Project Report

Student Feedback Classifier

Name: Sushen Grover
Reg No: 23BCE1728
University: VIT Chennai
Course: GEN AI Using IBM Watsonx
Submission Date: July 3, 2025

Table of Contents

Introduction

Student feedback is a crucial element in understanding and improving the overall quality of higher education. It provides firsthand insights into a wide range of campus life areas such as academics, faculty engagement, infrastructure quality, hostel accommodations, extracurricular activities, health services, and transportation.

In this project, we aim to bridge that gap by leveraging the power of Generative AI, specifically through the FLAN-T5 model hosted on IBM Watsonx.ai, to automate the classification of open-ended student feedback. The primary goal is to categorize each textual input into one of ten meaningful labels such as Academics, Faculty, Hostel & Accommodation, Health & Wellness, and so on.

The project begins by synthesizing a dataset of 500 realistic feedback entries, each ranging from 10 to 100 words and tagged with a predefined category. The model predicts each review's category, and the results are compared against the ground truth labels to evaluate performance. The methodology is simple, transparent, and adaptable to other classification problems in the education domain and beyond.

Objective

The project focuses on:

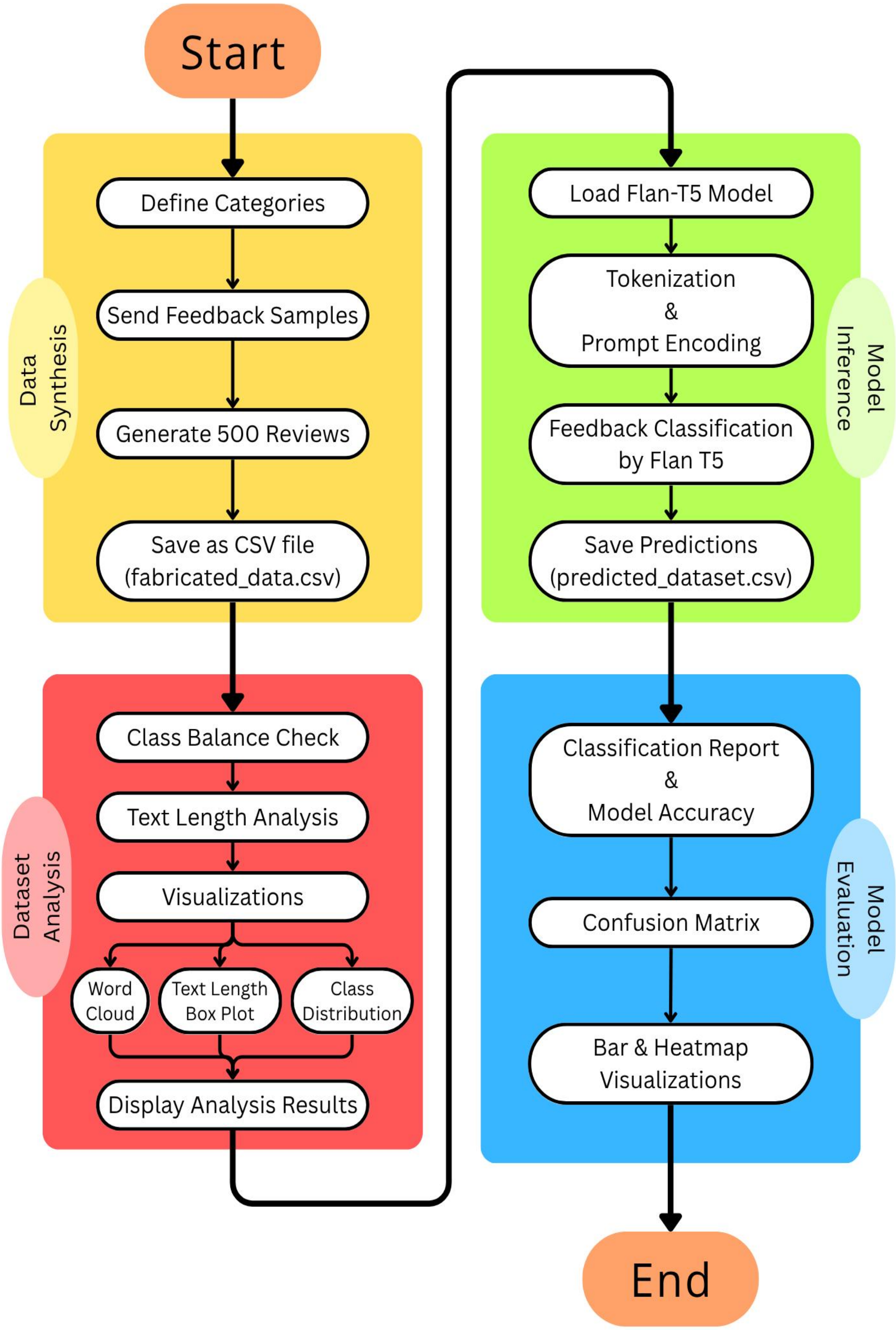
- Synthesizing a balanced dataset of student feedback entries.
- Deploying FLAN-T5 model to infer categories with high semantic understanding.
- Evaluating and showcasing the model's classification accuracy, reliability, and potential for real-world use in institutional feedback analysis.

