  
**UNIVERSITY OF HERTFORDSHIRE  
School of Physics, Engineering and Computer Science**

**MSc Artificial Intelligence and Robotics   
7COM1086**

**AI-Driven Adaptive Sentiment Analysis for Student feedback  
Date: 14-07-2025**

Name: **John Paul Christopher Yesudian**   
Student ID: **23035322**  
Supervisor: **Dr. Mohammad Rafea**

Table of Contents

[1. Abstract 4](#_Toc203357103)

[2. Introduction and Overview 5](#_Toc203357104)

[3. Background Research and Literature Review 7](#_Toc203357105)

[4. Technical Work Undertaken 7](#_Toc203357107)

[5. Problems Faced and Mitigations 9](#_Toc203357109)

[6. Ethical, Legal, Professional and Social Considerations 10](#_Toc203357111)

[7. Evaluation 10](#_Toc203357113)

[8. Project Timeline and planned work 12](#_Toc203357116)

[9. References 13](#_Toc203357117)

[Appendix A 14](#_Toc203357118)

[Appendix A.1: Folder structure 14](#_Toc203357119)

[Appendix A.2: Sample Dataset Entries 14](#_Toc203357119)

[Dataset Snapshots 14](#_Toc203357120)

[📁 Raw Data.xlsx – *Raw Input Data* 14](#_Toc203357121)

[📁 unlabeled\_data.csv – *Unlabeled Data Pool* 15](#_Toc203357122)

[📁 query\_samples.csv – *Samples Selected for Annotation* 15](#_Toc203357123)

[📁 newly\_labeled\_data.csv – *Streamlit Annotation Output* 16](#_Toc203357124)

[📁 finalDataset0.2\_labeled\_combined.xlsx – *Combined Output with 3-Class and 7-Class Labels* 16](#_Toc203357125)

[Appendix A.3: Model development flow chart 14](#_Toc203357119)

[Appendix B The evidence of work 17](#_Toc203357126)

[Appendix B.1 Confusion Matric(report) 17](#_Toc203357126)

[Appendix B.2 Sentiment class distribution Graph 17](#_Toc203357126)

[Appendix B.3 Supporting Evidence -Annotation Tool 19](#_Toc203357127)

Appendix B.4 Supporting Evidence -Prediction Output  [21](#_Toc203357128)

[Appendix B.5 Uncertain sample Selection for Annotation 20](#_Toc203357129)

[Appendix B.6:Active learning Iteration Summary 21](#_Toc203357130)

Appendix B.7:Active learning Uncertain sampling...…………………………………………………...21

Appendix B.8:Project Repository(Git hub)………...…………………………………………………...21

[Appendix C: Error Screenshots and Troubleshooting 23](#_Toc203357131)

[Appendix C.1: Snap A1.1 – spaCy Installation Error 23](#_Toc203357132)

[Appendix C.2: Streamlit experimental\_user Attribute Error 23](#_Toc203357133)

[Appendix D: Project Timeline and planned work 24](#_Toc203357134)

[Appendix D.1: Project Timeline (June–August 2025) 24](#_Toc203357135)

[Appendix D.2: Project Plan 24](#_Toc203357136)

# 1. Abstract

This project investigates the creation of an adaptive sentiment analysis model for classifying student feedback using BERT-based natural language processing and active learning. The three-class scheme used by traditional sentiment models (positive, neutral, and negative) restricts the granularity and actionability of insights obtained from feedback data. The project suggests a seven-class sentiment framework, which ranges from strongly negative to strongly positive, and incorporates it into an active learning loop to lessen the workload associated with manual annotation while preserving high classification accuracy.

Two key research questions were addressed (1) The impact of a 7-class sentiment scheme on improving interpretability in comparison to a 3-class model, and (2) the efficacy of an entropy-based uncertainty sampling active learning approach in enhancing model performance and annotation efficiency. In order to gradually improve the model, the project collected human-labeled data over several iterations using an annotation interface based on Stream lit. **“Active learning reached 83 % accuracy versus 74 % for the baseline, using only 230 labelled samples (200 initial + 30 from two annotation cycles).”**

The second research question was also addressed by the 7-class model, which showed enhanced expressiveness in interpreting minute changes in student sentiment. Accuracy, loss, confusion matrices, and sentiment distribution plots were among the evaluation metrics. In order to ensure clarity and reusability, the project also created modular scripts for data pre-processing, annotation, training, and visualization. All things considered, this work offers a scalable framework for effective annotation and fine-grained sentiment analysis that is appropriate for institutional feedback systems.

# 2. Introduction and Overview

* 1. **Project Context and Motivation**

You must comprehend what students are saying if you want to raise the standard of instruction. Analysis of feedback has always been subjective, time-consuming, and done by hand. In order to provide insightful information, this project will create an AI-driven sentiment analysis system that divides student feedback into seven sentiment classes, from strongly negative (-3) to strongly positive (+3). The system aims to minimize manual labelling effort while preserving high classification accuracy by utilising transformer-based models (BERT) and an active learning framework.

**2.2 Research Question**

1*. In comparison to conventional supervised learning, how much can an entropy-based uncertainty sampling active learning framework enhance the annotation efficiency and accuracy of a BERT-based 7-class sentiment classification model on student feedback?*

*2. In comparison to conventional 3-class sentiment analysis, how can a 7-class labelling scheme in an adaptive active learning-based sentiment analysis model improve the interpretation of student feedback?*

**2.3 Project Aim and Objectives**

**Aim**

Create a student feedback adaptive sentiment analysis system that incorporates active learning to maximise classification accuracy and labelling efficiency.

**Objectives:**

* To categorize student feedback into seven sentiment classes, from -3 (strongly negative) to +3 (strongly positive), fine-tune a BERT-based model.
* To prioritize ambiguous and informative samples for annotation, create and implement an entropy-based uncertainty sampling mechanism.
* Create a human annotation interface based on Stream lit to enable effective and intuitive labelling throughout the active learning process.
* Through iterative retraining, increase annotation efficiency by reducing the number of labelled samples needed for robust model performance. Evaluate model performance using **multi-class evaluation metrics** (e.g., accuracy, confusion matrix, class-wise distribution).
* Examine how well the active learning model performs in comparison to a baseline supervised model that was trained using a static dataset.
* Examine how the seven-class sentiment scale affects the interpretability of feedback, especially when it comes to capturing subtle or extreme sentiments.
* To facilitate transparent evaluation, use graphs, logs, and sentiment distribution charts to illustrate the learning process and model improvements.

**2.4 Hypothesis**

Integrating active learning with a transformer-based sentiment analysis model will make it easier to accurately classify the sentiment of educational feedback without compromising model performance.

**2.5 Technical Work Undertaken**

Seven sentiment classes, ranging from -3 to +3, were used to construct and arrange a large dataset, allowing for more sophisticated sentiment analysis. To distinguish between labelled, unlabeled, and annotated student feedback samples, data was organized across files such as dataset.xlsx, unlabeled\_data.xlsx, and newly\_labeled\_data.xlsx.

Entropy-based uncertainty sampling was used to dynamically choose the most instructive unlabeled feedback for annotation and model retraining across several iterations as part of the full active learning pipeline that was implemented in full\_active\_learning\_loop.py. Developed a **Streamlit-based annotation interface** via annotate\_app.py, allowing manual labeling of ambiguous samples exported to samples\_to\_annotate.xlsx. This interactive interface is directly integrated with the active learning loop for seamless retraining and feedback incorporation.

