. Users can generate a variety of charts, graphs, and dashboards to display prediction model results, teaching evaluations, feedback analysis, and other important information. The module has dynamic filtering, drill-down functionality, and export choices to help in data discovery and sharing.

## 3.3 User Interface Design

The Student Evaluation System's user interface is designed to provide straightforward navigation and clear information presentation. It has a homepage that provides quick access to modules such as student prediction, instructor assessment, and feedback analysis. Each module is intended to be user-friendly, with forms for entering necessary data and result pages that present predictions or analysis results. The interface has a clean and well-organized layout, with colors and fonts used to improve readability and aesthetics. Furthermore, interactive components like as dropdown menus and buttons are used to enhance user involvement and experience. Overall, the interface is intended to simplify the process of accessing and evaluating performance data for both students and teachers.

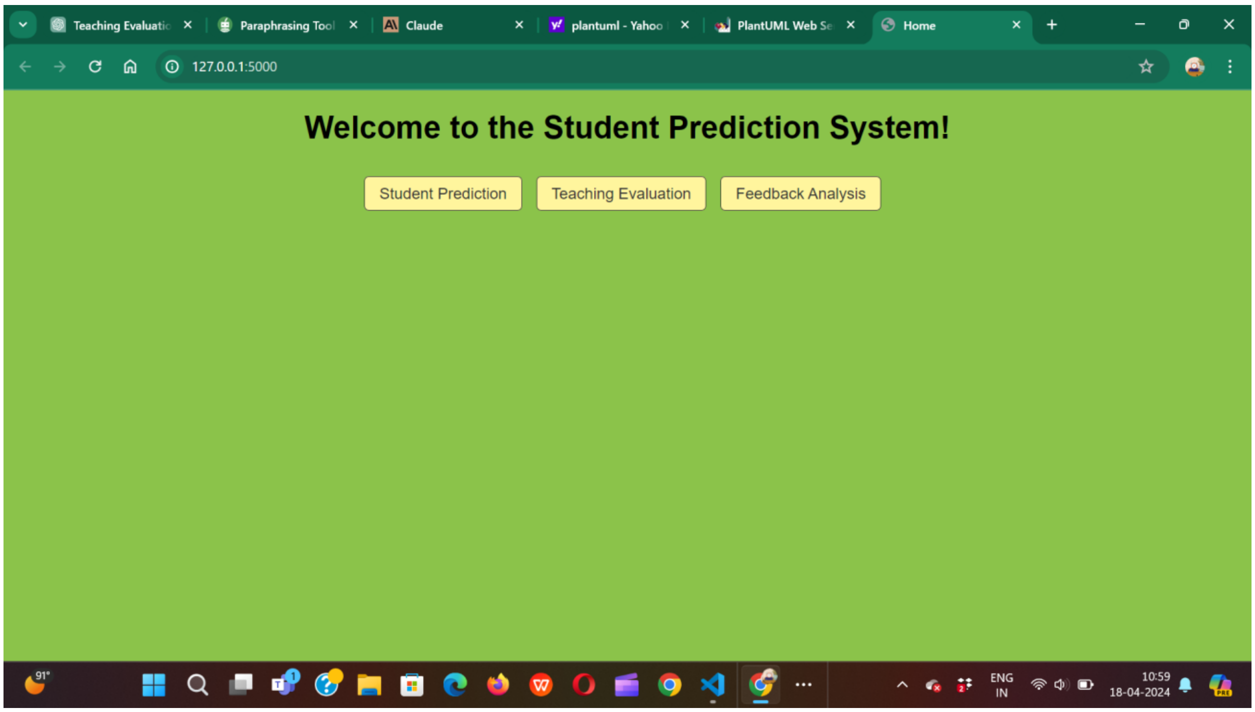



**Figure 3.5**: Home page

Student Prediction appears to be the system's core function or module, which generates student performance.
Teaching Evaluation is another important component that will most likely focus on evaluating and assessing teaching methods and practices.
Feedback Analysis aspect indicates that the system may evaluate feedback, such as student or feedback comments, to inform the overall process.

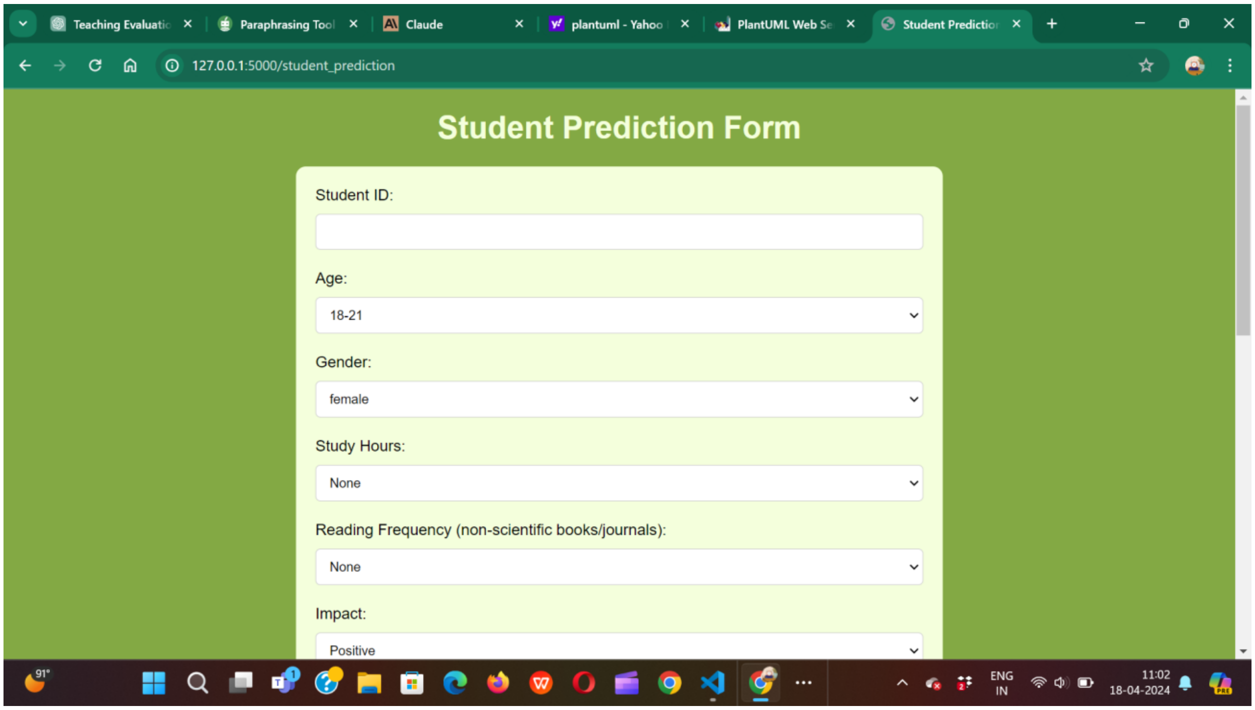



**Figure 3.6:** Student prediction form1

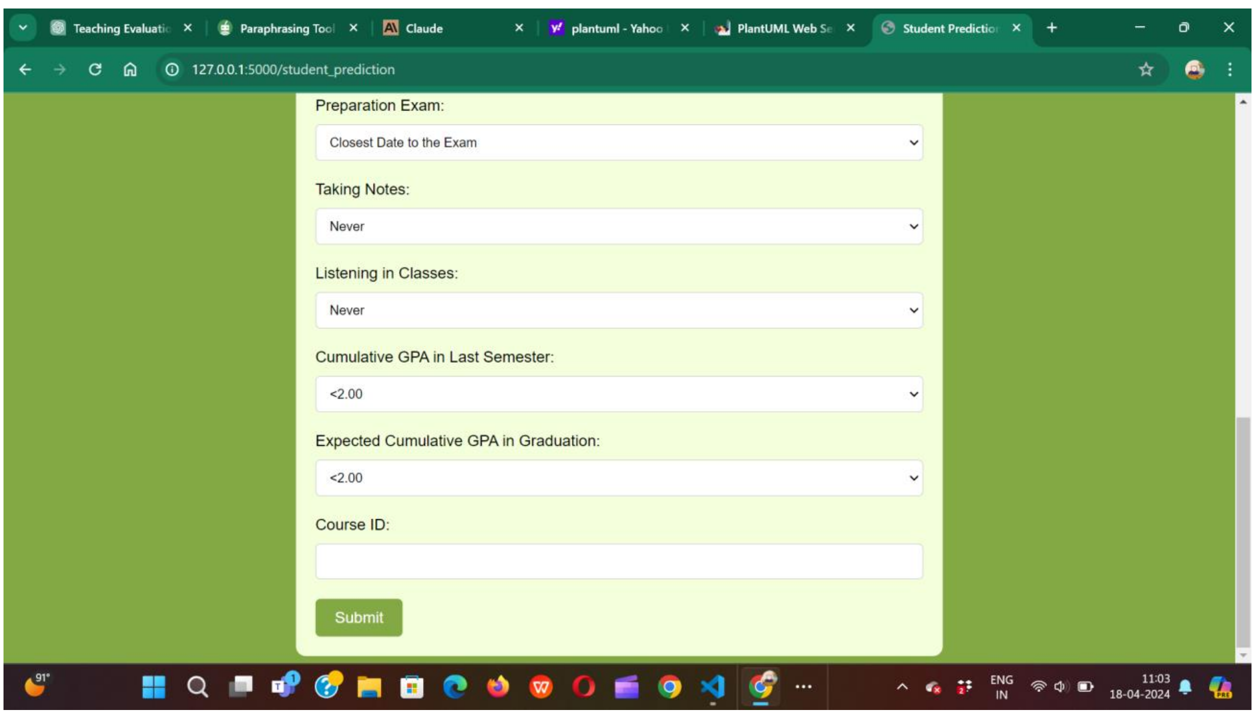



**Figure 3.7:** Student prediction from2

# CHAPTER 4

# IMPLEMENTAION AND RESULTS

## 4.1 About Software

The Student Performance Analysis System is a complete platform that addresses the various areas of student performance evaluation. The system has an effective layout that allows for efficient data processing, analysis, and visualization. It was built using Python and the Flask web framework. Using a variety of libraries such as pandas, scikit-learn, matplotlib, and seaborn, the software allows users to delve deep into student data and extract significant insights to guide teaching techniques.

At its core, the system includes several capabilities that cater to various stakeholders in the educational ecosystem. The software's modular design allows administrators to see overall trends while educators can adjust teaching techniques. Each module, rigorously designed and deployed, has a specific purpose, ranging from predictive analytics for student performance to real-time feedback analysis.

The predictive analytics module demonstrates the system's advanced capabilities. The software estimates student outcomes using machine learning algorithms, considering a variety of criteria such as previous academic performance, extracurricular activity, and socioeconomic background. These projections not only help educators identify at-risk pupils, but they also enable proactive actions to promote student achievement. Furthermore, the teaching evaluation component provides instructors with crucial information about their educational performance. Instructors get a comprehensive understanding of their educational approaches by analyzing advanced indicators such as student engagement, assessment performance, and course evaluation. With such knowledge, instructors can fine-tune their teaching methods and create a more conducive learning atmosphere.

In parallel, the feedback analysis module facilitates positive conversation between students and educators. The approach promotes a culture of continuous development by collecting and analyzing input from a variety of sources, including questionnaires, quizzes, and peer assessments. Furthermore, the use of sentiment analysis algorithms allows for the automatic categorization of feedback, indicating areas of strength and areas for improvement.

