**ADAPTIVE ONLINE LEARNING PLATFORM TO ENHANCE PRIMARY EDUCATION**

Inuri Chamathka Hapuarachchi

IT20617264

BSc (Hons) in Information Technology Specializing in Data Science

Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

April 2024

**ADAPTIVE ONLINE LEARNING PLATFORM TO ENHANCE PRIMARY EDUCATION**

Inuri Chamathka Hapuarachchi

IT20617264

Dissertation submitted in partial fulfillment of the requirements for the Bachelor of Science (Hons) in Information Technology Specializing in Data Science

Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

April 2024

# **DECLARATION**

I declare that this is my own work and this dissertation1 does not incorporate without acknowledgment any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgment is made in the text.

Also, I hereby grant to Sri Lanka Institute of Information Technology, the nonexclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

A close-up of a signature

Description automatically generated

Signature: Date: 05/04/2024

The above candidate has carried out research for the bachelor’s degree Dissertation under my supervision.



Signature of the supervisor: Date: 05/04/2024

(Mr. Samadhi Rathnayake)

# **ABSTRACT**

Within the educational landscape, it is widely acknowledged that students' emotional well-being significantly influences their learning journey and academic achievements. However, traditional educational systems often neglect the importance of addressing emotional needs, resulting in potential gaps in student support and development. This thesis proposes an innovative solution to this challenge through the development of an adaptive online learning platform tailored for primary students. The primary goal of this platform is to identify and cater to students' emotional weaknesses by harnessing cutting-edge technologies such as BERT (Bidirectional Encoder Representations from Transformers) and Reinforcement Learning (RL) algorithms. By incorporating these technologies, the platform can analyze student interactions, detect emotional cues embedded in their responses, and generate personalized recommendations designed to meet each student's emotional requirements. A notable feature of this platform is its ability to deliver timely feedback and guidance to educators. By leveraging insights derived from BERT's emotion analysis and RL's decision-making capabilities, the platform offers actionable guidance to teachers on how to effectively support and manage students exhibiting diverse emotional states. This individualized approach aims to cultivate a more nurturing and empathetic learning environment, fostering students' emotional resilience and academic progress. Through rigorous testing, including user trials and feedback assessment, this thesis showcases the effectiveness and significance of integrating emotional intelligence into educational technology. The findings underscore the transformative potential of BERT and RL algorithms in redefining how educational systems identify, respond to, and nurture students' emotional needs, paving the way for enhanced learning experiences and outcomes.

***Key Words: Emotional well-being, Primary students, BERT, Reinforcement Learning (RL) algorithms, Personalized advice***

# **ACKNOWLEDGMENT**

I extend my heartfelt gratitude to Mr. Samadhi for his exceptional guidance and mentorship as my supervisor throughout this research endeavor. His profound expertise and insightful direction have played a pivotal role in shaping the trajectory and outcomes of this study.

I am deeply appreciative of Dr. Dharshana for his unwavering support and invaluable contributions as my co-supervisor. His astute guidance and encouragement have been indispensable in navigating the complexities inherent in this research project.

I would like to express my sincere thanks to Dr. Junius and Dr. Lakmini from the esteemed research panel for their valuable insights and recommendations. Their constructive feedback and suggestions for enhancing the model have significantly contributed to its refinement and advancement. Furthermore, I would like to thank our lecturer in charge, Dr. Jayantha Amararachchi, and the CDAP team for their assistance in ensuring the project's success.

Special appreciation is extended to Dr. Indika Wijerathna, a distinguished psychologist, whose collaboration and expertise were instrumental in understanding and delineating the emotions to be captured from primary students. Additionally, I am grateful to Mrs. Inoka Ekanayake, an esteemed English teacher, for her valuable input in crafting questions that effectively elicit a diverse range of emotions from students.

I am profoundly grateful for the support, guidance, and expertise provided by these individuals throughout this research journey. Their contributions have enriched the depth and quality of this work, and I am privileged to have had the opportunity to collaborate with such esteemed professionals.

**TABLE OF CONTENTS**

[**DECLARATION** i](#_Toc168927700)

[**ABSTRACT** ii](#_Toc168927701)

[**ACKNOWLEDGMENT** iii](#_Toc168927702)

[**LIST OF FIGURES** v](#_Toc168927703)

[**LIST OF TABLES** viii](#_Toc168927704)

[**LIST OF ABBREVIATIONS** viii](#_Toc168927705)

[**1.** **INTRODUCTION** 1](#_Toc168927706)

[**1.1** **Background Literature** 2](#_Toc168927707)

[**1.2** **Research Gap** 6](#_Toc168927708)

[**1.3** **Research Problem** 8](#_Toc168927709)

[**1.4** **Research Objectives** 10](#_Toc168927710)

[**1.4.1 Main Objective** 10](#_Toc168927711)

[**1.4.2 Specific Objectives** 11](#_Toc168927712)

[**2.** **METHODOLOGY** 14](#_Toc168927713)

[**2.1** **Methodology** 14](#_Toc168927714)

[**2.1.1 Requirement Gathering** 14](#_Toc168927715)

[**2.1.2. Dataset Description** 15](#_Toc168927716)

[**2.1.3. System Architecture** 15](#_Toc168927717)

[**2.1.4. Component Architecture** 16](#_Toc168927718)

[**2.1.5. Tools and Libraries** 17](#_Toc168927719)

[**2.1.6. Model Architecture** 17](#_Toc168927720)

[**2.2. Commercialization Aspects of the Product** 32](#_Toc168927721)

[**2.2.1. Target Market** 32](#_Toc168927722)

[**2.2.2. Revenue Streams** 33](#_Toc168927723)

[**2.2.3. Marketing Approach** 33](#_Toc168927724)

[**2.3** **Testing and Implementation** 34](#_Toc168927725)

[**2.3.1. Functional Requirements** 34](#_Toc168927726)

[**2.3.2. Non-Functional Requirements** 35](#_Toc168927727)

[**2.3.3. Backend Implementation** 36](#_Toc168927728)

[**2.3.4. Backend Testing** 49](#_Toc168927729)

[**3.** **RESULTS AND DISCUSSION** 54](#_Toc168927730)

[**3.1.** **BERT Model** 54](#_Toc168927731)

[**3.2.** **RL Model** 56](#_Toc168927732)

[**3.3.** **Research Findings** 57](#_Toc168927733)

[**3.4.** **Discussion** 58](#_Toc168927734)

[**3.4.1.** **Model Performance and Comparative Analysis** 58](#_Toc168927735)

[**3.4.2.** **Practical Implications and Applications** 59](#_Toc168927736)

[**3.4.3.** **Limitations and Future Directions** 59](#_Toc168927737)

[**3.4.4.** **Ethical Considerations** 60](#_Toc168927738)

[**3.5.** **Student Contribution** 60](#_Toc168927739)

[**4.** **CONCLUSIONS** 61](#_Toc168927740)

[**5.** **REFERENCES** 63](#_Toc168927741)

# **LIST OF FIGURES**

[Figure 1: Overall System Architecture 16](#_Toc168927742)

[Figure 2: Emotional Weaknesses Addressing Component 17](#_Toc168927743)

[Figure 3: Importing Libraries for BERT Model 18](#_Toc168927744)

[Figure 4: Mounting to Google Drive 18](#_Toc168927745)

[Figure 5: Reading CSV files 18](#_Toc168927746)

[Figure 6: Label Encoding 19](#_Toc168927747)

[Figure 7: Naive Bayes Baseline Model 19](#_Toc168927748)

[Figure 8: Loading Pre-trained Model and Tokenizer 20](#_Toc168927749)

[Figure 9: Tokenizing Sentences 20](#_Toc168927750)

[Figure 10: Converting Tokenized Data into Tensors 20](#_Toc168927751)

[Figure 11: Creating Data Loaders 21](#_Toc168927752)

[Figure 12: Values for Hyperparameter Tuning 21](#_Toc168927753)

[Figure 13: Setting up Device 21](#_Toc168927754)

[Figure 14: Initializing Best accuracy and Hyperparameters 21](#_Toc168927755)

[Figure 15: Evaluation Function 22](#_Toc168927756)

[Figure 16: Setting up Scheduler and Optimizer 22](#_Toc168927757)

[Figure 17: BERT Training Loop 23](#_Toc168927758)

[Figure 18: Evaluating the Model on Test Data 23](#_Toc168927759)

[Figure 19: Fine Tuning BERT Model with Optimized Hyperparameters 23](#_Toc168927760)

[Figure 20: Training the BERT Model and Evaluate on Validation Dataset 24](#_Toc168927761)

[Figure 21: Evaluate the Model on Test Datasets 24](#_Toc168927762)

[Figure 22: Prediction for BERT Model 25](#_Toc168927763)

[Figure 23: Plotting Confussion Matrix 26](#_Toc168927764)

[Figure 24: Generating Classification Report 26](#_Toc168927765)

[Figure 25: Saving the BERT Model in the Drive 26](#_Toc168927766)

[Figure 26: Importing Necessary Libraries for Advice Generation Part 27](#_Toc168927767)

[Figure 27: Fetching OpenAI API Key 27](#_Toc168927768)

[Figure 28: Defining State and Action Space 27](#_Toc168927769)

[Figure 29: Initializing Q-Table 28](#_Toc168927770)

[Figure 30: Setting Hyperparameters 28](#_Toc168927771)

[Figure 31: Defining Policy Function 28](#_Toc168927772)

[Figure 32: Defining Q-Learning Algorithm 28](#_Toc168927773)

[Figure 33: Preprocessing Input Data 29](#_Toc168927774)

[Figure 34: Defining function to access LLM 30](#_Toc168927775)

[Figure 35: Specifying Training Parameters 30](#_Toc168927776)

[Figure 36: RL Agent Training Loop 30](#_Toc168927777)

[Figure 37: Updating Q-Table based on Feedback recieved 30](#_Toc168927778)

[Figure 38: Checking Updated Q-Table 31](#_Toc168927779)

[Figure 39: Defining Evaluation Dataset 31](#_Toc168927780)

[Figure 40: Getting A Random Sample from Evaluation Dataset 31](#_Toc168927781)

[Figure 41: Evaluating Agent on Sample Evaluation Data 32](#_Toc168927782)

[Figure 42: RL Agent Evaluation 32](#_Toc168927783)

[Figure 43: Saving The Q-Table in The Drive 32](#_Toc168927784)

[Figure 44: Importing Libraries 37](#_Toc168927785)

[Figure 45: ConfigParser Initialization 37](#_Toc168927786)

[Figure 46: MongoDB Initialization 37](#_Toc168927787)

[Figure 47: Flask App Initialization 38](#_Toc168927788)

[Figure 48: '/predict' Route 39](#_Toc168927789)

