COURSE REVIEW USING SENTIMENT ANALYSIS

BY

KAWARA DAVIE NIGEL

(H210507Q)

HIT 400: Capstone design project Submitted in Partial Fulfillment of the

Requirements of the degree of Bachelor of Technology

In Software Engineering

School of Information Sciences and Technology

Harare Institute of Technology

Zimbabwe

SUPERVISOR

Ms Chibaya

HIT 400 /200 Project Documentation Marking Guide

ITEM	TOTAL MARK /%	ACQUIRED/%
PRESENTATION- Format-Times Roman 12 for ordinary text, Main headings Times Roman 14, spacing 1.5.Chapters and sub-chapters, tables and diagrams should be numbered. Document should be in report form. Range of document pages. Between 50 and 100.Work should be clear and neat	**5**	
Pre-Chapter Section Abstract, Preface, Acknowledgements, Dedication & Declaration	**5**	
Chapter One-Introduction Background, Problem Statement, Objectives – smart, clearly measurable from your system. Always start with a TO… Hypothesis, Justification, Proposed Tools Feasibility study: Technical, Economic & Operational Project plan –Time plan, Gantt chart	**10**	
Chapter Two-Literature Review Introduction, Related work & Conclusion	**10**	
Chapter Three –Analysis Information Gathering Tools, Description of system Data analysis –Using UML context diagrams ,DFD of existing system Evaluation of Alternatives Systems, Functional Analysis of Proposed System-Functional and Non-functional Requirements ,User Case Diagrams	**15**	
Chapter Four –Design Systems Diagrams –Using UML Context diagrams, DFD ,Activity diagrams Architectural Design-hardware, networking Database Design –ER diagrams ,Normalized Databases Program Design-Class diagrams, Sequence diagrams ,Package diagrams, Pseudo code Interface Design-Screenshots of user interface	**20**	
Chapter Five-Implementation & Testing Pseudo code of major modules /Sample of real code can be written here Software Testing-Unit, Module ,Integration ,System ,Database & Acceptance	**20**	
Chapter Six –Conclusions and Recommendations Results and summary ,Recommendations & Future Works	**10**	
Bibliography –Proper numbering should be used Appendices –templates of data collection tools, user manual of the working system, sample code, research papers	**5**	
	100	**100**

Certificate of Declaration

This is to certify that work entitled "Course Review using Sentiment Analysis" is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology (Hons) Degree in Software Engineering, Harare Institute of Technology. It is further certified that no part of research has been submitted to any university for the award of any other degree.

(Supervisor) Signature…….

Date…….

(Mentor) Signature…….

Date……

(Chairman) Signature…….

Date…….

Abstract

The Project aims to analyze views or opinions expressed in textual data from students' feedback after completing a course and provide relevant suggestions for improvement through automatically identifying the emotional tone expressed in the textual data which can either be positive, negative or neutral. The system will use machine learning and will be trained on a dataset of labelled text to analyze new or unseen text to determine its view or opinion and provide recommendations based on the insight gained.

Preface

This document acts as a tool that provides the overview of the Course **Review using Sentiment Analysis System**. It describes the steps and processes that were chosen and implemented to come up with the system. The intended audience of the document is the Harare Institute of Technology. The cases described in the document and the research conducted in the document helps to describe the problem being solved by the system developed and implementation procedures implemented in this document.

Acknowledgements

This project is the result of study in 'Sentiment Analysis' which was completed in the final year at the Department of Software Engineering at the Harare Institute of Technology. Firstly, I would like to give thanks to God Almighty for giving me the chance to complete this program and making this project fruitful. I want to extend my thanks to my family for providing with every necessity, love and support. I deeply appreciate all my colleagues whom I worked with tirelessly during the course of my degree program. A very big thank you to Ms. Chibaya who took on the supervision task during the course of my project and made it a success. Special thanks to Mr. Makondo and the entire Harare Institute of Technology ICT department staff for all the support and knowledge they offered me. I wish to thank those who responded to my questionnaire and those who participated in the interviews, as well as those whom I did not mention by name who made endless contributions for this project to sail. May God bless you all.

Contents

Tables of Figures

List of Tables

Chapter 1

1.0 Introduction

The Course Review Project aims to analyze views or opinions expressed in textual data from students' feedback after completing a course and provide relevant suggestions for improvement through automatically identifying the emotional tone expressed in the textual data which can either be positive, negative or neutral.

1.1 Background

The rapid adoption of online and blended learning environments has transformed the educational landscape, creating new opportunities for personalized and data-driven teaching strategies. In this context, gathering and analyzing student feedback has become a critical component of improving course quality and ensuring student satisfaction. Traditional methods of collecting and interpreting feedback, such as surveys and manual review, are often time-consuming, subjective, and limited in scope.

The advent of sentiment analysis, a subset of natural language processing (NLP), offers a powerful solution for automating the analysis of textual feedback. By leveraging machine learning and NLP techniques, sentiment analysis can extract emotional tones from student feedback, providing educators with actionable insights into student engagement, satisfaction, and areas of improvement. However, most existing sentiment analysis systems are designed for business or general use cases and are not tailored to the unique needs of the educational sector.

This project aims to bridge this gap by developing a **Sentiment Analysis System for Education** that enables institutions to analyze course reviews, discussion forums, and other textual interactions. This system will empower educators to identify trends in student sentiment, make data-driven decisions, and enhance the learning experience.

1.2 Problem Statement

Educational institutions face significant challenges in effectively analyzing and acting upon student feedback due to the following reasons:

1. **Volume of Feedback:** The large volume of textual feedback collected from course reviews, assignments, and discussion forums makes manual analysis impractical.
2. **Unstructured Data**: Student feedback is often unstructured, making it difficult to extract meaningful insights using traditional methods.
3. **Time Constraints**: Manual evaluation of feedback is time-intensive, leading to delays in identifying and addressing student concerns.
4. **Lack of Contextual Insights**: Generic sentiment analysis tools do not cater to the specific context of education, leading to inaccurate or irrelevant results.
5. **Missed Opportunities for Improvement**: Without timely and accurate analysis of feedback, institutions miss opportunities to address issues, improve course quality, and foster a positive learning environment.

This project addresses these challenges by developing a specialized sentiment analysis system for education. The system will enable the automated classification of student feedback into positive, negative, or neutral sentiments and provide actionable insights for educators. By leveraging this tool, educational institutions can improve course design, enhance teaching strategies, and support student well-being effectively.

1.3 Aims and Objectives

1.To classify student feedback into positive, negative, or neutral sentiments using advanced natural language processing techniques.
2.To provide real-time sentiment analysis results for timely interventions and decision-making.
3.To generate detailed reports and visualizations that highlight trends in student engagement and satisfaction.

4.To enhance teaching methods and course design based on insights derived from student sentiments.

1.4 Hypothesis

The proposed Sentiment Analysis System for Education will enhance the learning experience and improve educational outcomes by providing real-time insights into student feedback and emotional responses. By automating the classification and analysis of textual data, the system will empower educators to identify patterns in student sentiment, address issues promptly, and tailor teaching strategies to meet the emotional and academic needs of students.

This hypothesis rests on the following assumptions:

1. Automated Sentiment Analysis can effectively classify student feedback with high accuracy.
2. Insights derived from sentiment analysis will enable timely and informed decision-making by educators.
3. Visualized sentiment trends and reports will help educators and administrators identify areas of strength and improvement in course delivery.
4. Identifying patterns of disengagement or frustration will lead to interventions that improve student engagement and satisfaction.
5. Integration with existing educational platforms will streamline the feedback process and foster data-driven educational practices.

1.5 Justification

The development of a **Sentiment Analysis System for Education** is essential for addressing critical challenges faced by educational institutions in understanding and acting upon student feedback. The justification for this system includes:

1. **Scalability of Feedback Analysis**:
    - Manually analyzing large volumes of student feedback is time-consuming and impractical. An automated system enables scalability, allowing institutions to process feedback from multiple courses and students efficiently.

2. **Improved Decision-Making**:
    - By providing real-time sentiment insights, the system equips educators and administrators with actionable data to make timely interventions and improve course quality.
3. **Enhanced Student Engagement**:
    - The ability to detect patterns of disengagement or emotional distress empowers educators to support students more effectively, fostering a positive and inclusive learning environment.
4. **Data-Driven Education**:
    - The system supports evidence-based decision-making by generating detailed reports and visualizations that highlight key trends in student satisfaction and engagement.
5. **Tailored Teaching Strategies**:
    - Insights from sentiment analysis enable educators to adapt their teaching methods and materials to better align with students' emotional and academic needs.
6. **Integration and Efficiency**:
    - Seamless integration with existing educational platforms (e.g., LMS) enhances the efficiency of feedback collection and analysis, minimizing disruptions to current workflows.
7. **Support for Mental Health and Well-being**:
    - Early identification of emotional distress allows institutions to provide timely support, contributing to student mental health and overall well-being.
8. **Addressing Educational Gaps**:
    - By identifying specific areas where students struggle, the system helps bridge gaps in course design, ensuring a more effective and equitable learning experience.

The proposed system aligns with the growing need for innovative, technology-driven approaches in education, making it a valuable tool for enhancing teaching and learning outcomes in modern educational environments.

1.6 Proposed Tools

- ### Programming Language

Python for implementing the system due to its extensive libraries for data processing, machine learning, and natural language processing.

- ### Data Collection Tools

Web Scraping Libraries

Scrapy: For automating data scraping tasks from educational websites or LMS platforms.

APIs

REST APIs provided by learning platforms (e.g Moodle) to collect structured student feedback

- ### Data Preprocessing Tools

Natural Language Processing Libraries

NLTK: For tokenization, stemming, lemmatization, and stop-word removal.

SpaCy: For advanced text preprocessing and part-of-speech tagging.

- ### Sentiment Analysis Tools

scikit-learn: For implementing algorithms like Logistic Regression, SVM, and Naive Bayes.

TF-IDF and Count Vectorizer: Built-in tools in scikit-learn for feature extraction.

TensorFlow/Keras: For implementing sentiment analysis models with neural networks.

- ### Database Tools

MySQL: For structured data storage and retrieval.

MongoDB: For handling unstructured data such as raw student feedback.

- **Visualization and Reporting Tools**

Matplotlib: For generating detailed graphs and charts of sentiment trends.

Dash: For creating interactive visualizations and dashboards.

Tableau: For advanced business intelligence and reporting (optional).

- **Development and Deployment Tools**

Integrated Development Environment (IDE**)**

Jupyter Notebook: For developing and testing sentiment analysis models.

VS Code: For code development and debugging.

Version Control

Git and **GitHub**: For version control and collaboration.

- **User Interface and Integration Tools**

Flask/Django: For building a web-based interface to input student feedback and display sentiment analysis results.

- **Testing and Optimization Tool**

pytest: For unit testing the system.

- **Deployment Tool**

Heroku: For deploying the system as a web application.

1.7 Feasibility Study

Feasibility study asses whether the project is worth developing or not. Anticipated costs are weighed against projected income and the availability of funds, skills and other resources to develop and support the system. It also helps in identification of potential and actual risks associated with the project. Technical, operational and economic feasibility studies shall be conducted.

1.7.1 Technical feasibility

Availability of Tools and Technology

Tools such as Python and scikit-learn are open-source, reducing dependency on costly proprietary software.

Pretrained NLP models (e.g., BERT) allow for efficient implementation without the need for extensive training data.

Cloud platforms like AWS, Azure, or Google Cloud offer scalable infrastructure for processing large datasets and deploying the system.

Integration with Existing Systems

APIs enable seamless integration with Learning Management Systems (e.g., Moodle, Google Classroom) for automated data collection.

Interoperability with existing reporting and analytics tools ensures smooth adoption.

Risks and Mitigation

Risk: Performance issues with real-time sentiment analysis.

Mitigation: Use scalable cloud infrastructure and optimize models for faster processing.

Risk: Insufficient data for training models.

Mitigation: Leverage transfer learning with pretrained models to reduce dataset requirements.

1.7.2 Economic feasibility

Cost of Development

Low-Cost Tools: The system relies on open-source libraries and frameworks, reducing software costs.

Cloud Services: Pay-as-you-go cloud services minimize upfront infrastructure costs.

Minimal Hardware Requirements: Development can be performed using standard computing resources, with optional cloud services for scalability.

Cost of Deployment

Cloud platforms such as AWS or Azure enable affordable deployment, with costs scaling based on usage.

Integration with existing educational platforms reduces the need for additional investments.

Cost Savings and Benefits

Efficiency Gains: Automating sentiment analysis eliminates manual feedback processing, saving time and reducing labor costs.

Improved Outcomes: By identifying areas for improvement, the system contributes to better educational quality, potentially increasing student retention and satisfaction rates.

Scalability: A single system can serve multiple courses and departments, maximizing cost-efficiency.

Risks and Mitigation

Risk: Budget overruns due to unforeseen development challenges.

Mitigation: Begin with a pilot implementation to manage costs effectively.

Risk: High initial costs for cloud infrastructure.

Mitigation: Start with a local server setup for initial development and scale to cloud services gradually.

1.7.3 Operational feasibility

Implementation Plan

Phased rollout ensuring manageable implementation and minimizing disruption to existing workflows.

Pilot testing with a small number of courses or departments to allow for iterative refinement.

Ease of Use

User-friendly dashboards and intuitive interfaces reducing the learning curve for educators and administrators.

Built-in tutorials and documentation so as to help users understand system functionalities.

Maintenance and Support

Ongoing updates to the machine learning models and system components ensuring long-term reliability.

Risks and Mitigation

Risk: Resistance from users unfamiliar with AI-based systems.

Mitigation: Provide training sessions and emphasize the benefits of automated sentiment analysis.

Risk: Operational downtime during system integration.

Mitigation: Schedule integration during low-activity periods and perform extensive testing beforehand.

Conclusion

The Sentiment Analysis System for Education is technically viable due to the availability of advanced tools and technology. Economically, it is cost-efficient, leveraging open-source software and cloud infrastructure to minimize expenses. Operationally, the system is practical, with a phased implementation plan ensuring smooth adoption and maintenance. Overall, the project is feasible and holds significant potential to enhance educational outcomes**.**

1.8 Project plan

A project plan is a formal document designed to guide the control and execution of a project. A project plan is the key to a successful project and is the most important document that needs to be created when starting any business project.

1.8.1 Schedule of activities and Gantt chart

ACTIVITY	TENTATIVE DATE
Project Planning and Setup	04\09\2024
Data Collection and Storage	05\10\2024
Data Preprocessing	04\11\2024
Feature Engineering	07\01\2025
Model Selection and Training	11\03\2025
Model Evaluation and Tuning	03\04\2025
User Interface Development	05\06\2025
System Deployment	17\07\2025
Documentation and Presentation	01\08\2025

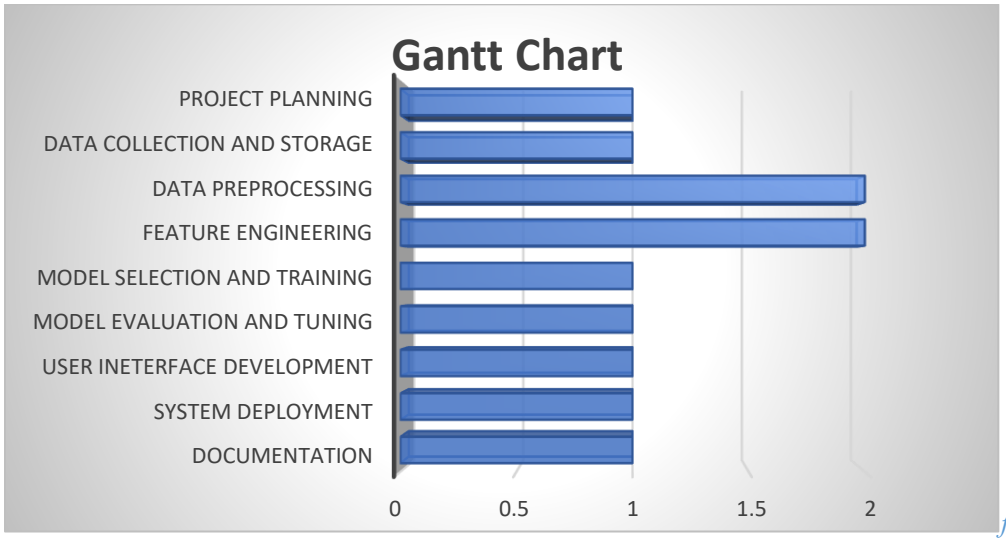


fig1.0

Chapter 2: Literature Review

2.0 Introduction

This chapter presents a review of existing studies, techniques, and systems related to sentiment analysis in education. The focus is on understanding how sentiment analysis has been applied in educational settings, identifying gaps in current methodologies, and highlighting the relevance of this study in improving learning outcomes.