In addition to its analytical capabilities, the Student Performance Analysis System has a user-centric interface that is intended to optimize usability and improve user experience. Intuitive navigation pathways, along with visually appealing dashboards and interactive visualizations, ensure seamless interaction for users at all skill levels. Furthermore, the system's adaptable design makes it accessible across a variety of devices, allowing stakeholders to interact with the platform at any time and from any location.

Furthermore, to its main functionalities, the system provides extensive support for exploratory data analysis (EDA), allowing users to obtain a deeper understanding of the underlying patterns and trends in data. Using interactive visualization tools and descriptive statistical analysis, educators and administrators can discover hidden relationships, outliers, and patterns, creating the framework for data-driven decision-making.

Overall, the Student Performance Analysis System exemplifies educational technology innovation by bridging the gap between data science and pedagogy. By leveraging the power of data analytics and machine learning, the software provides educators and administrators with the resources they need to support student achievement, drive instructional improvement, and cultivate a culture of ongoing learning and development.

## 4.2 Code Implementation

**Data Preprocessing:**

The code loads student data from CSV files ('train.csv' and 'test.csv'). It uses basic exploratory data analysis (EDA) to visualize the distribution of score characteristics and essay lengths. Missing values in the dataset are detected using the 'isna()' method to ensure data integrity. A heatmap is used to investigate the association between several data, revealing probable links.

**Vectorization and Modeling:**

NLTK packages are used to perform text preprocessing techniques including tokenization, stemming, and filtering on essays. The Doc2Vec (D2V) model is used to encode the writings into numerical vectors, which contain semantic information. Each essay is labelled with its matching score attributes, which aids in supervised learning. The processed essays are separated into training and testing sets, and tagged appropriately for model training.

**Model Training:**

Six models are trained to predict certain score features. The LightGBM regressor was chosen as the principal model because of its efficiency and effectiveness in dealing with huge datasets. To evaluate the models' predictive performance, appropriate regression metrics are used. Hyperparameter tweaking techniques may be used to improve the model's performance even further.

**User Interface Development**

While not directly implemented in the offered code, user interface development usually entails generating input forms where users may enter essay text and other pertinent information. The interface will show the projected grades for various aspects of the essay, giving consumers practical feedback. Visualization tools, such as histograms or scatter plots, could be integrated into the interface to improve data display and interpretation.

**Deployment:**

The code snippet does not cover application deployment, which involves packaging trained models and interfaces into a web application or API. The backend may be developed using technologies such as Flask, while the frontend would be developed using HTML, CSS, and JavaScript. Deployment to a web server would ensure users' internet access. Encryption, authentication, and permission would be used to protect user data and ensure compliance with privacy laws.

### 4.2.1 Complete Project Coding

**app.py**

```
import pandas as pd
from flask import Flask, render_template, request
from sklearn.metrics.pairwise import euclidean_distances
app = Flask(__name__)
# Load the dataset
df = pd.read_csv('student_prediction.csv')
# Function to predict the SUMM value based on input data
def predict_summ(data):
    try:
        # Extract input values from the form data
        input_values = [
            int(data['age']),
            int(data['gender']),
            int(data['study_hrs']),
            int(data['read_freq']),
            int(data['impact']),
            int(data['attend']),
            int(data['prep_study']),
            int(data['prep_exam']),
            int(data['notes']),
            int(data['listens']),
            int(data['cuml_gpa']),
            int(data['exp_gpa'])
        ]
        # Select relevant columns from the dataset
        relevant_columns = [
            'AGE', 'GENDER', 'STUDY_HRS', 'READ_FREQ', 'IMPACT', 'ATTEND',
            'PREP_STUDY', 'PREP_EXAM', 'NOTES', 'LISTENS', 'CUML_GPA', 'EXP_GPA'
        ]
       # Subset the dataset with relevant columns
        relevant_data = df[relevant_columns]
       # Calculate distances between input values and dataset records
        distances = euclidean_distances(relevant_data, [input_values])
        # Find the index of the record with the smallest distance
        closest_index = distances.argmin()
        # Get the SUMM value from the closest matching record
```

```
        summm = df.iloc[closest_index]['SUMM']
       return summ
    except KeyError as e:
        return f"Error: {e}"
@app.route('/')
def index():
    return render_template('index.html')
@app.route('/student_prediction', methods=['GET', 'POST'])
def student_prediction():
    if request.method == 'POST':
        # Get form data
        form_data = request.form
        # Predict SUMM
        summ = predict_summ(form_data)
        # Render template with the SUMM result
        return render_template('result.html', summ=summ)
    else:
        # Render the form template for GET requests
        return render_template('student_prediction_form.html')
@app.route('/teaching_evaluation')
def teaching_evaluation():
    return render_template('teaching_evaluation.html')
@app.route('/text_analysis')
def text_analysis():
    return render_template('text_analysis.html')
if __name__ == '__main__':
    app.run(debug=True)
```

**home.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home</title>
    <style>
        /* Reset default margin and padding */
        body, ul {
```

```
        margin: 0;
        padding: 0;
      }
  body {
        font-family: Arial, sans-serif;
        background-color: #8bc34a; /* Background color for the body */
        color: #000; /* Text color */
      }
  h1 {
        color: #000; /* Text color for the h1 element */
        text-align: center;
        padding: 20px 0;
        margin: 0;
        background-color: #8bc34a; /* Background color for the h1 element */
      }
ul {
        list-style-type: none;
        text-align: center;
        margin-top: 20px; /* Spacing between heading and list */
      }
li {
        display: inline-block; /* Display list items as inline blocks */
        margin-right: 10px;
      }
 a {
        text-decoration: none;
        color: #333; /* Link color */
        padding: 8px 15px; /* Add padding to links */
        border: 1px solid #333; /* Add border to links */
        border-radius: 5px; /* Add border radius to links */
        transition: background-color 0.3s ease; /* Smooth transition for background color */
      }
a:hover {
        background-color: #555; /* Hover background color for links */
        color: #fff; /* Text color on hover */
      }
   </style>
</head>
<body>
```

```
    <h1>Welcome to the Student Prediction System!</h1>
    <ul>
        <li><a href="/student_prediction">Student Prediction</a></li>
        <li><a href="/teaching_evaluation">Teaching Evaluation</a></li>
        <li><a href="/text_analysis">Feedback Analysis</a></li>
    </ul>
</body>
</html>
```

**Index.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home</title>
    <style>
        /* Reset default margin and padding */
        body, ul {
            margin: 0;
            padding: 0;
        }
body {
            font-family: Arial, sans-serif;
            background-color: #8bc34a; /* Background color for the body */
            color: #000; /* Text color */
        }
h1 {
            color: #000; /* Text color for the h1 element */
            text-align: center;
            padding: 20px 0;
            margin: 0;
            background-color: #8bc34a; /* Background color for the h1 element */
        }

        ul {
            list-style-type: none;
            text-align: center;
```

```
            margin-top: 20px; /* Spacing between heading and list */
        }
 li {
            display: inline-block; /* Display list items as inline blocks */
            margin-right: 10px;
        }
a {
            text-decoration: none;
            color: #333; /* Link color */
            padding: 8px 15px; /* Add padding to links */
            border: 1px solid #333; /* Add border to links */
            border-radius: 5px; /* Add border radius to links */
            transition: background-color 0.3s ease; /* Smooth transition for background color */
            background-color: #fff59d; /* Default background color for links */
        }
a:hover {
            background-color: #fff; /* Hover background color for links */
            color: #333; /* Text color on hover */
        }
    </style>
</head>
<body>
    <h1>Welcome to the Student Prediction System!</h1>
    <ul>
        <li><a href="/student_prediction">Student Prediction</a></li>
        <li><a href="/teaching_evaluation">Teaching Evaluation</a></li>
        <li><a href="/text_analysis">Feedback Analysis</a></li>
    </ul>
</body>
</html>
```

**Result.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Result</title>
    <style>
```

```
        body {
            font-family: Arial, sans-serif;
            background-color: #fff59d; /* Background color */
            color: #000; /* Text color */
            margin: 0;
            padding: 0;
        }
   h1 {
            color: #000; /* Heading color */
            text-align: center;
            padding: 20px 0;
        }
        p {
            font-size: 18px;
            text-align: center;
            margin: 20px 0;
        }
    </style>
</head>
<body>
    <h1>Prediction Result</h1>
    <p>The predicted SUMM value is: {{ summ }}</p>
</body>
</html>
```

**Student_prediction_from.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Student Prediction Form</title>
     <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #83a944;
            color: #000;
        }
        h1 {
```

```
            text-align: center;
            color: #f4ffdc;
        }
        form {
            width: 50%;
            margin: 0 auto;
            background-color: #f4ffdc;
            padding: 20px;
            border-radius: 10px;
        }
        label {
            display: block;
            margin-bottom: 10px;
        }
        select, input[type="text"], input[type="number"] {
            width: 100%;
            padding: 10px;
            margin-bottom: 20px;
            border: 1px solid #ccc;
            border-radius: 5px;
            box-sizing: border-box;
        }
        button {
            padding: 10px 20px;
            background-color: #83a944;
            color: #fff;
            border: none;
            border-radius: 5px;
            cursor: pointer;
            font-size: 16px;
        }
        button:hover {
            background-color: #628436;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Student Prediction Form</h1>
```

```
  <form action="/student_prediction" method="post">
  <label for="student_id">Student ID:</label>
  <input type="text" id="student_id" name="student_id" required><br>
<label for="age">Age:</label>
  <select id="age" name="age">
    <option value="1">18-21</option>
    <option value="2">22-25</option>
    <option value="3">above 26</option>
  </select><br>
  <label for="gender">Gender:</label>
  <select id="gender" name="gender">
    <option value="1">female</option>
    <option value="2">male</option>
  </select><br>
  <label for="study_hrs">Study Hours:</label>
  <select id="study_hrs" name="study_hrs">
    <option value="1">None</option>
    <option value="2">&lt;5 hours</option>
    <option value="3">6-10 hours</option>
    <option value="4">11-20 hours</option>
    <option value="5">more than 20 hours</option>
  </select><br>
  <label for="read_freq">Reading Frequency (non-scientific books/journals):</label>
  <select id="read_freq" name="read_freq">
    <option value="1">None</option>
    <option value="2">Sometimes</option>
    <option value="3">Often</option>
  </select><br>
<label for="impact">Impact:</label>
  <select id="impact" name="impact">
    <option value="1">Positive</option>
    <option value="2">Negative</option>
    <option value="3">Neutral</option>
  </select><br>
  <label for="attend">Attendance:</label>
  <select id="attend" name="attend">
    <option value="1">Always</option>
    <option value="2">Sometimes</option>
    <option value="3">Never</option>
```

```
 </select><br>
 <label for="prep_study">Preparation Study:</label>
 <select id="prep_study" name="prep_study">
    <option value="1">Alone</option>
    <option value="2">With Friends</option>
    <option value="3">Not Applicable</option>
 </select><br>
 <label for="prep_exam">Preparation Exam:</label>
 <select id="prep_exam" name="prep_exam">
    <option value="1">Closest Date to the Exam</option>
    <option value="2">Regularly During the Semester</option>
    <option value="3">Never</option>
 </select><br>
<label for="notes">Taking Notes:</label>
 <select id="notes" name="notes">
    <option value="1">Never</option>
    <option value="2">Sometimes</option>
    <option value="3">Always</option>
 </select><br>
<label for="listens">Listening in Classes:</label>
 <select id="listens" name="listens">
    <option value="1">Never</option>
    <option value="2">Sometimes</option>
    <option value="3">Always</option>
 </select><br>
 <label for="cuml_gpa">Cumulative GPA in Last Semester:</label>
 <select id="cuml_gpa" name="cuml_gpa">
    <option value="1">&lt;2.00</option>
    <option value="2">2.00-2.49</option>
    <option value="3">2.50-2.99</option>
    <option value="4">3.00-3.49</option>
    <option value="5">Above 3.49</option>
 </select><br>
 <label for="exp_gpa">Expected Cumulative GPA in Graduation:</label>
 <select id="exp_gpa" name="exp_gpa">
    <option value="1">&lt;2.00</option>
    <option value="2">2.00-2.49</option>
    <option value="3">2.50-2.99</option>
    <option value="4">3.00-3.49</option>
```

```
        <option value="5">Above 3.49</option>
      </select><br>

      <label for="course_id">Course ID:</label>
      <input type="number" id="course_id" name="course_id" required><br>
        <button type="submit">Submit</button>
      </form>
    </div>
</body>
</html>
```