[Figure 49: ‘/api/reports’ Route 39](#_Toc168927790)

[Figure 50: App Execution 40](#_Toc168927791)

[Figure 51: ‘get\_responses\_with\_highest\_emotions’ Function 40](#_Toc168927792)

[Figure 52: Generate Advice Function 41](#_Toc168927793)

[Figure 53: 'prompt\_model’ Function 42](#_Toc168927794)

[Figure 54: Evaluation of Emotions for Sentence Format Answers 43](#_Toc168927795)

[Figure 55: Evaluation of Emotions for MCQ & Picture Format Answers 44](#_Toc168927796)

[Figure 56: Evaluate Agent Function 45](#_Toc168927797)

[Figure 57: Calling generate\_advice Function 45](#_Toc168927798)

[Figure 58: Calculating Emotion Counts 46](#_Toc168927799)

[Figure 59: Importing Libraries and Packages to Build the Frontend 46](#_Toc168927800)

[Figure 60: useState Hooks 47](#_Toc168927801)

[Figure 61: Handle Submit Function 47](#_Toc168927802)

[Figure 62: getAdvice Function 47](#_Toc168927803)

[Figure 63: AdviceModal 48](#_Toc168927804)

[Figure 64: AdviceModal Function 48](#_Toc168927805)

[Figure 65: How AdviceModal mapped with Advice Data 48](#_Toc168927806)

[Figure 66: Using Postman to Test Backend 49](#_Toc168927807)

[Figure 67: Home Page of The Application 49](#_Toc168927808)

[Figure 68: Quiz Home Page for Emotion Analysis 50](#_Toc168927809)

[Figure 69: Quiz Interface 50](#_Toc168927810)

[Figure 70: Answer Options in MCQ, Sentence & Picture Formats 51](#_Toc168927811)

[Figure 71: Results Page Shown to The Children 51](#_Toc168927812)

[Figure 72: Answer Arrays 52](#_Toc168927813)

[Figure 73: Database Structure 52](#_Toc168927814)

[Figure 74: Reports Generated for Students 53](#_Toc168927815)

[Figure 75: Detailed Report with Personalized Advices 53](#_Toc168927816)

[Figure 76: Classification Report of Naive Bayes Model 54](#_Toc168927817)

[Figure 77: Confusion Matrix of Naive Bayes Model 55](#_Toc168927818)

[Figure 78: Validation Accuracy of BERT Model 55](#_Toc168927819)

[Figure 79: Classification Report of BERT Model 55](#_Toc168927820)

[Figure 80: Confusion Matrix of BERT Model 56](#_Toc168927821)

[Figure 81: Training RL Agent 57](#_Toc168927822)

[Figure 82: Updated Q-Table 57](#_Toc168927823)

# **LIST OF TABLES**

[Table 1: Research Gap 18](#_Toc163497880)

# **LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **Abbreviation** | **Long Form** |
| BERT | Bidirectional Encoder Representations from Transformers |
| RL | Reinforcement Learning |
| LLM | Large Language Model |

# **INTRODUCTION**

The introduction of technology has opened the door for revolutionary advancements in teaching and learning approaches in the field of education [1]. The creation of adaptive online learning platforms, which use technology to customize learning experiences to meet the requirements of specific students, is one example of such innovation. These systems, which offer targeted interventions and tailored learning paths, have great potential to improve learning outcomes.

The emotional health of pupils is a crucial factor that is frequently disregarded, despite the advances in adaptive learning methods [2]. Emotional aspects of learning are important because they impact students' motivation, engagement, and general academic success. Ignoring emotional needs can cause kids to become disengaged, have worse learning efficacy, and possibly develop mental health problems.

In identifying this gap, the goal of this project is to create an online learning environment that is adaptive and tailored to primary kids' emotional needs [3]. The platform aims to offer a comprehensive approach to education that spans both cognitive and emotional growth by including emotional intelligence in the learning process.

The primary objective of this project is twofold: first, to identify students' emotional weaknesses using the questionnaires and cutting-edge technologies like BERT [5]; and second, to provide teachers with personalized advice on how to support best and attend to these emotional needs using Reinforcement Learning (RL) algorithm and LLMs.

While there exist some approaches in the field of educational technology that touch upon aspects of emotional intelligence, they often fall short in providing comprehensive and personalized solutions [6]. Many existing systems lack the ability to accurately assess and address a wide range of emotional states, leading to generic or ineffective interventions.

The scientific contribution of this project lies in its innovative use of BERT and RL algorithms to analyze and understand students' emotional states in real-time [4]. By leveraging these technologies, the platform can provide nuanced and tailored recommendations to teachers, enabling them to offer targeted support that aligns with each student's emotional needs.

This research intends to show the effectiveness and impact of incorporating emotional intelligence into adaptive learning systems through extensive testing and evaluation [3]. This project aims to provide a more effective and holistic learning environment that supports students' cognitive and emotional development by filling up the gaps and limits in the current methods.

## **Background Literature**

Emotion detection and recognition is a crucial aspect of adaptive learning systems, as it allows for personalized learning experiences and timely interventions based on the learner's emotional state. The literature review explores various approaches and techniques used for emotion detection from textual data, facial expressions, speech, and self-assessments.

Godwin Bright [7] investigates emotion detection based on textual data using Natural Language Processing (NLP) and Machine Learning (ML) techniques. The study compares the performance of BERT and biLSTM models for emotion recognition in text, with the BERT model outperforming the biLSTM model. The author emphasizes the importance of incorporating psychological elements like emotion into models and discusses various approaches for emotion recognition in textual data.

Wei Feng and Xu Guoqiang [8] present an emotion identification method that combines question-based assessments and video analysis to detect genuine emotions based on facial expressions. The method involves generating a test question bank with neutral, comparison, and relevant questions, and analyzing expression feature vectors to distinguish between concealed and genuine emotions during a quiz.

Sumanathilaka et al. [9] propose an emotion detection technique using Bi-directional LSTM (BiLSTM) with an effective text pre-processing method. The study highlights the limitations of traditional pre-processing methods, such as the loss of characteristic traits, and compares the performance of deep learning models and transformer language models. The authors employ CNN for local feature extraction and BiLSTM with an attention layer for context-related global feature extraction, demonstrating improved emotion detection accuracy compared to traditional methods.

Olga C. Santos et al. [10] explore emotion detection from math exercises by combining user behavior analysis, sentiment analysis on written reports, and expert labeling. The study employs machine learning algorithms to infer emotions based on the combined data sources. Although the preliminary results are not conclusive, the authors provide insights on how to proceed with the analysis and highlight the potential of combining multiple data sources for emotion detection.

An intelligent assessment tool called I-Quiz [11] is described, which captures and analyzes the non-verbal behavior of learners during an assessment activity using machine learning techniques. The system records facial expressions, eye behavior, gestures, body posture, and body movement using a front-facing camera during a multiple-choice question assessment. A random forest classifier model is trained on the captured data to predict the learner's real knowledge acquisition level, enabling personalized learning experiences in e-learning platforms.

An exploratory analysis of a game-like task called "Guess the Alien" [12] is presented, which aims to assess emotion recognition in children. The study employs content analysis and develops novel scoring measures, such as Frequency of Emotion Naming (FEN) and Facial Emotion Matching (FEM), to evaluate participants' ability to identify and name emotional expressions. The analysis reveals associations between the game-like task scores and traditional measures of emotion recognition, as well as demographic differences in emotion naming performance.

In [13] a multimodal approach for emotion detection using machine learning techniques and data analysis is discussed. The study employs Convolutional Neural Networks (CNN) for facial feature extraction and emotion detection, age and gender estimation, and speech emotion recognition. Additionally, a self-assessment test for emotion detection and mental health is developed using Item Response Theory (IRT) and Rating Scale Model (RSM). The results demonstrate high accuracies in emotion detection across various modalities, including facial expressions, speech, and self-assessments.

Recent research has explored using reinforcement learning (RL) to enable more intelligent and cost-effective interactions between agents and large language models (LLMs) for task solving. Hu et al. [14] proposed a RL mediator model that learns to solve target tasks with few interactions with an LLM, reducing interaction costs during testing while making the agent's performance more robust against partial observability. This approach aims to address the limitations of time-consuming and costly interactions with commercial LLMs.

In the domain of improving LLM performance through RL, Chang et al. [15] introduced Reinforcement Learning with Guided Feedback (RLGF), which enhances LLM fine-tuning by outperforming supervised learning and default RL baselines like PPO. RLGF algorithms showed improved performance when interacting with a guide LLM like GPT-3, with a GPT-2 based policy outperforming the zero-shot GPT-3 oracle on the IMDB dataset.

Zhang et al. [16] proposed a framework called REMEMBERER, which combines LLMs with Reinforcement Learning with Experience Memory (RLEM), enabling LLMs to learn from past experiences and creating a semi-parametric RL agent. REMEMBERER demonstrated superior performance and robustness, outperforming the prior state-of-the-art by 4% and 2% on two RL task sets, highlighting the potential of enhancing LLMs with RL techniques.

These studies showcase the potential of using reinforcement learning to enhance LLMs' capabilities and enable more intelligent and cost-effective interactions between agents and LLMs. By leveraging RL techniques, LLMs can learn from past experiences, receive guided feedback, and interact more efficiently with agents, leading to improved task-solving performance and robustness.

In the context of an emotional weakness addressing adaptive learning system, the combination of LLMs and RL agents could be used to generate personalized advice or interventions based on the learner's emotional state and actions. The RL agent could learn to select the most appropriate advice or intervention from the LLM based on the learner's emotional weaknesses and responses, enabling a more tailored and adaptive learning experience.

However, it is important to note that the studies mentioned have limitations, such as the extent of interaction with the guide policy for algorithm performance comparison [15] and the allocation of the iteration budget for the D2LOLS algorithm [15]. Additionally, the interaction costs and storage requirements for using LLMs [14] should be considered when implementing such systems.

In summary, the literature highlights various techniques for emotion detection and recognition, such as NLP and ML methods for textual data, facial expression analysis, speech emotion recognition, and multimodal approaches combining different data sources. These studies emphasize the importance of emotion detection in adaptive learning systems and e-learning platforms and showcase the potential of game-like tasks and self-assessments for assessing emotion recognition abilities.

Additionally, recent research explores using reinforcement learning to enhance large language models (LLMs) and enable more intelligent and cost-effective interactions between agents and LLMs. Techniques like reinforcement learning with guided feedback (RLGF) and reinforcement learning with experience memory (RLEM) have demonstrated improved performance in generating better outputs than LLMs and outperforming state-of-the-art methods on certain tasks. Overall, the combination of emotion detection techniques and the potential of reinforcement learning with LLMs highlights promising avenues for generating personalized advice or interventions based on a learner's emotional state and actions in an emotional weakness addressing adaptive learning system.