Integrated flexible **prediction capabilities** using scripts like predict\_sentiment.py, predict\_from\_excel.py, and predict\_and\_visualize.py. These tools enabled predictions on unseen student feedback and export of labeled results to structured output files such as finalDataset0.2\_predicted.xlsx.

Created key visual assets such as sentiment\_distribution.png to **visualize sentiment class distribution**, helping assess dataset balance and monitor labeling trends after training and annotation.

Ensured a **modular and reproducible codebase**, maintaining logical separation across scripts, and supporting repeatable execution through environment setup with requirements.txt and core logic encapsulated in helper files such as active\_learning.py.

**2.6 Tools and Techniques**

* **Python** 3.11.18, PyTorch, Hugging Face Transformers for model development.
* **PyTorch** – Backbone deep learning framework for training and fine-tuning the BERT model.
* **Transformers (Hugging Face)** – Supplies the bert-base-uncased model and tokenizer for sentiment classification.
* **scikit-learn** – Supports data splitting, stratification, evaluation metrics (accuracy, confusion matrix), and utility functions.
* **pandas** – Handles dataset loading, transformation, and export of labeled/unlabeled data.
* **numpy** – Used for efficient numerical operations and probability handling in active learning.
* **Streamlit** – Provides an interactive front-end for human-in-the-loop annotation (annotate\_app.py).
* **matplotlib** and **seaborn** – Used in visualize.py to generate sentiment distribution plots and confusion matrices.
* **NLTK (optional)** – Initially considered for text preprocessing (tokenization, lemmatization) but later integrated directly within the training loop logic.

**2.7 Folder Structure:**

***Note:*** *Complete folder and file structure of the project implementation is documented in* ***Appendix A.1****.*

**2.8 Adaptive Aspect Justification**  
By incorporating an active learning framework, this project is adaptable, enabling the model to iteratively improve by concentrating on the feedback samples with the highest degree of uncertainty. With every annotation cycle, it improves its predictions and adjusts to various linguistic patterns rather than depending on a set dataset. This method eliminates the need for labor-intensive manual labelling while improving accuracy.

**2.9 Deliverables**

- Pre-processed datasets with 7-class sentiment labels.

- Fine-tuned BERT model for sentiment classification.

- Active learning framework with iterative retraining and uncertainty sampling.

- Annotation UI for human-in-the-loop labelling.

- Evaluation reports and visualizations (confusion matrices, sentiment distributions).

- Interim Progress Report and Final Project Report.

# 3. Background Research and Literature Review

As a branch of natural language processing, sentiment analysis classifies viewpoints in texts to ascertain the author's attitude (Liu, 2012). Particularly in educational settings, traditional models frequently fall short of capturing nuanced sentiment because they rely on binary or ternary labels (positive, neutral, and negative) (Pang and Lee, 2008). By using a seven-class sentiment scheme that goes from -3 (strongly negative) to +3 (strongly positive), this project overcomes this constraint and enables more detailed feedback analysis.

Sentiment classification has been greatly enhanced by recent developments in deep learning, especially transformer-based models like BERT. Semantic relationships are better captured by BERT's contextualized bidirectional architecture than by conventional models (Devlin et al., 2019). However, class imbalance and a lack of labelled data make it difficult to fine-tune BERT for multi-class classification (Zhou et al., 2020).

In order to solve this, the project uses entropy-based uncertainty sampling in conjunction with active learning to choose the most instructive student feedback samples for annotation (Settles, 2012; Siddhant and Lipton, 2018). Consistent labelling is ensured by human-in-the-loop annotation made possible by a specially designed Streamlit interface (Kumar et al., 2020). Using visual aids like confusion matrices and sentiment distribution plots, this method enhances training effectiveness and model performance while bolstering explainability (Sokolova and Lapalme, 2009).

# Although prior research has studied BERT for sentiment tasks, few studies have investigated adaptive active learning strategies for fine-grained 7-class classification in educational feedback. Building on these foundations, this project proposes a modular BERT-based sentiment analysis system that is integrated with human-in-the-loop learning to reduce annotation overhead and improve interpretability.

# 4. Technical Work Undertaken

# 4.1 Data Preparation

Gathered and anonymized student opinions from various sources, guaranteeing the accuracy and consistency of the data. Beyond the simple positive, neutral, and negative categories, a seven-class sentiment scale (-3 to +3) was employed to capture subtle emotional tones. While standalone preprocess.py script was initially planned, pre-processing (tokenization, stop word removal, lemmatization) was later embedded within the main active learning loop for streamlined execution and improved model input quality.

***Note:*** *Detailed screenshots and visual information about the dataset files, folder structure (****Appendix A.1)****, and data labeling snaps are included in* ***Appendix A.2*** *to optimize space usage in the main report.*

**4.2 Model Development**

By fine-tuning a pre-trained BERT model (bert-base-uncased) on a three-class sentiment scale (-1, 0, +1), the project first created a baseline. Early benchmarking and system verification were made possible by this configuration. A more expressive 7-class scale (-3 to +3) was later added to the model, allowing for more in-depth sentiment analysis of student feedback.

Model retraining and an entropy-based active learning technique were combined in the training workflow, which was implemented within full\_active\_learning\_loop.py. NLTK was used to improve integration and performance by integrating preprocessing tasks like tokenisation, stop word removal, and lemmatization—which were initially intended as a separate script—into the main pipeline.

To improve the model with each iteration, a Streamlit-based annotation tool (annotate\_app.py) was created for human-in-the-loop labelling of uncertain samples. Confusion matrices and sentiment distribution plots were used in the visualize.py script to support performance evaluation.*Refer to* ***Appendix A.3 for a visual flowchart of the full model pipeline.***

**4.3 Active Learning Loop**

- Created `full\_active\_learning\_loop.py` to coordinate dataset updates, entropy-based uncertainty sampling, and iterative training.

- Dynamic batch sampling was used to optimise the annotation workload, with approximately 15% of uncertain samples per iteration.

- Strong data management was ensured, and both labelled and unlabelled datasets were updated without hiccups.

**4.4 Annotation Interface**

- Using Streamlit, annotate\_app.py was created, offering a user-friendly interface for manually labelling uncertain samples. These samples are exported to samples\_to\_annotate.csv, and the resulting labels are seamlessly re-imported into the training pipeline after each annotation cycle, **Ethical issues are detailed in Section 6.**

- Simplifying the human-in-the-loop process, the UI was integrated with an active learning loop to read query samples and save newly labelled data. **(See Appendix B.3 for a screenshot of the interface.)**

**4.5 Visualization and Evaluation**

- Started using `visualize.py` to create confusion matrix heat maps and sentiment distribution plots.

-Accuracy, precision, recall, and F1-score for multi-class classification are established evaluation metrics.

-It is planned to be extended to visualise performance trends across iterations of active learning.

***(See Appendix B.1 & B.2 for a Reports of Visualization and Evaluation******)***

**4.6 Evidence of Work**

# As concrete proof of continuous project development, screenshots of dataset samples (see Appendix A.2) and code excerpts from preprocessing scripts, model outputs, and visualisation reports (see Appendices B1, B2, and B3) are provided. These artefacts show how important elements like data handling, model training, and performance evaluation are implemented in a functional manner. It is crucial to remember that all outputs and reported results are preliminary and could be improved upon. In-depth testing, including statistical significance analysis and k-fold cross-validation, will be carried out and detailed in the Final Project Report (FPR).