Thus, this work highlights how Generative AI models can be adapted for text classification tasks with minimal labeled data.

Tools & Technologies Used

VS Code / Jupyter	Code development and experimentation
Python	Primary programming language
Pandas	Data handling and preprocessing
Matplotlib	Data visualization
Seaborn	Enhanced plotting and heatmaps
Scikit-learn	Evaluation metrics
Transformers	Loading FLAN-T5 tokenizer and mode
CSV	Saving/loading the dataset
Random	Dataset generation and sampling
WordCloud	Visualizing frequent words in feedback

Methodology / Working



The goal of this project is to build an automated classification system that leverages a pre-trained generative language model (Flan-T5) to categorize open-ended student feedback into ten distinct categories. The methodology follows a modular pipeline comprising data generation, preprocessing, model inference, and evaluation.

1. Synthetic Dataset Generation

- **Category Definition**: Identified 10 key feedback categories including Academics, Faculty, Facilities, and more.
- **Seed Data Creation**: Curated 5 representative feedback samples manually for each category.
- **Data Augmentation**: Applied sentence extension techniques using random suffix phrases to simulate natural language variation.
- **Dataset Assembly**: Generated 50 unique samples per category, totaling 500 reviews.
- **Storage**: Saved the synthesized dataset as a CSV file (fabricated_data.csv).

2. Exploratory Data Analysis

- **Class Distribution Check**: Used seaborn to plot review counts per category and ensure balance.
- **Text Length Analysis**: Calculated character lengths for all reviews; visualized using histograms and box plots.
- **Lexical Diversity**: Combined all review text to create a word cloud of frequent terms using the wordcloud library.
- **Summary Statistics**: Computed key metrics like average, max, and min review lengths.

3. Text Classification using Flan-T5

- **Model Initialization**: Loaded google/flan-t5-base using HuggingFace's transformers library.
- **Prompt Design**: Created dynamic prompts instructing the model to classify a given feedback into one of the 10 categories.
- **Inference Pipeline**: Tokenized input, fed it to the model, and generated output using generate() method.
- **Postprocessing**: Mapped model outputs to standardized category names for consistency.
- **Storage**: Saved predictions to a new CSV file (predicted_dataset.csv).

4. Evaluation and Visualization

- **Metric Computation**: Used sklearn to generate a detailed classification report (precision, recall, F1-score).
- **Confusion Matrix**: Constructed a confusion matrix to visualize misclassifications across categories.
- **Performance Plots**: Rendered bar plots for metrics and a heatmap for the confusion matrix to assess strengths and weaknesses.

Code Snippets with Explanation

1. Prompt Engineering with FLAN-T5

A key strength of this project lies in the prompt engineering strategy employed to adapt the FLAN-T5 model for zero-shot classification. Below is a segment of the code where a natural language prompt is generated dynamically for each student review. The prompt is then passed to the model to infer the appropriate category.

```
def classify_feature(review, prompt_type):
    prompt = (
        f"Review: {review}\n"
        f"Classify this review into one of the following categories: {',
'.join(categories)}.\n"
        "Category:"
    )
    inputs = tokenizer(prompt, return_tensors="pt", truncation=True)
    outputs = model.generate(**inputs, max_new_tokens=10)
    predicted_text = tokenizer.decode(outputs[0], skip_special_tokens=True)
    return predicted_text
```

This approach allows for a flexible and human-readable query to be interpreted by the model without requiring fine-tuning. The prompt specifies the task, presents context (the review), and expects a direct category response from the model.

2. Word Cloud Visualization

To understand the most commonly used words in the synthesized feedback, a word cloud was generated. This helped verify whether the artificial feedback retained contextual relevance and diversity. The code for this visualization is shown below:
from wordcloud import WordCloud

```
text = " ".join(df['feedback'].values)
wordcloud = WordCloud(width=1000, height=600,
background_color='white').generate(text)
plt.figure(figsize=(15, 8))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title("Common Words in Feedback")
plt.show()
```

The output image revealed terms like 'faculty', 'infrastructure', 'canteen', and 'labs' as frequent across categories, indicating strong coverage of domain-specific language.