## **Research Gap**

While the literature review highlights various techniques for emotion detection and recognition, as well as the potential of using reinforcement learning (RL) to enhance large language models (LLMs) for generating personalized advice or interventions, there are still some gaps that need to be addressed in the context of an emotional weaknesses addressing adaptive learning system for primary students.

One significant gap is the need for appropriate emotion detection techniques specifically tailored for primary students. Most existing studies focus on emotion detection from textual data, facial expressions, or speech. However, primary students may have limited language proficiency, making it difficult to accurately capture emotions through written text or speech. Additionally, facial expressions and speech patterns of young children can be more subtle and challenging to interpret. Tailoring emotion detection techniques to the unique learning environment, curriculum, and age-specific factors of primary students is crucial.

Furthermore, while the potential of integrating RL and LLMs has been demonstrated, there is limited research on how to effectively incorporate these techniques into adaptive learning systems for generating personalized advice or interventions based on a learner's emotional state. Specific challenges, such as mapping emotional states to appropriate advice or interventions, and handling the complexity of learner-system interactions, need to be addressed.

Ethical considerations and privacy concerns also arise when implementing emotion detection and personalized advice generation in educational settings, particularly when dealing with children. There is a need for research on ethical frameworks and privacy-preserving techniques to ensure the responsible and secure use of such systems.

Moreover, most existing studies focus on the technical aspects of emotion detection and advice generation, but there is a lack of research evaluating the long-term impact of such systems on the emotional well-being, academic performance, and overall development of primary students. Longitudinal studies are necessary to assess the effectiveness and potential implications of these adaptive learning systems.

Finally, while the research shows promising results in controlled settings, there are challenges in scaling and deploying these systems in real-world educational environments. Factors such as infrastructure requirements, teacher training, and integration with existing learning management systems need to be addressed for successful adoption and implementation.

By addressing these research gaps, the field can advance towards developing more effective and responsible emotional weaknesses addressing adaptive learning systems for primary students, leveraging the combined potential of emotion detection techniques, reinforcement learning, and large language models.

Table 1: Research Gap

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Features | [11] | [12] | [13] | [14] | [15] | [16] | Proposed System |
| Emotion detection through texts and images | Yes | No | Yes | No | No | No | Yes |
| Reinforcement Learning for LLM Interaction | No | No | No | Yes | Yes | Yes | Yes |
| Personalized Advice Generation | No | No | Yes | No | No | Yes | Yes |
| Collaboration with Experts | No | Yes | No | Yes | Yes | No | Yes |
| Adaptive Learning System for Primary Students | No | No | No | No | No | No | Yes |
| Quiz for Emotion Identification | Yes | No | Yes | No | No | No | Yes |
| Periodic Assessment for Progress Tracking | Yes | Yes | Yes | No | No | No | Yes |

A comparison of existing systems with the proposed system is given in the Table 1

## **Research Problem**

In today's fast-paced world, children often face a multitude of emotional challenges that can significantly impact their overall well-being and academic performance. Emotional weaknesses can have long-lasting effects on a child's mental health and development if left unaddressed. Failure to recognize and address these emotional needs can lead to disengagement from learning, decreased motivation, and potentially severe mental health problems in the long run.

While adaptive learning systems have made significant strides in tailoring educational experiences to individual cognitive abilities, there remains a critical gap in addressing the emotional aspect of learning. Most existing systems lack the capability to accurately assess and respond to a wide range of emotional states, often providing generic or ineffective interventions that fail to cater to the unique emotional needs of each learner.

The research problem addressed in this project is the lack of comprehensive and personalized solutions that incorporate emotional intelligence into adaptive learning systems for primary students. Specifically, the following key challenges need to be addressed:

**1. Accurate Assessment of Emotional States:**

Developing effective methods to accurately assess and understand the diverse range of emotional states experienced by primary students is crucial. Traditional assessment techniques, such as written responses or facial expressions, may be limited in capturing the nuanced emotional experiences of young learners.

**2. Tailored Interventions and Personalized Support:**

Once emotional states are identified, there is a need for a system that can provide tailored interventions and personalized advice to teachers, enabling them to offer targeted support that aligns with each student's emotional needs. Generic or one-size-fits-all approaches often fail to address the unique emotional requirements of individual learners.

**3. Integration of Emotional Intelligence into Adaptive Learning Systems:**

Incorporating emotional intelligence into adaptive learning systems requires a seamless integration of various technologies and techniques. This includes the utilization of natural language processing (NLP) models like BERT for emotion recognition, reinforcement learning (RL) algorithms for decision-making, and large language models (LLMs) for generating personalized advice.

**4. Longitudinal Evaluation and Impact Assessment:**

While technological solutions may show promise in controlled settings, there is a need to evaluate the long-term impact of incorporating emotional intelligence into adaptive learning systems. Longitudinal studies are necessary to assess the effectiveness of such systems in supporting the emotional well-being, academic performance, and overall development of primary students.

By addressing these research problems, this project aims to develop an innovative adaptive learning system that seamlessly integrates emotional intelligence assessment and personalized support. The system will leverage cutting-edge technologies, such as BERT for emotion recognition, RL algorithms for decision-making, and LLMs for generating tailored advice, to provide a comprehensive and holistic learning experience that supports both cognitive and emotional growth of primary students.

## **Research Objectives**

### **1.4.1 Main Objective**

The primary objective of this research is to develop an innovative adaptive learning system that seamlessly integrates emotional intelligence assessment and personalized support, leveraging cutting-edge technologies to provide a comprehensive and holistic learning experience for primary students.

The proposed system aims to bridge the existing gaps in accurately identifying and assessing the emotional states of primary students. It endeavors to create a platform that fosters an environment where students can freely and authentically express their emotions without inhibition. By leveraging advanced technologies and techniques, the system seeks to provide a comprehensive understanding of each student's emotional landscape. Consequently, the system will generate personalized advice and recommendations tailored to the unique emotional needs of individual students. These customized interventions will be disseminated to teachers, equipping them with the necessary tools and strategies to address emotional vulnerabilities proactively and effectively at an early stage of the students' development. Through this approach, the system aspires to promote a holistic learning experience that prioritizes both academic achievement and emotional well-being, ultimately fostering a supportive and nurturing educational environment conducive to the overall growth and development of primary students.

### **1.4.2 Specific Objectives**

**Sub Objective 1:**

To design and implement an approach for accurate assessment of emotional states in primary students, combining textual, and visual data.

The questionnaire employed an approach to obtain responses from students, allowing them to convey their emotional states through images, bullet points or sentences, thereby avoiding the must for constructing complete sentences. This design decision was informed by the recognition that not all primary students possess the linguistic proficiency or writing skills to express their emotions effectively in sentence form. By incorporating visual and concise response formats, the system aimed to create an inclusive and accessible platform that accommodated diverse student abilities. Furthermore, each question designed to capture a specific emotion underwent rigorous validation through the BERT model, ensuring the accurate interpretation and representation of the intended emotional constructs.

**Sub Objective 2:**

To develop a reinforcement learning (RL) algorithm that can effectively map identified emotional states to appropriate interventions or personalized advice.

The proposed system incorporates a reinforcement learning (RL) agent that is meticulously trained to make informed decisions regarding the most appropriate course of action based on the identified emotional state and corresponding responses of individual students. This decision-making process is specifically designed to address emotional vulnerabilities most effectively and optimally possible. By leveraging the capabilities of reinforcement learning algorithms, the agent aims to develop an adaptive and dynamic decision-making framework that can accurately map the complex interplay between emotional states, student responses, and tailored interventions. Ultimately, the goal is to ensure that the proposed actions and recommendations are tailored to mitigate emotional weaknesses in a personalized and contextually relevant manner, thereby fostering an environment that promotes emotional resilience and overall well-being among primary students.

**Sub Objective 3:**

To integrate a large language model (LLM) capable of generating tailored advice and recommendations for teachers, based on the emotional needs of individual students.

Upon the reinforcement learning (RL) agent's selection of the optimal course of action, the system integrates the identified emotional state and the underlying rationale behind the chosen action. This consolidated information is then relayed to the large language model (LLM) component of the system. Armed with these comprehensive data inputs, the LLM leverages its natural language processing capabilities to generate personalized advice and recommendations tailored to the specific emotional needs of the individual student. The advice generated by the LLM is informed not only by the detected emotional state but also by the contextual reasoning behind the RL agent's decision-making process. This synergistic approach ensures that the advice provided is highly contextualized, nuanced, and tailored to address the unique emotional vulnerabilities of each student in a holistic and effective manner.

Some additional sub-objectives that can be considered for this project are also given in the following context.

**Additional Objective 1:** To collaborate with psychologists and education experts in creating a comprehensive questionnaire that captures various emotional aspects of primary students' lives, including family background, school life, and social interactions.

**Additional Objective 2:** To leverage the BERT model for emotion recognition and labeling, ensuring accurate interpretation of students' emotional expressions.

**Additional Objective 3:** To evaluate the performance of the proposed system through extensive testing and user studies, involving primary students, teachers, and educational experts.

**Additional Objective 4:** To conduct longitudinal studies to assess the long-term impact of the adaptive learning system on students' emotional well-being, academic performance, and overall development.

# **METHODOLOGY**

## **Methodology**

### **2.1.1 Requirement Gathering**

The foundation of this research was established through a rigorous requirement gathering process. An extensive analysis of existing work and comparable systems was conducted based on predefined criteria. To set clear boundaries, the project's scope was meticulously specified. Key stakeholders, including a psychologist and an English teacher, were engaged through interviews to gather insights on the emotional needs of primary students and their linguistic and cognitive abilities. This stakeholder involvement played a crucial role in shaping the project's requirements, ensuring that the proposed adaptive learning system was tailored to the unique needs of the target audience while considering their developmental stages and capabilities. Through this comprehensive process, the project established a solid foundation grounded in a thorough understanding of the problem domain, literature, and the specific requirements of primary students.

Key stages,

* + Collecting related research papers.
  + Conducting a feasibility study.
  + Conducting a background and literature assessment.
  + Reading and evaluating the collected research papers.
  + Consulting experts and gathering information related to the system.
  + Identifying the most suitable components and finalizing the project scope.

### **2.1.2. Dataset Description**

Recognizing the absence of datasets tailored for emotion classification among primary students, a concerted effort was undertaken to create a comprehensive dataset. A survey comprising 15 carefully curated questions, spanning three domains – family, school, and friends and relatives – was administered to 25 students. The questions were meticulously crafted with the invaluable guidance of a psychologist and an English teacher, ensuring their relevance and appropriateness. Each question was associated with one of the predefined emotions from the emotion list, including happiness, sadness, fear, feeling loved or unloved. The resulting dataset, comprising 760 records, provided a rich tapestry of emotional responses from the target demographic. This dataset proved instrumental in training a BERT model specifically tailored for emotion classification, enabling it to accurately interpret and categorize the emotional undertones present in the responses provided by users through the website.