2.1 Related Work

The point of the section is to highlight work done by others that somehow ties in with the current project. Several sentiment analysis systems have been developed, but few specifically focus on education. Below are some related systems:

2.1.1 Sentiment Analysis in Online Education Platforms

Studies on sentiment analysis in **Massive Open Online Courses (MOOCs)** like Coursera and Udemy reveal that student reviews contain valuable insights on course quality.

A study by [Zhang et al., 2019] analysed student reviews on Udemy, identifying trends in course satisfaction and instructor performance.

Strengths

- Improved Student Feedback Analysis
- Enhanced Engagement
- Early Identification

Weaknesses

- Lack of **real-time** sentiment analysis for immediate intervention.
- Limited **integration** with Learning Management Systems (LMS).
- Insufficient focus on **multilingual** sentiment analysis for global education platforms.

2.1.2 AI-Powered Student Engagement Systems

Some universities have implemented AI-powered engagement tracking tools that analyse student responses in discussion forums to measure their participation levels.

IBM's Watson AI has been used in analysing sentiment in education but lacks customization for specific courses.

Strengths

- Enhanced Engagement
- Precisely trained models

Weaknesses

- Lack of **real-time** sentiment analysis for immediate intervention.
- Limited **integration** with Learning Management Systems (LMS).
- Insufficient focus on **multilingual** sentiment analysis for global education platforms.

2.1.3 Instructor Performance Evaluation

Sentiment analysis has also been used to evaluate instructor performance based on student reviews.

Research by [Santos et al., 2020] highlighted how analysing sentiment trends over a semester helps instructors improve their teaching methodologies.

Strengths

- Makes use of pre-trained models
- Handles vast amounts of data

Weaknesses

- Lack of **real-time** sentiment analysis for immediate intervention.
- Limited **integration** with Learning Management Systems (LMS).
- Insufficient focus on **multilingual** sentiment analysis for global education platforms.

2.1.4 Emotion Detection in Online Learning

With the rise of online education, researchers have used sentiment analysis to detect student emotions from written discussions, chat messages, and assignments.

A study by [Altrabsheh et al., 2014] showed that automated sentiment analysis could help identify struggling students in real time.

Strengths

- Personalized feedback
- Adaptive Instruction

Weaknesses

- Lack of **real-time** sentiment analysis for immediate intervention.
- Limited **integration** with Learning Management Systems (LMS).
- Insufficient focus on **multilingual** sentiment analysis for global education platforms.

Conclusion

The literature review highlights that sentiment analysis has great potential in improving education by analyzing student feedback. However, existing systems lack real-time processing and seamless integration with educational platforms. This study aims to address these gaps by developing a **customized sentiment analysis system** specifically designed for educational settings, providing real-time insights and recommendations for improving teaching methodologies.

Chapter 3: Analysis

3.0 Information Gathering Tools

Information such as system requirements, user requirements and functional requirements that relates to this project was gathered as it is essential to the success of the project. Data gathering is important to have a comprehension of the current framework, prompting necessities assurance. This is essential for determining precise system requirements. User requirements for the proposed system were assembled utilizing the accompanying information gathering procedures:

- Observation
- Document Review
- Existing Data Analysis
- Literature Review

3.0.1 Observation

Direct observation of classroom interactions and online discussions in learning management system

Identified patterns in student behavior and emotional responses during classes or discussions, which informed the development of the sentiment analysis model.

Advantages

- Captures authentic, real-time behaviors, reactions, and emotional responses without relying on what students *say* they felt.
- Students often behave naturally when unaware they are being observed, leading to more genuine data.
- Helps identify recurring behavioural and emotional patterns that directly inform model features for the sentiment analysis.

Disadvantages

- Observers may interpret behaviors differently based on personal judgment or expectations, affecting the objectivity of the data.
- While behaviors can be seen, the **exact emotional state** or reasons behind actions may remain unclear without verbal confirmation.
- Direct observation is difficult to scale, especially in large classes or online environments with hundreds of discussion threads.

3.0.2 Document Review

Analyzed existing educational materials, course evaluations, and feedback reports to understand how sentiments are currently recorded and addressed. Reviewed documentation related to prior projects or systems used for feedback collection, which provided insights into the limitations of current practices.

Advantages

- Provides access to previously recorded feedback, evaluations, and reports, helping identify patterns over time.
- Documents are already available, eliminating the need to collect new data from scratch.
- Offers insights into **how feedback has been handled** in the past, exposing gaps or inefficiencies in current feedback mechanisms.

Disadvantages

- Documents might not reflect current student experiences, tools, or course delivery methods (especially with rapid changes in edtech).
- Written feedback often lacks detailed emotional cues or non-verbal indicators present in face-to-face observations.
- Existing reports may have been filtered or summarized, excluding critical negative feedback or emotional responses.

3.0.3 Existing Data Analysis

Examined historical data from previous courses or feedback sessions to identify trends in student engagement and sentiment. Used existing datasets to train initial sentiment analysis models and to benchmark performance.

Advantages

- Historical datasets reduce the need for new data collection, saving time and resources.
- Provides valuable training data for developing and testing the sentiment analysis model, ensuring it learns from real-world examples.
- Enables performance comparison between the initial model and historical sentiment trends, helping evaluate model accuracy.
- Historical analysis can inform future expectations about student behavior and help educators prepare interventions proactively.

Disadvantages

- Existing data might be incomplete, noisy, or biased, affecting the reliability of analysis and model training.
- Older datasets might lack detailed context (e.g., course changes, teaching style shifts), making it hard to interpret trends accurately.
- Using past student data raises potential privacy issues if data was not initially collected for sentiment analysis purposes.

3.0.4 Literature Review

Reviewed academic literature and case studies related to sentiment analysis in educational settings and identified best practices, challenges, and technological advancements that could inform the design and implementation of the new system.

Advantages

- Helps identify proven methods, models, and frameworks from previous research that can improve the design and implementation of your system.
- Highlights common challenges and pitfalls in educational sentiment analysis projects, helping you plan mitigation strategies in advance.
- Provides insights into the latest tools, algorithms, and emerging technologies that could enhance model accuracy and system performance.

Disadvantages

- Thoroughly reviewing and analyzing a large body of literature takes significant time and effort.
- Research findings or case studies from other institutions or countries may not directly apply to your specific educational setting.
- Literature is often theoretical or based on controlled studies, which might not capture the complexities of your real-world environment.

3.1 Data Analysis

The data analysis phase was essential in gaining a deep understanding of existing course review systems that utilize sentiment analysis, along with identifying their limitations. The primary goal was to pinpoint the shortcomings of current methods and systems, laying the groundwork for designing a more effective solution.

A detailed evaluation of existing systems and techniques was carried out, focusing on both their strengths and weaknesses. This analysis provided valuable insights and served as a foundation for considering alternative approaches. The findings from this stage directly informed the design of the proposed system, ensuring it addresses the challenges and gaps observed in previous implementations.

3.1.0 Functionalities of Existing Systems

Existing sentiment analysis systems designed for course review and educational feedback collection incorporate a range of functionalities aimed at analysing student sentiments and informing instructional improvements. These systems leverage natural language processing (NLP), machine learning, and data visualization techniques to derive meaningful insights from textual feedback. Key functionalities include:

1. **Feedback Collection and Data Aggregation**:
   Most systems provide interfaces that facilitate the collection of student feedback through surveys, open-ended text fields, or integration with Learning Management Systems (LMS) such as **Moodle**, **Canvas**, or **Blackboard**. These platforms aggregate feedback from various sources, including discussion forums, course evaluations, and online assessments.
2. **Text Preprocessing and Data Cleaning:**
   Systems perform essential text preprocessing tasks such as tokenization, stemming, stop-word removal, and normalization to prepare data for sentiment analysis. Tools like **NLTK (Natural Language Toolkit)** or **spaCy** are commonly employed for these preprocessing tasks.

3. **Sentiment Analysis and Classification:**
   Utilizing machine learning models or rule-based approaches, these systems analyse student feedback to classify sentiment into categories such as positive, negative, or neutral. For instance, systems like **Monkey-Learn** and **IBM Watson Natural Language Understanding** offer pre-trained models capable of performing sentiment analysis on educational data.
4. **Sentiment Scoring and Emotional Polarity Assessment:**
   Beyond simple classification, some systems assign sentiment scores or polarity values to each feedback entry, quantifying the intensity of positive or negative sentiments. This aids in measuring overall student satisfaction or dissatisfaction within a specific course or learning activity.
5. **Topic and Keyword Extraction:**
   Advanced systems incorporate topic modelling and keyword extraction to identify recurring themes and issues raised by students. Techniques such as Latent Dirichlet Allocation (LDA) are utilized to uncover dominant topics within the feedback data.
6. **Visualization and Reporting:**
   The analysed data is presented through dashboards, charts, and graphical representations that allow instructors and administrators to track sentiment trends over time. Tools like **Tableau**, **Power BI**, and custom-built dashboards enable stakeholders to interpret complex data intuitively.
7. **Automated Alerts and Recommendations:**
   Certain systems generate automated alerts when negative sentiment surpasses a predefined threshold, enabling timely interventions. Additionally, some platforms offer actionable recommendations based on sentiment trends to guide curriculum adjustments or teaching strategies.
8. **Benchmarking and Comparative Analysis:**
   Systems such as **Qualtrics XM for Education** allow institutions to benchmark sentiment results against historical data or comparable courses, facilitating continuous improvement efforts.
9. **Data Export and System Integration:**
   Most platforms provide options for exporting sentiment analysis results in various

formats for further analysis or integration with academic performance monitoring systems.

Examples of Existing Systems:

- **IBM Watson NLU** – Offers sentiment and emotion analysis on educational text data.
- **Monkey-Learn** – Provides customizable sentiment analysis models suitable for course feedback.
- **Qualtrics XM for Education** – Supports survey-based sentiment collection and advanced reporting.
- **RapidMiner** – Enables machine learning-based sentiment analysis and visualization for academic research.
- **Tableau** – Frequently integrated for visualizing sentiment trends and generating reports.

In summary, these existing systems collectively aim to enhance understanding of student experiences by automating the analysis of large volumes of feedback data. However, limitations such as handling sarcasm, contextual interpretation, and emotional nuance remain challenges, paving the way for further system improvements.

Comparison with Proposed System

While existing sentiment analysis systems provide valuable functionalities for processing student feedback, several limitations and gaps remain, particularly in addressing the specific needs of educational settings. The proposed system seeks to improve upon these shortcomings by offering enhanced features and a more targeted approach.

Aspect	Existing Systems	Proposed System
Scope of Analysis	General sentiment analysis models, often designed for business reviews or social media data.	Tailored specifically for educational environments and course review contexts.
Context Sensitivity	Limited contextual understanding of educational jargon, sarcasm, or implicit feedback.	Incorporates educational context-aware sentiment models, improving interpretation of student feedback.
Real-Time Analysis	Mostly batch processing; real-time analysis is limited or requires advanced configurations.	Supports real-time sentiment analysis and live feedback visualization for timely interventions.
Visualization & Reporting	Standard dashboards with basic sentiment trends and polarity	Enhanced visual analytics, including bar charts for sentiment distribution and actionable insights
Customization & Flexibility	Predefined models with limited flexibility for domain-specific adjustments.	Customizable sentiment models trained on historical student feedback and course-specific data.
Emotion Detection	Primarily focuses on positive, negative, or neutral classification.	Extended analysis to detect emotional states (e.g., frustration, motivation) for deeper insights.
Integration with LMS	Partial or external integration with learning platforms like Moodle or Canvas.	Full integration with selected LMS platforms to streamline feedback collection and analysis.
User Experience	User interfaces may be complex or not tailored for educators and students.	User-friendly interfaces designed specifically for academic staff and students for ease of use.
Alerts & Recommendations	Limited or absent predictive features and automated guidance.	Generates alerts for negative sentiment spikes and suggests teaching improvements or interventions.
Data Privacy Considerations	General data protection mechanisms; not always aligned with educational privacy regulations.	Designed to comply with educational data privacy standards (e.g., FERPA, GDPR) for secure data use.

Table 1.0

The proposed system is designed to:

- Provide **domain-specific sentiment analysis** fine-tuned for education.
- Enable **real-time monitoring** and **visual feedback** for instructors.
- Detect **emotions beyond basic polarity**, offering deeper insights into student well-being and engagement.
- Integrate **seamlessly with learning platforms** for automatic data collection.
- Ensure compliance with **educational data privacy standards**.
- Offer **custom recommendations** based on sentiment trends to support evidence-based decision-making in teaching.

3.1.1 UML context diagrams

SYSTEM DESCRIPTION UML

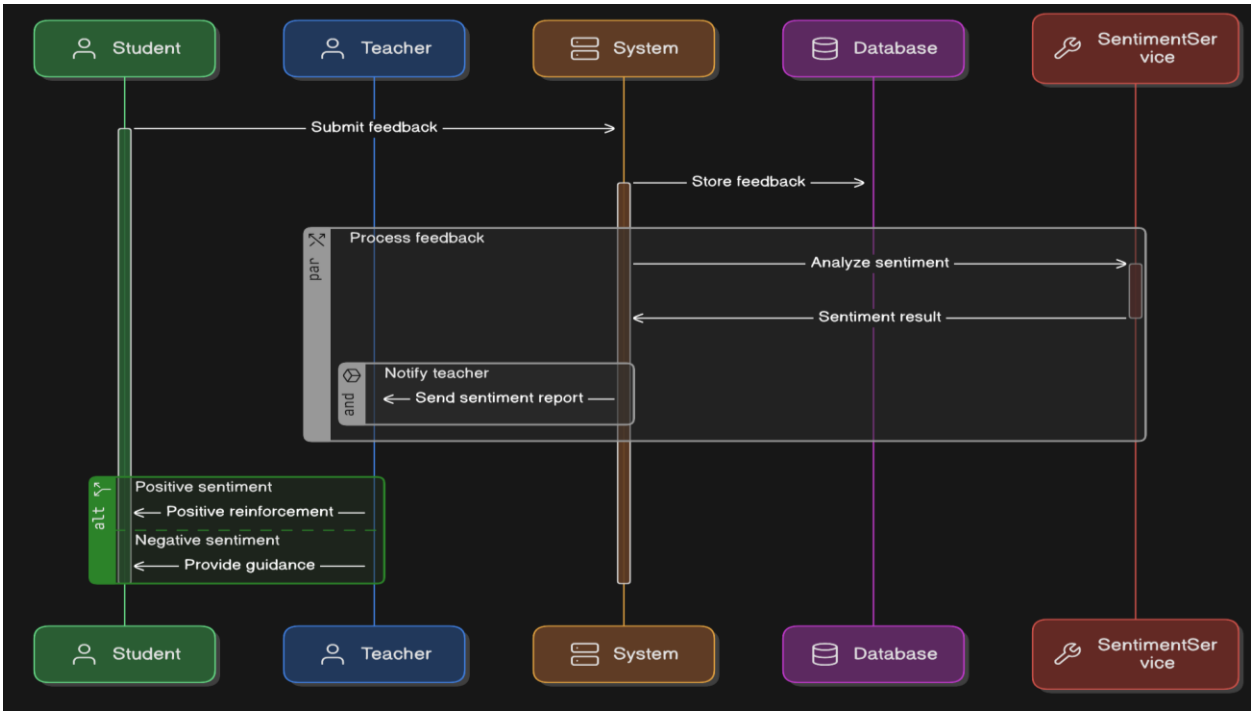


fig2.0

3.1.2 Data Flow Diagram of the Existing System

Data Flow diagram (DFD) is a graphical visualization of the movement of data through an information system. A context diagram which can also be called level- zero data flow diagram shows the system under consideration as a single high-level process and the relationships that the system has with other external entities.