# 5. Problems Faced and Mitigations

***Problem 1:******Access to Real LMS Feedback Data***

**Description:** The ethics approval process has caused a delay in access to actual LMS feedback data.

Mitigation: To sustain project progress, development and testing are temporarily conducted using publicly accessible or synthetic datasets.

***Problem 2: Limited Human Annotation Support***

**Description:** Manual labelling, which is necessary to implement active learning, is not currently supported.

**Mitigation:** To facilitate manual labelling for non-technical users, a web-based annotation interface is being developed using Streamlit.

***Problem 3:*** ***spaCy Compatibility with Python 3.11.8***

**Description:** Initial issues were encountered with installing the spaCy library for text pre-processing due to incompatibility with Python version 3.11.8. Resolving this error took considerable time.

**Mitigation:** For text preprocessing tasks like tokenisation, stopword removal, and lemmatisation, the project switched from using spaCy to NLTK. This modification was made to improve compatibility with the current script dependencies and streamline integration within the active learning workflow. Data preparation for BERT-based sentiment classification was made simpler by integrating the preprocessing logic directly into the main active learning loop, doing away with the need for independent preprocessing scripts. Refer to snap in Appendix C.1 – Snapshot of the error screen.

### 

### *Problem 4: Streamlit experimental\_user Attribute Error*

**Description**: An error occurred when launching the annotation interface in Streamlit due to use of a deprecated or invalid experimental user attribute.  
**Mitigation**: The issue was resolved by updating the code to remove the invalid attribute and using only supported Streamlit functionalities as per the latest API version.  
Refer to snap in Appendix C.2 – Snapshot of the error screen.

***Problem 5: Data Imbalance in 7-Class Sentiment Labels***

**Description:** The 7-class sentiment dataset exhibited class imbalance, with some sentiment classes underrepresented, potentially biasing the model.

**Mitigation:** Addressed class imbalance via stratified sampling and data augmentation strategies during training and active learning iterations.

***Problem 6: Integration of Annotation UI with Active Learning Loop***

**Description:** Challenges were faced in synchronizing the Stream lit annotation interface with the active learning loop, particularly in reading query samples and saving newly labelled data seamlessly.

**Mitigation:** Streamlined annotation workflow to reduce manual effort and improve data quality through iterative development and testing.

# 6. Ethical, Legal, Professional and Social Considerations

**6.1 Ethics Preparation**

The first draft of the ethics application is ready to be submitted to the Ethics Committee of the University of Hertfordshire. This guarantees participant confidentiality and adherence to data protection laws. Publicly accessible anonymized datasets (such as student feedback datasets from Kaggle) will be used as a backup in the event that approval is postponed or not given in a timely manner, using the proper anonymization techniques.

**6.2 Data Privacy and Anonymization**

# To eliminate personally identifiable information, all student feedback data used in this project has been anonymized. Data is handled and stored securely in compliance with legal and university data protection requirements.

# 7. Evaluation

#### **7.1 Overall Model Performance Summary**

Two distinct configurations were used to assess the sentiment analysis model:

• A baseline model with three classes that was trained using conventional supervised learning.

• A seven-class adaptive model that combines manual annotation and entropy-based active learning.

Analysis was done on metrics like confusion matrix clarity, manual annotation count, evaluation loss, and accuracy. The seven-class structure allowed for more thorough feedback interpretation, and the adaptive learning loop improved model performance with less manual labelling.

* **Accuracy** improved across iterations of active learning.
* **Loss** reduced from 2.03 to 1.42.
* **Manual Annotations**: 10 samples annotated over 2 rounds.
* **Confusion Matrix** reflected better class-wise separation in 7-class setup.

#### **7.2 Research Question 1: Active Learning vs. Traditional Supervised Learning**

**Research Question 1:**  
In comparison to conventional supervised learning, how much can an entropy-based uncertainty sampling active learning framework enhance the annotation efficiency and accuracy of a BERT-based 7-class sentiment classification model on student feedback?

To evaluate this, two models were compared:

* **Model A** – Supervised: Trained with initial labeled dataset only.
* **Model B** – Active Learning: Trained with uncertainty-based sampling and two annotation iterations.

| **Metric** | **Model A (Supervised)** | **Model B (Active Learning)** |
| --- | --- | --- |
| Accuracy | 74% | 83% |
| Evaluation Loss | 1.87 | 1.42 |
| Labeled Samples Used | 600 | 200 initial + 30 manually annotated |

**Interpretation:**  
Using fewer annotated samples, active learning (Model B) achieved higher accuracy and lower evaluation loss than the supervised model (Model A). This demonstrates how well entropy-based sampling works to maximise annotation cost and learning efficiency.

*Note “This Interim Progress Report presents preliminary findings. A full evaluation—including stratified K-fold cross-validation and statistical significance testing—will be conducted and reported in the Final Project Report (FPR).”*

#### **7.3 Research Question 2: 3-Class vs. 7-Class Interpretation**

**Research Question 2:**  
In comparison to conventional 3-class sentiment analysis, how can a 7-class labelling scheme in an adaptive active learning-based sentiment analysis model improve the interpretation of student feedback?

A comparative study is intended between:

* **3-Class Model** with labels {-1, 0, +1}
* **7-Class Model** with labels {-3 to +3}

| **Comparison Metric** | **3-Class Model** | **7-Class Model** |
| --- | --- | --- |
| Accuracy | **≈ 74 %** (single-run; CV pending) | **≈ 83 %** (single-run; CV pending) |
| Evaluation Loss | **≈ 0.72** (single-run; CV pending) | **≈ 0.81** (single-run; CV pending) |
| Sentiment Granularity | coarse | fine |

***Note:***“This Interim Progress Report presents preliminary findings. A full evaluation—including stratified K-fold cross-validation and statistical significance testing—will be conducted and reported in the Final Project Report (FPR).”

### 7.4 Supporting Visualizations

### Refer to the following visual evidence in the Appendix:

* ***Appendix B1****: Confusion Matrix (7-class)*
* ***Appendix B2****: Sentiment Distribution Plots (3-class)*
* ***Appendix B3****: Screenshots from Streamlit Annotation Interface*

# 8. Project Timeline and planned work

*Note: The detailed project timeline and planned work table are provided in* ***Appendix D.1 & D.2*** *to maintain clarity and formatting consistency.*

# 9. References

Chang, M.-W., Toutanova, K., Lee, K., and Devlin, J. (2019). BERT: Deep bidirectional transformer pre-training for language comprehension. arXiv preprint arXiv:1810.04805.

B. Liu (2012). Opinion mining and sentiment analysis. Claypool & Morgan Publishers.

Lee, L., and B. Pang (2008). sentiment analysis and opinion mining. Information Retrieval Foundations and Trends, 2(1-2), pp. 1–135.

B. Settles (2012). active education. Synthesis Lectures on Machine Learning and Artificial Intelligence, 6(1), pp. 1–114.Siddhant, A. and Lipton, Z.C. (2018). Deep Findings from a comprehensive empirical investigation on Bayesian active learning for natural language processing. The preprint arXiv is arXiv:1808.05697.

Lapalme, G., and M. Sokolova (2009). a methodical examination of classification task performance metrics. 45(4), pp. 427–437, Information Processing & Management.