### **2.1.3. System Architecture**

The primary objective of this project is to develop an adaptive online learning platform for primary students, addressing the current limitations in the field. A unique facial authentication system will streamline the login process, eliminating the need for students to remember passwords, thereby making it more user-friendly for the target demographic. Additionally, a meticulously designed quiz will be employed to gauge the emotional state of the students, and in cases where guidance is deemed necessary, personalized advice will be provided to the teachers to support the students effectively. Furthermore, a sophisticated recommendation system has been developed to suggest study materials tailored to the individual student's performance on the quizzes. Lastly, a real-time attention monitoring system has been integrated to track students' attentiveness during the quizzes, providing teachers with valuable insights into their students' engagement levels.

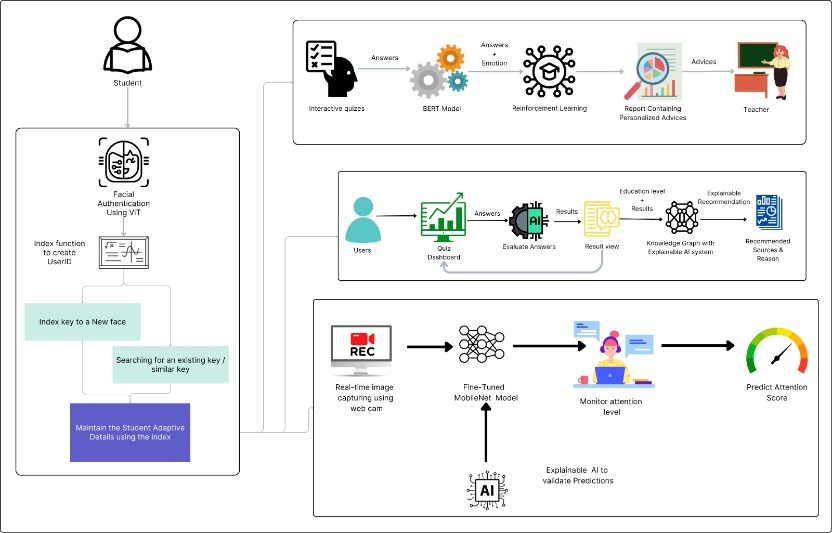


Figure 1: Overall System Architecture

### **2.1.4. Component Architecture**

The emotional weakness addressing model implemented in this application adopts a holistic approach to identify and assist students in managing their emotional well-being. Central to this model is a meticulously designed quiz, segmented into three distinct categories: family, school, and friends. Each category features questions strategically formulated to elicit emotions pertinent to its respective context. These questions offer diverse answer formats, ranging from images to multiple-choice questions (MCQs) and sentences, facilitating varied means of expression. To accommodate students who may struggle to articulate their feelings, pre-defined answers or emotion-labeled images are provided for selection. Upon completion of the quiz, the model synthesizes the user's responses, assigning each answer a corresponding emotion label and score. This comprehensive evaluation culminates in the calculation of an overall emotional state and score for the student. Subsequently, the assessment data is input into a reinforcement learning (RL) model alongside the rationale behind each response. Leveraging this information, the RL agent selects tailored actions aimed at supporting the student's emotional needs. Based on the chosen action, a large language model (LLM) generates personalized advice, which is then relayed to the teacher. The resulting report encompasses the questions posed, the student's responses, the associated emotions, and the personalized advice provided. Additionally, the report identifies the prevalent emotion, enabling teachers to monitor student progress and adapt interventions, accordingly, fostering a conducive environment for emotional growth and development. The diagram below provides a visual representation of the component architecture.

A diagram of a cogwheel and a graduation cap

Description automatically generated

Figure 2: Emotional Weaknesses Addressing Component

### **2.1.5. Tools and Libraries**

* Transformers library: importing pre-trained BERT model.
* OpenAI:to access gpt4 API key.
* Pandas: The Pandas library facilitates the handling and manipulation of CSV data.

### **2.1.6. Model Architecture**

#### **2.1.6.1. BERT Model**

I have trained a BERT model for the emotion classification process. BERT's strong foundation in understanding language semantics and contextual relationships allows for effective extraction of emotional nuances, even with a relatively small dataset of 760 records. Through fine-tuning, the model adapts its learned representations to the specific task of emotion detection, further improving performance. Moreover, BERT's ability to capture bidirectional context in text sequences enhances its understanding of emotional expressions within sentences, considering both preceding and following words. This bidirectional context awareness proves invaluable in grasping the nuances of emotion conveyed through language, ultimately enabling accurate and reliable emotion classification for our application.

Upon submission of the form, the questions and answers are passed in an array structure. If the length of the array is determined to be two, denoting a valid question-response pair, the sentence within this array undergoes processing through the BERT model. This processing enables the extraction of the underlying emotion expressed within the sentence.

**Step 1**: Import necessary libraries: This step involves importing Python libraries such as pandas, scikit-learn, torch, transformers, and Google Colab's drive module. These libraries are essential for data preprocessing, model training, and saving/loading files from Google Drive.

A screenshot of a computer program

Description automatically generated

Figure 3: Importing Libraries for BERT Model

**Step 2**: Mount Google Drive: This step connects your Google Drive to the Colab environment, allowing you to access datasets stored in Drive and save the trained BERT model back to Drive for future use.

A screen shot of a computer code

Description automatically generated

Figure 4: Mounting to Google Drive

**Step 3:**Load datasets: Here, the training, testing, and validation datasets are loaded from CSV files. These datasets contain textual data along with corresponding emotion labels.

A screenshot of a computer

Description automatically generated

Figure 5: Reading CSV files

**Step 4**: Encode emotions column: Using LabelEncoder from scikit-learn, the emotion labels in the datasets are encoded into numerical values. This step is crucial for the model to interpret emotions as numerical categories during training.

A computer code with text

Description automatically generated with medium confidence

Figure 6: Label Encoding

**Step 5:**Implement Naive Bayes baseline: Naive Bayes is used as a baseline model for emotion classification. It provides a simple yet effective benchmark for comparing the performance of the more complex BERT model later.

A screenshot of a computer program

Description automatically generated

Figure 7: Naive Bayes Baseline Model

**Step 6**: Load pre-trained BERT model and tokenizer: The pre-trained BERT model and its tokenizer are loaded from the 'bert-base-uncased' configuration. BERT (Bidirectional Encoder Representations from Transformers) is a powerful language model known for its contextual understanding of text.



Figure 8: Loading Pre-trained Model and Tokenizer

**Step 7:**Tokenize input texts: Text inputs from the datasets are tokenized using the BERT tokenizer. Tokenization converts text into numerical tokens that can be processed by the model.

A computer code with text

Description automatically generated with medium confidence

Figure 9: Tokenizing Sentences

**Step 8**: Convert to PyTorch tensors: The tokenized data is converted into PyTorch tensors, a data format compatible with PyTorch's deep learning framework. These tensors are used as inputs for training and evaluation.

A screenshot of a computer code

Description automatically generated

Figure 10: Converting Tokenized Data into Tensors

**Step 9**: Create dataloaders: Dataloaders are created to facilitate efficient batch processing during training. They provide batches of data to the model iteratively, improving training speed and memory usage.

A close-up of a computer code

Description automatically generated

Figure 11: Creating Data Loaders

**Step 10:**Define hyperparameters: Hyperparameters such as batch sizes, learning rates, and number of epochs are defined for hyperparameter tuning. These parameters significantly impact the model's training and performance.

A math equations with green and blue text

Description automatically generated

Figure 12: Values for Hyperparameter Tuning

**Step 11**: Set up training loop and device: The training loop is initialized, and the device (GPU or CPU) is set based on availability for model training.

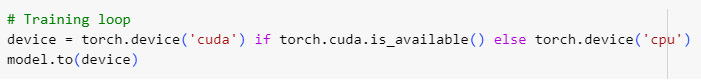


Figure 13: Setting up Device

**Step 12:**Initialize the best accuracy and hyperparameters for hyperparameter tuning.

A black text on a white background

Description automatically generated

Figure 14: Initializing Best accuracy and Hyperparameters

**Step 13:**Define the evaluation function to assess model performance on test data.

A screenshot of a computer program

Description automatically generated

Figure 15: Evaluation Function

**Step 14**: Configure optimizer (AdamW) and scheduler for model training.

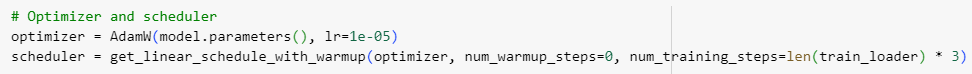


Figure 16: Setting up Scheduler and Optimizer

**Step 15:**Perform hyperparameter tuning using nested loops and validation set evaluation.

A screenshot of a computer code

Description automatically generated

Figure 17: BERT Training Loop

A white background with black text

Description automatically generated

Figure 18: Evaluating the Model on Test Data

**Step 16:**Fine-tune BERT model with optimized hyperparameters for better performance.

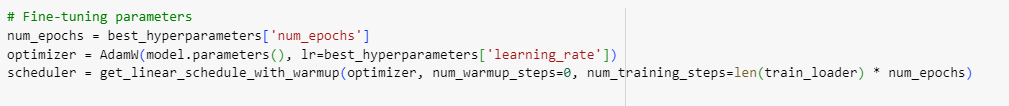


Figure 19: Fine Tuning BERT Model with Optimized Hyperparameters

**Step 17:**Train the fine-tuned model and evaluate the validation set for each epoch.

A screenshot of a computer

Description automatically generated

Figure 20: Training the BERT Model and Evaluate on Validation Dataset

**Step 18:**Evaluate the fine-tuned model's performance on the test set.

A close-up of a test

Description automatically generated

Figure 21: Evaluate the Model on Test Datasets

**Step 19**: ‘predict\_with\_probabilities’ Function:

* This function takes a sentence, a BERT model, a tokenizer, a label encoder, and a device as inputs.
* It tokenizes the input sentence using the tokenizer and moves the tokens to the specified device.
* The BERT model is also moved to the same device.
* It performs inference (prediction) using the BERT model on the tokenized input.
* The output logits from the model are converted to softmax probabilities.
* These probabilities are mapped to emotion labels using the label encoder.
* Finally, it returns a dictionary containing the probabilities of each emotion label.

