

**TITLE: -** College Feedback Classifier

**NAME: -** K. Sri Anjani

**Roll NO: -** 23BCE20001

**College Name: -** Vellore Institute of Technology (VIT), Amaravati

**Branch: -** Computer science and Engineering

**Date of Submission: -** 08-07-2025

**Table of Content**

|  |  |  |
| --- | --- | --- |
| S.NO | Topic | Page.NO. |
| 1. | Introduction | 3 |
| 2. | Objectives | 3 |
| 3. | Tools | 3 |
| 4. | Methodology | 4 |
| 5. | Code snippets | 5-11 |
| 6. | GitHub link | 11 |
| 7. | Conclusion | 12 |

**1. INTRODUCTION**

Feedback from students plays a vital role in improving the academic and administrative functions of any educational institution. However, manually categorizing open-ended feedback can be inefficient and inconsistent. The **College Feedback Classifier** is an AI-powered system that automates this process.

This project uses few-shot prompting techniques with IBM watsonx.ai foundation models to classify free-text feedback into meaningful categories like *Academics*, *Facilities*, or *Administration*. It enables academic institutions to identify key strengths and problem areas based on student input.

**2. OBJECTIVES**

* To automate the classification of textual student feedback.
* To map feedback into high-level themes for easier analysis.
* To generate structured summaries and reports.
* To support data-driven decision-making for institutional improvement.

**3. TOOLS**

|  |  |
| --- | --- |
| Category | Tools/Technologies |
| Programming Language | Python |
| Libraries | pandas, sklearn, IBM WML SDK |
| AI Model | FLAN-T5 / Mistral |
| Platform | IBM watsonx.ai |
| Cloud Storage | IBM Cloud Object Storage (COS) |
| IDEs | Google Colab |

**4. Methodology:**

1. **Data Preparation**  
A dataset of student feedback is collected in CSV format. Each row contains free-text feedback and optionally a manually assigned category.

2. **Prompt Engineering**  
Few-shot examples are provided to the model, showing how feedback should be mapped to categories.

3. **Model Setup**  
IBM watsonx.ai foundation models such as flan-t5-xxl are initialized using project credentials.

4. **Classification**  
Each feedback entry is passed through the prompt for classification into a relevant category.

5. **Evaluation**  
The predicted categories are compared with ground truth labels to assess accuracy.

6. **Export**  
Results are saved to a CSV for further reporting and analysis.

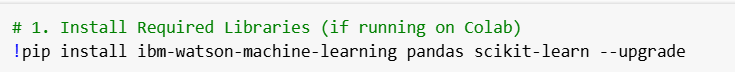
**5. Code Snippets**

Step 1: Install Required Libraries

Installs necessary Python packages for your project.

These include:

* IBM-Watson-machine-learning: To use IBM Watson foundation models.
* pandas: For data handling and processing.
* scikit-learn: For model evaluation tools like classification report.



**Step 2: Import Required Libraries**

* These libraries allow:
  + Reading data (pandas)
  + Splitting data into train/test sets
  + Evaluating results
  + Communicating with IBM Watson’s AI services

A screenshot of a computer program

AI-generated content may be incorrect.

**Step 3: IBM Cloud Credentials Setup**

Takes your **API key** and **Project ID** as input so you can access IBM Watson foundation models securely.

A close-up of a computer screen

AI-generated content may be incorrect.

Creates a connection (client) to your IBM Cloud project using the credentials.

A screen shot of a computer code

AI-generated content may be incorrect.

**Step 4: Load Feedback Dataset**

Loads the dataset (CSV file) containing feedback and categories.

Removes any rows that have missing values.

Displays the first few rows to confirm the data was loaded correctly.

A computer screen shot of a computer

AI-generated content may be incorrect.

**Step 5: Split Data into Train and Test**

Splits the dataset:

* X: Feedback text (input)
* y: Category (label)

80% of the data is used for training, 20% for testing.

A close-up of a computer code

AI-generated content may be incorrect.

**Step 6: Prepare Few-shot Examples**

Selects **one feedback example per category** to teach the model what each category looks like.

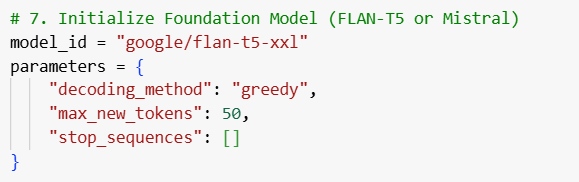
This is called **few-shot learning**, where a few examples are given to the AI model before asking it to classify new text.

A computer code with text

AI-generated content may be incorrect.

**Step 7: Initialize Foundation Model**

Chooses the AI model: FLAN-T5-XXL, a large language model from Google, hosted on IBM Watson.



Connects to the selected model using the credentials and settings.

A screenshot of a computer program

AI-generated content may be incorrect.

**Step 8: Define Classification Function**

This function:

1. Combines the few-shot examples with the new feedback.
2. Sends the full prompt to the model to **predict the category**.
3. Returns the predicted category.

A screen shot of a computer code

AI-generated content may be incorrect.

**Step 9: Run Predictions on Test Set**

For each feedback in the test set, the function sends it to the model and stores the predicted category in y\_pred.

A close-up of text

AI-generated content may be incorrect.

**Step 10: Evaluate Model Accuracy**

Compares predicted categories (y\_pred) with actual categories (y\_test).

Prints precision, recall, and F1-score for each category.

A close-up of a report

AI-generated content may be incorrect.

**Step 11: Analyze Distribution**

Counts how many times each category was predicted.

Helps you see if the model is biased toward any category.

A close-up of words

AI-generated content may be incorrect.

**Step 12: Save Output**

Creates a new table (DataFrame) that includes:

* Feedback text
* Actual category
* Predicted category

Saves this as a CSV file for your records.

A close-up of a computer screen

AI-generated content may be incorrect.

**FINAL OUTPUT**

**A close up of a text

AI-generated content may be incorrect.**

**A close up of a document

AI-generated content may be incorrect.**

**6. GitHub link**

<https://github.com/srianjani2005/College-Feedback-Classifier>

**7. Conclusion**

The **College Feedback Classifier** project successfully automates the task of categorizing student feedback using advanced foundation models. It replaces time-consuming manual processes with AI-powered classification, making it easier for college administrators to identify strengths and issues across academic, administrative, and facility domains. The use of IBM watsonx.ai ensures highly relevant and context-aware predictions. This system can be further extended to real-time dashboards and alert systems in future versions.