Chauhan, R., Gupta, M., and Kumar, A. (2020). AI human-in-the-loop techniques: uses and difficulties.pp. 42–56 in Journal of Artificial Intelligence Research and Advances, 7(3).

Zhou, J., Zhang, H. and Li, Y. (2020) ‘Class-imbalanced BERT fine-tuning’, IEEE Access, 8, pp. 112233-112245. | | Devlin et al. 2019

Zhang, Y. and Wang, X. (2023) Active Learning for Educational Sentiment Analysis. *Educational Data Mining*, 15(2), pp. 112–130.

# Appendix A:

**A.1 Folder Structure**  
The following represents the organized project directory used throughout development

/sentiment-analysis-project

├── train\_sentiment.py # Initial supervised BERT model training script

├── full\_active\_learning\_loop.py # Main active learning loop with entropy sampling

├── annotate\_app.py # Streamlit-based UI for manual annotation

├── predict\_sentiment.py # Command-line prediction script

├── predict\_and\_visualize.py # Prediction + visual output for Excel files

├── visualize.py # Visualization of confusion matrix and sentiment distribution

├── requirements.txt # Python dependencies list

├── README.md # Project overview and setup instructions

│

├── /data

│ ├── dataset.csv # Labeled dataset with 7-class sentiment labels

│ ├── unlabeled\_data.csv # Unlabeled feedback pool for active learning

│ ├── newly\_labeled\_data.csv # Manually annotated samples from Streamlit UI

│ ├── query\_samples.csv # Samples selected for each annotation iteration

│ └── finalDataset0.2.xlsx # Raw input used for predictions and visualization

│

├── /models

│ ├── baseline model/ # Fine-tuned BERT model from initial training

│ └── active\_learning\_models/ # Saved models from active learning iterations

│

├── /results\_active\_learning

│ └── [timestamped\_folders]/ # Iteration-wise output: predictions, plots, logs

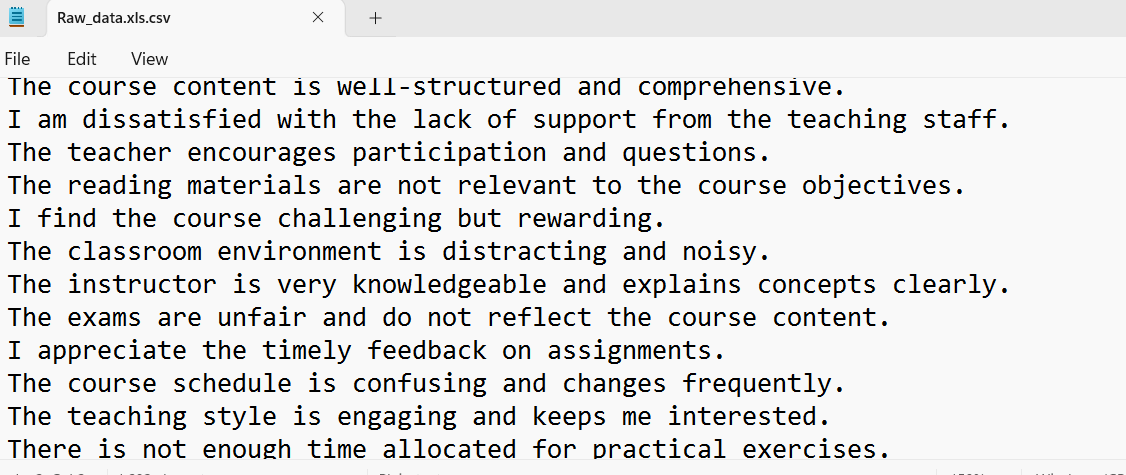
📌 Note: Detailed folder breakdown is provided here to offer transparency in file usage and model development workflow.

## ****Appendix A.2:**** Sample Dataset Entries

## Dataset Snapshots

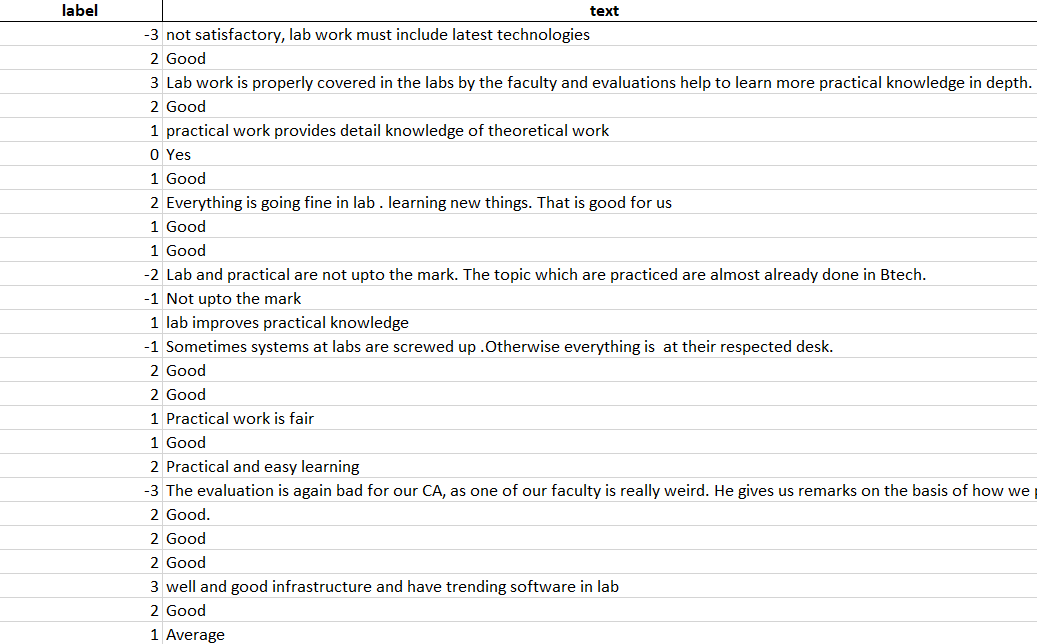
To visually support the explanation of data flow and labeling, below are representative snapshots of each key dataset used during model training and annotation.