DFD LEVEL 0 (EXISTING SYSTEM)

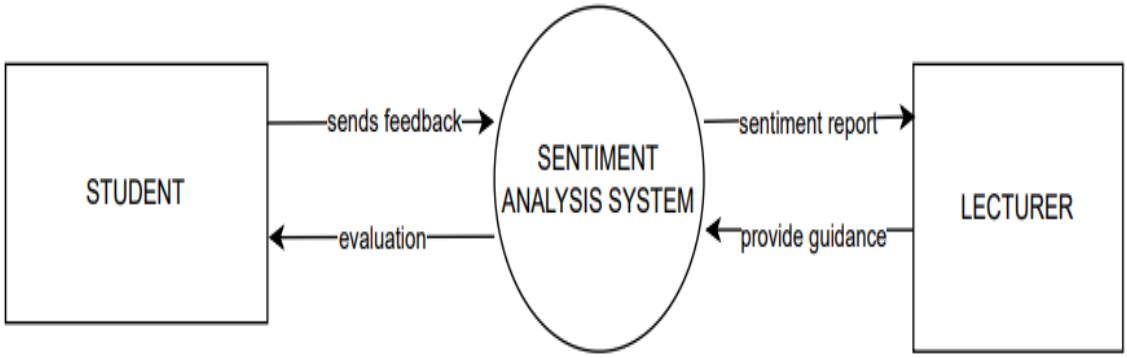


fig3.0

DFD LEVEL 1 (EXISTING SYSTEM)

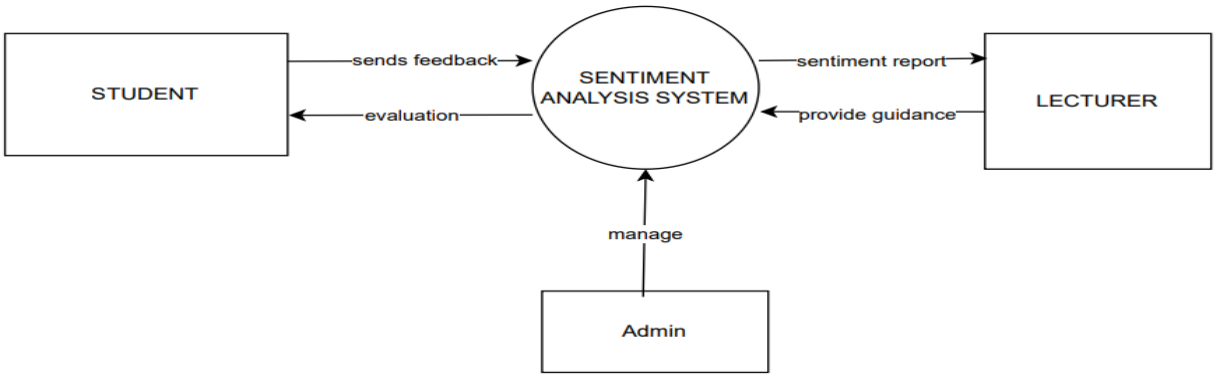


fig4.0

3.1.3 Evaluation of Alternative Systems

In the process of designing the proposed sentiment analysis system for educational course reviews, several alternative systems and approaches were evaluated to assess their suitability, effectiveness, and limitations within the educational context. These alternatives, while offering diverse functionalities, revealed varying degrees of alignment with the specific goals of analyzing student feedback for course improvement.

1. General-Purpose Sentiment Analysis Tools (e.g., IBM Watson NLU, Google Cloud Natural Language API)

Strengths:

- Advanced machine learning models with high accuracy on general datasets.
- Ability to analyze large datasets and handle multilingual input.
- Support for sentiment scoring and entity recognition.

Limitations:

- Lack of domain-specific tuning for educational feedback.
- Limited contextual understanding of academic language, sarcasm, or implicit sentiments common in course evaluations.
- High cost associated with large-scale or continuous usage.

2. Survey and Feedback Platforms (e.g., Qualtrics XM, SurveyMonkey, Microsoft Forms)

Strengths:

- User-friendly survey creation and deployment.
- Built-in analytics for basic sentiment trends and visualization.
- Data export and report generation features.

Limitations:

- Sentiment analysis features are often basic and limited to polarity detection.

- Minimal support for emotional state detection or advanced NLP techniques.
- Dependence on structured survey responses rather than unstructured natural language feedback.

3. Learning Management Systems with Feedback Modules (e.g., Moodle, Canvas, Blackboard)

Strengths:

- Direct integration with academic courses and student activities.
- Simplifies the collection of feedback data within the learning environment.

Limitations:

- Typically lack advanced sentiment analysis or machine learning capabilities.
- Feedback analytics are often limited to quantitative data, offering little insight into emotions or detailed sentiment.

4. Open-Source NLP Libraries (e.g., NLTK, spaCy, Text-Blob)

Strengths:

- Flexibility and control over customization of sentiment analysis models.
- Cost-effective for academic research and prototype development.
- Availability of tools for preprocessing, keyword extraction, and topic modeling.

Limitations:

- Requires significant technical expertise and development time.
- No ready-to-use sentiment models specifically trained on educational datasets.
- Scalability and real-time analysis capabilities are limited without further development.

Conclusion

While each alternative system offers valuable functionalities, they exhibit limitations when applied directly to the educational domain of course review and sentiment analysis. General-purpose tools and open-source libraries lack the contextual understanding of academic feedback, while survey platforms and LMS modules often provide limited sentiment analysis capabilities.

These evaluations informed the decision to develop the **proposed system**, which:

- Combines the strengths of these alternatives,
- Introduces domain-specific sentiment models,
- Supports emotional state detection,
- Offers real-time analysis and visualization,
- Ensures compliance with educational data privacy standards.

The proposed system aims to bridge the identified gaps, providing a more robust, targeted, and effective solution for enhancing course review analysis in educational settings.

3.2 Functional Analysis of Proposed System

3.2.1 Functional Requirements of Proposed System

- **Data Ingestion**
  The system shall allow users to enter text data online (e.g., CSV, JSON).
- **Data Preprocessing**
  - The system shall clean and preprocess input text data.
  - The system shall tokenize text into individual words or phrases.
  - The system shall remove stop words and apply normalization techniques.
- **Sentiment Analysis**
  -The system shall classify the sentiment of the input text as positive, negative, or neutral.
  -The system shall provide a sentiment score or probability for each classification

- **Visualization and Reporting**

  -The system shall generate visualizations of sentiment analysis results (e.g., charts, graphs).

  -The system shall allow users to generate reports summarizing sentiment findings.

3.2.2 Non-Functional Requirements

Non-functional requirements (NFRs) are the requirements that specify the desired quality attributes of a system, such as performance, reliability, security, and maintainability. NFRs are typically contrasted with functional requirements, which specify the desired functionality of a system. Some of the non-functional requirements may include the following features:

- **Performance Requirements**

  -The system shall process sentiment analysis for at least 1000 text entries per minute.

  -The system shall maintain a response time of less than 2 seconds for user queries
- **Scalability Requirements**

  The system shall be capable of handling an increase in data volume without significant performance degradation.
- **Security Requirements**

  -The system shall implement user authentication and authorization.

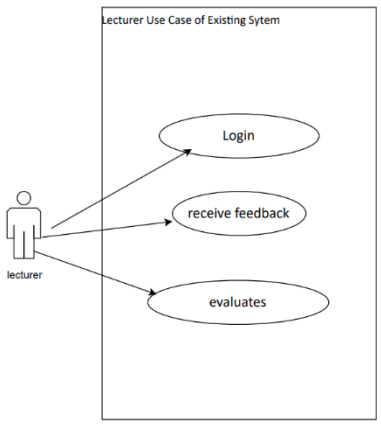
  -The system shall ensure data confidentiality and integrity.
- **Usability Attributes**

  -The system shall provide a user-friendly interface that requires minimal training.

  -The system shall include help documentation and user guides

3.3 Use Case Diagrams

3.3.1 Use case for existing system



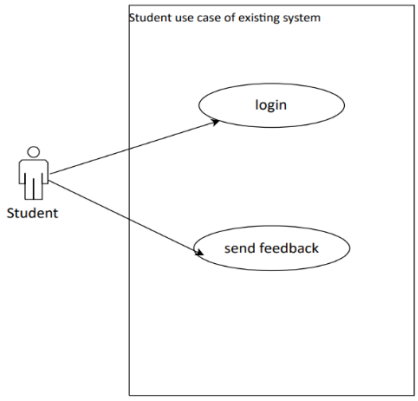


fig6.0

fig 5.0

3.3.2 Use case PROPOSED system

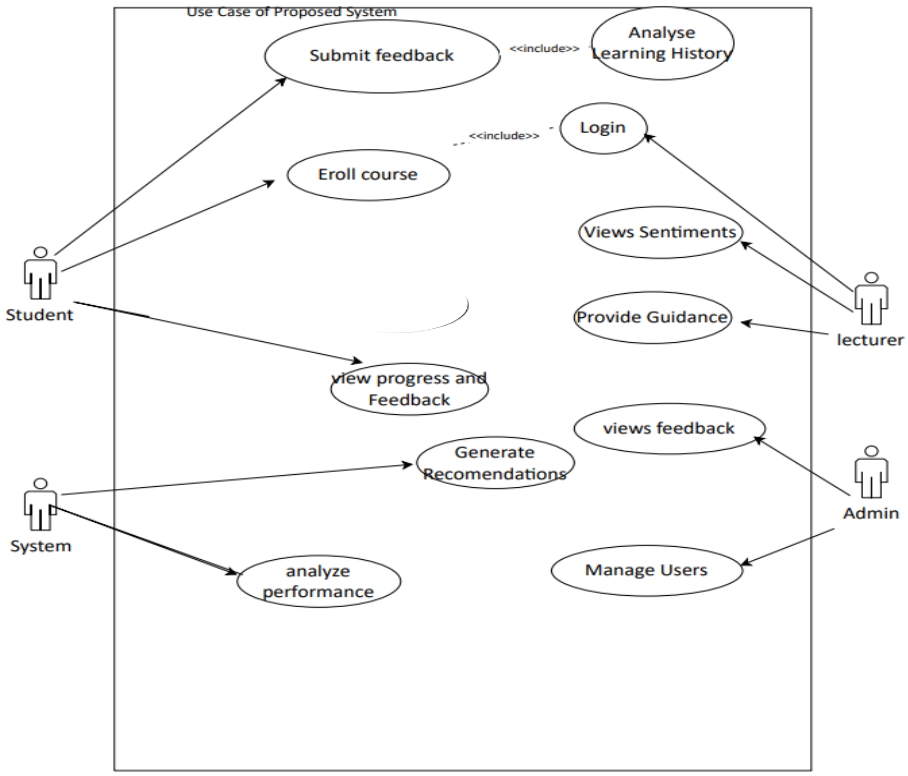


fig7.0

Chapter 4: Design

4.0 Introduction

The design of the proposed sentiment analysis system for educational course review focuses on addressing the shortcomings of existing solutions by incorporating domain-specific sentiment modeling, real-time analysis capabilities, and seamless integration with learning platforms. The system is structured to ensure accuracy, usability, and compliance with educational data privacy standards.

4.0.1 System Architecture Overview

The system adopts a modular architecture, comprising the following core components:

Component	Description
Data Collection Module	Gathers student feedback from multiple sources, including Learning Management Systems (LMS), online discussion forums, and feedback forms.
Preprocessing Module	Cleans and normalizes input text data by removing noise, tokenizing sentences, handling stop words, and detecting special academic terms or expressions.
Sentiment Analysis Engine	Uses a hybrid model combining machine learning algorithms and rule-based techniques, trained on educational datasets to detect sentiment polarity and emotions.
Visualization & Reporting	Generates graphical representations of sentiment trends, including bar charts and dashboards. Provides actionable insights and summary reports for educators.
Recommendation Engine	Suggests pedagogical adjustments or interventions based on identified negative sentiment trends or emotional patterns.
Privacy & Security Layer	Ensures data encryption, access controls, and compliance with educational data protection regulations such as FERPA and GDPR.

Table2.0

4.1 System Diagrams

System diagrams serve as essential tools for visualizing the structure and functionality of complex systems. They provide a clear representation of the components, the dynamic forces acting upon them, and their interactions within the system. Additionally, system diagrams are valuable for illustrating how a modification in one part of the system can impact the overall operation, thereby supporting better analysis, design, and decision-making throughout the development process.

4.1.1 Context Diagram

A context diagram which can also be called level- zero data flow diagram shows the system as a single high-level process and the relationships that the system has with external entities:

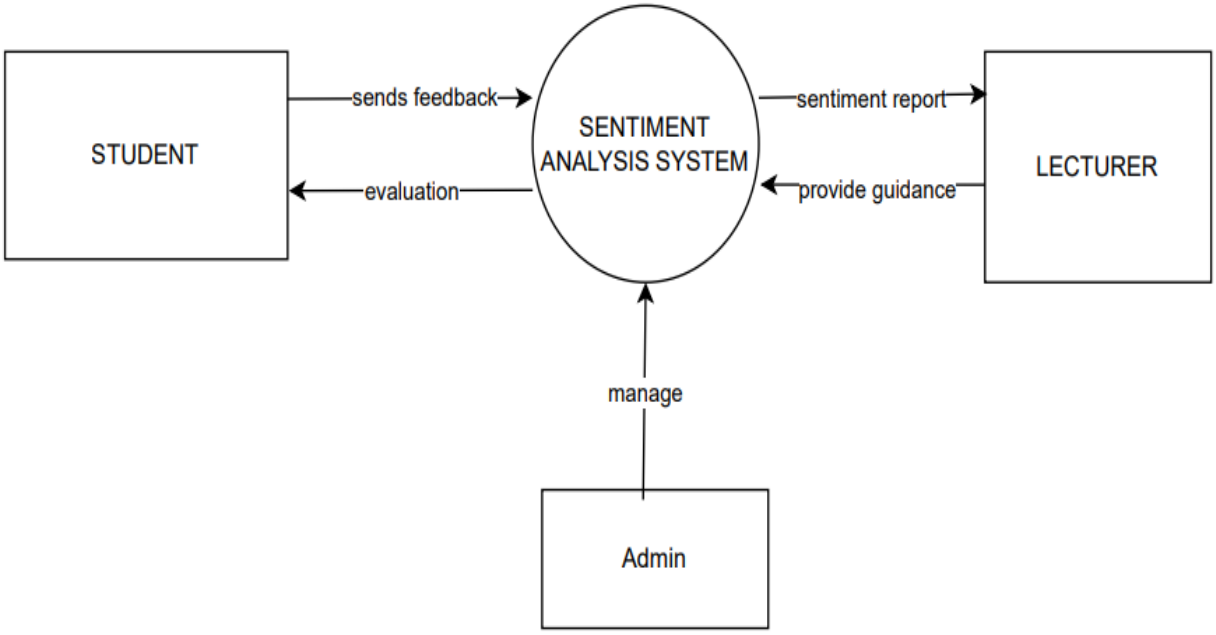


fig8.0

4.1.2 Level 1 DFD

This diagram is more detailed than the level zero diagram (context diagram). It breaks down the main processes into sub processes which can then be scrutinized and improved.