3. Controlled Data Synthesis

Creating a balanced and varied dataset from a small set of seeds required intelligent augmentation. The following snippet shows how variation was introduced into the seed data by appending random suffixes, ensuring each sentence remained unique yet logically valid.

```
def generate_feedback(base):
    extra_phrases = [
        "Overall, the experience has been mixed.",
        "This needs urgent attention.",
        "Highly recommended improvements.",
        "I'm hopeful for future changes.",
        "This aspect exceeded expectations."
    ]
    base = base.strip('.')
    return f"{base}. {random.choice(extra_phrases)}"
```

By mixing core feedback with stochastic post-fixes, the dataset maintained linguistic variety while preserving semantic intent. This was critical in simulating realistic feedback.

4. Evaluation Pipeline with Scikit-Learn

To assess classification performance, metrics such as precision, recall, and F1-score were computed. Below is the evaluation pipeline that prints a classification report and plots a confusion matrix.

```
from sklearn.metrics import classification_report, confusion_matrix
import seaborn as sns

# Evaluation report
print(classification_report(df['category'], df['predicted_category']))

# Confusion matrix visualization
cm = confusion_matrix(df['category'], df['predicted_category'])
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=categories,
yticklabels=categories)
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```

The classification report provided per-class and macro-average metrics, while the confusion matrix visually highlighted misclassification trends, particularly between semantically close categories.

5. Handling Model Output Noise

Generative models sometimes produce unexpected outputs like typos, repeated words, or off-label predictions. To mitigate this, basic postprocessing was applied to clean the results and map them to valid category labels.