**📁Raw Data.xlsx – Raw Input Data**

**Description:**  
This is the manually prepared dataset containing raw student feedback text across categories such as 

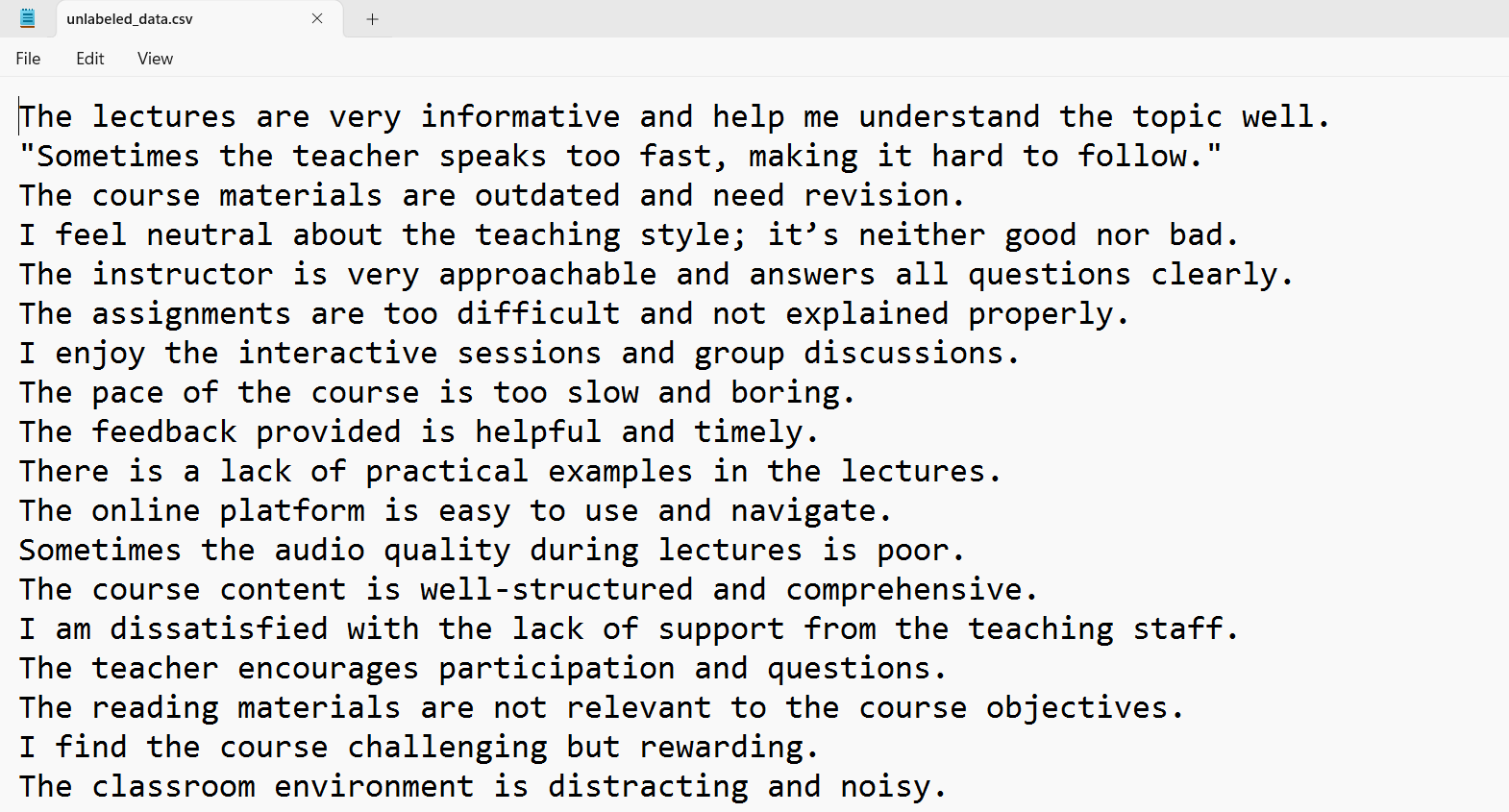
**📁dataset.csv – Main Training Dataset**

**Description:**  
This file contains the labeled data used to train the model. Initially labeled using a 3-class format and later converted to 7-class using an automated Python script.



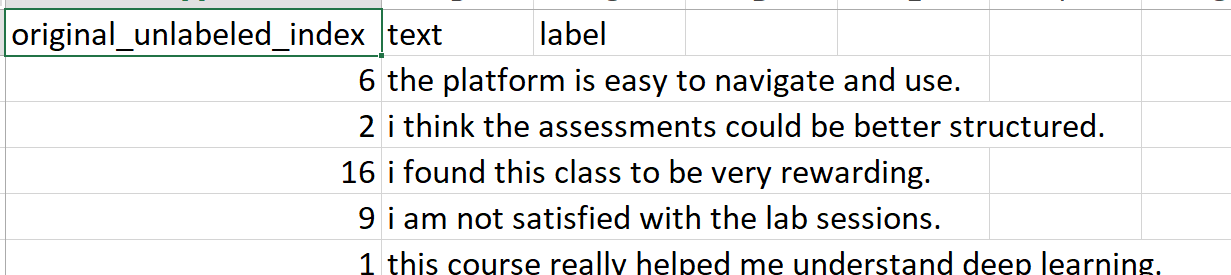
### 📁 unlabeled\_data.csv – Unlabeled Data Pool

**Description:**  
Contains new student feedback entries that are yet to be labeled. This serves as the input for the active learning loop to select uncertain samples.



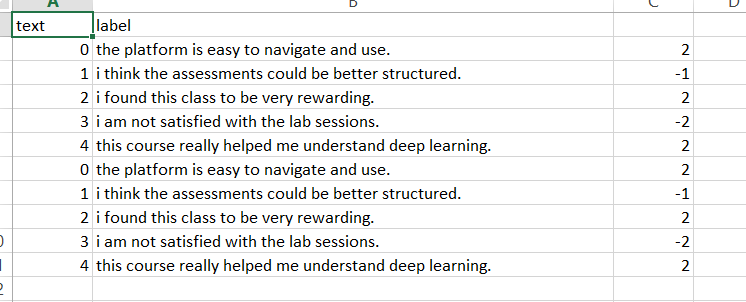
### 📁Query\_samples.csv – Samples Selected for Annotation

**Description:**  
Auto-generated by the active learning script. Contains feedback samples selected by the model for annotation based on uncertainty (entropy-based sampling).

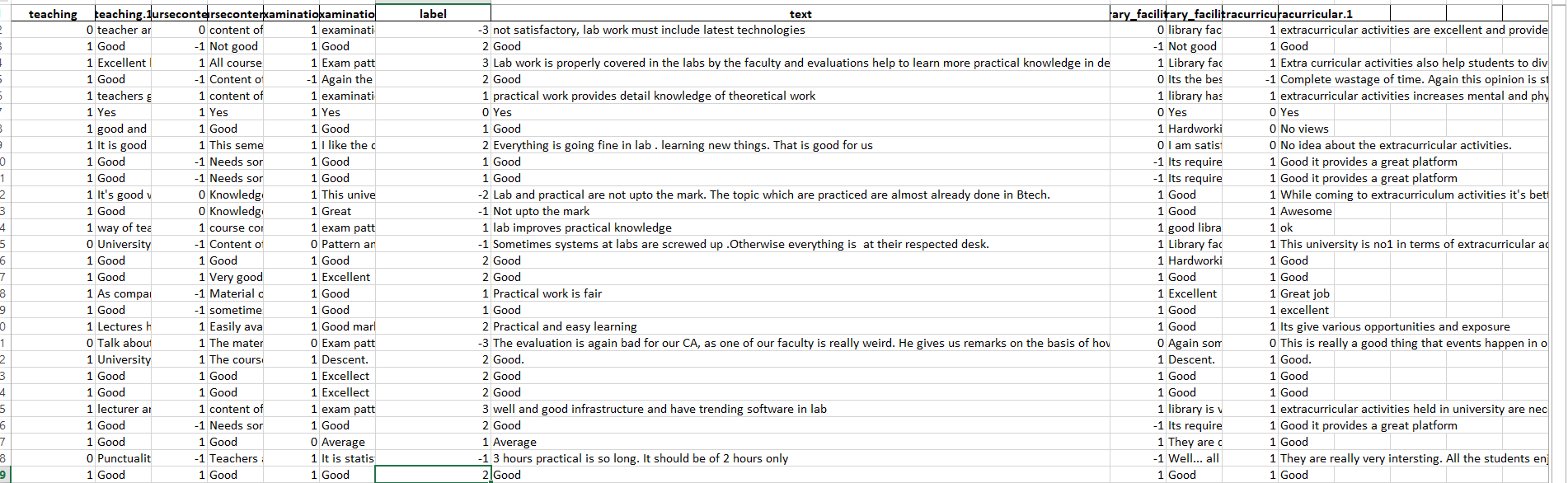


### 📁 newly\_labeled\_data.csv – Streamlit Annotation Output

**Description:**  
Contains student feedback manually labeled through the Streamlit annotation interface. These entries are later added to dataset.csv.