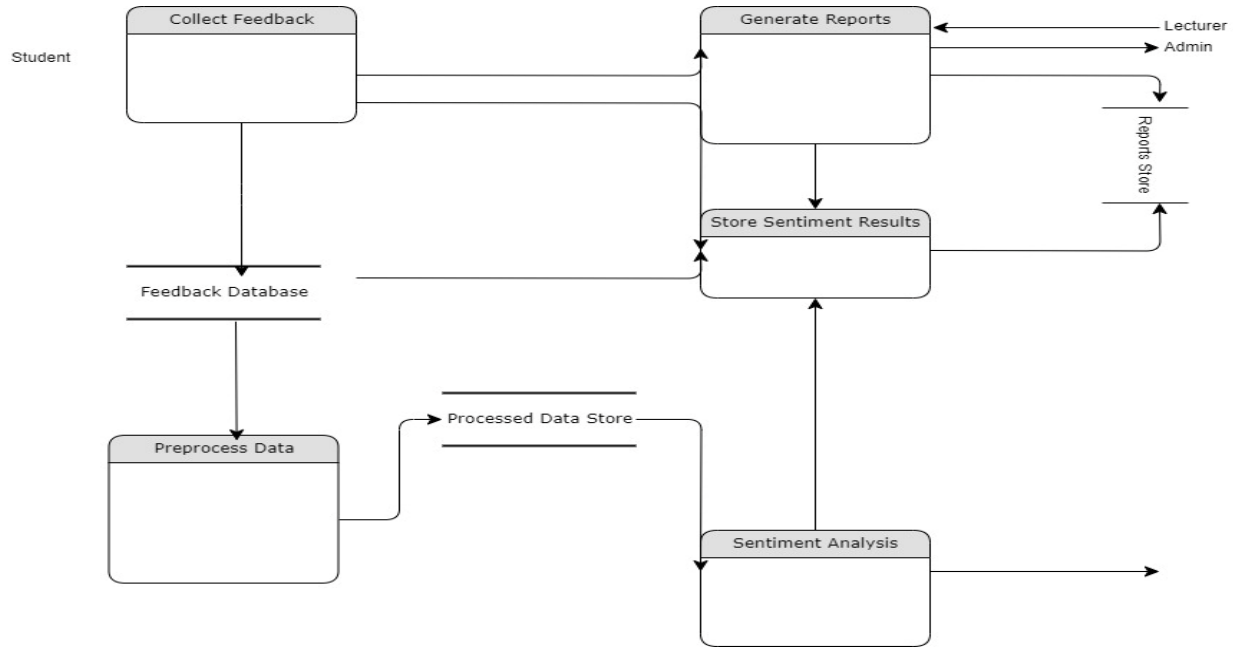


fig9.0

4.1.3 Activity Diagram

An activity diagram is basically a flow chart which shows flow of one activity to another.

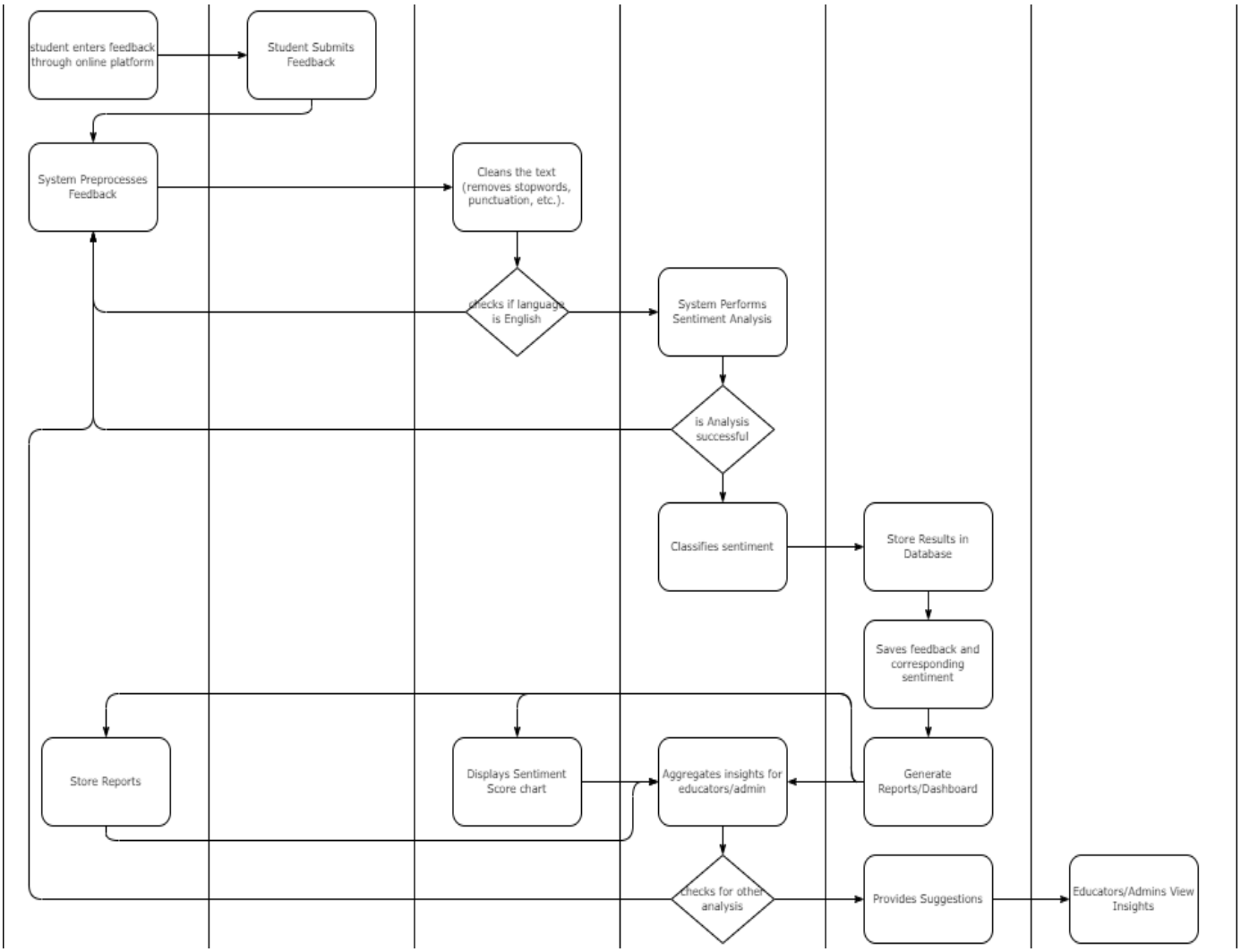


fig10.0

4.2 Architectural Design

The architectural design outlines the fundamental structure of the system by defining the key hardware and software components along with their interactions and interfaces. It serves as the blueprint for system development, ensuring that all elements work together seamlessly to achieve the desired functionality and performance.

This design incorporates both **hardware** and **networking considerations**, detailing the infrastructure required to support data collection, processing, storage, and visualization. The architecture ensures efficient communication between modules, scalability for handling large datasets, and robust performance to support real-time sentiment analysis in educational environments.

Hardware Components

- **Client Devices:** Computers, tablets, or smartphones used by students and instructors to access the system, submit feedback, and view sentiment reports.
- **Application Server:** A central server responsible for running the sentiment analysis engine, and handling user requests. This server is equipped with high-performance CPUs and GPUs to support machine learning operations.
- **Database Server:** A dedicated storage server used to store raw feedback data, pre-processed text, sentiment scores, and generated reports. It ensures data integrity, backup, and secure storage.

Networking Components

- **Internet Connectivity:** Provides remote access for instructors and students, allowing feedback submission and sentiment monitoring from any location.
- **Cloud Integration:** Supports scalability by integrating with cloud services for storage, processing, and backup when dealing with large volumes of data or distributed users.

This architectural design ensures that the system is scalable, secure, and capable of handling real-time sentiment analysis tasks. The combination of powerful servers, reliable networking, and

secure cloud options allows the system to function efficiently within an educational environment, providing educators with actionable insights for course improvement.

4.3 Database Design

Database design refers to the systematic organization of data based on a chosen database model. It involves identifying the types of data to be stored and defining the relationships between various data elements. A database model establishes the logical structure of the database, guiding how data is organized, connected, and accessed.

Logical database design focuses on defining entities, their attributes, and the relationships between them, often represented through entity-relationship diagrams (ERDs). In contrast, physical database design involves specifying the actual database tables, columns, data types, primary and foreign keys, and indexing strategies to optimize data storage and retrieval.

This structured approach ensures data integrity, consistency, and efficient query performance within the system.

4.3.1 Entity Relationship Diagram

An entity relationship diagram (ERD) shows the relationships between entities stored in a system database. Before designing an ERD one need to first at least identify the entities, attributes and relationships.

Entity	Attributes
Student	Student-ID, Name, Email, Course-ID
Lecturer	Lecturer-ID, Name, Email
Course	Course-ID, Lecturer-ID, Course-Name
Feedback	Feedback-ID, Course-ID, Feedback Text, Student-ID
Analysis (Sentiment)	Analysis-ID, Feedback-ID, Sentiment Score, Sentiment Label
Reports	Report-ID, Lecturer-ID, Course-ID, Summary

Table3.0

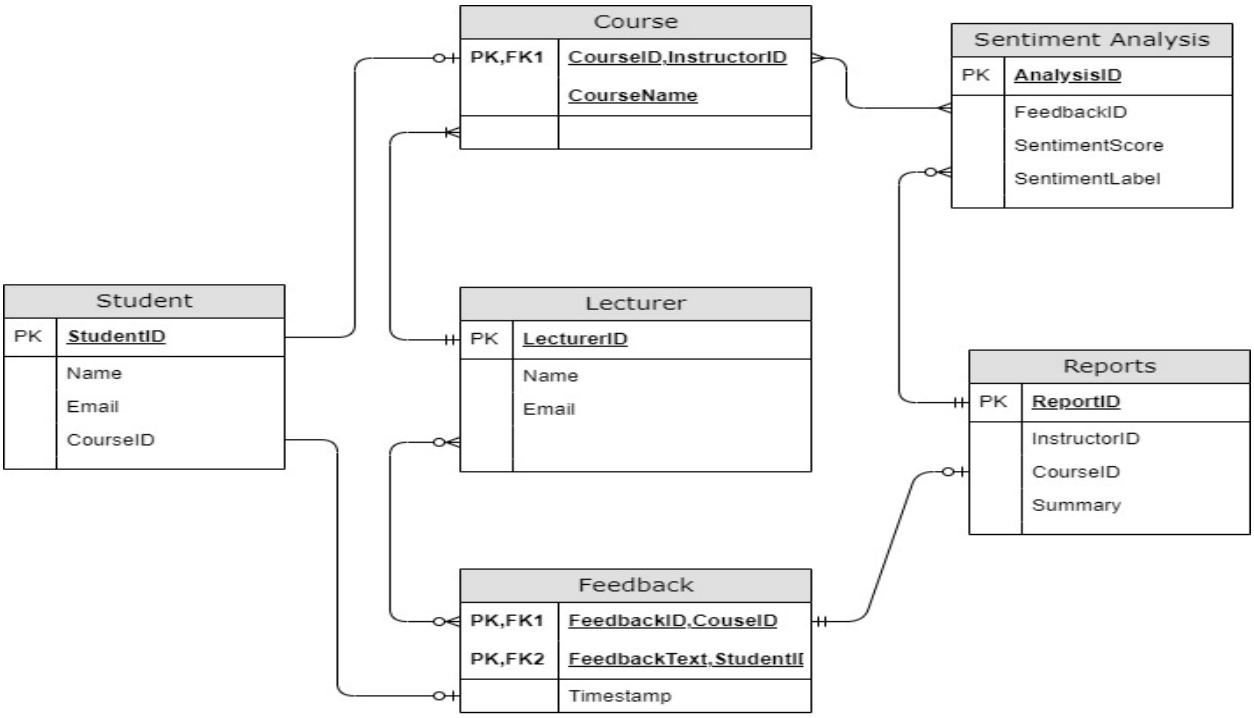


fig11.0

4.3.2 Normalized Databases

Normalization is done to:

- Prevent data redundancy
- Prevent anomalies (creation and delete)

From the attributes, the database was normalized to third normal form (3NF) and the following relations were achieved.

Students Table

Field Name	Field Type & Size	Description
Student ID	Integer (10)	Unique identifier for students
Name	Varchar (50)	Name of student
Email	Varchar (50)	Students email address

Lecturers Table

Field Name	Field Type & Size	Description
Lecturer ID	Integer (10)	Unique identifier for lecturers
Name	Varchar (50)	Name of Lecturer
Email	Varchar (50)	Lecturers email address

Courses Table

Field Name	Field Type & Size	Description
Course ID	Integer (10)	Unique identifier for Courses
Course Name	Varchar (50)	Name of Course
Lecturer ID	Integer (10)	Foreign Key

tables 4.0,5.0and 6.0

Feedback Table

Field Name	Field Type & Size	Description
Feedback ID	Integer (10)	Unique identifier for Feedback Text
Student ID	Integer (10)	Foreign Key
Course ID	Integer (10)	Foreign Key
Feedback Text	Varchar (100)	Feedback from student

Analysis Table

Field Name	Field Type & Size	Description
Analysis ID	Integer (10)	Unique identifier for Analysis
Feedback ID	Integer (10)	Foreign Key
Sentiment Score	Boolean	Scale from 0 to 1
Sentiment Label	Boolean	Positive or Negative Result

Reports Table

Field Name	Field Type & Size	Description
Report ID	Integer (10)	Unique identifier for Reports
Lecturer ID	Integer (10)	Foreign Key
Course ID	Integer (10)	Foreign Key
Summary	Varchar (100)	Overall results

table 7-9

4.4 Program Design

4.4.0 Class Diagram

Class diagrams is graphical notation used to construct and visualize object-oriented system.

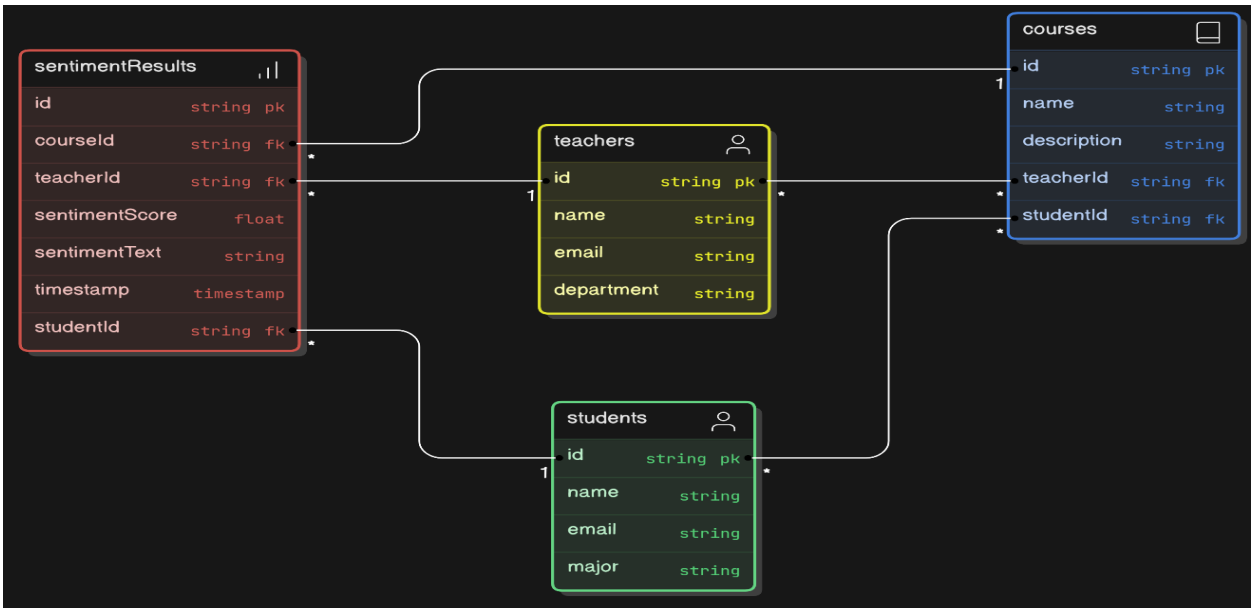


fig12.0

4.4.1 Sequence Diagram

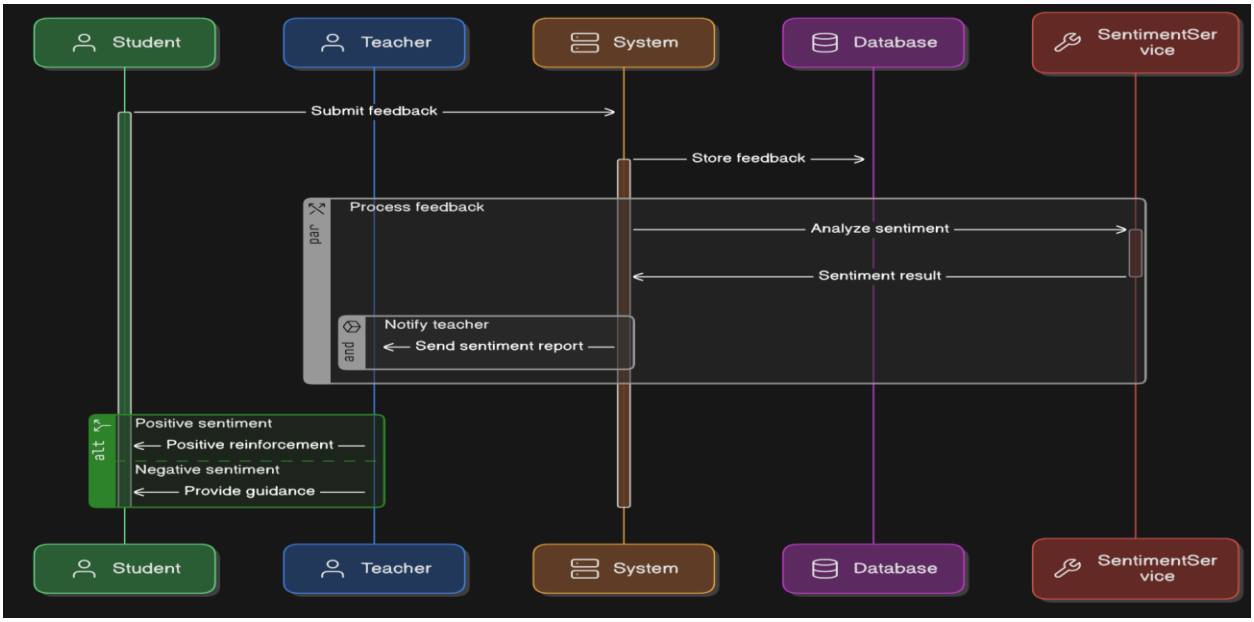


fig 13.0

4.4.2 Package Diagram

Package diagrams are structural diagram which is commonly used to simplify complex class diagrams and organize classes into packages. A package is a collection of related UML elements including diagrams, documents, classes, and event packages (Norlyn,2021).