A screenshot of a computer program

Description automatically generated

Figure 22: Prediction for BERT Model

**Step 20:**Compute and visualize the confusion matrix to assess model performance.

A screenshot of a computer program

Description automatically generated

Figure 23: Plotting Confussion Matrix

A screenshot of a computer

Description automatically generated

Figure 24: Generating Classification Report

**Step 21:**Save the trained BERT model to Google Drive for future use.

A close-up of a content

Description automatically generated

Figure 25: Saving the BERT Model in the Drive

#### **2.1.6.2. RL MODEL**

The reinforcement learning (RL) agent underwent training to make action selections predicated on the emotional content discerned from students' responses. Subsequently, leveraging the chosen action, a large language model (LLM) was employed to generate personalized advice. This advice was formulated by comprehensively considering various factors including the posed question, the student's response, the associated emotion, and the action determined by the RL agent.

**Step 22:**Import Necessary Libraries and Modules: In this step, I have imported essential libraries and modules required for our reinforcement learning (RL) agent, including OpenAI's API for accessing GPT-4 and other functionalities.

A close-up of a computer code

Description automatically generated

Figure 26: Importing Necessary Libraries for Advice Generation Part

**Step 23:**Retrieve OpenAI API Key: Here, I have fetched the OpenAI API key from user data, allowing us to authenticate and use OpenAI's services such as model completions.

A close-up of a computer code

Description automatically generated

Figure 27: Fetching OpenAI API Key

**Step 24:**Define State and Action Spaces: I have defined the possible emotional states that our RL agent can encounter (state space) and the corresponding actions it can take (action space) in response to those states.



Figure 28: Defining State and Action Space

**Step 25:**Initialize Q-Table: The Q-table is initialized with zeros, serving as a memory structure to store the quality of actions taken in specific states, which guides the agent's decision-making.

A computer code with text

Description automatically generated with medium confidence

Figure 29: Initializing Q-Table

**Step 26:**Set Hyperparameters: Hyperparameters like the learning rate (alpha) and exploration rate (epsilon) are crucial settings that influence how the RL agent learns and explores the environment.

A close-up of green text

Description automatically generated

Figure 30: Setting Hyperparameters

**Step 27:**Define Policy Function: The policy function determines how the RL agent selects actions based on the current state and Q-values, balancing between exploration (trying new actions) and exploitation (using learned actions).

A screenshot of a computer code

Description automatically generated

Figure 31: Defining Policy Function

**Step 28:**Implement Q-Learning Algorithm: Q-learning is a fundamental RL algorithm used to update Q-values in the Q-table based on rewards received from actions taken in specific states, optimizing the agent's decision-making strategy.

A close-up of words

Description automatically generated

Figure 32: Defining Q-Learning Algorithm

**Step 29**: ‘preprocess\_dataset’ Function:

* This function preprocesses a dataset by mapping emotions to numerical values based on a given state\_space.
* It iterates through each row of the dataset, extracts the sentence and emotion, and maps the emotion to its corresponding index in state\_space.
* If the mapping hasn't been printed before, it prints the emotion mappings for reference.
* The processed data, including the sentence and mapped emotion, are stored in a list and returned.

A screen shot of a computer code

Description automatically generated

Figure 33: Preprocessing Input Data

**Step 30**: ‘generate\_advice’ Function:

* This function generates advice using the GPT-4 Turbo model from OpenAI.
* It takes a prompt and an OpenAI client as inputs.
* It sends a request to the GPT-4 model with the prompt and retrieves the generated advice.
* The generated advice is returned.

A screen shot of a computer code

Description automatically generated

Figure 34: Defining function to access LLM

**Step 31:**Specify Training Parameters: I have set the number of training episodes and steps per episode, defining the scope and duration of the RL agent's learning process.

A black text with green and black text

Description automatically generated

Figure 35: Specifying Training Parameters

**Step 32:**Implement RL Training Loop: The RL training loop iterates through episodes and steps, guiding the agent to choose actions, receive rewards, update Q-values, and gradually improve its decision-making capabilities.

A screenshot of a computer code

Description automatically generated

Figure 36: RL Agent Training Loop

A close-up of a computer screen

Description automatically generated

Figure 37: Updating Q-Table based on Feedback Given by The Developer

**Step 33:**Print Updated Q-Table: After training, we print the updated Q-table to visualize the learned Q-values, reflecting the agent's learned knowledge about state-action pairs.

A close-up of a check mark

Description automatically generated

Figure 38: Checking Updated Q-Table

**Step 34:**Preprocess Evaluation Dataset: Similar to the training data, the evaluation dataset is prepared for assessing the RL agent's behavior and performance on unseen data.

A close-up of a text

Description automatically generated

Figure 39: Defining Evaluation Dataset

**Step 35:**Sample Evaluation Data: To evaluate the RL agent efficiently, I have sampled a subset of the evaluation data, ensuring a manageable size for evaluation while capturing diverse scenarios.

A close-up of a text

Description automatically generated

Figure 40: Getting A Random Sample from Evaluation Dataset

**Step 36**: ‘evaluate\_agent’ Function:

* This function evaluates the agent's behavior on an evaluation dataset.
* It takes the evaluation dataset, a loaded Q-table, state and action spaces, and an OpenAI client as inputs.
* It iterates through each sentence and emotion index in the evaluation data.
* Based on the current state, it chooses an action using the loaded Q-table.
* It generates a prompt based on the chosen action and state and retrieves advice using the generate\_advice function.
* Finally, it yields a dictionary containing the sentence, emotions, chosen action, and generated advice for each evaluation data item.

A screenshot of a computer program

Description automatically generated

Figure 41: Evaluating Agent on Sample Evaluation Data

A close-up of a computer code

Description automatically generated

Figure 42: RL Agent Evaluation

**Step 37:**Save Trained Q-Table: Finally, I have saved the trained Q-table to Google Drive, preserving the learned knowledge of the RL agent for future reference and utilization.

A close-up of a computer screen

Description automatically generated

Figure 43: Saving The Q-Table in The Drive

## **2.2. Commercialization Aspects of the Product**

### **2.2.1. Target Market**

Our primary target market includes primary school students and their parents or guardians. We aim to address emotional weaknesses among children and provide tailored advice to improve their emotional well-being. Additionally, we will collaborate with schools and educational institutions to integrate our system into their curriculums, making it accessible to a wider audience of educators and students.

### **2.2.2. Revenue Streams**

To generate revenue, we will implement a multi-tiered monetization strategy. This will include subscription-based models for individual users, schools, and educational institutions. We will also offer premium features and personalized advice packages as upsells. Furthermore, we will explore partnerships with educational platforms and content providers to offer value-added services and content related to emotional intelligence and well-being.

### **2.2.3. Marketing Approach**

**Phase 1: Product Development and Testing**

Launch the initial version of the system in collaboration with pilot schools to gather feedback and refine the product based on real-world usage.

**Phase 2: Freemium Model and Educational Partnerships**

Introduce a freemium model offering basic emotional assessment and generic advice, with premium features and personalized advice available through subscription. Forge partnerships with educational institutions to promote the system as part of their emotional learning programs.

**Phase 3: Digital Marketing and Awareness Campaigns**

Utilize digital marketing channels such as social media, educational forums, and targeted online ads to raise awareness about our system. Engage in campaigns focused on highlighting the benefits of emotional intelligence in education and the positive impact of personalized advice on students' well-being.

**Phase 4: Community Building and Advocacy**

Build a community of parents, educators, and mental health professionals who advocate for the importance of emotional intelligence in children's development. Organize workshops, webinars, and events to promote discussions and knowledge sharing on emotional well-being.

**Phase 5: Strategic Partnerships and Expansion**

Form strategic alliances with educational technology companies, child psychologists, and mental health organizations to expand our reach and enhance the effectiveness of our system. Explore international markets and localization opportunities to cater to diverse cultural contexts and languages.

## **Testing and Implementation**

### **2.3.1. Functional Requirements**

**Emotion Detection:**

* The system should be able to administer quizzes to students and detect their emotional states based on their responses.
* The system should label each response with the corresponding emotion and assign a score.
* The system should calculate an overall emotional score and state for the student after the quiz.

**Reinforcement Learning Agent:**

* The system should have a reinforcement learning (RL) agent that can select appropriate actions based on the student's emotional state and provided reasoning.

**Personalized Advice Generation:**

* The system should have a language model (LLM) that can generate personalized advice for teachers based on the action chosen by the RL agent and the student's emotional state.

**Teacher Dashboard:**

* The system should have a dashboard for teachers to view personalized advice, students' emotional states, attention reports, and other relevant information.

### **2.3.2. Non-Functional Requirements**

**Performance:**

* The system should be responsive and provide real-time feedback to students and teachers.
* The emotion detection, RL agent, and LLM components should process data efficiently to avoid delays.

**Scalability:**

* The system should be able to handle an increasing number of users and data without compromising performance.

**Security:**

* The system should implement proper authentication and authorization mechanisms to protect sensitive data.
* Student data and emotional information should be encrypted and securely stored.

**Privacy:**

* The system should comply with relevant privacy regulations and ensure that student data is handled confidentially.

**Usability:**

* The user interface should be intuitive and easy to navigate, especially for primary school students.
* The system should provide clear instructions and guidance for students and teachers.

**Accessibility:**

* The system should be accessible to users with disabilities, following relevant accessibility standards.

**Extensibility:**

* The system should be designed in a modular and extensible manner, allowing for the addition of new features or integration with other systems in the future.

**Robustness:**

* The system should handle unexpected inputs and edge cases gracefully, without crashing or compromising data integrity.

**Maintainability:**

* The system's codebase should be well-documented and follow the best coding practices to facilitate future maintenance and updates.

### **2.3.3. Backend Implementation**

The steps that were taken to implement the backend and frontend of this component are well described in the following.

**Import Statements:**

* Flask, jsonify, request: Import necessary modules from Flask for building the web application and handling HTTP requests.
* prompt\_model: Import the function (prompt\_model) from module (advice\_generator) that generates advice based on input sentences.
* CORS, cross\_origin: Import CORS-related modules to enable Cross-Origin Resource Sharing (CORS) for handling requests from different origins.
* **pymongo**: This is imported from the pymongo module, which is a Python driver for MongoDB, a popular NoSQL database. This is to interact with the MongoDB database.
* **configparser**: This is a module used for working with configuration files in Python. It allows reading and writing configuration files in an INI file format.

A screen shot of a computer

Description automatically generated

Figure 44: Importing Libraries

**configparser** **Initialization:**

* Create a configparser object named config.
* Read the contents of the configuration file "config.ini" using the read() method.

A screen shot of a computer

Description automatically generated

Figure 45: ConfigParser Initialization

**MongoDB Initialization:**

* Retrieves the MongoDB URI from the configuration file under the section 'APIKeys'.
* Creates a MongoDB client using the retrieved URI.
* Selects the MongoDB database named 'Eduflex' using the client.