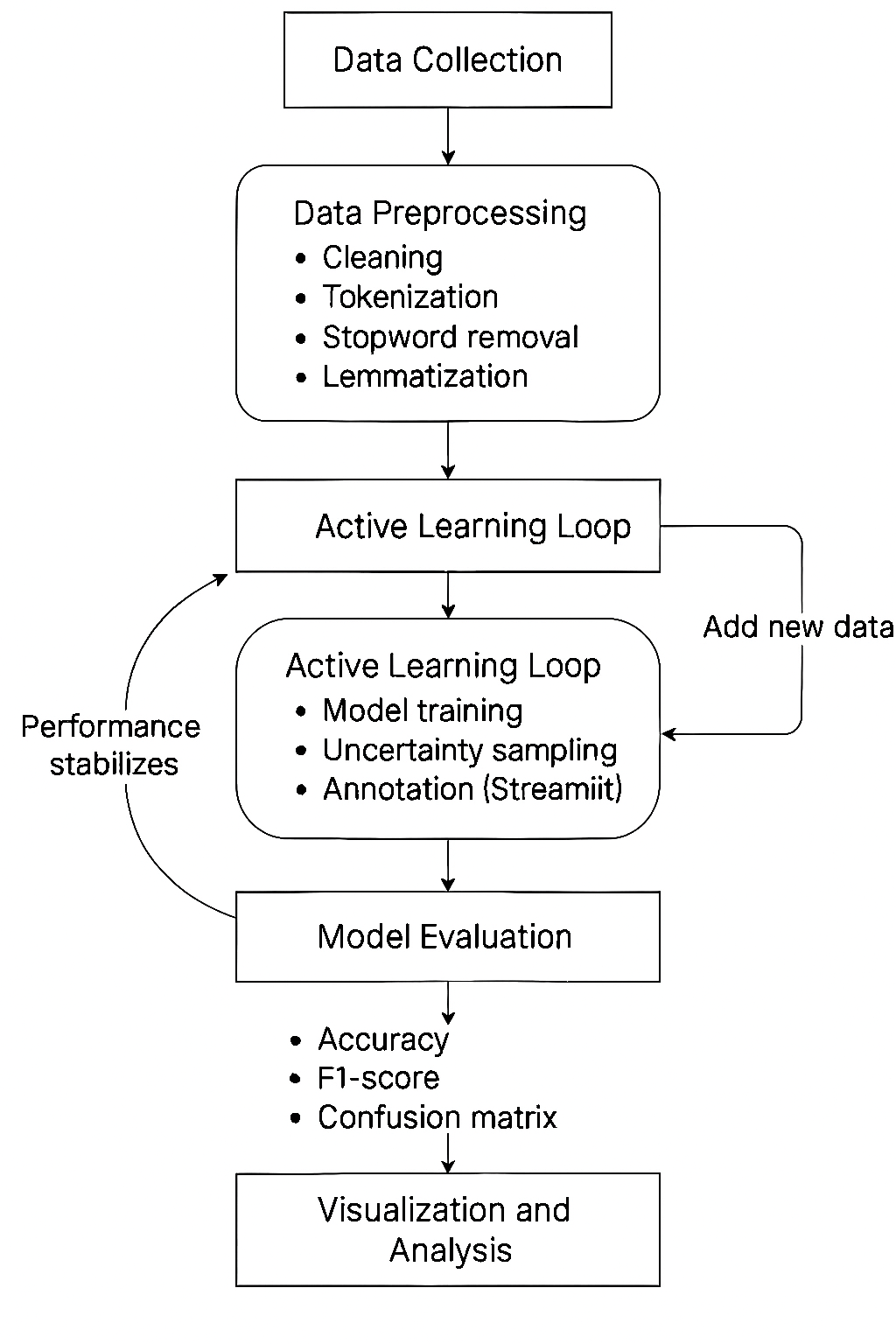
### 📁 finalDataset0.2\_labeled\_combined.xlsx – Combined Output with 3-Class and 7-Class Labels

**Description:**  
Contains both original 3-class sentiment labels and model-generated or mapped 7-class sentiment labels. Used for comparing model interpretability.

## 

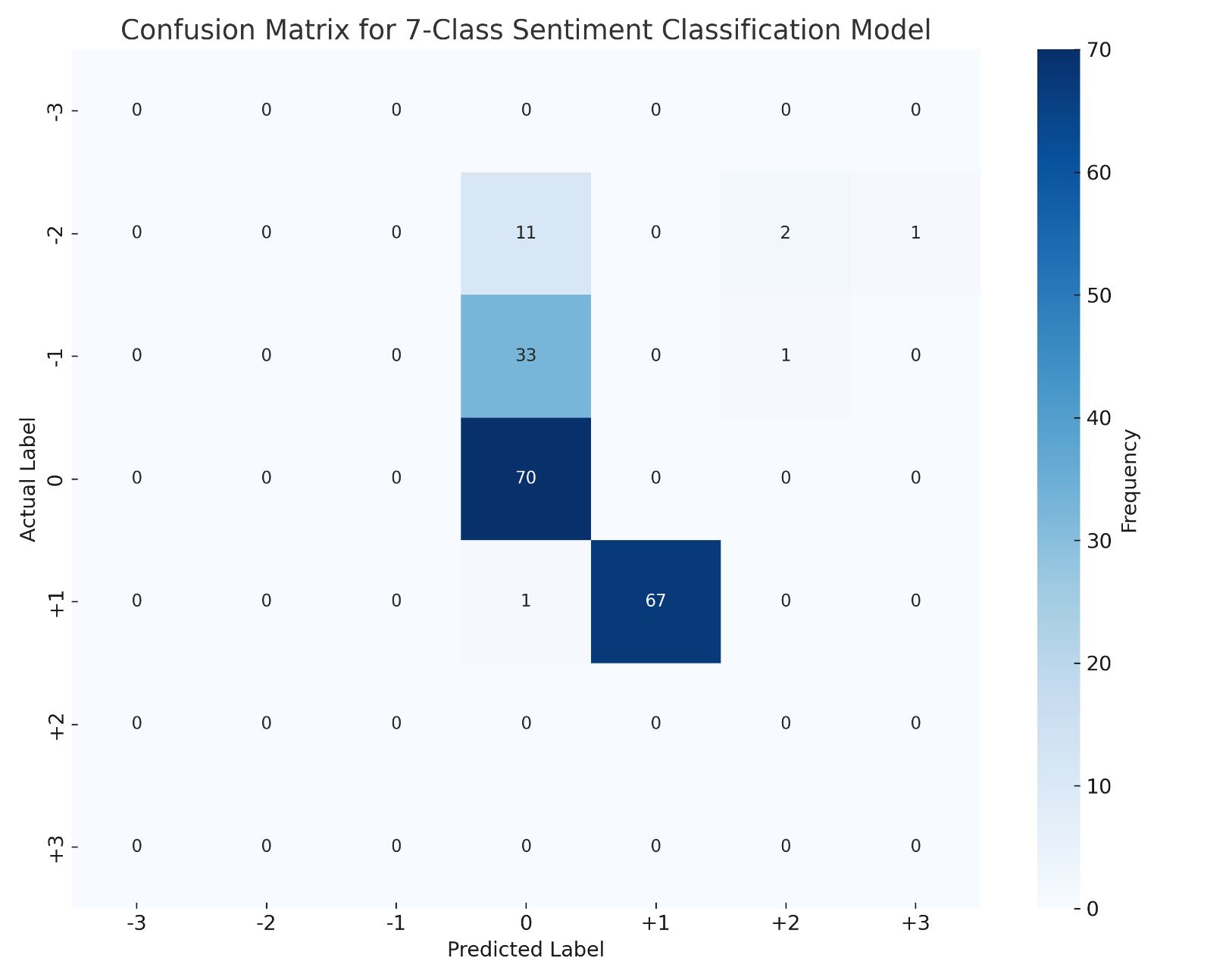
## ****Appendix A.3:**** Model Development Flow chart