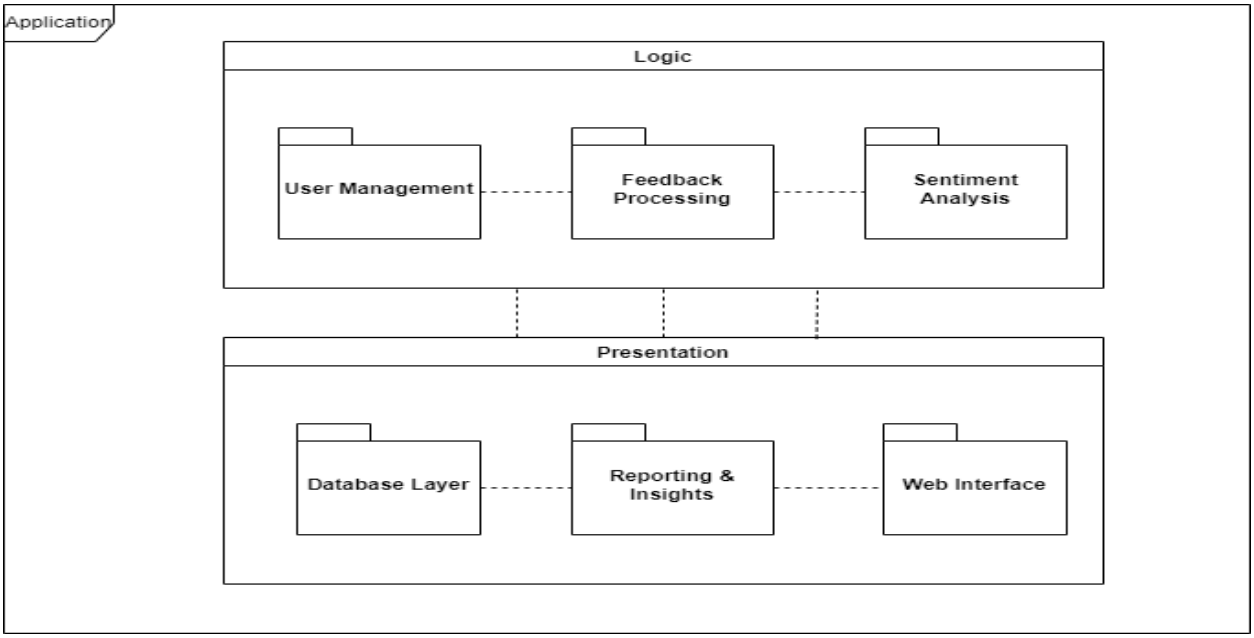


fig14.0

4.4.3 Pseudo Code

Start the system

Collect Feedback

- Prompt the user to enter feedback text
- Store the feedback text in the database

Preprocess the Feedback

- Clean the feedback text (remove punctuation, convert to lowercase, remove stop words)
- Tokenize the feedback text into individual words

Analyze Sentiment

- Initialize a sentiment score to zero
- For each word in the tokenized feedback:

- If the word is found in the positive word list, increase the sentiment score by 1
- If the word is found in the negative word list, decrease the sentiment score by 1

Classify the Sentiment

- If the sentiment score is greater than 0, classify the feedback as **Positive**
- If the sentiment score is less than 0, classify the feedback as **Negative**
- If the sentiment score is equal to 0, classify the feedback as **Neutral**

Store the Result

- Save the classified sentiment result in the database

Display the Result

- Output the sentiment analysis result to the user or instructor

End the process

4.5 Interface Design

A user interface (UI) is the front-end application view to which a client associates with when using the system. User interface design is done with a focus on maximizing efficiency, responsiveness and aesthetics to provide a good user experience. The user interface should allow the user to perform required tasks without complications.

4.5.0 Input Design

Input design is the process of designing the input to a computer system. This includes the design of the user interface, the format of the input data, and the validation of the input data (Kendall,2018). It involves the designing of forms used to capture or to input data.

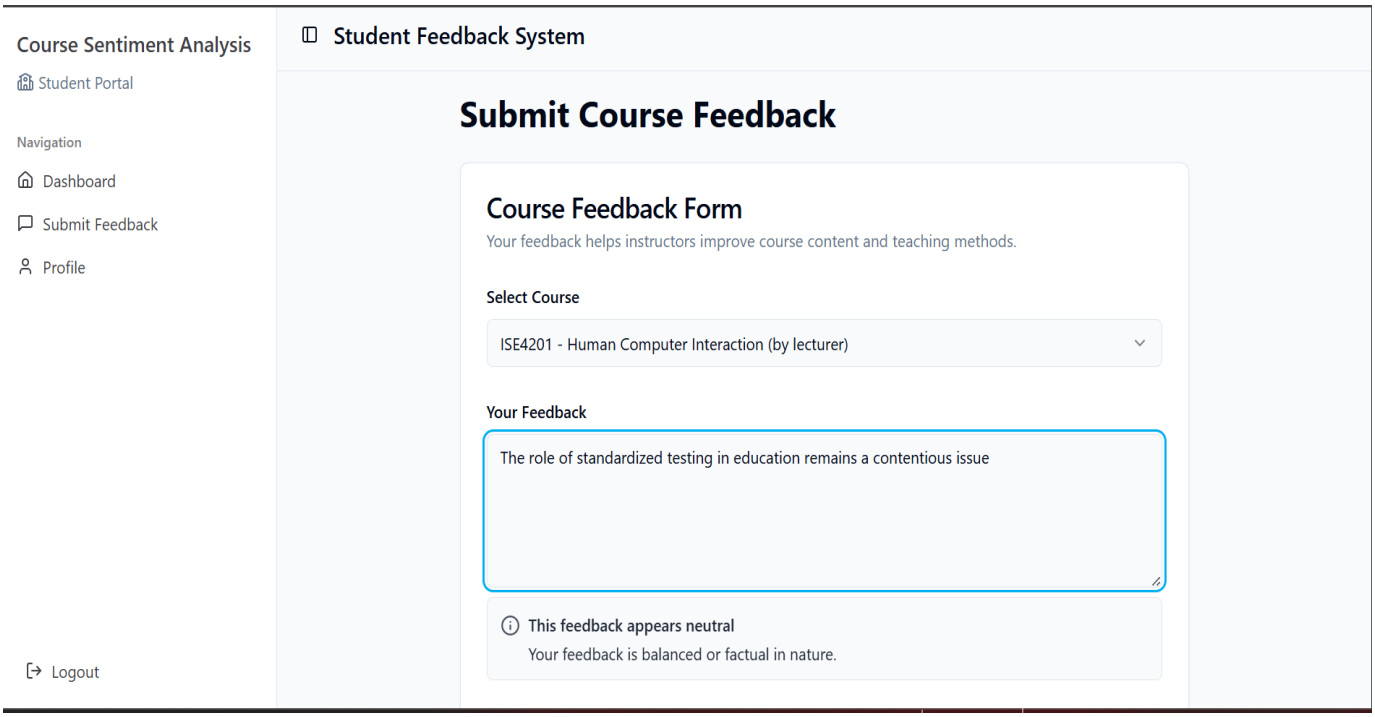


Fig 15.0: Course Selection

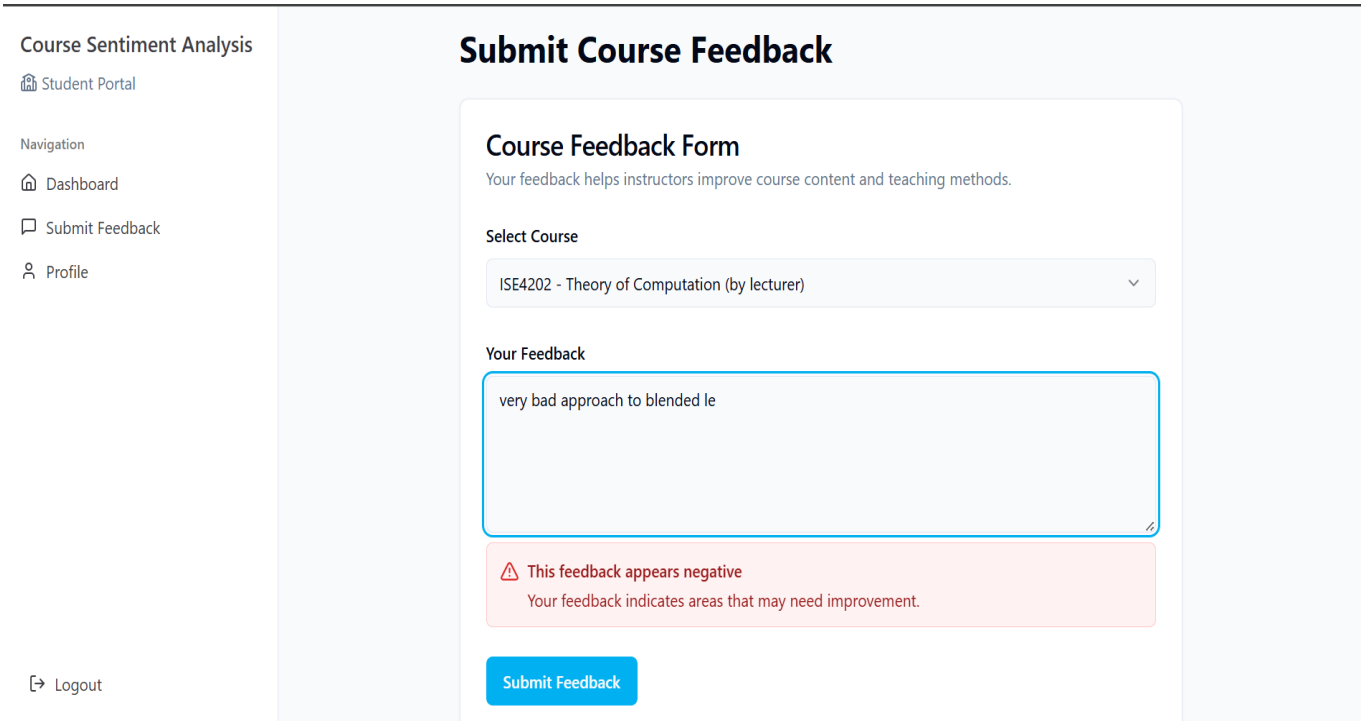


Fig 16.0: Feedback Submission

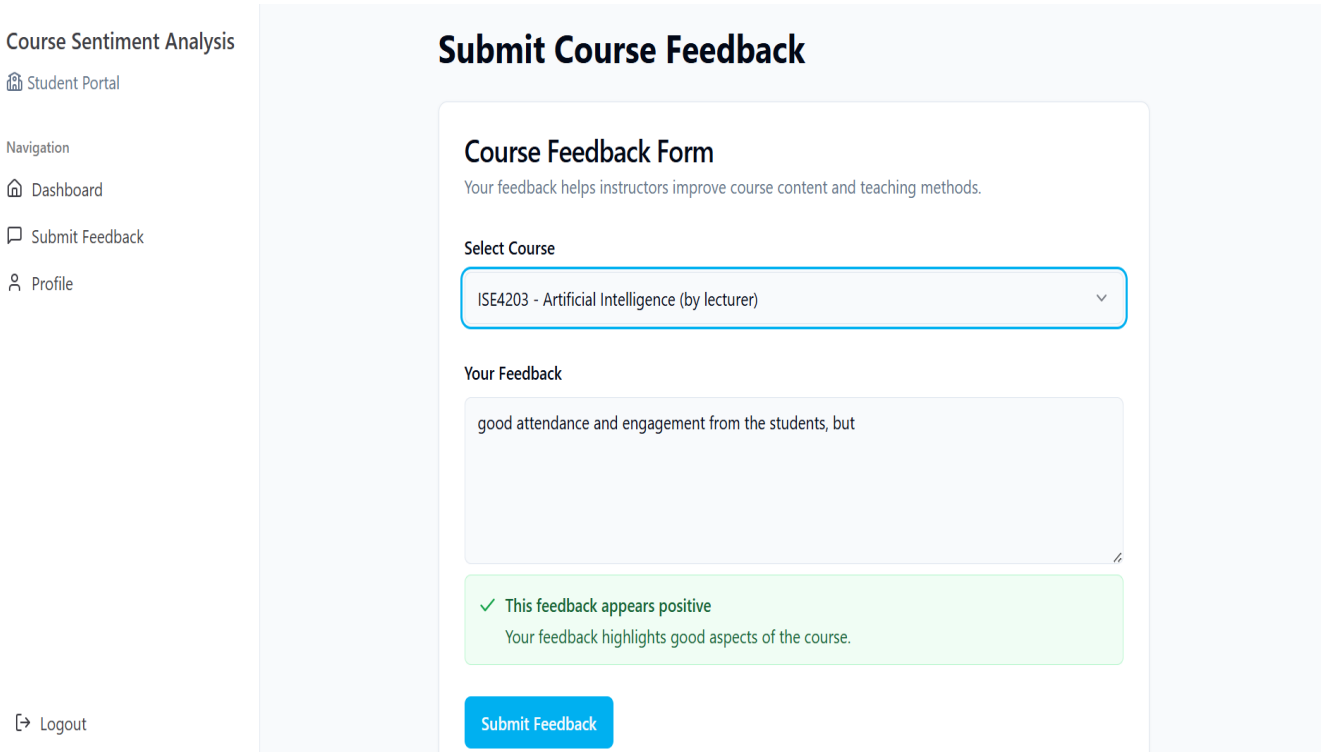


Fig 17.0 Feedback Submission

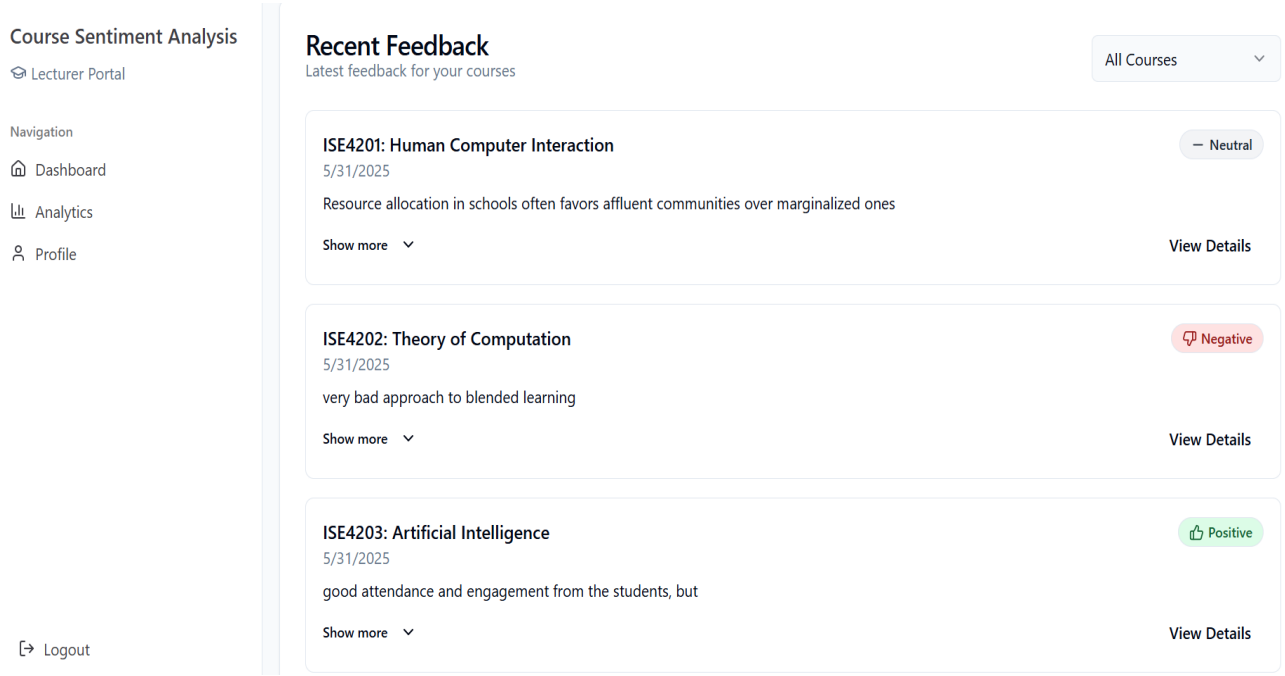


Fig17.1: stored feedback

4.5.1 Output Design

Designing of the mechanism by which the information processed by the system will be delivered to the user. The process of output is designed to meet the following goals:

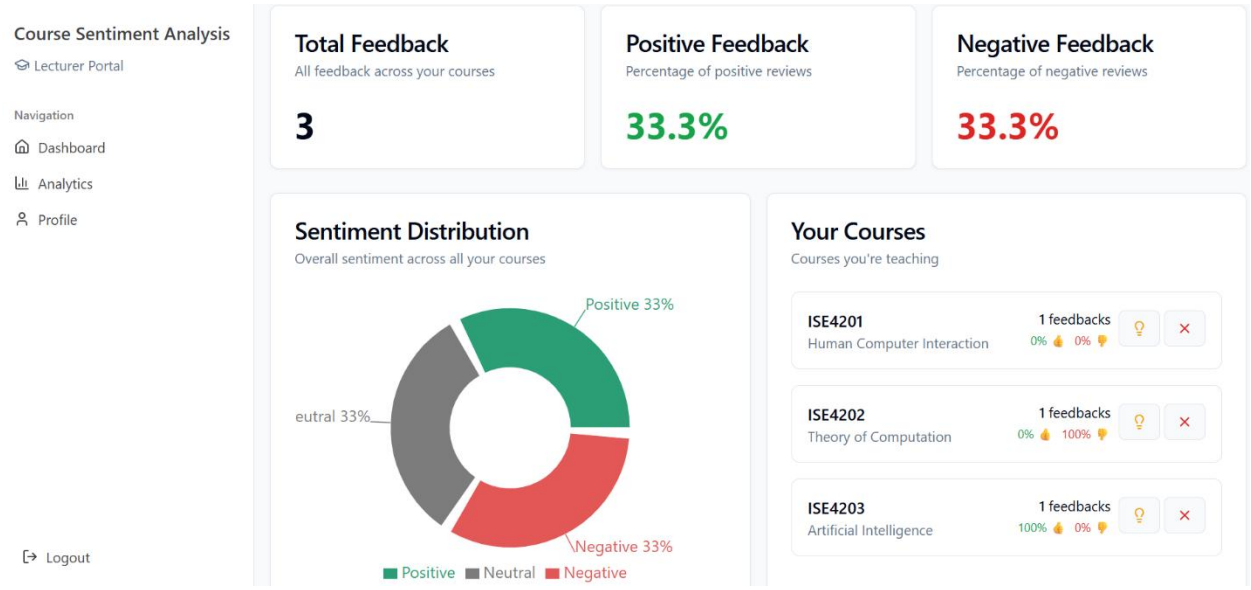


Fig 18.0: Overall Sentiment Analysis and Distribution

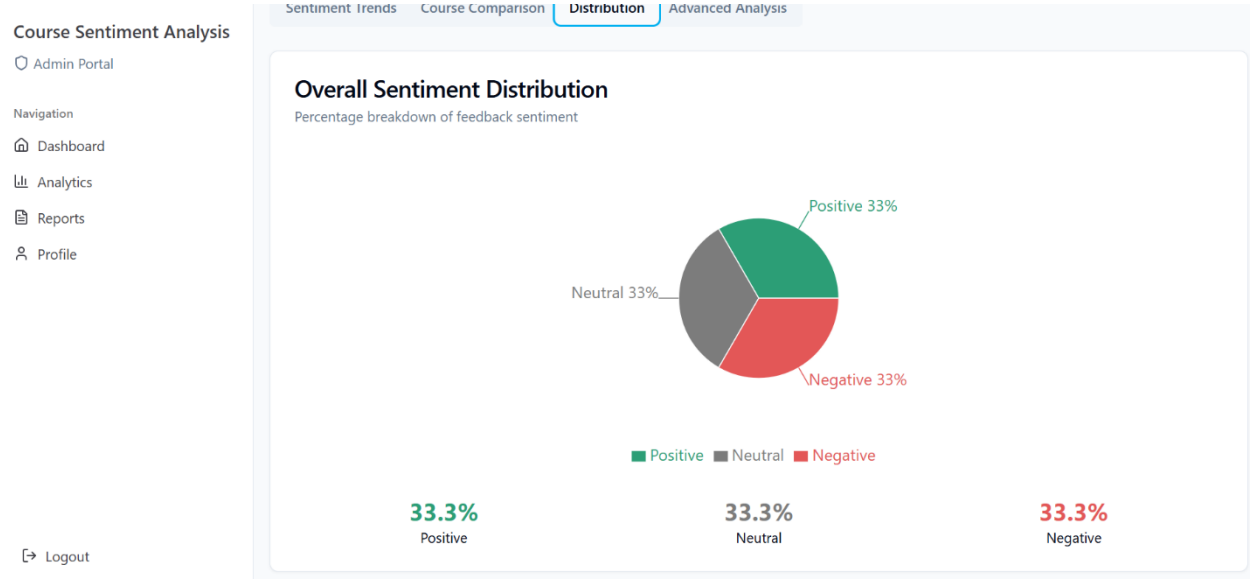


Fig18.1: Sentiment Distribution

fig19.0: System Suggestion

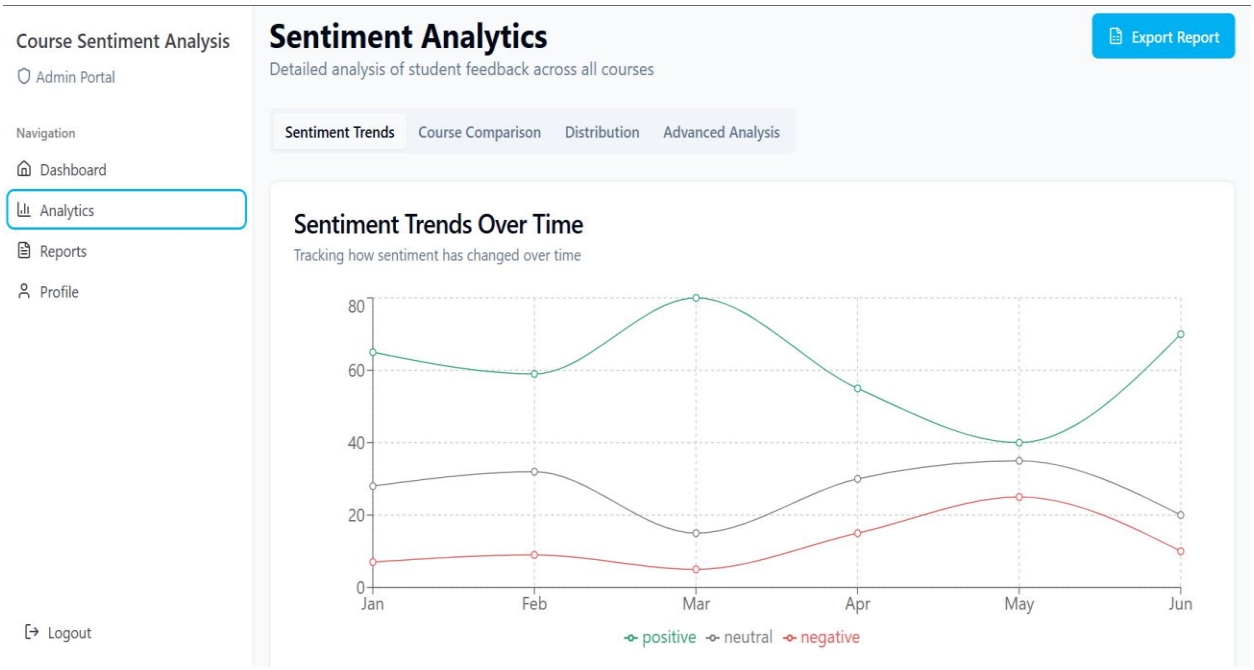


Fig19.1: Sentiment trends over time

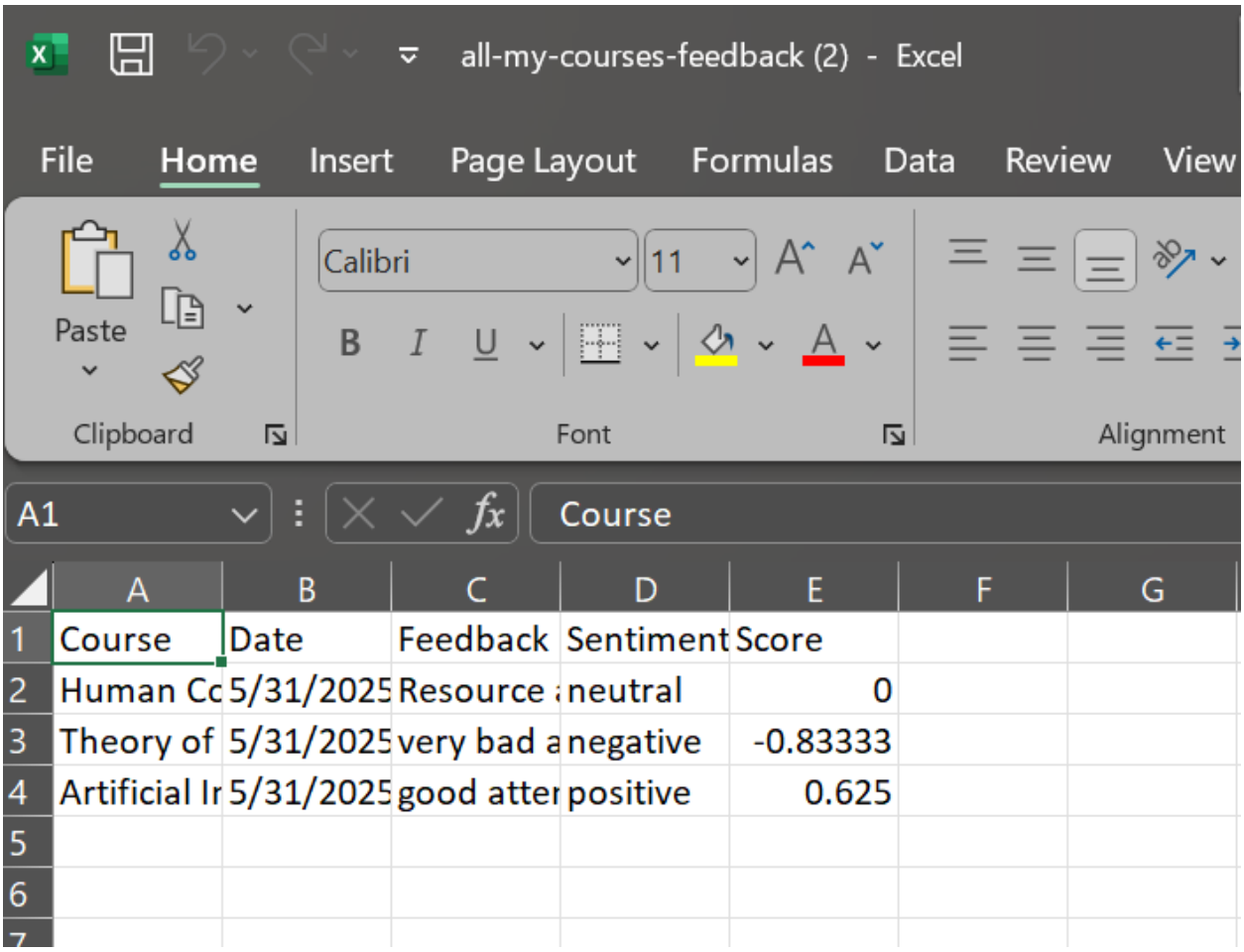


Fig19.2: Exported Excel

4.6 Conclusion

The designing of the proposed system was carried out during this phase. The main Inputs, processes and outputs of the proposed system were looked to in detail and were designed with the main purpose of meeting the functional and non-functional requirements stated in chapter 3.

Chapter 5: Implementation and testing

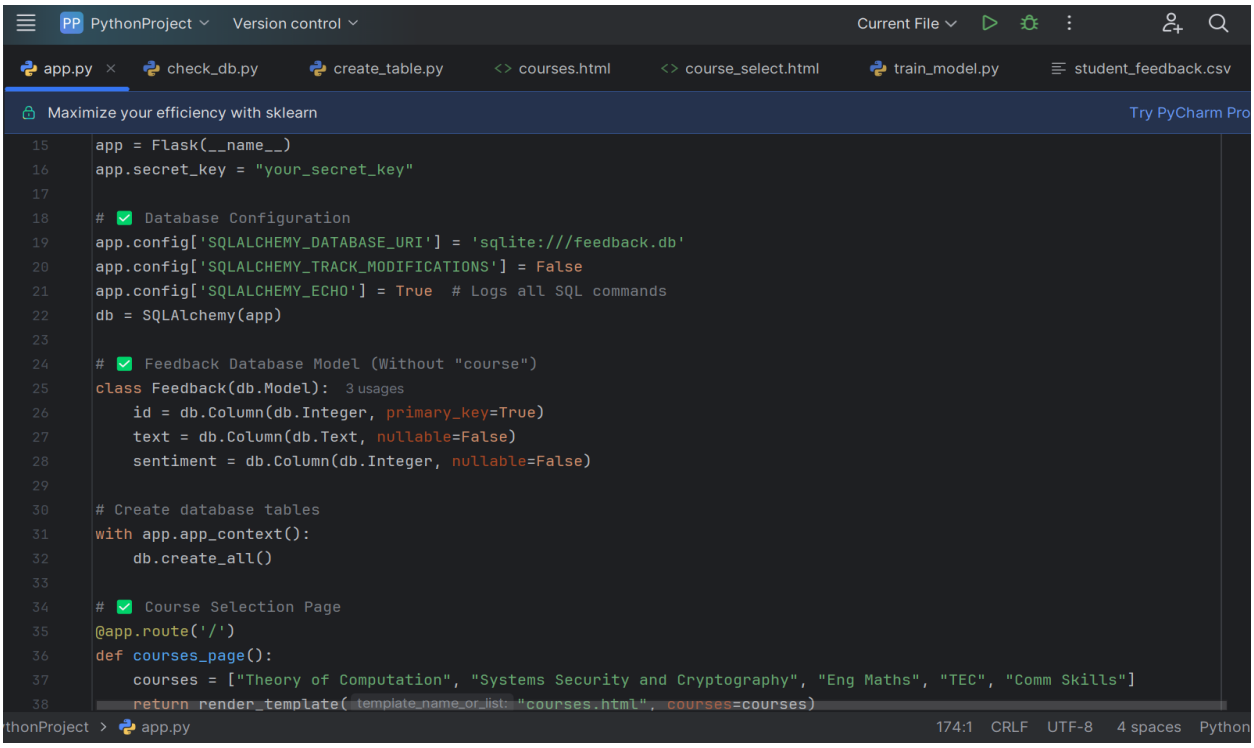
5.0 Introduction

This chapter presents the detailed implementation and testing process of the sentiment analysis system designed to evaluate student feedback in educational settings. The primary goal of the system is to gather qualitative feedback from students, analyse the sentiment expressed, and provide educators with visual insights to improve learning experiences.