**Teaching_evaluation.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Teaching Evaluation</title>
    <style>
      body {
        font-family: Arial, sans-serif;
        background-color: #fff59d; /* Background color */
        color: #333; /* Text color */
        margin: 0;
        padding: 0;
      }
      h1 {
        color: #000; /* Heading color */
        text-align: center;
        padding: 20px 0;
      }. image-container {
        text-align: center;
        margin-bottom: 20px;
      }.image-container img {
        max-width: 50%; /* Set maximum width of image */
        height: 50%%; /* Maintain aspect ratio */
      }.caption {
        font-style: italic;
        font-size: 18px;
```

```
            margin-top: 5px;
        }
    </style>
</head>
<body>
    <h1>Teaching Evaluation Results</h1>
        <div class="image-container">
        <img src="{{ url_for('static', filename='teaching_results/countplotodsumm.png') }}" alt="Count Plot of SUMM">
        <p class="caption">Count Plot of SUMM: This plot illustrates the distribution of SUMM values among students.</p>
    </div>
    <div class="image-container">
        <img src="{{ url_for('static', filename='teaching_results/gendervaluecountofdata.png') }}" alt="Gender Value Count of Data">
        <p class="caption">Gender Value Count of Data: This plot shows the count of each gender category in the dataset.</p>
    </div>
    <div class="image-container">
<img
src="{{url_for('static',filename='teaching_results/plotrepresentsthecourse,exp_gpa,cuml_gpa,prep_exam.png') }}" alt="Plot Representing Course, EXP_GPA, CUML_GPA, Prep_Exam">
        <p class="caption">Plot Representing Course, EXP_GPA, CUML_GPA, Prep_Exam: This plot represents the relationship between course, expected GPA, cumulative GPA, and preparation for the exam.</p>
    </div>
      <div class="image-container">
        <img src="{{ url_for('static',filename='teaching_results/plotrepresentwork,scholarship.png') }}" alt="Plot Representing Work, Scholarship">
        <p class="caption">Plot Representing Work, Scholarship: This plot illustrates the correlation between work and scholarship factors.</p>
    </div>
    </body>
</html>
```

**Text_analysis.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Text Analysis</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #fff59d; /* Background color */
            color: #333; /* Text color */
            margin: 0;
            padding: 0;
        }

        h1 {
            color: #000; /* Heading color */
            text-align: center;
            padding: 20px 0;
        }
        .image-container {
            text-align: center;
            margin-bottom: 20px;
        }
        .image-container img {
            max-width: 50%; /* Set maximum width of image */
            height: 50%%; /* Maintain aspect ratio */
        }.caption {
            font-style: italic;
            font-size: 18px;
            margin-top: 5px;
        }
    </style>
</head>
<body>
    <h1>Feedback Analysis Results</h1>
    <div class="image-container">
```

```
        <img
src="{{url_for('static',filename='teaching_results/text_analysis_results/findingthecoorelationofte
xt.png') }}" alt="Finding the Corelation of Text">
        <p class="caption">Finding the Corelation of Text: This image shows the analysis of text
data to find correlations between different variables.</p>
    </div> <div class="image-container">
        <img
src="{{url_for('static', filename='teaching_results/text_analysis_results/score_featureoftext.png')
}}" alt="Score Feature of Text">
        <p class="caption">Score Feature of Text: This image displays the scores assigned to
various features extracted from text data.</p>
    </div>
</body>
</html>
```

### 4.2.2 Comments and Description of Coding Segments

**app.py**

**import pandas as pd:**
Imports the pandas library under the alias pd, which is commonly used for data manipulation.
**from flask import Flask, render_template, request:**
Imports necessary modules from the Flask framework. Flask is used to create the web application instance, render_template is used to render HTML templates, and request is used to handle incoming requests.
**app = Flask(__name__)**
This line initializes a Flask application instance named app using the constructor of the Flask class. __name__ is a special Python variable that represents the name of the current module.
**df = pd.read_csv('student_prediction.csv')**
This line reads a CSV file named 'student_prediction.csv' containing student data into a pandas DataFrame named df. The dataset is assumed to be in the same directory as the script.
**def predict_summ(data):**
**# Function body**
This function, predict_summ, takes form data as input and predicts the 'SUMM' value based on the provided data. It extracts relevant input values, calculates Euclidean distances between the input values and dataset records, finds the closest record, and returns the corresponding 'SUMM' value.
**@app.route('/')**
**def index():**
**return render_template('index.html')**
This route handles requests to the root URL ('/'). It renders the 'index.html' template, which likely serves as the homepage of the web application.
**@app.route('/student_prediction', methods=['GET', 'POST'])**
**def student_prediction(): # Function body**
This route ('/student_prediction') handles both GET and POST requests. For POST requests, it collects form data, predicts the 'SUMM' value, and renders the 'result.html' template with the prediction result. For GET requests, it renders the 'student_prediction_form.html' template, likely containing a form for inputting data.
**@app.route('/teaching_evaluation')**
**def teaching_evaluation():**
**return render_template('teaching_evaluation.html')**
This route ('/teaching_evaluation') renders the 'teaching_evaluation.html' template, presumably for conducting teaching evaluations.
**@app.route('/text_analysis')**

```
def text_analysis():
    return render_template('text_analysis.html')
```

This route ('/text_analysis') renders the 'text_analysis.html' template, likely used for text analysis purposes.

```
if __name__ == '__main__':
    app.run(debug=True)
```

This block ensures that the Flask application is run when the script is executed directly (not imported as a module into another script). The debug=True parameter enables debug mode, which provides helpful debugging information and automatically reloads the application when changes are made to the code.

**Home.html**

```
<!DOCTYPE html>
```

This is a document type declaration, indicating that the document is an HTML5 document.

```
<html lang="en">
```

This is the opening tag for the HTML document. The `lang` attribute specifies the language of the document, which is English.

```
<head>
```

This is the opening tag for the head section of the HTML document, where meta-information about the document is placed.

```
    <meta charset="UTF-8">
```

This meta tag specifies the character encoding for the document, ensuring that it is interpreted correctly by browsers.

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

This meta tag sets the viewport properties, ensuring that the webpage renders properly on various devices with different screen sizes.

```
    <title>Home</title>
```

This sets the title of the webpage, which is displayed in the browser's title bar or tab.

```
    <style>
```

This is the opening tag for the `<style>` element, where CSS rules for styling the HTML content are defined**.**

```
        /* Reset default margin and padding */
        body, ul {
            margin: 0;
            padding: 0;
        }
```

These CSS rules reset the default margin and padding for the `body` and `ul` elements to zero, ensuring consistent spacing.

```
        body {
```

```
    font-family: Arial, sans-serif;
    background-color: #8bc34a;
    color: #000;
}
```

These CSS rules set the font family to Arial or a generic sans-serif font, set the background color to a shade of green (#8bc34a), and set the text color to black (#000) for the `body` element.

```
h1 {
    color: #000;
    text-align: center;
    padding: 20px 0;
    margin: 0;
    background-color: #8bc34a;
}
```

These CSS rules style the `h1` element, setting its color to black, centering the text, adding padding above and below the heading, removing margin, and setting a background color like the body.

```
ul {
    list-style-type: none;
    text-align: center;
    margin-top: 20px;
}
```

These CSS rules style the `ul` element, removing the default list-style type (bullets), centering the text, and adding top margin for spacing.

```
li {
    display: inline-block;
    margin-right: 10px;
}
```

These CSS rules style the `li` (list item) elements, displaying them as inline blocks and adding right margin for spacing between items.

```
a {
    text-decoration: none;
    color: #333;
    padding: 8px 15px;
    border: 1px solid #333;
    border-radius: 5px;
    transition: background-color 0.3s ease;
}
```

These CSS rules style the anchor (`<a>`) elements, removing the default underline (`text-decoration: none`), setting the text color to dark gray (#333), adding padding around the text, applying a border with rounded corners, and defining a smooth transition effect for background color changes.

```
    a:hover {
      background-color: #555;
      color: #fff;
    }
  </style>
```

These CSS rules define the styles for anchor elements when hovered over by the mouse. It changes the background color to a darker shade of gray (#555) and the text color to white (#fff).

**</head>**

This is the closing tag for the head section of the HTML document.

**<body>**

This is the opening tag for the body section of the HTML document, where the visible content of the webpage is placed.

```
  <h1>Welcome to the Student Prediction System!</h1>
```

This is a level 1 heading (`<h1>`) displaying the text "Welcome to the Student Prediction System!". It serves as the main heading of the webpage.

```
  <ul>
```

This is the opening tag for an unordered list (`<ul>`), which will contain navigation links.

```
    <li><a href="/student_prediction">Student Prediction</a></li>
    <li><a href="/teaching_evaluation">Teaching Evaluation</a></li>
    <li><a href="/text_analysis">Feedback Analysis</a></li>
```

These are list items (`<li>`) within the unordered list (`<ul>`), each containing an anchor (`<a>`) element representing a navigation link. The `href` attribute specifies the URL or route to which each link points.

```
  </ul>
```

This is the closing tag for the unordered list (`<ul>`).

**</body>**

This is the closing tag for the body section of the HTML document.

**</html>**

This is the closing tag for the HTML document.

**Index.html**

**<!DOCTYPE html>**

Declares the document type and version of HTML being used, in this case, HTML5.

**<html lang="en">**

Defines the root element of the HTML document with the language attribute set to English.

**<head>**

Contains meta-information about the HTML document, such as character encoding, viewport settings, and title.

**<meta charset="UTF-8">**

Specifies the character encoding of the document as UTF-8, which supports a wide range of characters**.**

**<meta name="viewport" content="width=device-width, initial-scale=1.0">**

Sets the viewport properties for responsive web design, ensuring the page renders properly on various devices**.**

**<title>Home</title>**

Sets the title of the HTML document to "Home," which appears in the browser's title bar or tab.

**<style>**

Begins the block for embedding CSS styles directly within the HTML document.

**/* Reset default margin and padding */:**

Comment indicating the purpose of the following CSS rules.

**body, ul { margin: 0; padding: 0; }**

Resets the default margin and padding for the `body` and `ul` elements to zero, removing any browser defaults**.**

**body { ... }**

Defines styles for the `body` element, including font family, background color, and text color.

**h1 { ... }**

Defines styles for the `h1` element, such as text color, alignment, padding, margin, and background color.

**li { ... }**

Defines styles for the `li` (list item) element, setting it to display as inline-block to appear horizontally, and adding right margin for spacing between list items.

**a { ... }**

Defines styles for anchor (`<a>`) elements, removing default underline, setting text color, padding, border, border radius, and transition effect for smooth background color change.