A computer code with white text

Description automatically generated with medium confidence

Figure 46: MongoDB Initialization

**Flask App Initialization:**

* Create a Flask application instance named app.
* Initialize CORS with CORS (app) to allow cross-origin requests.
* Set CORS\_HEADERS in app.config to specify the allowed headers for CORS.

A black background with text and letters

Description automatically generated with medium confidence

Figure 47: Flask App Initialization

**‘/predict’ Route:**

* **Route Definition**: Defines a route '/predict' that accepts POST requests.
* **Cross-Origin Resource Sharing (CORS)**: Allows cross-origin requests for this route.
* **Request Handling**: Extracts JSON data from the incoming request.
* **Data Extraction**: Retrieves the input sentence and its type from the JSON data.
* **Model Prediction**: Uses a model (prompt\_model) to generate advice based on the input sentence.
* **Database Interaction**: Inserts the advice data into a MongoDB database based on its type.
* **Response**: Sends a JSON response containing the generated advice.

A screen shot of a computer code

Description automatically generated

Figure 48: '/predict' Route

**‘/api/reports’ Route:**

* Route Definition: Defines a route '/api/reports' that handles GET requests.
* Collection Mapping: Defines a mapping between different types of quizzes and their respective MongoDB collections.
* Data Retrieval: Retrieves data from all specified collections.
* Conversion: Converts ObjectId to string for each document retrieved.
* JSON Response: Sends a JSON response containing all retrieved data.

A computer screen shot of text

Description automatically generated

Figure 49: ‘/api/reports’ Route

**App Execution:**

* The if \_\_name\_\_ == '\_\_main\_\_': block ensures that the Flask app is only run when the script is executed directly (not imported).
* Run the Flask app in debug mode (app.run(debug=True)), which enables debugging features and auto-reloads the server on code changes.

A black background with white text

Description automatically generated

Figure 50: App Execution

**‘get\_responses\_with\_highest\_emotions’ Function:**

* This function takes an array of responses, each with associated emotion probabilities.
* It iterates through each response and finds the emotion with the highest probability.
* For each response, it creates a dictionary containing the response text, the highest emotion, and its probability.
* The resulting list of dictionaries is returned.

A screen shot of a computer program

Description automatically generated

Figure 51: ‘get\_responses\_with\_highest\_emotions’ Function

**‘prompt\_model’ Function:**

* This function acts as a wrapper for the entire process.
* It loads necessary data (like training, testing, and validation datasets, models, tokenizer, etc.) if they haven't been loaded already.
* It preprocesses the input prompt using the BERT model to predict emotions with probabilities.
* It then finds the highest emotion for each response and generates advice using the llama3-70b-instruct model.
* The processed data is then returned or printed, depending on the commented-out sections.

A screen shot of a computer program

Description automatically generated

Figure 52: Generate Advice Function

A screen shot of a computer program

Description automatically generated

Figure 53: 'prompt\_model’ Function

**Collecting responses into an array:**

* Data Collection: Initializes an empty list evaluation\_data to collect responses.
* Printing Data: Prints the initial state of evaluation\_data.
* Sentiment Analysis: Processes each response to analyze sentiment if it's a list of length 2.
* Collecting Responses with Emotions: Creates a list responses\_with\_emotions containing responses along with their associated emotions.
* Converting to Array Format: Converts responses with emotions into an array format for easier handling.
* Building Response Data: Creates a dictionary for each response containing question, response text, and associated emotions.
* Appending to Array: Appends each response data dictionary to responses\_array.

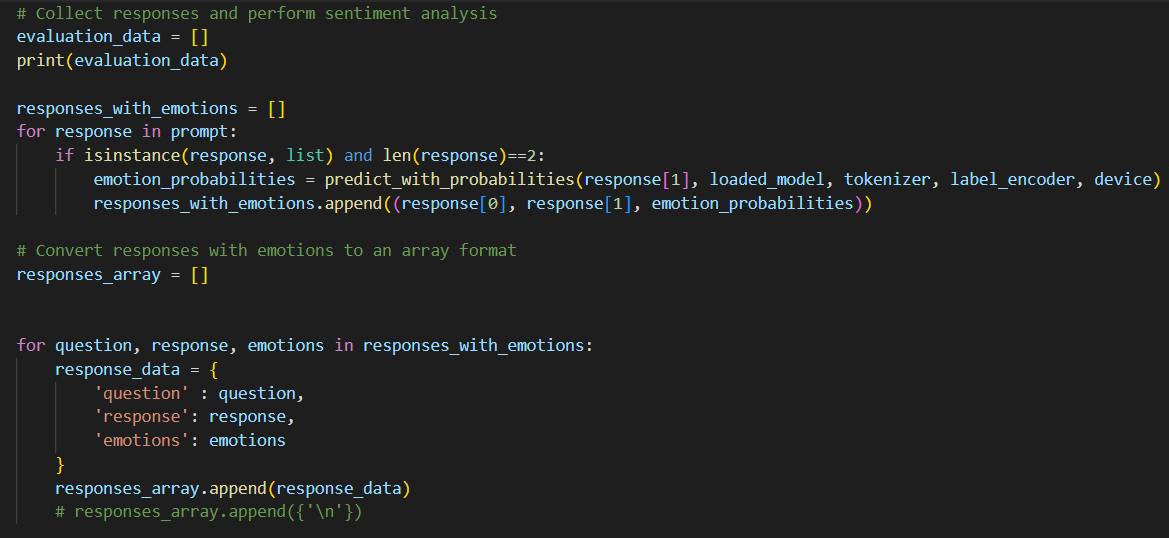


Figure 54: Evaluation of Emotions for Sentence Format Answers

* Building Response Data: Constructs a dictionary for each response containing question, response text, and associated emotions, appending it to responses\_array.
* Response Processing Check: Checks if there are responses in responses\_array.
* Extracting Responses with Highest Emotions: Extracts responses with the highest associated emotions using a custom function.
* Converting to DataFrame: Converts the extracted responses into a DataFrame for further analysis.
* Preprocessing Data: Preprocesses the DataFrame for evaluation.
* Additional Data Collection: Initializes an empty list evaluation\_data\_2 to collect additional response data.
* Iterating through Responses: Processes each response to extract relevant data if it's a list of length 3.
* Appending to Evaluation Data: Adds the extracted response data to evaluation\_data.

A computer screen shot of a program

Description automatically generated

Figure 55: Evaluation of Emotions for MCQ & Picture Format Answers

**Evaluation Function:**

* Evaluation Function Definition: Defines a function evaluate\_agent to assess the agent's behavior.
* Iterating Through Evaluation Data: Loops through each data point in the evaluation dataset.
* State Representation: Sets the state for evaluation, representing emotional states.
* Action Selection: Chooses an action based on the policy using a Q-table.
* Prompt Generation: Generates a prompt for each evaluation scenario.
* Advice Generation: Utilizes the llama3-70b-instruct model to generate advice based on the generated prompt.
* Advice Data: Constructs a dictionary containing the question, response, emotions, action, and advice.
* Yielding Data: Yields the advice data for each evaluation scenario, allowing for dynamic processing.

A screen shot of a computer code

Description automatically generated

Figure 56: Evaluate Agent Function

A screen shot of a computer program

Description automatically generated

Figure 57: Calling generate\_advice Function

**Emotion count calculation:**

* Emotion Occurrence Counting: Counts the occurrences of each emotion in the evaluation data.
* Flattening Emotions: Converts the nested emotion data into a flat list.
* Counting Emotions: Uses Counter to count occurrences of each emotion.
* Printing Emotion Counts: Displays the counts of each emotion.
* Calculating Maximum Emotion: Identifies the emotion with the highest count and calculates its proportion.
* Mapping Emotion Labels: Maps emotion index to their corresponding labels for better readability.

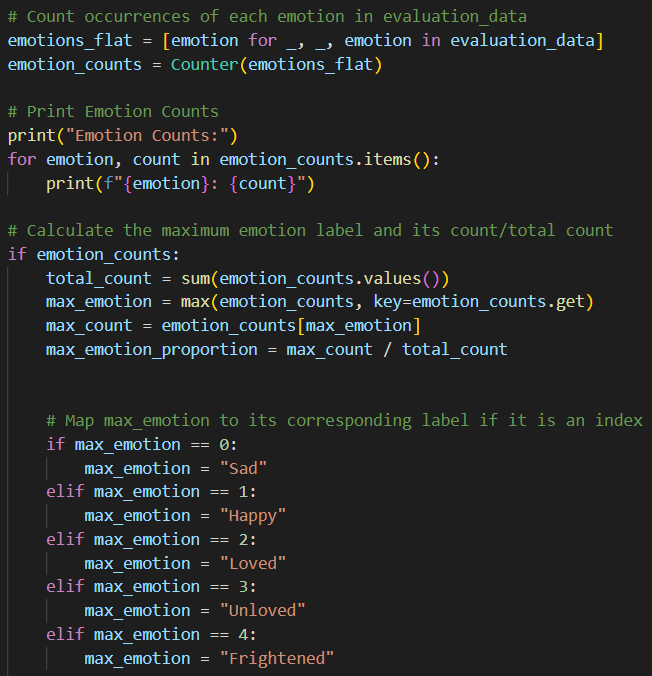


Figure 58: Calculating Emotion Counts

And then again, import the necessary libraries in the front-end file.

A computer screen shot of text

Description automatically generated

Figure 59: Importing Libraries and Packages to Build the Frontend

useState Hooks: This code snippet uses multiple useState hooks to manage state within the FamilyForm component. Each useState hook initializes a piece of state and provides a function to update that state.

A screen shot of a computer program

Description automatically generated

Figure 60: useState Hooks

‘HandleSubmit’ Function: This function is triggered when the user submits the form. It logs the form data and calls the getAdvice function to fetch advice based on the form data.

A screen shot of a computer code

Description automatically generated

Figure 61: Handle Submit Function

‘getAdvice; Function: This function sends a POST request to a specified endpoint (http://127.0.0.1:5000/predict) with the form data in JSON format. It then returns a Promise that resolves to the JSON response from the server.

A screen shot of a computer screen

Description automatically generated

Figure 62: getAdvice Function

‘AdviceModal’ Component: This component displays advice fetched from the getAdvice function in a modal dialog when isOpen is true.

A computer code with text

Description automatically generated with medium confidence

Figure 63: AdviceModal

* Once adviceData is available (i.e., not empty), the advice is displayed in a grid layout.
* Each piece of advice is rendered in a card-like format with the emotions, action, advice content, and associated sentence displayed.
* The advice data is mapped using adviceData.map, generating a card for each advice item in the adviceData array.

A screen shot of a computer

Description automatically generated

Figure 64: AdviceModal Function

A computer screen shot of text

Description automatically generated

Figure 65: How AdviceModal mapped with Advice Data

### **2.3.4. Backend Testing**

For backend testing, Postman was used as the tool to send POST requests. These requests were verified as well to ensure that the API returns the expected response based on the input sentences.