```
def clean_prediction(pred, categories):
```

```
pred = pred.strip().lower()
for cat in categories:
    if cat.lower() in pred:
        return cat
return "Unknown"
```

This function ensures robustness by identifying the closest valid label from the prediction, increasing the model's reliability without needing additional training.

Screenshots / Output Results

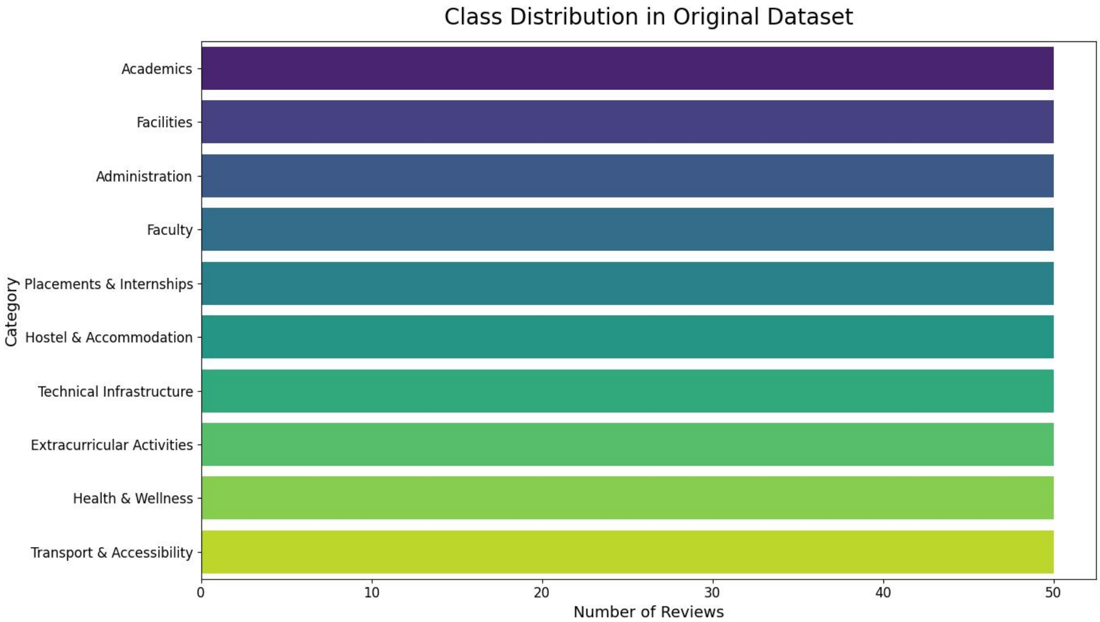
fabricated_dataset.csv

	A	B	C
1	**id**	**feedback**	**category**
2	1	The curriculum should include more practical and industry-relevant courses. Overall, the experi	Academics
3	2	Exams are conducted smoothly, but sometimes the questions are not aligned with lectures. It m	Academics
4	3	The syllabus is vast, but the professors do a great job explaining every topic clearly. Such impro	Academics
5	4	Exams are conducted smoothly, but sometimes the questions are not aligned with lectures. It's	Academics
6	5	I appreciate how we are encouraged to think critically in our assignments. This needs urgent att	Academics
7	6	The syllabus is vast, but the professors do a great job explaining every topic clearly. Overall, the	Academics
8	7	The curriculum should include more practical and industry-relevant courses. It's a step in the ri	Academics
9	8	The curriculum should include more practical and industry-relevant courses. It makes a real diff	Academics
10	9	The syllabus is vast, but the professors do a great job explaining every topic clearly. This needs	Academics
11	10	Some classes are too theoretical and lack interactive learning methods. This needs urgent atten	Academics
12	11	Some classes are too theoretical and lack interactive learning methods. Such improvements wo	Academics
13	12	I appreciate how we are encouraged to think critically in our assignments. It makes a real differ	Academics
14	13	Exams are conducted smoothly, but sometimes the questions are not aligned with lectures. It m	Academics
15	14	The curriculum should include more practical and industry-relevant courses. Overall, the experi	Academics
16	15	Exams are conducted smoothly, but sometimes the questions are not aligned with lectures. Thi	Academics
17	16	I appreciate how we are encouraged to think critically in our assignments. Overall, the experien	Academics
18	17	The syllabus is vast, but the professors do a great job explaining every topic clearly. This needs	Academics
19	18	I appreciate how we are encouraged to think critically in our assignments. It's a step in the right	Academics
20	19	I appreciate how we are encouraged to think critically in our assignments. Such improvements	Academics

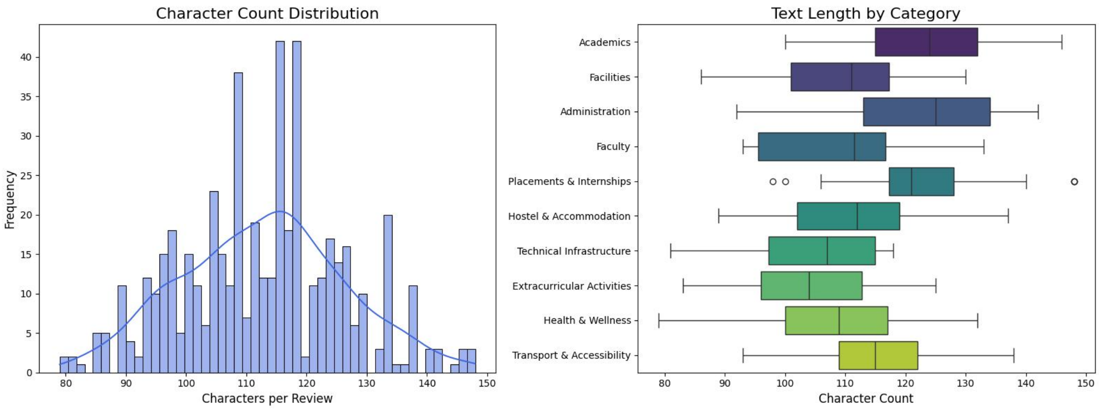
predicted_dataset.csv (after classification Google Flan T-5)

	A	B	C	D
1	id	feedback	category	predicted_category
2	1	The curriculum should include more practical and industry-relevant courses. Overall, th	Academics	Faculty
3	2	Exams are conducted smoothly, but sometimes the questions are not aligned with lectu	Academics	Academics