**a:hover { ... }**

Defines styles for anchor elements when hovered over, changing background color and text color to create a hover effect.

**</style>**

Ends the block of embedded CSS styles.

**</head>**

Closes the head section of the HTML document.

**<body>**

Begins the body section of the HTML document.

**<h1>Welcome to the Student Prediction System!</h1>**

Displays a heading with the specified text.

**<ul>**

Begins an unordered list.

**<li><a href="/student_prediction">Student Prediction</a></li>**

Creates a list item with a hyperlink to "/student_prediction" and displays the text "Student Prediction."

**<li><a href="/teaching_evaluation">Teaching Evaluation</a></li>**

Creates another list item with a hyperlink to "/teaching_evaluation" and displays the text "Teaching Evaluation."

**<li><a href="/text_analysis">Feedback Analysis</a></li>**

Creates another list item with a hyperlink to "/text_analysis" and displays the text "Feedback Analysis."

**</ul>**

Closes the unordered list.

**</body>**

Ends the body section of the HTML document.

**</html>**

Closes the HTML document.

**Result.html**

**<!DOCTYPE html>**

Declares the document type and version of HTML being used, in this case, HTML5.

**<html lang="en">**

Defines the root element of the HTML document with the language attribute set to English.

**<head>**

Contains meta-information about the HTML document, such as character encoding, viewport settings, and title.

**<meta charset="UTF-8">**

Specifies the character encoding of the document as UTF-8, which supports a wide range of characters.

**<meta name="viewport" content="width=device-width, initial-scale=1.0">**

Sets the viewport properties for responsive web design, ensuring the page renders properly on various devices.

**<title>Result</title>**

Sets the title of the HTML document to "Result," which appears in the browser's title bar or tab.

**<style>**

Begins the block for embedding CSS styles directly within the HTML document.

**body { ... }**

Defines styles for the `body` element, including font family, background color, text color, margin, and padding.

**h1 { ... }**

Defines styles for the `h1` element, such as text color, alignment, and padding.

**p { ... }**

Defines styles for the `p` (paragraph) element, setting font size, alignment, and margin.

**</style>**

Ends the block of embedded CSS styles.

**</head>**

Closes the head section of the HTML document.

**<body>**

Begins the body section of the HTML document.

**<h1>Prediction Result</h1>**

Displays a heading with the specified text ("Prediction Result") in the center of the page.

**<p>The predicted SUMM value is: {{ summ }}</p>**

Displays a paragraph with dynamic content. The value of `{{ summ }}` is provided by the Flask application when rendering this template. It represents the predicted SUMM value obtained from the prediction process.

**</body>**

Ends the body section of the HTML document.

**</html>**

Closes the HTML document.

**Student_prediction.html**

**<!DOCTYPE html>**

This line declares the document type and version of HTML being used, which in this case is HTML5.

**<html lang="en">**

This line opens the HTML document and specifies the language of the document (English).

**<head>**

This line starts the header section of the HTML document, where metadata and links to external resources are typically placed.

**<meta charset="UTF-8">**

This line specifies the character encoding of the document, which is UTF-8. UTF-8 is widely used and supports various languages and characters.

**<meta name="viewport" content="width=device-width, initial-scale=1.0">**

This line sets the viewport properties for responsive design. It ensures that the width of the viewport is equal to the width of the device and sets the initial zoom level to 1.0.

**<title>Student Prediction Form</title>**

This line defines the title of the HTML document, which is displayed in the browser's title bar or tab.

**<style>**

This line starts the internal CSS styles section, where CSS rules for styling the HTML elements are defined.

```
body {
   font-family: Arial, sans-serif;
   background-color: #83a944;
color: #000;
}
```

These CSS rules define the styles for the `<body>` element. They set the font family to Arial or a sans-serif font, set the background color to a shade of green (#83a944), and set the text color to black (#000).

```
/* CSS styles for heading */
h1 {
   text-align: center;
color: #f4ffdc;
}
```

These CSS rules define the styles for the `<h1>` element. They center-align the text and set the color to a shade of off-white (#f4ffdc).

**/* CSS styles for form */This line starts a CSS comment that describes the purpose of the following CSS rules.**

```
form {
   width: 50%;
   margin: 0 auto;
   background-color: #f4ffdc;
   padding: 20px;
```

```
  border-radius: 10px;
}
```

These CSS rules define the styles for the `<form>` element. They set the width to 50% of its containing element, center-align it horizontally, set the background color to a shade of off-white (#f4ffdc), add padding, and apply a border-radius of 10px for rounded corners.

**/* CSS styles for form elements */This line starts a CSS comment that describes the purpose of the following CSS rules.**

```
label {
  display: block;
  margin-bottom: 10px;
}
```

These CSS rules define the styles for `<label>` elements. They set the display to block to ensure each label is on its own line and add a bottom margin of 10px to create spacing between labels.

```
select, input[type="text"], input[type="number"] {
  width: 100%;
  padding: 10px;
  margin-bottom: 20px;
  border: 1px solid #ccc;
  border-radius: 5px;
  box-sizing: border-box;
}
```

These CSS rules define the styles for `<select>`, `<input type="text">`, and `<input type="number">` elements. They set the width to 100% to fill the width of their container, add padding, a bottom margin for spacing, a 1px solid border, a border-radius of 5px for rounded corners, and box-sizing to include padding and border in the element's total width and height.

```
button {
  padding: 10px 20px;
  background-color: #83a944;
color: #fff;
  border: none;
  border-radius: 5px;
  cursor: pointer;
  font-size: 16px;
}
```

These CSS rules define the styles for `<button>` elements. They set the padding, background color to a shade of green (#83a944), text color to white (#fff), remove the border, apply a border-radius of 5px for rounded corners, change the cursor to a pointer on hover, and set the font size to 16px.

```
button:hover {
   background-color: #628436;
}
```

These CSS rules define the styles for `<button>` elements when hovered over. They change the background color to a darker shade of green (#628436).

```
</style>
```

This line closes the internal CSS styles section.

```
</head>
```

This line closes the header section of the HTML document.

```
<body>
```

This line starts the body section of the HTML document, where the visible content is placed.

```
<!-- Heading -->
<h1>Student Prediction Form</h1>
```

This line displays an `<h1>` heading with the text "Student Prediction Form".

```
<!-- Form for submitting data -->
<form action="/" method="POST">
```

This line starts a form element with the action attribute set to "/" (the root URL) and the method attribute set to "POST", indicating that the form data will be sent to the server using the POST method.

```
<!-- Input fields and labels -->
<label for="student_id">Student ID:</label>
<input type="text" id="student_id" name="student_id" required><br>
```

These lines display a label "Student ID:" with an input field for entering the student ID. The input field is of type "text", has an ID of "student_id", a name of "student_id", and is required.

```
<!-- Other input fields and labels -->
```

These lines represent additional input fields and labels for other student information. They are omitted here for brevity.

```
<!-- Submit button -->
<button type="submit">Submit</button>
```

This line displays a submit button for submitting the form data.

**</form>**
**</body>**
**</html>**

This line closes the form, body, and html section of the HTML documen

**Teaching_evaluation.html**
**<!DOCTYPE html>**

This line specifies the document type and version of HTML being used, which is HTML5 in this case.

**<html lang="en">**

This tag indicates the beginning of the HTML document and specifies the language attribute as English.

**<head>**

This section contains metadata about the HTML document, such as character set, viewport settings, and title.

**<meta charset="UTF-8">**

This meta tag specifies the character encoding used in the document, which is UTF-8 (Unicode Transformation Format 8-bit).

**<meta name="viewport" content="width=device-width, initial-scale=1.0">**

This meta tag sets the viewport properties for responsive design, ensuring that the width of the viewport is equal to the device width and the initial zoom level is set to 1.0.

**<title>Teaching Evaluation</title>**

This tag defines the title of the HTML document, which is displayed in the browser's title bar or tab.

**<style>**

This section contains embedded CSS styles for styling the HTML elements within the document.

```
body {
    font-family: Arial, sans-serif;
    background-color: #fff59d; /* Background color */
```

```
    color: #333; /* Text color */
    margin: 0;
    padding: 0;
}
```

This CSS rule sets the font family, background color, text color, margin, and padding for the body element.

```
h1 {
    color: #000; /* Heading color */
    text-align: center;
    padding: 20px 0;
}
```

This CSS rule sets the color, text alignment, and padding for the h1 element.

```
.image-container {
    text-align: center;
    margin-bottom: 20px;
}
```

This CSS rule defines the styles for elements with the class "image-container", setting text alignment to center and adding a margin at the bottom.

```
.image-container img {
    max-width: 50%;
    height: 50%;
}
```

This CSS rule sets the maximum width and height of images within elements with the class "image-container" to 50% of their parent container.

```
.caption {
    font-style: italic;
    font-size: 18px;
    margin-top: 5px;
}
```

This CSS rule defines the styles for elements with the class "caption", setting font style to italic, font size to 18px, and adding a margin at the top.

```
</style>
```

```
</head>
```

This closing tag marks the end of the styles section within the head of the document.

```
<body>
```

This section contains the main content of the HTML document.

**<h1>Teaching Evaluation Results</h1>**

This tag defines the main heading of the page, which displays the text "Teaching Evaluation Results".

**<div class="image-container">**

This tag creates a division (or container) with the class "image-container", which will hold an image and its caption.

**<img src="{{ url_for('static', filename='teaching_results/countplotodsumm.png') }}" alt="Count Plot of SUMM">**

This tag inserts an image with the specified source (URL), which is generated dynamically using Flask's `url_for` function to access a static file named "countplotodsumm.png". The alt attribute provides alternative text for the image.

**<p class="caption">Count Plot of SUMM: This plot illustrates the distribution of SUMM values among students.</p>**

This tag creates a paragraph element with the class "caption", which serves as a caption for the preceding image. It describes the content of the image.

**</div>**

This closing tag marks the end of the first image container.

**Text_analysis.html**

**<!DOCTYPE html>**
This line specifies the document type and version of HTML being used, which is HTML5 in this case.
**<html lang="en">**
This tag indicates the beginning of the HTML document and specifies the language attribute as English.
**<head>**
This section contains metadata about the HTML document, such as character set, viewport settings, and title.
**<meta charset="UTF-8">**
This meta tag specifies the character encoding used in the document, which is UTF-8 (Unicode Transformation Format 8-bit).
**<meta name="viewport" content="width=device-width, initial-scale=1.0">**

This meta tag sets the viewport properties for responsive design, ensuring that the width of the viewport is equal to the device width and the initial zoom level is set to 1.0.

```
<title>Text Analysis</title>
```

This tag defines the title of the HTML document, which is displayed in the browser's title bar or tab.

```
<style>
```

This section contains embedded CSS styles for styling the HTML elements within the document.

```
body {
    font-family: Arial, sans-serif;
    background-color: #fff59d; /* Background color */
    color: #333; /* Text color */
    margin: 0;
    padding: 0;
}
```

This CSS rule sets the font family, background color, text color, margin, and padding for the body element.

```
h1 {
    color: #000; /* Heading color */
    text-align: center;
    padding: 20px 0;
}
```

This CSS rule sets the color, text alignment, and padding for the h1 element.

```
.image-container {
    text-align: center;
    margin-bottom: 20px;
}
```

This CSS rule defines the styles for elements with the class "image-container", setting text alignment to center and adding a margin at the bottom.

```
.image-container img {
    max-width: 50%;
    height: 50%;
}
```

This CSS rule sets the maximum width and height of images within elements with the class "image-container" to 50% of their parent container.

```
.caption {
    font-style: italic;
    font-size: 18px;
    margin-top: 5px;
}
```

This CSS rule defines the styles for elements with the class "caption", setting font style to italic, font size to 18px, and adding a margin at the top.