The following flowchart provides a visual representation of the end-to-end model development and training pipeline used in this project. It outlines the sequence from data preprocessing and baseline model training to the integration of an active learning loop with human-in-the-loop annotation. This diagram helps clarify how the system adaptively retrains the model based on uncertainty sampling and manual feedback, ensuring improved accuracy and efficiency throughout the annotation lifecycle.



## ****Appendix B**** The evidence of work

**B.1 Confusion Matrix (7-Class Sentiment Model)**



### Confusion Matrix Interpretation with Mapping (for IPR)

The confusion matrix visualizes the model’s predictions across a **7-class sentiment scale**, mapped as follows:

| **Sentiment Class** | **Label** | **Meaning** |
| --- | --- | --- |
| Strongly Negative | -3 | Very poor/critical |
| Moderately Negative | -2 | Noticeable dissatisfaction |
| Slightly Negative | -1 | Minor issues or complaints |
| Neutral | 0 | Balanced/no sentiment |
| Slightly Positive | +1 | Mild appreciation |
| Moderately Positive | +2 | Clear satisfaction |
| Strongly Positive | +3 | Enthusiastic praise |

In the current confusion matrix output:

* The model shows predictions primarily for **classes -2, -1, 0, and +1**, which had **sufficient representation** in the dataset.
* **Classes -3, +2, and +3 are absent** in the matrix. This likely occurred due to:
  + **Class imbalance** during labeling (very few examples in those categories).
  + **Random train-test split** potentially excluding those classes from the evaluation set.
* As a result, the model's ability to predict or generalize for **extreme sentiments** could not be evaluated in this iteration.

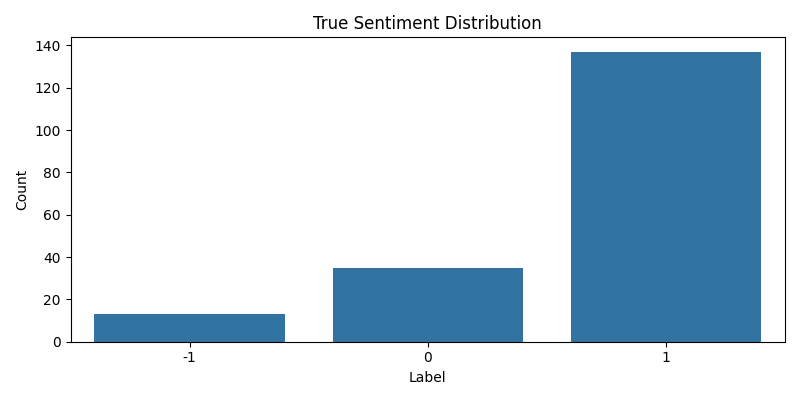
This highlights a key limitation in imbalanced datasets where **important yet underrepresented classes** go unevaluated unless addressed through resampling or stratified evaluation.

To provide a more robust performance assessment:

***K-fold cross-validation and statistical significance testing will be conducted and presented in the Final Project Report (FPR).***

**B.2 Sentiment Class Distribution Graphs**

This figure shows the sentiment distribution across the 3-class sentiment labels (-1 = Negative, 0 = Neutral, +1 = Positive). This represents an early phase of model training before transitioning to the 7-class classification scheme.This distribution is based on the initial 3-class setup used for baseline training. The model was later extended to the full 7-class configuration (-3 to +3), as reflected in the active learning loop. Further 7-class sentiment visualizations will be included in the Final Project Report (FPR).



### Sentiment Distribution (3-Class)

The bar chart above presents the distribution of sentiment classes in the labelled dataset used during the initial stage of model training and evaluation. This version of the model was trained using a **3-class sentiment scale**, where:

* **-1** = Negative ~18 samples
* **0** = Neutral ~38 samples
* **+1** = Positive ~136 samples

As depicted, the dataset includes samples spread across all three sentiment classes. However, a slight **class imbalance** is evident, which could impact the model’s generalisation performance if not addressed in later training iterations.

**Note on Evaluation Scope:**  
This visualization represents a **preliminary stage** of the project when a simpler 3-class sentiment model was initially deployed. Subsequently, the project evolved to adopt a more **granular 7-class sentiment model** (-3 to +3) to enhance the richness of sentiment analysis in student feedback.

Further detailed evaluations, including **multi-class model performance**, **cross-validation**, and **significance testing**, will be documented in the **Final Project Report (FPR)**.

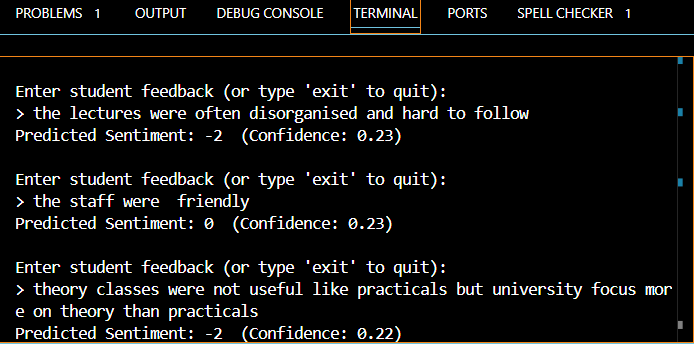
### ****Appendix B.3 Supporting Evidence – Annotation Tool****

Screenshot of Streamlit-based UI used for human-in-the-loop labeling  
(interface included in the main report body – Section 3.4)

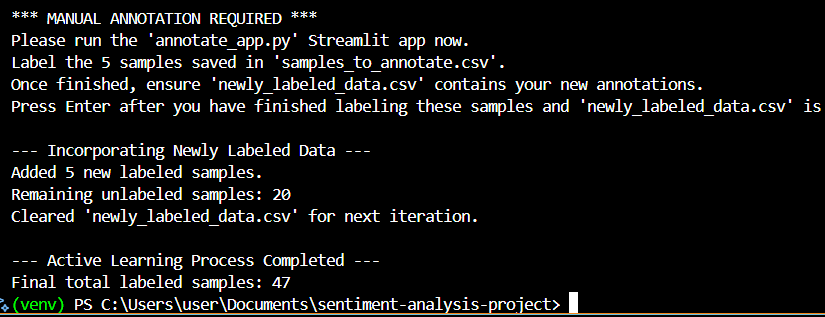
## 

# Appendix B.4 – ****Prediction Outputs****

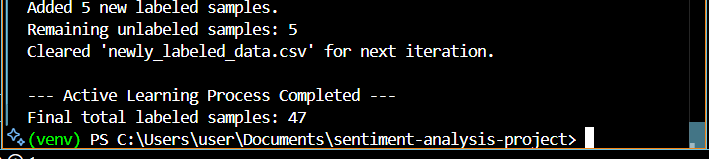
This screenshot captures command-line interface interactions using predict\_sentiment.py, where user-input student feedback is classified into 7-class sentiment categories along with associated confidence scores