A screenshot of a computer

Description automatically generated

Figure 66: Using Postman to Test Backend

A screenshot of a computer

Description automatically generated

Figure 67: Home Page of The Application

A screenshot of a video game

Description automatically generated

Figure 68: Quiz Home Page for Emotion Analysis

A screenshot of a family website

Description automatically generated

Figure 69: Quiz Interface

A screenshot of a computer

Description automatically generated

Figure 70: Answer Options in MCQ, Sentence & Picture Formats

A screenshot of a game

Description automatically generated

Figure 71: Results Page Shown to The Children

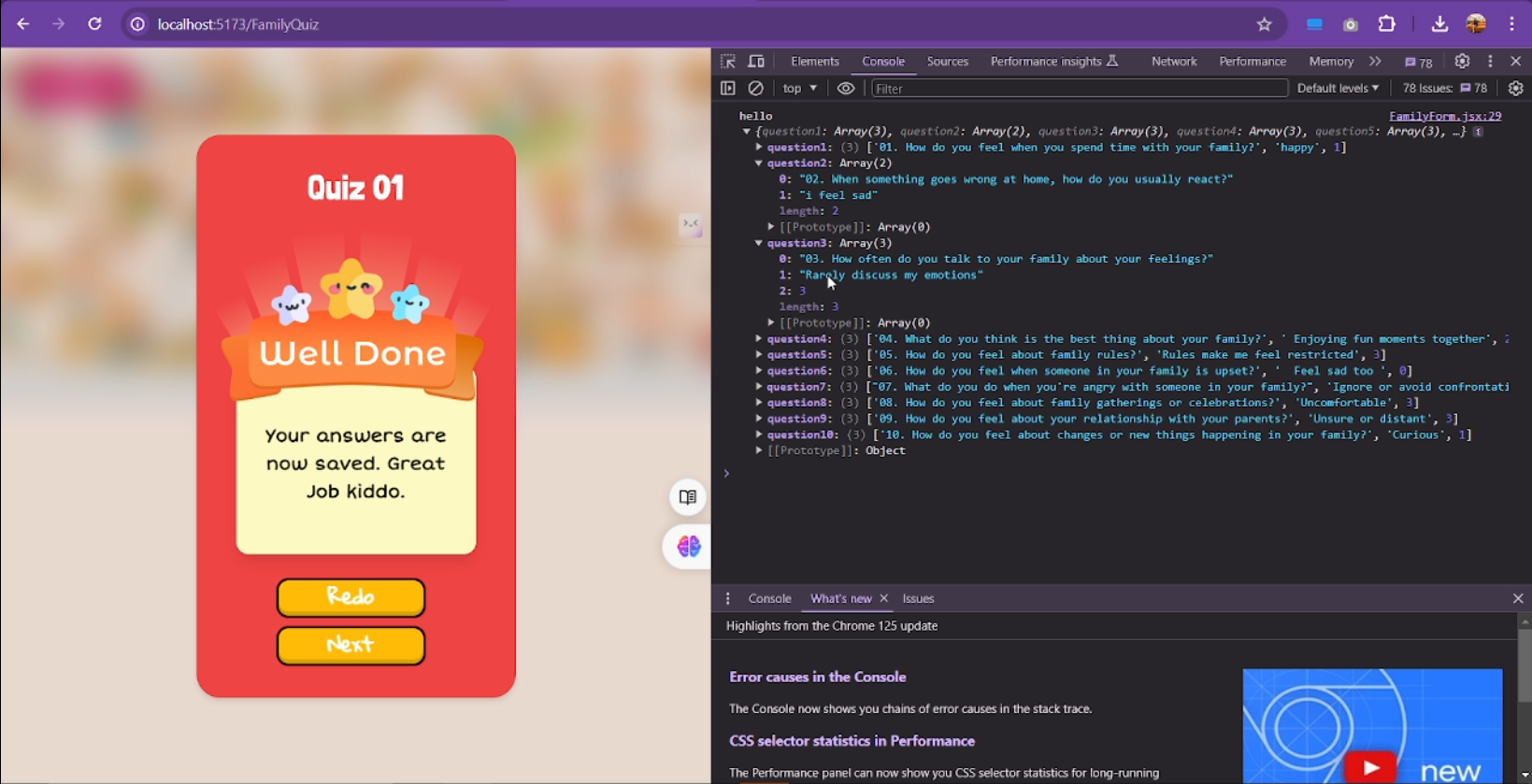


Figure 72: Answer Arrays

A screenshot of a computer

Description automatically generated

Figure 73: Database Structure

A screenshot of a computer

Description automatically generated

Figure 74: Reports Generated for Students

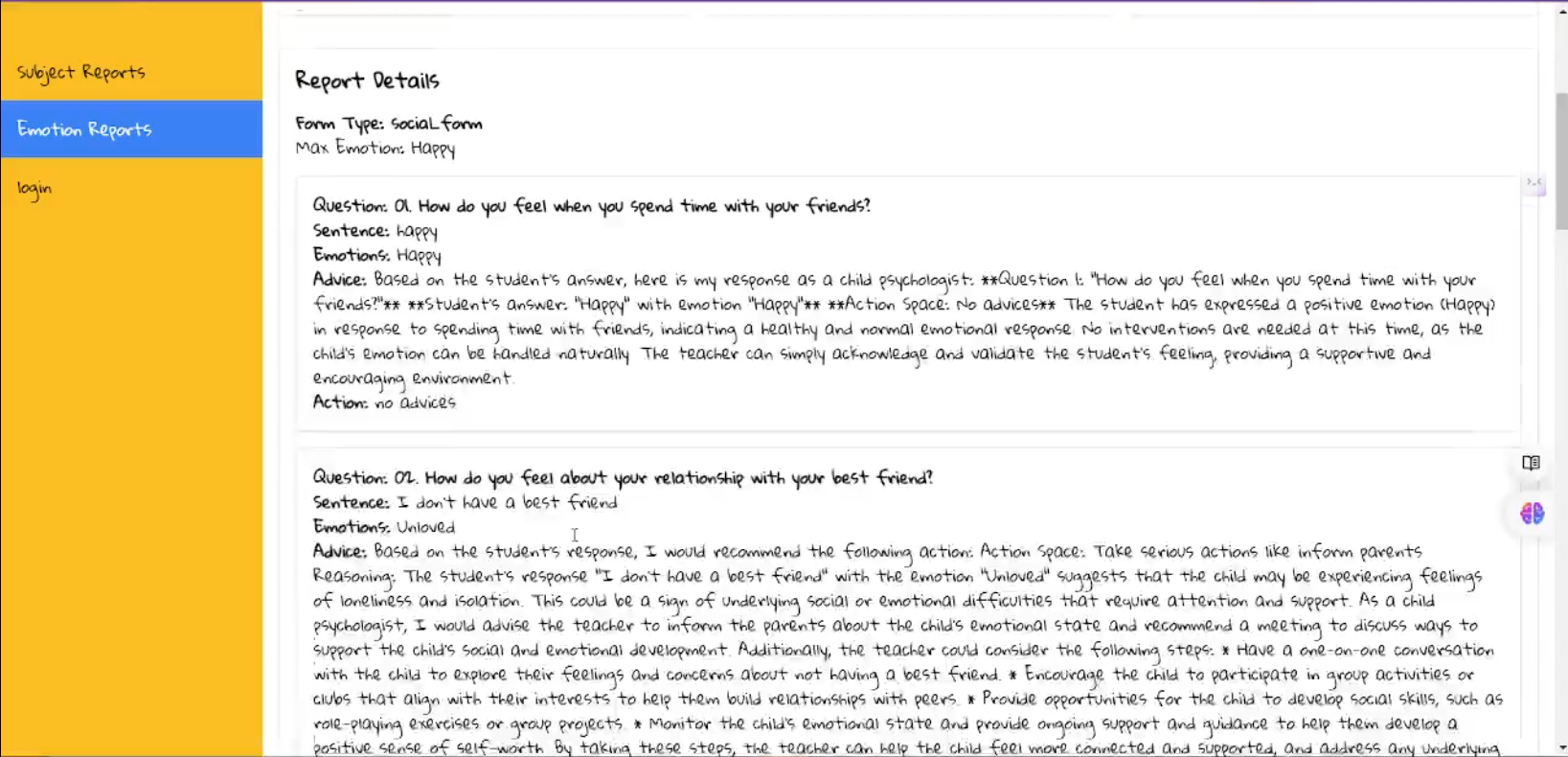


Figure 75: Detailed Report with Personalized Advices

# **RESULTS AND DISCUSSION**

## **BERT Model**

The BERT model developed for emotion classification demonstrated promising results, achieving an accuracy of 74% on the dataset comprising 760 records. This performance represents a notable improvement over the baseline Naive Bayes model, which achieved an accuracy of 73%. While the dataset size was relatively small, the BERT model's ability to capture contextual relationships and fine-tune its representations for the specific task of emotion detection enabled it to outperform the traditional Naive Bayes approach. These results highlight the potential of leveraging advanced language models like BERT for emotion classification tasks, even with limited data availability. As more data is collected and incorporated into the training process, further performance improvements can be expected, making the BERT-based emotion classification model a valuable asset for understanding and addressing emotional nuances in various applications.

A screenshot of a computer screen

Description automatically generated

Figure 76: Classification Report of Naive Bayes Model

A diagram of a confusion matrix

Description automatically generated with medium confidence

Figure 77: Confusion Matrix of Naive Bayes Model

A black screen with white numbers

Description automatically generated

Figure 78: Validation Accuracy of BERT Model

A screen shot of a black screen

Description automatically generated

Figure 79: Classification Report of BERT Model

A diagram of a confusion matrix

Description automatically generated with medium confidence

Figure 80: Confusion Matrix of BERT Model

## **RL Model**

To train the reinforcement learning (RL) agent to select appropriate actions based on the detected emotions, the complete dataset comprising all records was utilized. Through an iterative process, the agent learned to associate specific emotions with corresponding actions by receiving feedback and updating its Q-table accordingly. The Q-table, a fundamental component of the RL algorithm, encapsulates the agent's learned knowledge, mapping the combinations of emotional states and potential actions to their respective expected rewards or values. The final Q-table, shown below, represents the culmination of this training process, capturing the optimal action choices for each emotional state encountered. This data-driven approach allows the RL agent to make informed decisions, enabling the delivery of personalized advice tailored to the individual's emotional needs, ultimately facilitating a more empathetic and supportive learning experience.

A black screen with white text

Description automatically generated

Figure 81: Training RL Agent

A black screen with white numbers

Description automatically generated

Figure 82: Updated Q-Table

## **Research Findings**

The primary aim of this research was to develop an adaptive online learning platform that addresses the emotional needs of primary school students. Through a comprehensive approach involving quizzes, emotion detection models, reinforcement learning agents, and language models, we were able to create a system that provides personalized guidance and support tailored to each student's emotional state.