**</style>**

**</head>**

This closing tag marks the end of the styles section within the head of the document.

**<body>**

This section contains the main content of the HTML document.

**<h1>Feedback Analysis Results</h1>**

This tag defines the main heading of the page, which displays the text "Feedback Analysis Results".

**<div class="image-container">**

This tag creates a division (or container) with the class "image-container", which will hold an image and its caption.

**<img src="{{url_for('static',filename='teaching_results/text_analysis_results/findingthecoorelationoftext.png') }}" alt="Finding the Coorelation of Text">**

This tag inserts an image with the specified source (URL), which is generated dynamically using Flask's `url_for` function to access a static file named "findingthecoorelationoftext.png". The alt attribute provides alternative text for the image.

**<p class="caption">Finding the Coorelation of Text: This image shows the analysis of text data to find correlations between different variables.</p>**

This tag creates a paragraph element with the class "caption", which serves as a caption for the preceding image. It describes the content of the image.

**</div>**

This closing tag marks the end of the first image container.

## _students_from__using_text.ipynb

**Importing Libraries:**

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from tqdm import tqdm
tqdm.pandas()
from lightgbm import LGBMRegressor
from sklearn import model_selection
from sklearn import metrics
```

```
import string
from nltk.corpus import stopwords
import nltk
from nltk.stem.snowball import SnowballStemmer
import gensim
from gensim.models.doc2vec import Doc2Vec, TaggedDocument
```

**`numpy`, `pandas`, `matplotlib`, `seaborn`**: Standard libraries for numerical operations, data manipulation, and visualization.

**`tqdm`**: A progress bar library for tracking loops.

**`LGBMRegressor`**: A regressor model from LightGBM library.

**`model_selection`, `metrics`**: Modules from scikit-learn for model selection and evaluation.

**`string`, `stopwords`, `nltk`, `SnowballStemmer`**: Natural language processing tools from NLTK for text preprocessing.

**`gensim`**: A library for topic modeling and document similarity.

**`Doc2Vec`, `TaggedDocument`**: Classes from Gensim for training Doc2Vec models.

**Loading Data**

```
train = pd.read_csv(r"C:\Users\mmeen\Downloads\student_learning_outcomes\train.csv")
testpd.read_csv(r"C:\Users\mmeen\Downloads\student_learning_outcomes\test.csv")
sample=pd.read_csv(r"C:\Users\mmeen\Downloads\student_learning_outcomes\sample_submission.csv")
```

`train`, `test`, and `sample` are DataFrames containing training data, test data, and a sample submission, respectively. They are read from CSV files located at specific paths.

**Exploratory Data Analysis (EDA)**

```
score_features = list(train.columns)
score_features.remove('text_id')
score_features.remove('full_text')
plt.rcParams["figure.figsize"] = (20, 3)
fig, ax = plt.subplots(1, 6)
count = 0
for f in score_features:
    ax[count].hist(list(train[f].values))
    ax[count].set_title(f)
    count += 1
plt.show()
```

**`score_features`**: Extracts the column names from the training data, excluding 'text_id' and 'full_text'.

**`plt.rcParams["figure.figsize"]`**: Sets the size of the figures for plotting.

**`fig, ax = plt.subplots(1, 6)`**: Creates subplots to visualize histograms of score features. Loops through each score feature, plots its histogram, and sets the title.

**`plt.show()`**: Displays the histograms.

**Essay Length Analysis**

```
def get_length_of_text(x):
   return len(x)
print(f'Averagelength{train.full_text.apply(lambdax:get_length_of_text(x)).mean():0.2f}')
print(f'Std length: {train.full_text.apply(lambda x: get_length_of_text(x)).std():0.2f}')
print(f'Min length: {train.full_text.apply(lambda x: get_length_of_text(x)).min():0.2f}')
print(f'Max length: {train.full_text.apply(lambda x: get_length_of_text(x)).max():0.2f}')
train.full_text.apply(lambda x: get_length_of_text(x)).hist();
```

Defines a function `get_length_of_text` to calculate the length of an essay.

Prints statistics (average, standard deviation, minimum, maximum) of essay lengths.
Plots a histogram of essay lengths.

**Checking for Missing Values (NaNs)**

```
train.isna().sum()
```

Checks for missing values in the training dataset using `.isna()` and `.sum()`.

**Checking Correlations**

```
colormap = sns.color_palette("Blues")
sns.heatmap(train.corr(), annot=True, cmap=colormap);
```

Generates a heatmap to visualize correlations between different features in the training data.

**Tokenization and Stemming**

```
stemmer = SnowballStemmer("english")
def tokenization_and_stemming(text):
   tokens = []
   pattern = r'''(?x)
      (?:[A-Z]\.)+
     | \w+(?:-\w+)*
     | \$?\d+(?:\.\d+)?%?
     | \.\.\.
     | [][.,;"'?():_`-]
   for word in nltk.regexp_tokenize(text, pattern):
      tokens.append(word.lower())
   filtered_tokens = []
   for token in tokens:
      if token.isalpha():
         filtered_tokens.append(token)
```

```
    stems = [stemmer.stem(t) for t in filtered_tokens]
    return stems
```

Initializes a SnowballStemmer for English language.

Defines a function `tokenization_and_stemming` for tokenizing and stemming text.

**`pattern`**: Regular expression pattern for tokenization.

Loops through tokens, converts them to lowercase, filters out non-alphabetic tokens, and performs stemming.

**Processing Essays**

```
processed_essays_train = []
processed_essays_test = []
for ess in list(train['full_text']):
    processed_essays_train.append(tokenization_and_stemming(ess))
for ess in list(test['full_text']):
    processed_essays_test.append(tokenization_and_stemming(ess))
processed_essays = []
processed_essays.extend(processed_essays_train)
processed_essays.extend(processed_essays_test)
```

Creates empty lists `processed_essays_train` and `processed_essays_test`.Iterates over essays in the training and test datasets, applies tokenization and stemming, and appends the processed essays to the respective lists.

Combines processed essays from both datasets into a single list `processed_essays`.

**Tagging Essays for Doc2Vec**

```
processed_essays_train_tagged=[TaggedDocument(processed_essays_train[i],list(train.iloc[i, 2:8])) for i in range(0, len(processed_essays_train))]
```

Tags each processed essay from the training set with its corresponding score categories.

Uses a list comprehension to create a list of TaggedDocument objects.

**StudentsEvaluation of Teaching .ipynb**

**Importing Libraries**

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier as dtc
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
```

**from sklearn.svm import SVC**
**from sklearn.metrics import confusion_matrix, classification_report**
**from sklearn.model_selection import cross_val_predict, train_test_split**
**from sklearn.feature_selection import mutual_info_classif**

Imports necessary libraries for data manipulation, visualization, clustering, and machine learning models.

**Loading Data and Initial Data Exploration**

**df=pd.read_csv(r"C:\Users\mmeen\Downloads\student_evaluation_of_learningoutcomes\student_prediction.csv")**

Loads the dataset from the specified CSV file path.

**df.head()**
**df.tail()**
**df.info()**
**df.shape**
**df.describe().T.style.background_gradient(cmap = "Oranges")**
**df["COURSE ID"].unique()**
**df.describe(include=object)**

Displays the first few rows, last few rows, information, shape, and descriptive statistics of the dataset.

Checks unique values in the 'COURSE ID' column and describes object-type columns.

**Data Preprocessing**

**df = df.drop('STUDENTID', axis=1)**
**duplicate = df[df.duplicated()]**

Drops the 'STUDENTID' column as it seems to be a unique identifier for each student.

Checks for duplicate rows in the dataset.

**Data Visualization**

**sns.countplot(df['SUMM'],label="Count")**
**plt.show()**

Plots a count of the 'SUMM' column using Seaborn's countplot.

**Feature Selection and Mutual Information Calculation**

**X = df.drop('SUMM', axis=1)**
**y = df['SUMM']**
**discrete_features = X.dtypes == int**
**mi_scores = make_mi_scores(X, y, discrete_features)**

Separates the features (X) and target variable (y).
Calculates Mutual Information (MI) scores for feature selection.

**Data Visualization**

```
palette = sns.color_palette('Set2')
data = df['GENDER'].value_counts(normalize=True) * 100
fig = plt.figure(figsize=(14, 5), constrained_layout=True)
plt.subplot(121)
plt.pie(data, labels=data.index, startangle=90, counterclock=False, autopct='%1.1f%%',
colors=palette)
data = df.groupby('GENDER')['SUMM'].mean().sort_values(ascending=False)
plt.subplot(122)
bp = sns.barplot(x=data.index, y=data, palette=palette)
annotate_bar_plot(barplot=bp)
plt.xlabel('')
plt.ylabel('Average Aumm')
plt.yticks([])
plt.box(False)
fig.suptitle('GENDER', fontsize=18)
plt.show()
```

Plots a pie chart of gender distribution (`'GENDER'` column) and a bar plot of average 'SUMM' score by gender.
Uses Seaborn's `barplot` and a custom function `annotate_bar_plot` to annotate the bar plot.

**Further Data Visualization**

```
fig, axarr  = plt.subplots(2,2,figsize=(10,10))
sns.barplot(x='SUMM', y='COURSE ID', data=df, ax=axarr[0,0])
sns.barplot(x='SUMM', y='EXP_GPA', data=df, ax=axarr[0,1])
sns.barplot(x='SUMM', y='CUML_GPA', data=df, ax=axarr[1,1])
sns.barplot(x='SUMM', y='PREP_EXAM', data=df, ax=axarr[1,0])
```

Creates subplots to visualize the relationship between 'SUMM' score and other variables ('COURSE ID', 'EXP_GPA', 'CUML_GPA', 'PREP_EXAM') using bar plots.

**Feature Selection**

```
def drop_uninformative(df, mi_scores):
   return df.loc[:, mi_scores > 0]
X = drop_uninformative(X, mi_scores)
```

Defines a function `drop_uninformative` to drop features with MI scores less than or equal to 0.

Applies the function to drop uninformative features from the feature matrix `X`.

**Modeling and Evaluation**

```
kmeans = KMeans(n_clusters=8, random_state=0)
X["Cluster"] = kmeans.fit_predict(X)
decision_tree = dtc(random_state=0)
decision_tree.fit(X, y)
predict = cross_val_predict(estimator=decision_tree, X=X, y=y, cv=5)
print("Classification Report: \n", classification_report(y, predict))
```

# Similar steps for other classifiers: RandomForestClassifier, KNeighborsClassifier, GaussianNB, SVC

Fits a KMeans clustering model to the feature matrix `X` and appends cluster labels to `X`.

Trains a Decision Tree Classifier (`DecisionTreeClassifier`) on the modified feature matrix `X` and evaluates its performance using cross-validation.

Similar steps are followed for other classifiers such as RandomForestClassifier, KNeighborsClassifier, GaussianNB, and SVC.