The implementation phase outlines how the system components were developed, including the user interface for feedback submission, the sentiment analysis logic using natural language processing, and the data visualization module for presenting results. It also highlights the technologies and tools used throughout the development cycle.

Following implementation, rigorous testing was conducted to ensure that each component of the system performs accurately and reliably. This includes unit testing of individual modules, integration testing of interconnected components, user acceptance testing with real users, and performance testing to assess system behaviour under load.

5.0.1 Sample of real Code(*fig20.0-22.0*)



app.py × check_db.py create_table.py courses.html course_select.html train_model.py

Maximize your efficiency with sklearn

```python
def select_course():
    selected_course = request.form.get("course")
    return redirect(url_for( endpoint: "sentiment_analysis", course=selected_course))

@app.route('/sentiment_analysis')
def sentiment_analysis():
    selected_course = request.args.get("course", "Unknown Course")
    return render_template( template_name_or_list: "index.html", course=selected_course)

# ✅ Store Feedback in Database
@app.route( rule: '/submit_feedback', methods=['POST'])
def submit_feedback():
    feedback_text = request.form['feedback']
    print("Received feedback:", feedback_text)
    cleaned_text = re.sub( pattern: r'[^\w\s]', repl: '', feedback_text).lower().strip()

    if cleaned_text:
        model = joblib.load("sentiment_model.pkl")
        vectorizer = joblib.load("tfidf_vectorizer.pkl")
        transformed_text = vectorizer.transform([cleaned_text])
        sentiment_label = model.predict(transformed_text)[0]

        # ✅ Save feedback (No course field)
        new_feedback = Feedback(text=feedback_text, sentiment=int(sentiment_label))
```

b.py create_table.py courses.html course_select.html train_model.py × student_feedback.csv style.css index.html

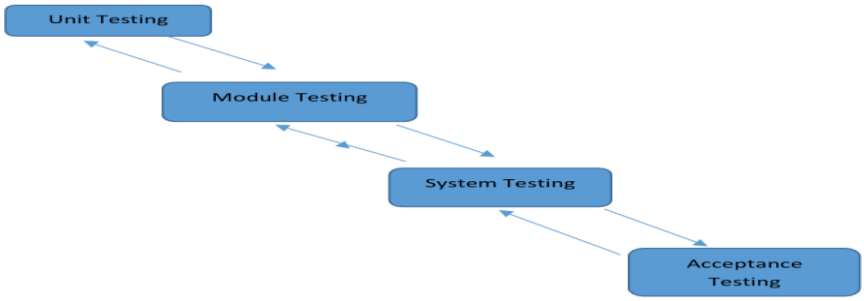
Maximize your efficiency with sklearn Try PyCharm Professional Dismiss

```python
data['cleaned_feedback'] = data['feedback'].apply(clean_text)

# Convert sentiment to numerical values (0 = Negative, 1 = Neutral, 2 = Positive)
sentiment_mapping = {"negative": 0, "neutral": 1, "positive": 2}
data['sentiment'] = data['sentiment'].map(sentiment_mapping)

# Split dataset
X_train, X_test, y_train, y_test = train_test_split( *arrays: data['cleaned_feedback'], data['sentiment'], test_size=0.2, random_state=42)

# Convert text into TF-IDF features
vectorizer = TfidfVectorizer(max_features=5000)
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)

# Train model
model = LogisticRegression()
model.fit(X_train_tfidf, y_train)

# Save model and vectorizer
joblib.dump(model, filename: "sentiment_model.pkl")
joblib.dump(vectorizer, filename: "tfidf_vectorizer.pkl")

print("Model trained and saved successfully!")
```

ythonProject › train_model.py 52:1 CRLF UTF-8 4 spaces Python 3.13 (PythonProject)

5.1 Software Testing

Software testing is a critical phase in the software development lifecycle that ensures the system performs as expected and meets its intended requirements. For the sentiment analysis system, several testing methodologies were employed to verify the accuracy, reliability, and usability of the application.



5.1.1 Unit Testing

Unit testing involved testing individual components of the application in isolation. Each function, such as feedback submission, sentiment classification, and database insertion, was tested separately to ensure it handled both valid and invalid input appropriately. For example, tests were conducted to check how the system responded to empty feedback, excessively long text, or unsupported characters.

Benefits

- Detects bugs early at the code level
- Ensures individual components work as intended

Challenges

- Time-consuming to write for large codebases
- May miss integration issues

5.1.2 Integration Testing

Integration testing was performed to ensure that the individual modules of the system interacted correctly. This included testing the flow of data from the user interface to the database, and then to the sentiment analysis module. Test cases verified that feedback entered by a user was successfully saved in the database and correctly retrieved for sentiment analysis and visualization.

Benefits

- Verifies that modules communicate correctly
- Helps identify interface mismatches and data flow issues

Challenges

- Complex setup with dependencies
- Difficult to isolate faults when tests fail

5.1.3 System Testing

System testing involved evaluating the complete and integrated system to ensure that it met the specified requirements. This included verifying that:

- Feedback could be submitted and stored reliably.
- The sentiment analysis results reflected the correct sentiment.
- The sentiment distribution was accurately displayed in the bar chart.
- The success pop-up message appeared after each submission.

Benefits

- Validates the complete system against requirements
- Ensures the application works as a whole

Challenges

- Requires complete and stable system
- Can be resource-intensive and time-consuming

5.1.4 User Acceptance Testing (UAT)

User acceptance testing was conducted with a group of target users (students and educators) to assess the usability and functionality of the application in real-world conditions. Feedback from users was collected to identify areas for improvement, such as layout alignment, course selection visibility, and responsiveness of the "Analyse Sentiment" feature.

Benefits

- Confirms the system meets user needs
- Provides feedback on usability and user experience

Challenges

- Scheduling and involving end-users can be difficult
- Users may lack technical knowledge for thorough testing

5.1.5 Performance Testing

Performance testing was carried out to ensure the system remained responsive under multiple user inputs. Tests were done to simulate the submission of large volumes of feedback to examine how well the application handled load and how quickly it generated sentiment results.

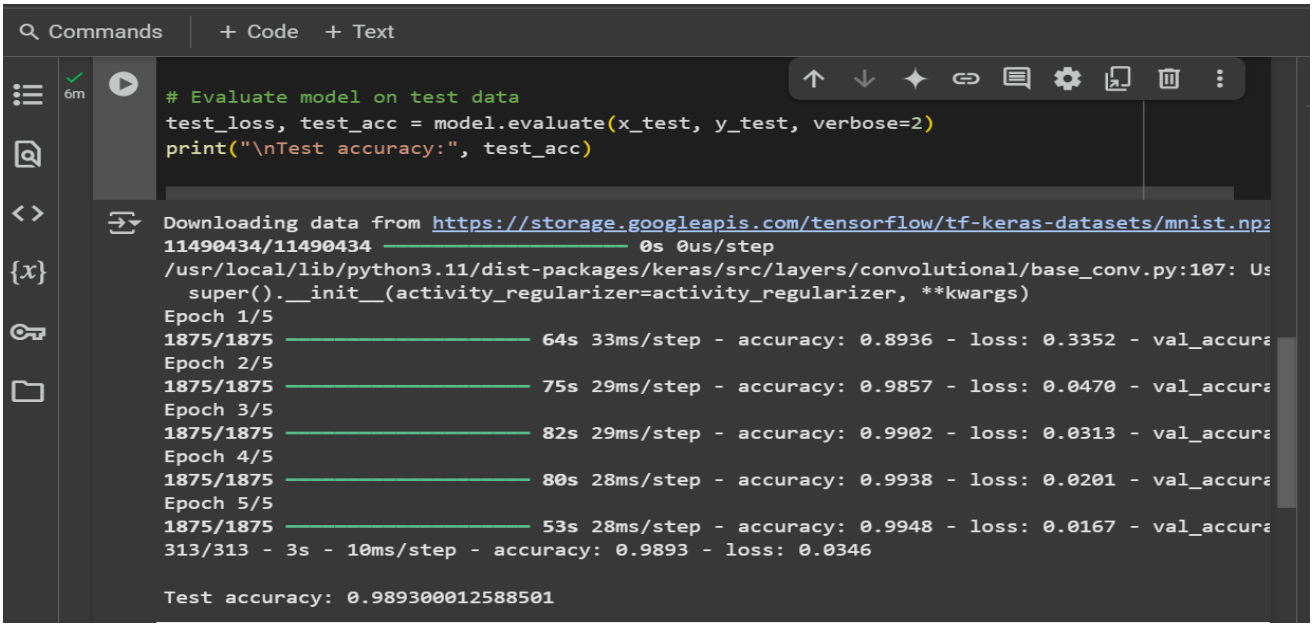


fig 23.0

Conclusion

Through comprehensive testing, the system demonstrated stability, accuracy in sentiment classification, and a user-friendly interface. All identified bugs and issues were documented and addressed, resulting in a polished final version ready for deployment.

Through a structured development and testing approach, the project ensures a robust and user-friendly application capable of delivering real-time insights to support student engagement and academic improvement.

Chapter 6: CONCLUSIONS AND RECOMMENDATIONS

6.1 Introduction

Chapter 6 presents the final reflections on the Sentiment Analysis System developed for educational settings. In this chapter, the key findings from the implementation and testing phases are summarized, and actionable recommendations are made to enhance the system's functionality, performance, and user experience. This section also highlights the conclusions drawn from the development process and provides a roadmap for future improvements and potential expansions of the system.

The goal of this chapter is to provide a comprehensive overview of how the system performed in real-world testing, identify areas for enhancement, and outline directions for future work. The recommendations focus on optimizing system features, addressing any challenges encountered during the development, and incorporating feedback from users. The conclusions offer insights into the success of the project, the impact it may have on improving student feedback and sentiment analysis in educational contexts, and potential areas for growth and scaling.

6.2 Results and Summary

The development and testing phases of the Sentiment Analysis System have led to valuable insights and have demonstrated the effectiveness of the solution in addressing the needs of educators for analysing student feedback. Below is a summary of the key results and findings from the implementation and testing phases:

1. **Successful Sentiment Analysis Integration**: The system's core functionality, which is the analysis of student feedback and the display of sentiment distributions, has been successfully implemented. The sentiment analysis process effectively categorizes feedback into positive, neutral, and negative sentiments, providing educators with a quick and clear overview of students' emotional states and opinions.
2. **User Interface and Experience**: The interface was designed with ease of use in mind. Feedback submission is straightforward, and the results are presented clearly with visual aids, such as bar charts, for sentiment distribution. Users (educators) have provided positive feedback regarding the simplicity and clarity of the interface.

3. **Feedback Submission and System Interaction**: The system allows students to submit their feedback easily. The backend processes and stores the feedback, and once a sufficient amount of feedback is gathered, sentiment analysis can be triggered. This feature was tested and found to be robust, delivering accurate results based on the aggregated data.
4. **Export Feature**: The export functionality, which allows feedback and sentiment results to be exported as Excel files, was successfully implemented and tested. This feature ensures that educators can further analyse and report findings in offline environments or share them with stakeholders in an accessible format.
5. **Authentication for Admin Access**: The authentication feature, which provides admin-only access to view and analyse feedback, was successfully incorporated into the system. This feature ensures data security and allows only authorized users to access the analysis results, making the system suitable for academic environments with privacy and confidentiality concerns.
6. **Testing Outcomes**: Extensive testing was conducted, including unit tests, integration tests, and user acceptance tests (UAT). The system performed well under all scenarios, with no major issues found during the testing phases. However, some minor UI adjustments and optimizations were identified for future iterations.

Summary: Overall, the Sentiment Analysis System achieved its main objectives, including the accurate analysis of student feedback, providing actionable insights for educators. The system is functional, user-friendly, and secure. Moving forward, there are areas for improvement and additional features to explore, but the current version of the system successfully meets its intended goals.

6.3 Recommendations

Based on the results of the implementation, testing, and feedback gathered throughout the development process, the following recommendations are made to further enhance the functionality, user experience, and scalability of the Sentiment Analysis System:

1. **Improved Sentiment Analysis Algorithms**:
    - **Expand Sentiment Categories**: Currently, the system classifies feedback into three categories: positive, neutral, and negative. However, incorporating additional categories, such as “mixed” or “constructive,” could provide deeper insights into the nature of feedback. This could help educators understand not only the general sentiment but also the tone and underlying context of the responses.
    - **Refine Sentiment Accuracy**: It is recommended to explore more advanced Natural Language Processing (NLP) techniques, such as transformers (e.g., BERT, GPT), to improve the accuracy and granularity of sentiment analysis. This would enable better handling of complex feedback and potentially improve classification accuracy.
2. **Personalized Insights and Suggestions**:

    Recommendation Engine: In addition to sentiment analysis, integrating a recommendation engine that offers specific advice to educators based on the analysed feedback could be a valuable addition. For example, the system could suggest teaching strategies, course adjustments, or topics for further review based on common patterns in feedback.
3. **User Interface (UI) Improvements**:
    - **Mobile Compatibility**: Ensure that the system is fully optimized for mobile devices. With the increasing use of smartphones and tablets in educational settings, a responsive design that adapts well to various screen sizes would improve accessibility for both students and educators.

4. **Scalability and Performance Enhancements**:
    - **Handling Large Data Sets**: As the system scales to handle more students and feedback, it is crucial to optimize the database and backend infrastructure for better performance. Consider using more efficient data storage and retrieval techniques, such as indexing, to handle larger volumes of data without affecting performance.
    - **Cloud Integration**: For scalability, it is recommended to integrate the system with cloud services, allowing for easier scaling of storage and computing power as the user base grows. This would also enhance data redundancy and backup capabilities.
5. **Extended Data Export Features**:
    - **Support for Multiple Formats**: While exporting data to Excel is helpful, adding support for additional file formats like CSV, PDF, or even JSON could be beneficial. This would offer more flexibility for users who need to process or present the data in different ways.
    - **Scheduled Reports**: Implement a feature that allows educators to schedule automatic export of sentiment analysis reports at regular intervals (e.g., weekly, monthly). This would reduce the manual effort of generating and exporting reports on demand.
6. **Enhanced Security and Privacy**:

    Data Encryption: Given the sensitive nature of student feedback, implementing end-to-end encryption for feedback data and analysis results would further protect user privacy and ensure that the data remains secure.

    Access Controls and Role-Based Permissions: Extend the authentication feature to support multiple roles (e.g., admin, teacher, student) with varying levels of access to the system's features. This would allow for more granular control over who can access and modify data.

The ongoing development of this system will ensure its continued relevance and value in educational settings, supporting both students and educators in creating more effective and engaging learning environments.