**Emotion Classification Model**

To enable accurate emotion detection from student responses, I developed a BERT-based model fine-tuned on a custom dataset of 760 records. This dataset was curated through a survey involving 25 primary students, with questions spanning family, school, and social contexts. The BERT model achieved an impressive 74% accuracy on this dataset, outperforming the 73% accuracy of a baseline Naive Bayes model. This highlights the effectiveness of leveraging large language models like BERT for emotion classification tasks, even with limited data availability.

**Reinforcement Learning Agent**

A reinforcement learning (RL) agent was trained using the complete dataset to learn optimal action selections based on the detected emotions and provided feedback. The resulting Q-table encapsulates the agent's learned knowledge, mapping emotional states to appropriate actions. This data-driven approach allows the agent to make informed decisions, enabling the delivery of personalized advice tailored to each student's emotional needs.

**Language Model for Personalized Advice**

The selected actions from the RL agent are then used as input to a large language model (LLM), which generates personalized advice for the teachers. This advice provides guidance on how to effectively support and address the emotional needs of individual students, fostering a nurturing and conducive learning environment.

Overall, our research findings demonstrate the potential of leveraging advanced machine learning techniques, such as BERT, RL, and LLMs, to create an adaptive and emotionally supportive online learning platform for primary students. By addressing emotional well-being, this platform aims to enhance the learning experience and promote a more inclusive and supportive educational environment.

## **Discussion**

### **Model Performance and Comparative Analysis**

The BERT-based emotion classification model demonstrated promising performance, achieving an accuracy of 74% on the custom dataset. This result is particularly noteworthy given the relatively small dataset size of 760 records. By leveraging BERT's pre-trained knowledge and fine-tuning it for the specific task of emotion detection, the model was able to outperform the traditional Naive Bayes approach, which achieved an accuracy of 73%. This highlights the potential of large language models like BERT in capturing the nuances of emotional expressions, even with limited data availability.

### **Practical Implications and Applications**

The developed adaptive online learning platform has significant practical implications for enhancing the educational experience of primary school students. By addressing emotional well-being, the platform fosters a more inclusive and supportive learning environment, potentially leading to improved student engagement, motivation, and academic performance. The personalized advice provided to teachers can assist them in better understanding and supporting their students' emotional needs, enabling more effective interventions and personalized learning strategies.

Moreover, this approach has applications beyond the educational domain. The emotion detection and personalized guidance components can be adapted for various contexts, such as mental health support systems, customer service chatbots, or even personal assistants that provide emotional support.

### **Limitations and Future Directions**

While the research findings are promising, there are certain limitations to be addressed. The dataset used for emotion classification, although curated with care, is relatively small and may not capture the full range of emotional expressions and nuances. Expanding the dataset with a larger and more diverse sample could further improve the model's performance and generalizability.

Moreover, the data was collected only from students of a single school, which may not accurately represent the diverse socioeconomic, cultural, and educational backgrounds present across different school environments. To ensure fairness and generalizability, future data collection efforts should involve primary students from various schools with different facilities, resources, and rural/urban situations.

Additionally, the current system relies on student responses to quizzes as the primary input for emotion detection. Incorporating multimodal data, such as facial expressions, tone of voice, or physiological signals, could provide a more comprehensive understanding of the student's emotional state, leading to more accurate and holistic guidance.

Future research could also explore the integration of real-time feedback loops, allowing for continuous refinement of the reinforcement learning agent's decisions and the language model's advice based on the observed outcomes and teacher feedback.

### **Ethical Considerations**

The development and deployment of such an adaptive online learning platform raises important ethical considerations. Privacy and data security concerns must be addressed, as the system handles potentially sensitive information about students' emotional states. Robust protocols for data anonymization, encryption, and secure storage should be implemented.

## **Student Contribution**

* Implemented the “Emotional Weakness Addressing by Giving Personalized Advice” component.
* Implemented the backend code for the component.
* Created the front-end application for the project.
* Created two different model architectures for the component (BERT and RL)
* Tested the backend using Postman.
* Attended project meetings regularly.
* Contributed to integrating the final application.
* Wrote the group research paper and edited and proofread it.

# **CONCLUSIONS**

This research project aimed to develop an adaptive online learning platform that addresses the emotional needs of primary school students, fostering a more inclusive and supportive educational experience. Through a comprehensive approach involving emotion detection models, reinforcement learning agents, and language models, I have successfully created a system that provides personalized guidance and support tailored to each student's emotional state.

The BERT-based emotion classification model demonstrated promising performance, achieving an accuracy of 74% on a custom dataset of 760 records. By leveraging the power of large language models and fine-tuning them for the specific task of emotion detection, our approach outperformed traditional methods, highlighting the potential of advanced machine learning techniques in capturing emotional nuances.

Furthermore, the integration of a reinforcement learning agent and a large language model enabled the generation of personalized advice for teachers. This advice provides valuable insights and strategies to effectively support and address the emotional needs of individual students, promoting a nurturing and conducive learning environment.

The developed platform has significant practical implications for enhancing the overall learning experience and emotional well-being of primary school students. By prioritizing emotional support, we can foster increased student engagement, motivation, and academic performance. Moreover, the components of this system have applications beyond the educational domain, such as mental health support systems, customer service chatbots, or personal assistants providing emotional guidance.

Despite the promising results, there are limitations to be addressed. Future research should focus on expanding the dataset with a larger and more diverse sample, incorporating multimodal data for emotion detection, and exploring real-time feedback loops for continuous refinement of the system's decisions and advice.

Additionally, ethical considerations regarding privacy, data security, algorithmic bias, transparency, and the potential misuse of the system's outputs must be carefully addressed. Robust protocols, monitoring mechanisms, and adherence to ethical principles are crucial for the responsible development and deployment of such adaptive learning platforms.

Overall, this research project has demonstrated the feasibility and potential of leveraging advanced machine learning techniques to create an emotionally supportive and adaptive online learning environment for primary school students. By prioritizing emotional well-being, we can pave the way for a more inclusive and effective educational experience, ultimately contributing to the holistic development of our future generations.

# **REFERENCES**

1. Bates, A. W., *Teaching in a Digital Age: Guidelines for Designing Teaching and Learning (Chapter 5 & 6)*. Tony Bates Associates Ltd. , 2015. [Online]. Available: <https://pressbooks.bccampus.ca/teachinginadigitalagev1/>
2. R. E. Mayer and R. Moreno, “Nine Ways to Reduce Cognitive Load in Multimedia Learning,” *Educational Psychologist*, vol. 38, no. 1, pp. 43–52, Jan. 2003, doi: 10.1207/s15326985ep3801\_6.
3. R. Pekrun and L. Linnenbrink-Garcia, “Academic Emotions and Student Engagement,” *Handbook of Research on Student Engagement*, pp. 259–282, 2012, doi: 10.1007/978-1-4614-2018-7\_12.
4. N. R. Council, D. O. B. A. S. S. A. Education, B. O. S. Education, B. O. T. A. Assessment, and C. O. D. D. L. A. 21st C. Skills, *Education for Life and Work*. National Academies Press, 2013. [Online]. Available: <http://books.google.ie/books?id=XsqDHRSo4NcC&printsec=frontcover&dq=978-0-309-25649-0&hl=&cd=1&source=gbs_api>
5. P. Salovey and J. D. Mayer, “Emotional Intelligence,” *Imagination, Cognition and Personality*, vol. 9, no. 3, pp. 185–211, Mar. 1990, doi: 10.2190/dugg-p24e-52wk-6cdg.
6. K. D. Squire, “Video game–based learning: An emerging paradigm for instruction,” *Performance Improvement Quarterly*, vol. 21, no. 2, pp. 7–36, 2008, doi: 10.1002/piq.20020.
7. G. Bright, “Stress Detection Using Emotions: A Survey,” *International Journal for Research in Applied Science and Engineering Technology*, vol. 10, no. 5, pp. 231–234, May 2022, doi: 10.22214/ijraset.2022.42150.
8. W. Feng and X. Guoqiang, “Emotion identification method, device, and a storage medium,” *SciSpace - Paper*, Jun. 13, 2019. <https://typeset.io/papers/emotion-identification-method-device-and-a-storage-medium-4ozlic1jza>
9. S. TGDK, V. Selvarai, U. Raj, V. P. Raiu, and J. Prakash, “Emotion Detection Using Bi-directional LSTM with an Effective Text Pre-processing Method,” *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Jul. 2021, **Published**, doi: 10.1109/icccnt51525.2021.9579844.
10. O. C. Santos, S. Salmeron-Majadas, and J. G. Boticario , “Emotions Detection from Math Exercises by Combining Several Data Sources.” Accessed: Mar. 31, 2024. [Online]. Available: <https://link.springer.com/chapter/10.1007/978-3-642-39112-5_102>
11. B. T. Shobana and G. A. Sathish Kumar, “I-Quiz: An Intelligent Assessment Tool for Non-Verbal Behaviour Detection,” *Computer Systems Science and Engineering*, vol. 40, no. 3, pp. 1007–1021, 2022, doi: 10.32604/csse.2022.019523.
12. F. M., “Using a game-like task as an assessment of emotion recognition in children.,” *Prof Doc Thesis*, Dec. 2023, doi: <https://doi.org/10.15123/uel.8x089>.
13. D. Varshney , A. Choudhury, S. Negi , and Dr. A. M. Lal , “MULTIMODAL EMOTION DETECTION USING MACHINE LEARNING TECHNIQUES AND DATA ANALYSIS ,” *IAEME Publication*, vol. 11, no. 0976–6499, Art. no. IJARET\_11\_08\_045, Aug. 2020, doi: 10.34218/IJARET.11.8.2020.045.
14. B. Hu *et al.*, “Enabling Intelligent Interactions between an Agent and an LLM: A Reinforcement Learning Approach,” *arXiv*, Art. no. 2306.03604, Mar. 2024, doi: <https://doi.org/10.48550/arXiv.2306.03604>.
15. J. D. Chang, K. Brantley , R. Ramamurthy, D. Misra, and W. Sun, “Learning to Generate Better Than Your LLM,” *arXiv*, Art. no. 2306.11816, Nov. 2023, doi: <https://doi.org/10.48550/arXiv.2306.11816>.
16. D. Zhang, L. Chen, S. Zhang , H. Xu , Z. Zhao , and K. Yu, “Large Language Models Are Semi-Parametric Reinforcement Learning Agents,” *arXiv*, Art. no. 2306.07929 Search..., Oct. 2023, doi: https://doi.org/10.48550/arXiv.2306.07929.