## 4.3 Output Screenshots with Explanation and Interpretation

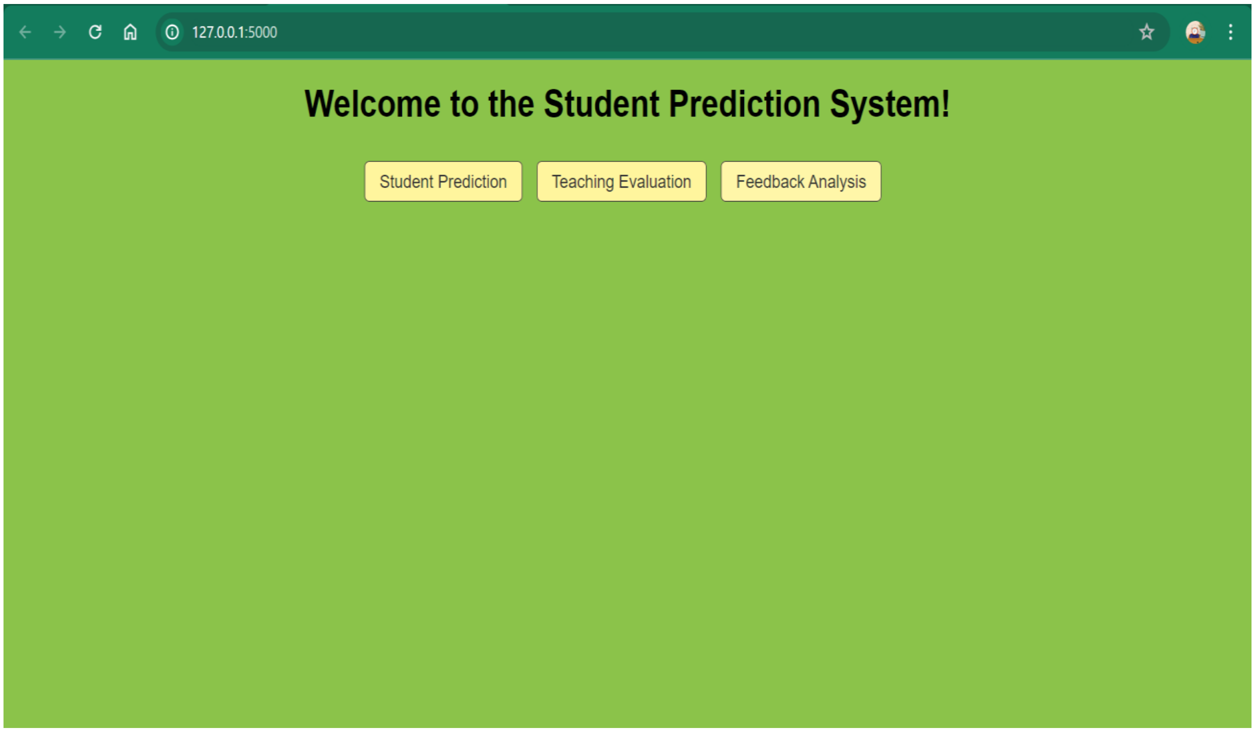



**Figure 4.1:** Welcome to the student prediction system

This image appears to be a web-based interface or dashboard for a student prediction system. The system seems to have three main components:

**Student Prediction**: This likely allows users to input student data and get predictions or forecasts about the students' performance, progress, or outcomes.

**Teaching Evaluation**: This component seems to be related to evaluating the teaching quality or effectiveness, perhaps as it relates to the students' performance.

**Feedback Analysis**: This section appears to be focused on analyzing feedback, likely from students, teachers, or other stakeholders, to provide insights and improvement opportunities.

A comprehensive platform for tracking, predicting, and analyzing various aspects of student performance and the teaching/learning environment. The green background and simple layout suggest a clean, user-friendly interface for educators, administrators, or students to access and utilize the system's features.

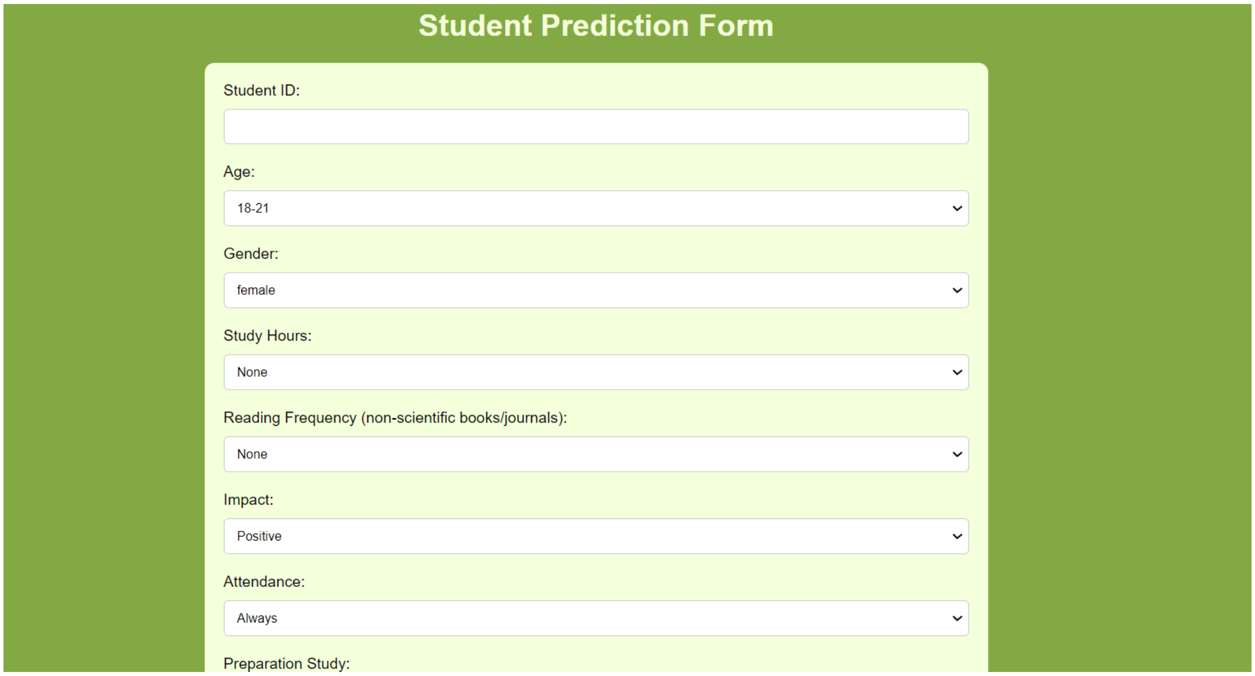



**Figure 4.2**: Student Prediction Form

This image shows a web-based form that allows users to input various student-related data to be used for predicting student outcomes or performance. The data can be extracted from student_prediction.csv if you fill the data, it predicts the SUMM according to the closest values present in the dataset. The form includes the following fields:

**Student ID**: This field likely allows the user to input a unique identifier for the student being evaluated.

**Age**: A dropdown menu that allows the user to select the student's age.

**Gender:** A dropdown menu that allows the user to select the student's gender.

**Study Hours**: A dropdown menu that allows the user to select the number of study hours for the student.

**Reading Frequency (non-scientific books/journals):** A dropdown menu that allows the user to select the student's reading frequency for non-academic materials.

**Impact:** A dropdown menu that allows the user to select the perceived impact or influence of the student's activities.

**Attendance:** A dropdown menu that allows the user to select the student's attendance record.

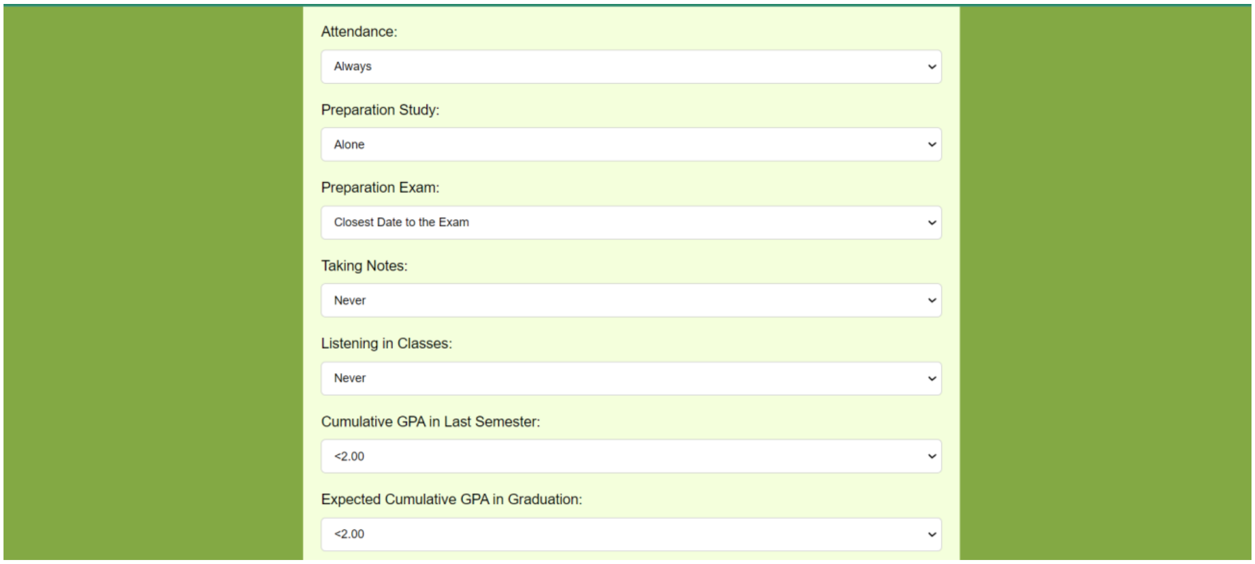



**Figure 4.3**: Student Prediction Form1

This image displays a series of additional data fields that can be filled out as part of the student prediction process. These fields include:

**Preparation Study**: A dropdown menu that allows the user to select the student's study preparation habits, with an option like "Alone".

**Preparation Exam**: A dropdown menu that allows the user to select the closest date to the student's exam.

**Taking Notes**: A dropdown menu that allows the user to select the student's note-taking habits, with an option like "Never".

**Listening in Classes**: A dropdown menu that allows the user to select the student's level of engagement in class, with an option like "Never".

**Cumulative GPA in Last Semester**: A dropdown menu that allows the user to select the student's cumulative GPA from the previous semester, with an option like "<2.00".

**Expected Cumulative GPA in Graduation**: A dropdown menu that allows the user to select the student's expected cumulative GPA upon graduation, with an option like "<2.00".