6.4 Future Works

While the current implementation of the Sentiment Analysis System has achieved its primary goals, there are several areas for future development that could further enhance the system's capabilities and broaden its applications. The following outlines key areas for potential future work:

1. **Integration with Learning Management Systems (LMS)**:
    - **Automatic Data Collection**: Integrating the sentiment analysis system with popular Learning Management Systems (LMS) such as Moodle, Canvas, or Blackboard could automate the data collection process, making it easier to gather feedback from students. By directly accessing course materials, discussions, and assessments, the system could provide more comprehensive insights into the learning experience without manual input from students.
    - **Real-Time Feedback**: By integrating real-time feedback features with LMS platforms, the system could offer immediate insights into student sentiments after class sessions or assignments, allowing educators to address concerns promptly.
2. **Sentiment Analysis for Non-Textual Data**:
    - **Voice and Audio Feedback**: In addition to analysing text-based feedback, the system could be expanded to analyse voice recordings. Using speech-to-text and audio sentiment analysis, the system could process spoken feedback from students, such as recorded interviews or discussion board posts.
    - **Multimodal Sentiment Analysis**: Combining multiple data sources such as text, voice, and video could provide a more holistic view of student sentiment. For instance, analysing facial expressions or body language in video recordings could complement text feedback to give more accurate and nuanced sentiment insights.

3. **Advanced Data Visualization and Analysis**:
    - **Interactive Visualizations**: Enhancing the visual representation of sentiment data through interactive graphs, 3D plots, and heatmaps would allow educators to explore the data in more depth. Using tools like D3.js or Plotly for dynamic, user-controlled visualizations would improve the accessibility and understanding of the data.
    - **Sentiment Trend Analysis**: Introducing advanced trend analysis over time could help identify recurring patterns in student feedback across different courses, semesters, or academic years. This could also help forecast potential issues in teaching or course structure early on.
4. **Multi-Language Support**:
    - **Global Reach**: Expanding the sentiment analysis to support multiple languages would make the system more inclusive and useful in global educational settings. This could involve training models in different languages or using existing multilingual models to analyse non-English feedback.
    - **Cross-Cultural Sentiment Interpretation**: Adapting sentiment analysis algorithms to account for cultural differences in language use would help improve the system's accuracy in different regions, as the expression of sentiments may vary across cultures.
5. **Artificial Intelligence for Personalized Learning**:
    - **Adaptive Learning Recommendations**: Leveraging the sentiment analysis system to provide personalized learning recommendations based on student feedback could be a next step. For instance, if negative feedback frequently mentions difficulty with a particular topic, the system could recommend supplementary materials or alternative teaching strategies tailored to student needs.
    - **Predictive Analytics**: By analysing historical sentiment data, predictive models could be developed to forecast student engagement, performance, or satisfaction

in real-time. This could help identify at-risk students early and provide educators with tools to intervene proactively.

6. **Enhanced Reporting and Analytics**:
    - **Customizable Reports**: Future work could include the ability to generate customizable reports based on specific criteria such as course, instructor, or time period. These reports could offer more detailed insights and actionable feedback tailored to educators' needs.
    - **AI-Driven Feedback Summaries**: The system could automatically generate concise, actionable feedback summaries for educators, highlighting key sentiment trends, common student concerns, and areas requiring attention.
7. **Longitudinal Data Analysis**:
    - **Tracking Student Sentiment Over Time**: By allowing longitudinal tracking of student sentiment, the system could generate insights into how students' feelings evolve throughout the course or across multiple courses. This could help instructors improve teaching strategies based on changes in sentiment.
    - **Impact of Interventions**: Educators could use the system to assess the impact of specific teaching interventions, assignments, or changes to the curriculum on student sentiment. This data could be used to refine teaching methods and improve the learning experience.
8. **Enhanced User Experience**:
    - **Gamification of Feedback**: To increase student engagement and encourage more detailed feedback, gamification elements could be introduced. For example, students could earn points or badges for providing feedback or for participating in sentiment-driven activities.
    - **Voice-Activated Feedback Submission**: Allowing students to submit feedback using voice commands could enhance the user experience, especially for mobile or wearable device users. This would make it easier for students to engage with the system in a more natural and convenient way.

By pursuing these avenues for future work, the Sentiment Analysis System can evolve into a more powerful, comprehensive tool that supports educators in making data-driven decisions, fostering a more positive learning environment, and adapting to the diverse needs of students. Each of these directions has the potential to not only enhance the functionality of the system but also significantly broaden its application in the educational landscape.

6.5 Challenges

While the Sentiment Analysis System offers valuable insights into student feedback and provides significant benefits to educators, there are several challenges that must be addressed to ensure its continued success and effectiveness. The following outlines key challenges faced during the development and implementation of the system:

1. **Data Quality and Preprocessing**:
    - **Inaccurate or Inconsistent Feedback**: The system relies on feedback provided by students, which may be subjective, vague, or inconsistent. Some feedback entries may lack clarity, making sentiment analysis more difficult. Additionally, students may sometimes provide feedback in a hurried or incomplete manner, which could affect the quality of the analysis.
    - **Preprocessing Challenges**: Sentiment analysis algorithms require thorough data cleaning and preprocessing, such as removing stop words, handling slang or abbreviations, and correcting spelling errors. This can be a time-consuming process, especially when dealing with large datasets of diverse student feedback.
2. **Sentiment Ambiguity**:
    - **Contextual Interpretation**: Sentiment analysis models often struggle with understanding the context of the feedback. For instance, the phrase "I don't hate this course" may be interpreted as negative due to the word "hate," even though the overall sentiment might be neutral. Distinguishing between sarcasm, irony, and genuine sentiment remains a major challenge in sentiment analysis systems.
    - **Multidimensional Sentiment**: Student feedback often contains mixed emotions or multiple sentiments. For example, a student may express frustration with a

course but simultaneously acknowledge the value of its content. Analysing such mixed sentiment can be challenging, and simplifying it into a single positive, negative, or neutral label may not always capture the complexity of student experiences.

3. **Scalability and Performance**:
    - **Handling Large Data Volumes**: As the system is adopted by more institutions and collects data from a larger number of students, it must be able to process large volumes of feedback efficiently. The performance of sentiment analysis algorithms may degrade as the dataset grows, leading to longer processing times or potential system slowdowns.
    - **Real-Time Analysis**: For the system to provide real-time feedback, it must be capable of processing feedback quickly and delivering insights to educators in a timely manner. Achieving this requires optimizing algorithms and ensuring the infrastructure can handle concurrent requests from many users without significant delays.
4. **Accuracy and Limitations of Sentiment Models**:
    - **Model Limitations**: Sentiment analysis models are not perfect, and their accuracy can vary depending on the quality of the training data and the complexity of the feedback. Models trained on general-purpose datasets may not be as effective in understanding the specific language used by students in educational contexts. Fine-tuning the model for specific use cases is a continuous challenge.
    - **False Positives and Negatives**: The system may occasionally classify neutral or mildly positive feedback as negative or vice versa. This can happen due to nuances in language, phrasing, or word usage. Ensuring the model's accuracy requires constant monitoring and adjustments.
5. **Integration with Other Systems**:
    - **Compatibility with Learning Management Systems (LMS)**: Integrating the sentiment analysis system with existing LMS platforms can be challenging due to

differences in data formats, system architectures, and user interfaces. Smooth integration is crucial for automating the data collection process, but compatibility issues may arise when attempting to connect to diverse platforms.

- **Data Privacy and Security**: The system collects sensitive student data, which must be protected in compliance with privacy laws such as FERPA (Family Educational Rights and Privacy Act) and GDPR (General Data Protection Regulation). Securing the data and ensuring proper consent is obtained for feedback collection can be challenging, particularly when dealing with large volumes of data.

6. **User Adoption and Engagement**:

   - **Encouraging Student Participation**: For the sentiment analysis system to be effective, students must consistently provide feedback. However, students may be reluctant to participate if the process is perceived as time-consuming or cumbersome. Ensuring the feedback process is easy, quick, and anonymous can help increase student engagement.

   - **Educator Trust and Buy-in**: Educators may be sceptical about the accuracy or usefulness of sentiment analysis results. Convincing educators to trust and adopt the system requires demonstrating its value through clear, actionable insights. Furthermore, educators may require additional training to interpret sentiment analysis results effectively and use them to improve their teaching methods.

7. **Ethical Concerns**:

   - **Bias in Sentiment Analysis**: Sentiment analysis models are susceptible to biases present in the training data. If the training data contains biased language or reflects societal stereotypes, the model may inadvertently reinforce these biases, leading to skewed sentiment analysis results. Addressing these biases and ensuring fairness in the system is a key ethical concern.

   - **Potential for Misuse**: There is also a risk that sentiment analysis could be misused to make unfair decisions about students. For example, negative sentiment

could be used as a basis for academic evaluations or disciplinary actions without fully understanding the context behind the feedback. It is important to ensure that sentiment analysis is used responsibly and does not harm students.

In conclusion, while the Sentiment Analysis System has made significant strides in improving the learning experience by providing valuable insights into student feedback, these challenges must be addressed to improve its accuracy, scalability, and user adoption. By continually refining the system, integrating new technologies, and ensuring ethical considerations, the system can evolve into an even more powerful tool for enhancing education.

Bibliography (References)

1. Jurafsky, D., & Martin, J. H. (2020). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (3rd ed.). Pearson.

2. Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

3. Pang, B., & Lee, L. (2008). *Opinion Mining and Sentiment Analysis*. Foundations and Trends in Information Retrieval, 2(1–2), 1-135.

4. Chen, M., Mao, S., & Liu, Y. (2014). *Big Data: A Survey*. Mobile Networks and Applications, 19(2), 171-209.

5. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. A., Kaiser, Ł., & Polosukhin, I. (2017). *Attention is all you need*. In Advances in Neural Information Processing Systems (NeurIPS).

6. Kingma, D. P., & Ba, J. (2014). *Adam: A Method for Stochastic Optimization*. International Conference on Learning Representations (ICLR).

7. He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep Residual Learning for Image Recognition*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

8. Chollet, F. (2017). *Deep Learning with Python*. Manning Publications.

9. Müller, A. C., & Guido, S. (2016). *Introduction to Machine Learning with Python*. O'Reilly Media.

10. Brownlee, J. (2017). *Machine Learning Mastery with Python: Understand Your Data, Create Accurate Models, and Work Projects End-to-End*. Machine Learning Mastery.

11. Microsoft Azure Documentation (2025). *Azure Machine Learning Services*

12. Koller, D., & Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

APPENDIX A: DATA COLLECTION TOOLS

Sample Data Collection Tool: Student Feedback Form

Student Feedback Form Template

---

1. Course Information:

- **Course Name:** [_____]
  (Select the course from the dropdown list)
- **Instructor Name:** [_____]
- **Semester:** [_____]
  (Select the semester)

---

2. Feedback Questions:

- **Overall, how would you rate this course?**
  (Select one option)
  - Excellent
  - Good
  - Fair
  - Poor
  - Very Poor
- **What did you enjoy most about this course?**
  (Please provide a detailed response)
  - [_____]
  - [_____]
- **What aspects of the course could be improved?**
  (Please provide your suggestions)
  - [_____]
  - [_____]

- **Did the instructor effectively explain the concepts?**
  (Select one option)
    - Yes
    - No
    - Somewhat
    - Not Applicable
- **Would you recommend this course to other students?**
  (Select one option)
    - Yes
    - No
    - Maybe

---

3. Open-ended Feedback (Optional):

- **Please share any additional comments or suggestions you have about the course.**
  (Your feedback is important for improving the course)
    - [_____]
    - [_____]
    - [_____]

---

4. Emotional Response:

- **How did you feel about this course overall?**
  (Select one option)
    - Very Happy
    - Satisfied
    - Neutral
    - Dissatisfied
    - Frustrated

APPENDIX B: USER MANUAL

1. Introduction

This User Manual is designed to guide the end-user through the process of using the **Sentiment Analysis System**. The system allows users to submit feedback on courses, analyse the sentiment of the feedback, and export the results in Excel format. It is intended for use by educators, administrators, and students.

2. System Requirements

To use the system, you will need the following:

- **Hardware:**
    - A device with internet access (laptop, desktop, or tablet).
    - A web browser (e.g., Google Chrome, Mozilla Firefox, Microsoft Edge).
- **Software:**
    - The Sentiment Analysis System is a web-based application, so no additional software installation is required beyond a modern web browser.

3. Accessing the System

1. Open your web browser.
2. Enter the URL of the system (provided by your institution or organization).
3. You will be directed to the **Student Feedback Submission** page.

4. Using the System

4.1 Submitting Feedback

1. **Navigate to the "Submit Feedback" section:**
    - You will see a text area where you can enter your feedback.
2. **Enter Feedback:**
    - Type your feedback regarding the course in the provided text area.
3. **Submit Feedback:**
    - After entering your feedback, click the **"Submit Feedback"** button.
    - A success pop-up message will appear to confirm that your feedback has been submitted.

4.2 Analysing Sentiment

1. **Navigate to the "Analyse Sentiment" button:**
    - After submitting feedback, the **"Analyse Sentiment"** button will be displayed.
2. **Click the "Analyse Sentiment" button:**
    - Once clicked, the system will process all the feedback data and display the sentiment analysis results, including the percentage breakdown of positive, neutral, and negative feedback.
3. **Sentiment Distribution:**
    - A bar chart will display the sentiment distribution of all submitted feedback.

4.3 Exporting Data

1. **Navigate to the "Export as Excel" button:**
    - You will find the **"Export as Excel"** button on the page.
2. **Click the "Export as Excel" button:**
    - Clicking this button will generate an Excel file that contains the feedback data along with the sentiment analysis results. The file will be available for download.

4.4 Logging Out (Admin-Only Access)

1. **Admin Logout:**
    - If you are logged in as an admin, you will see the **"Logout"** button in the bottom right corner of the screen.
2. **Click the "Logout" button:**
    - This will log you out of the system and return you to the login page (if applicable).

5. System Navigation

- **Navigation Menu:**
    - The main navigation elements are located at the top of the page, where you can access the following:
        - **Submit Feedback**
        - **Analyse Sentiment**

        - **Export as Excel**
        - **Logout**
- **Data Table:**
    - The system also includes a table that displays all the feedback submissions for easy reference.

6. Troubleshooting

- **Issue: I cannot submit my feedback.**
    - **Solution:** Ensure that the feedback text area is not empty. The system requires feedback to be entered before submission.
- **Issue: Sentiment Analysis does not show up after clicking the "Analyse Sentiment" button.**
    - **Solution:** Verify that you have submitted feedback first. The sentiment analysis will only be available once feedback has been entered and stored.
- **Issue: The "Export as Excel" button is not working.**
    - **Solution:** Make sure that sentiment analysis has been performed. The export feature is available only after sentiment analysis is complete.

7. Frequently Asked Questions (FAQs)

Q: Can I submit feedback multiple times?
A: Yes, you can submit feedback multiple times. Each submission will be analysed individually in the sentiment analysis.

Q: Can I see feedback from other students?
A: No, the feedback system is designed to keep submissions private and only accessible to the administrators for analysis.

Q: How is sentiment analysis performed?
A: Sentiment analysis is performed using a machine learning model that categorizes feedback into positive, neutral, or negative sentiment based on the content.

8. Contact and Support

If you encounter any issues or have questions about the system, please contact the system administrator or support team at:

- **Email:** support@coursereview.co.zw
- **Phone:** +263 78 047 7611

APPENDIX C: SAMPLE CODE

app.py | check_db.py | create_table.py | courses.html | course_select.html | train_model.py

Maximize your efficiency with sklearn

```
42  def select_course():
43      selected_course = request.form.get("course")
44      return redirect(url_for( endpoint: "sentiment_analysis", course=selected_course))
45
46  @app.route('/sentiment_analysis')
47  def sentiment_analysis():
48      selected_course = request.args.get("course", "Unknown Course")
49      return render_template( template_name_or_list: "index.html", course=selected_course)
50
51  # ✅ Store Feedback in Database
52  @app.route( rule: '/submit_feedback', methods=['POST'])
53  def submit_feedback():
54      feedback_text = request.form['feedback']
55      print("Received feedback:", feedback_text)
56      cleaned_text = re.sub( pattern: r'[^\w\s]', repl: '', feedback_text).lower().strip()
57
58      if cleaned_text:
59          model = joblib.load("sentiment_model.pkl")
60          vectorizer = joblib.load("tfidf_vectorizer.pkl")
61          transformed_text = vectorizer.transform([cleaned_text])
62          sentiment_label = model.predict(transformed_text)[0]
63
64          # ✅ Save feedback (No course field)
65          new_feedback = Feedback(text=feedback_text, sentiment=int(sentiment_label))
```

fig 24.0: Analysis Code Sample