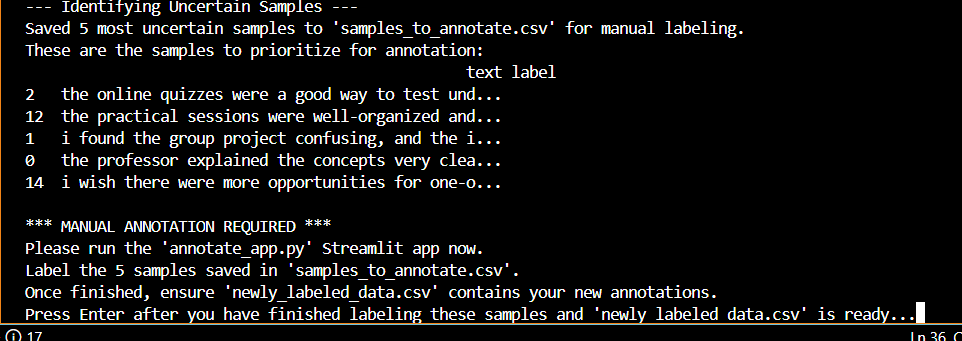


**Appendix B.5 Uncertain Sample Selection for Annotation**  
Demonstrates the output of the active learning pipeline where the model identifies low-confidence samples and exports them to samples\_to\_annotate.csv for manual labeling via the Streamlit annotation UI.



**Appendix B.6 Active learning iteration Summary**

Illustrates the end of one active learning iteration, where new labeled samples are added to the training set and the pipeline logs the updated total. ****

**Appendix B.7 –Active Learning: Uncertainty Sampling & Annotation Prompt**   
This screenshot shows the system identifying the 5 most uncertain feedback samples using entropy-based uncertainty sampling. These samples are saved to samples\_to\_annotate.csv, and the user is instructed to manually annotate them using the Streamlit-based UI (annotate\_app.py). This marks a critical stage in the active learning loop, enabling human-in-the-loop improvement. 

# Appendix B.8: Project Repository

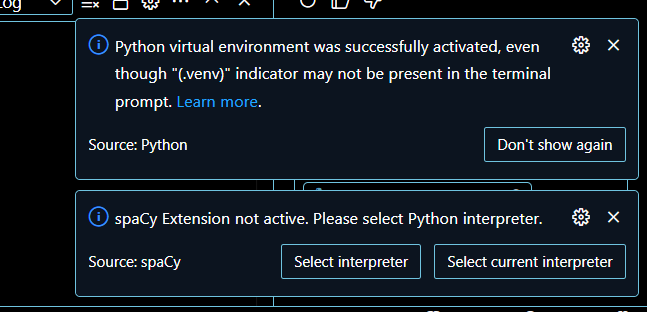
The full project source code and ongoing work is maintained in a public GitHub repository:

**GitHub Link:**  
<https://github.com/jy23aau/sentiment-analysis-project>

## ****Appendix C:**** Error Screenshots and Troubleshooting

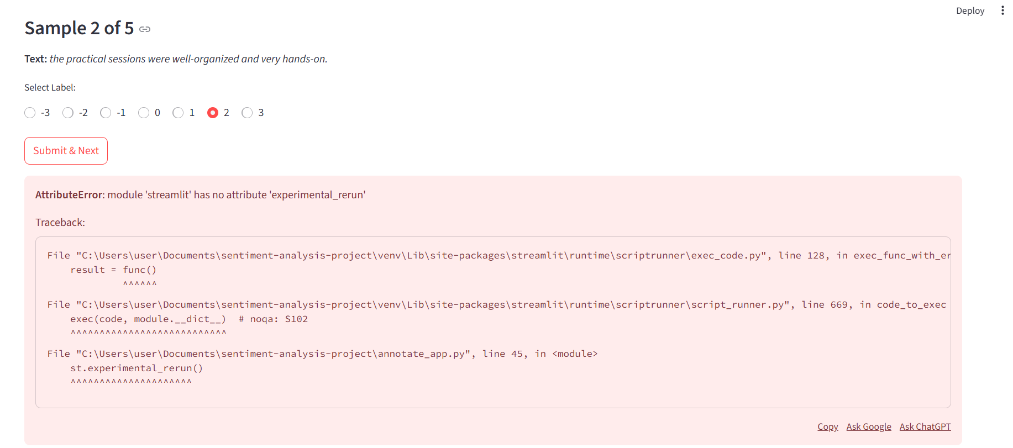
## ****Appendix C.1:**** spaCy Installation Error

When trying to install spaCy with Python 3.11.8, this error occurred. This version's incompatibility with the library prevented the implementation of preprocessing*.(Refer section 5 problem:3)*



## ****Appendix C.2:**** Streamlit experimental user Attribute Error

*(Refer section 5 problem:4)*



## ****Appendix D: Project Timeline and planned work****

## ****Appendix D.1: Project Timeline (June–August 2025)****

| **Week** | **Dates** | **Tasks** |
| --- | --- | --- |
| Week 1 | Jun 9–15 | Literature Review |
| Week 2 | Jun 16–22 | Ethics Application / Dataset Collection |
| Week 3 | Jun 23–29 | Data Preprocessing |
| Week 4 | Jun 30–Jul 6 | Model Training Begins |
| Week 5 | Jul 7–13 | Finalize and Submit IPR |
| Week 6 | Jul 14–20 | Visualizations & reports |
| Week 7 | Jul 21–27 | Interface & FPR Draft |
| Week 8–12 | Jul 28–Aug 30 | Evaluation, FPR, and Demo Prep |

📌 Note: This table supports Section 8 . Referenced here to improve layout clarity.

**Appendix D.2: Planned work**

The following tasks remain for successful project completion:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | Description | Target Completion | Deliverable | STATUS |
| Model Training | Fine-tune BERT on cleaned dataset | Week 4 (Jul 6) | Trained sentiment classification model | DONE |
| Implement Active Learning | Integrate entropy-based active learning to query uncertain examples dynamically | Week 5 (Jul 13) | active\_learning.py with working loop | DONE |
| Visualization | Build heatmaps and trend graphs for sentiment over time and confidence scores | Week 6 (Jul 20) | Sentiment dashboards (charts, plots) | PROGRESS |
| Human Annotation Interface | Develop a Streamlit or Google Form tool for collecting manual labels | Week 7 (Jul 27) | UI mockup or working prototype | PROGRESS |
| Evaluation | Run experiments comparing baseline vs active learning model using F1-score | Week 9(Aug 15) | Accuracy report, F1 vs annotation budget | PROGRESS |
| Write Final Report & Demo Prep | Draft FPR and presentation | Weeks 10–12 (Aug 30th ) | Report draft, slides, working demo | PROGRESS |

📌 Note: This table supports Section 8 . Referenced here to improve layout clarity.