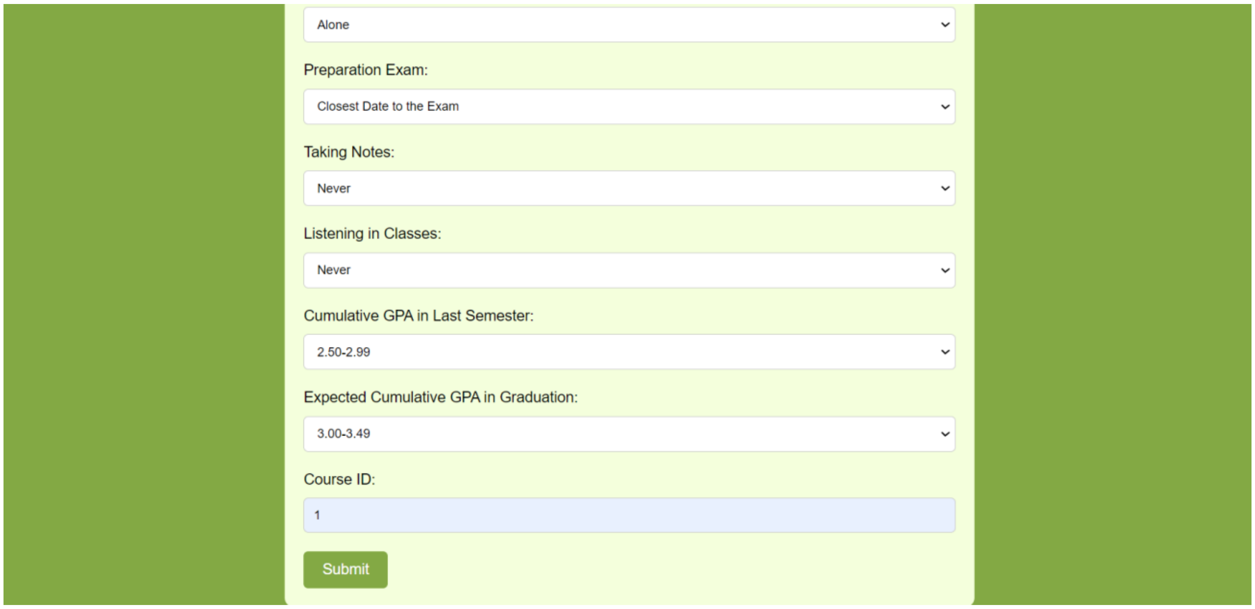



**Figure 4.4**: Student Prediction Form2

**Course ID:** A text field where the user can enter a course identifier.

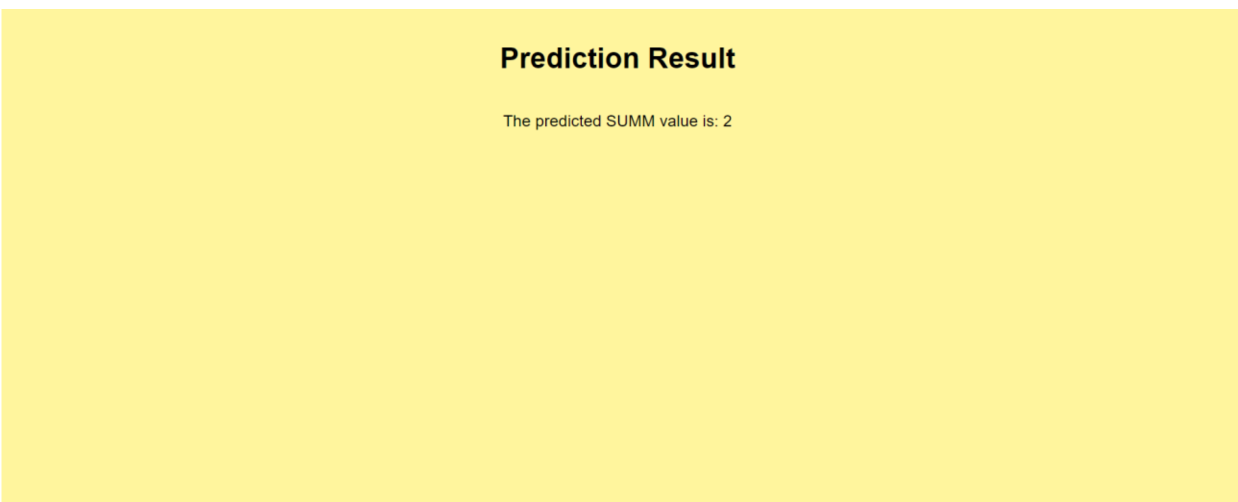



**Figure 4.5**: Prediction Result

The result displayed is "2" (if SUMM=0 FAIL, SUMM=1 DD, SUMM=2 DC, SUMM=3 CC, SUMM=4 CB, SUMM=5 BB, SUMM=6 BA, SUMM=7 AA)

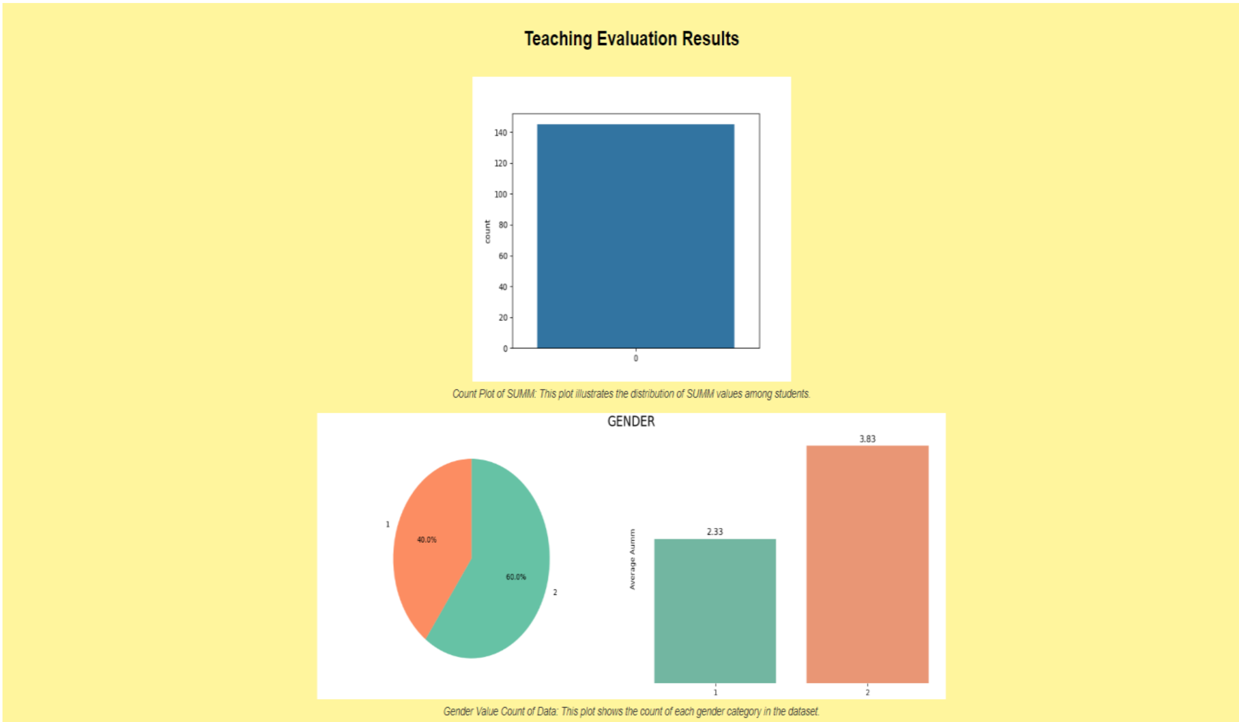


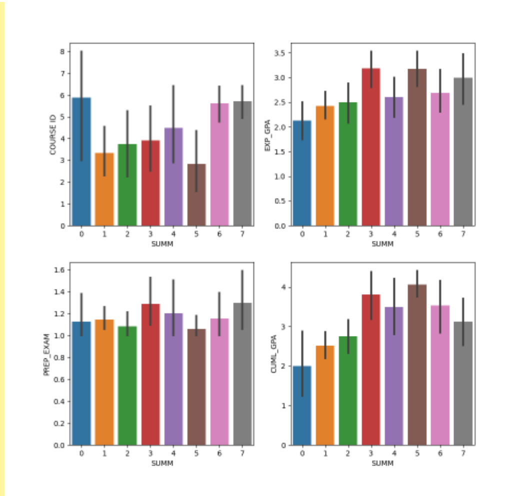

**Figure 4.6**: Teaching Evaluation Results

This image displays the outcomes of a teaching evaluation, which includes two major components:

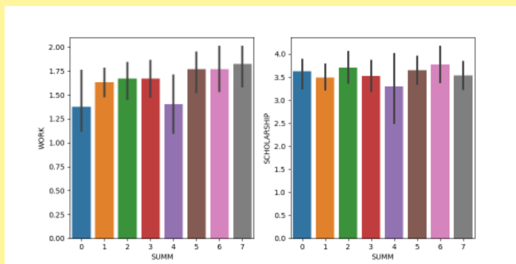
**SUMM Evaluation**: The bar graph depicts the distribution of SUMM (Student Learning Assessment Measure) values across students. SUMM is a metric for assessing student learning, and the graph depicts the number or frequency of various SUMM values.

**Gender Distribution**: The pie chart depicts the students' gender distribution, with 60.6% female and 39.4% male. The bar graph below the pie chart shows the SUMM values for each gender, with females at 2.33 and males at 3.43.

This image provides insights into the teaching evaluation results, such as the overall SLAM distribution and the differences in SLAM values between male and female students. The data can be used to evaluate the effectiveness of teaching methods and identify potential gender disparities in student performance.




Plot Representing Course, EXP_GPA, CUML_GPA, Prep_Exam: This plot represents the relationship between course, expected GPA, cumulative GPA, and preparation for the exam.




Plot Representing Work, Scholarship: This plot illustrates the correlation between work and scholarship factors.

**Figure 4.7**: Teaching Evaluation Results1

This image includes several plots depicting the relationship between various academic factors and student performance. Here is a detailed description of the information provided:

**Plot for Course, GPA, CGPA, GRA, and Prep_Exam**: This set of plots depicts the relationship between a student's course, expected GPA, cumulative GPA, GRA (Graduation Readiness Assessment), and exam preparation. The x-axis represents the course, and the y-axis displays the values for each factor. These plots show how the various academic measures are correlated across courses.

**Plot Representing Work and Scholarship**: This series of plots depicts the relationship between a student's work and scholarship factors. The x-axis represents the course, while the y-axis depicts the values for work and scholarship. These graphs can help identify any connections between a student's work responsibilities and their scholarship or financial aid. Overall, this image depicts a comprehensive data visualization of various academic variables and their relationships. The plots show how course selection, GPA, exam preparation, work, and scholarship affect student performance and outcomes. This information can help you understand the academic landscape and identify areas for improvement or support.

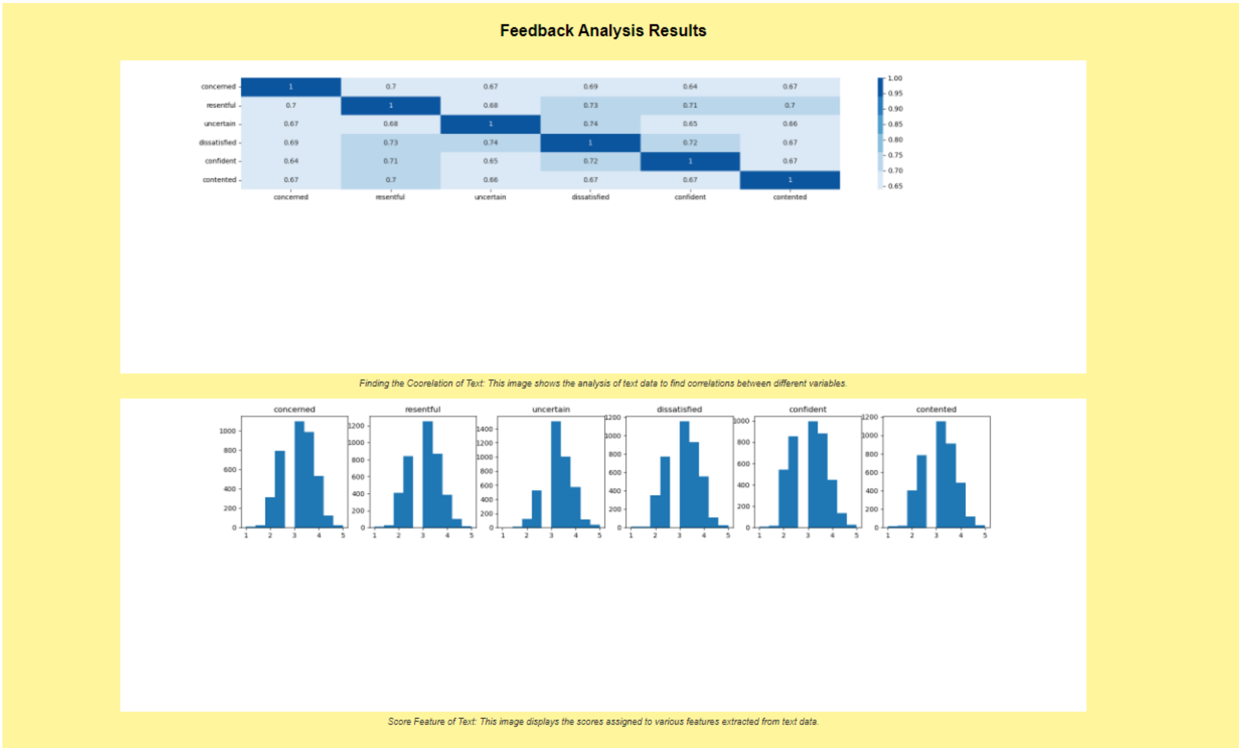



**Figure 4.8**: Feedback Analysis Results

This image depicts the results of a feedback analysis, specifically the correlations between various variables derived from the feedback data.

The top section of the image shows a heatmap that depicts the correlation coefficients between various feedback-related variables, such as concerned, respectful, and uncertain. The color scale shows the strength and direction of the correlations, with blue representing positive correlations and red representing negative correlations.

The bottom half of the image contains bar charts for each of the feedback-related variables, which provide a more detailed view of the distribution of values for each variable. These charts provide a better understanding of the underlying data and the relationships between the various factors.

Overall, this image provides a thorough analysis of the feedback data, enabling the identification of patterns, relationships, and potential insights that can be used to improve the feedback process or address any issues raised by the analysis.

# CHAPTER 5

# TESTING

## 5.1 Test Cases

**Boundary Testing:**

Test the system with the minimum and maximum values for each input field to ensure it handles extreme values correctly. For example, age ranging from 18 to 99 years, study hours ranging from 0 to 24 hours, GPA ranging from 0.0 to 4.0, etc.

**Equivalence Partitioning:**

Divide the input ranges into equivalence classes and select representative values from each class for testing. For example, select ages 20, 30, and 40 to represent different age groups, select study hours of 5, 10, and 15 to represent different study durations, etc.

**Edge Cases:**

Test the system with edge cases such as empty input fields, invalid input types (e.g., entering text in a numeric field), or submitting the form without selecting any options. Ensure that the system provides appropriate error messages or handles these cases gracefully.

**Combination Testing:**

Test various combinations of input values to ensure that the system produces correct predictions for different scenarios. For example, test combinations of different age groups, genders, study hours, GPA levels, etc.

**Regression Testing:**

After making any changes or updates to the system, rerun previous test cases to ensure that existing functionality has not been affected. Verify that the system still produces accurate predictions after any modifications.

**Load Testing:**

Simulate many simultaneous users submitting prediction requests to evaluate the system's performance under heavy load. Monitor response times, resource utilization, and any potential bottlenecks to ensure the system can handle expected levels of traffic.

**User Acceptance Testing (UAT):**

Involve real users, such as students or educators, to test the system's usability, functionality, and accuracy in a real-world environment. Gather feedback from users to identify any issues or areas for improvement.

By performing these test cases, you can ensure that your student prediction system functions

## 5.2 Validation Testing

Validation testing ensures that the student prediction system accurately predicts outcomes based on the provided input data. Here are some validation test cases:

**Age Validation:**

Test the system with valid age values within the specified range (e.g., 18-99 years).

Test with invalid age values such as negative numbers, decimals, or non-numeric characters to ensure proper error handling.

**Gender Validation:**

Test the system with valid gender options (e.g., male, female).

Test with invalid gender options or missing gender selection to verify error handling.

**Study Hours Validation:**

Test the system with valid study hours ranging from 0 to 24 hours.

Test with invalid study hour values, such as negative numbers, decimals, or non-numeric characters.

**Reading Frequency Validation:**

Test the system with valid options for reading frequency (e.g., none, sometimes, often).

Test with invalid or missing reading frequency selections to ensure proper error handling.

**Impact Validation:**

Test the system with valid impact options (e.g., positive, negative, neutral).

Test with invalid or missing impact selections to verify error handling.

**Attendance Validation:**

Test the system with valid attendance options (e.g., always, sometimes, never).

Test with invalid or missing attendance selections to ensure proper error handling.

**Preparation Study and Exam Validation:**

Test the system with valid options for preparation study and exam (e.g., alone, with friends, closest date to exam).

Test with invalid or missing selections to verify error handling.

**Notes and Listening Validation:**

Test the system with valid options for taking notes and listening in classes.

Test with invalid or missing selections to ensure proper error handling.

**Cumulative GPA Validation:**

Test the system with valid cumulative GPA values within the specified range (e.g., 0.0-4.0).

Test with invalid GPA values, such as negative numbers, values exceeding the range, or non-numeric characters.

**Expected Cumulative GPA Validation:**

Test the system with valid expected cumulative GPA values within the specified range (e.g., 0.0-4.0).

Test with invalid GPA values, such as negative numbers, values exceeding the range, or non-numeric characters.

**Form Submission Validation:**

Test the system by submitting the form with all valid input values to ensure accurate prediction results.

Test with incomplete or incorrect input data to verify error handling and validation messages.

By performing these validation test cases, you can ensure that the student prediction system accurately processes input data and produces reliable prediction outcomes.

## 5.3 Unit Testing

Unit testing involves testing individual units or components of your code in isolation to ensure they perform as expected. In the context of your provided project code, unit testing can be performed on the individual functions and methods. Let's break down the unit testing process for project:

**Unit Testing of `predict_summ` Function:**

Test the `predict_summ` function with different sets of input data to verify that it correctly filters the dataset and predicts the SUMM value.

Test with valid input data to ensure the function returns the expected SUMM value.

Test with invalid input data to verify that appropriate error handling mechanisms are triggered and error messages are returned.

**Unit Testing of `index` Function:**

Test the `index` function, which serves as the main entry point for handling HTTP requests.

Write test cases to simulate both GET and POST requests and ensure that the function behaves correctly in response to each type of request.

For GET requests, verify that the form template (`index.html`) is rendered properly.

For POST requests, test with different sets of form data to validate that the `predict_summ` function is called correctly and that the result is displayed in the template (`result.html`).

**Error Handling:**

Write test cases to cover error scenarios, such as missing or invalid input data.

Verify that the functions handle these error conditions gracefully and return appropriate error messages or responses.

**Test Coverage:**

Aim for high test coverage to ensure that the unit tests exercise most of the code paths within the functions being tested.

Monitor test coverage metrics to identify areas of the code that may require additional testing.

## 5.4 System Testing

System testing involves testing the entire integrated system as a whole to evaluate its compliance with specified requirements and assess its overall functionality. In the context of your project, system testing can be performed to ensure that all components work together as expected and meet the desired objectives. Here is how you can approach system testing for project:

**Functional Testing:**

Test the functionality of the entire system by interacting with the user interface and verifying that it behaves as expected. Test each feature of the system to ensure that it meets the functional requirements specified for the project. Perform end-to-end testing of the user journey, from submitting the prediction form to receiving the result, to ensure seamless functionality.

**Integration Testing:**

Test the integration between different components of the system, such as the Flask web application, the `predict_summ` function, and the HTML templates. Verify that data flows correctly between the frontend and backend components. Ensure that the frontend accurately reflects the predictions generated by the backend.

**User Acceptance Testing (UAT):**

Conduct UAT with stakeholders or end users to validate that the system meets their requirements and expectations. Have users interact with the application and provide feedback on its usability, performance, and overall satisfaction. Address any issues or concerns raised during UAT to improve the quality of the system.

**Performance Testing:**

Evaluate the performance of the system under various load conditions to ensure that it can handle expected levels of traffic.

Test the response time of the application for different types of requests and user interactions. Identify any bottlenecks or performance issues and optimize the system as needed.

**Security Testing:**

Conduct security testing to identify vulnerabilities and ensure that sensitive data is protected. Test for common security threats such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF). Implement security best practices to mitigate risks and protect the integrity of the system and user data.

**Usability Testing:**

Assess the usability of the system by evaluating how easily users can accomplish tasks and navigate through the application. Gather feedback from users on the clarity of instructions, intuitiveness of the interface, and overall user experience. Make improvements based on usability testing to enhance the accessibility and user-friendliness of the system.

By performing comprehensive system testing, you can ensure that your project meets the requirements and expectations of stakeholders, delivers a high-quality user experience, and operates reliably in production environments.

# CHAPTER 6

# CONCLUSION

The project is the result of extensive research, meticulous planning, and diligent implementation efforts to create a powerful educational data analysis system. The system is based on the core goal of leveraging data-driven insights to improve educational outcomes, and it represents an integration of innovative technologies, methodological approaches, and user-centered design principles. At its core, the project encapsulates the complexities of system architecture, functionality, and design, giving stakeholders a complete picture of the project's scope and objectives. From using Flask as the underlying web framework to incorporating machine learning algorithms for student prediction, teaching evaluation, and feedback analysis, each component is meticulously detailed to convey its role and significance within the system ecosystem. Furthermore, the project explains the user interface design considerations, focusing on usability, accessibility, and visual appeal. The interface is designed using HTML templates and CSS styling to facilitate intuitive navigation and seamless interaction, resulting in a positive user experience across diverse user demographics and preferences. The project goes beyond technical specifications to discuss the system's broader implications and potential impact on educational practices and outcomes. By providing educators, administrators, and stakeholders with actionable insights derived from data analytics and machine learning, the system enables evidence-based decision-making, proactive intervention strategies, and continuous improvement initiatives. In essence, the project demonstrates the collaborative efforts, innovation, and dedication put into realizing the vision of an advanced educational data analysis system. As a comprehensive reference guide, it provides stakeholders with the knowledge and insights they need to fully leverage the system's potential and drive positive educational transformation.

# CHAPTER 7

# FUTURE SCOPE OF THE PROJECT

This project's future scope is full of opportunities for growth, refinement, and innovation. As technology advances and educational paradigms shift, the system has several avenues for improving its capabilities and impact. The integration of additional data sources and analysis techniques is a significant opportunity for future development. By incorporating various data modalities such as student demographics, learning preferences, and socioeconomic factors, the system can provide a more comprehensive understanding of student behaviour and performance. Advanced analytical methods, such as natural language processing (NLP) for text analysis and deep learning for predictive modeling, can help the system generate deeper and more accurate insights. Furthermore, there is tremendous potential for improving the system's scalability and interoperability. By leveraging cloud computing resources and implementing a microservices architecture, the system can easily accommodate growing user bases and handle larger volumes of data. Seamless integration with existing educational platforms and systems, such as learning management systems (LMS) and student information systems (SIS), can improve data exchange and interoperability, resulting in a more cohesive and integrated educational environment. Another promising direction for future development is to improve user experience and engagement. User feedback mechanisms, usability studies, and iterative design processes can help the system evolve to better meet the needs and preferences of its diverse user base. Personalization features, adaptive interfaces, and interactive visualizations can tailor the user experience to specific preferences and learning styles, resulting in increased engagement and usability.

Furthermore, the project can investigate opportunities for community engagement and collaboration. By forming partnerships with educational institutions, research organizations, and industry stakeholders, the system can pool its expertise and resources to address pressing educational challenges. Collaborative research initiatives, data-sharing agreements, and joint development efforts can generate new insights, drive innovation, and increase the system's impact on educational outcomes.