4	3	The syllabus is vast, but the professors do a great job explaining every topic clearly. Suc	Academics	Academics
5	4	Exams are conducted smoothly, but sometimes the questions are not aligned with lectu	Academics	Academics
6	5	I appreciate how we are encouraged to think critically in our assignments. This needs u	Academics	Academics
7	6	The syllabus is vast, but the professors do a great job explaining every topic clearly. Ove	Academics	Academics
8	7	The curriculum should include more practical and industry-relevant courses. It's a step	Academics	Academics
9	8	The curriculum should include more practical and industry-relevant courses. It makes a	Academics	Academics
10	9	The syllabus is vast, but the professors do a great job explaining every topic clearly. Thi	Academics	Academics
11	10	Some classes are too theoretical and lack interactive learning methods. This needs urge	Academics	Academics
12	11	Some classes are too theoretical and lack interactive learning methods. Such improvem	Academics	Academics
13	12	I appreciate how we are encouraged to think critically in our assignments. It makes a re	Academics	Academics
14	13	Exams are conducted smoothly, but sometimes the questions are not aligned with lectu	Academics	Academics
15	14	The curriculum should include more practical and industry-relevant courses. Overall, th	Academics	Faculty
16	15	Exams are conducted smoothly, but sometimes the questions are not aligned with lectu	Academics	Academics
17	16	I appreciate how we are encouraged to think critically in our assignments. Overall, the	Academics	Academics
18	17	The syllabus is vast, but the professors do a great job explaining every topic clearly. Thi	Academics	Academics
19	18	I appreciate how we are encouraged to think critically in our assignments. It's a step in	Academics	Academics
20	19	I appreciate how we are encouraged to think critically in our assignments. Such improv	Academics	Academics

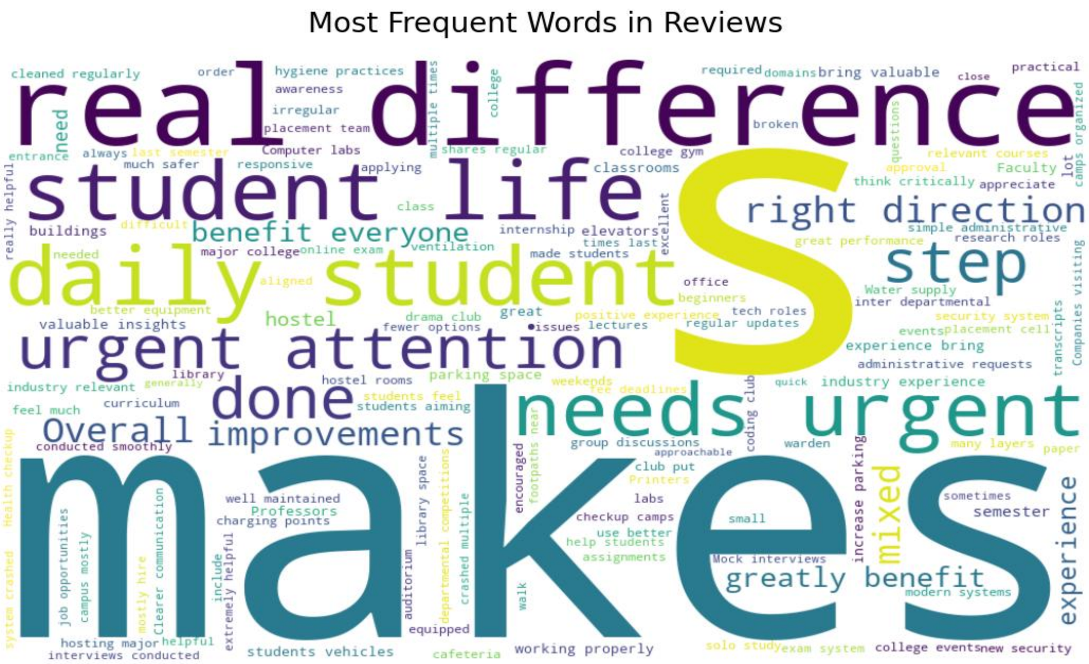
Class Distribution of fabricated data



Text Length Analysis for Fabricated Data



Word Cloud for most used words in Fabricated Data



Fabricated Dataset Summary

Dataset Summary

Total Reviews: 500 **Categories:** 10 **Average Text Length:** 113 characters **Longest Review:** 148 characters **Shortest Review:** 79 characters

Top 5 Categories:

	Count
category	
Academics	50
Facilities	50
Administration	50
Faculty	50
Placements & Internships	50

Google Flan T-5 Classification Report

Classification Report:

	precision	recall	f1-score	support
Academics	0.60	0.88	0.72	50.0
Facilities	0.37	1.00	0.54	50.0
Administration	1.00	0.72	0.84	50.0
Faculty	0.85	0.66	0.74	50.0
Placements & Internships	0.94	0.94	0.94	50.0
Hostel & Accommodation	1.00	0.86	0.92	50.0
Technical Infrastructure	0.84	0.32	0.46	50.0
Extracurricular Activities	1.00	0.72	0.84	50.0
Health & Wellness	1.00	0.56	0.72	50.0
Transport & Accessibility	1.00	0.36	0.53	50.0
micro avg	0.73	0.70	0.72	500.0
macro avg	0.86	0.70	0.72	500.0
weighted avg	0.86	0.70	0.72	500.0

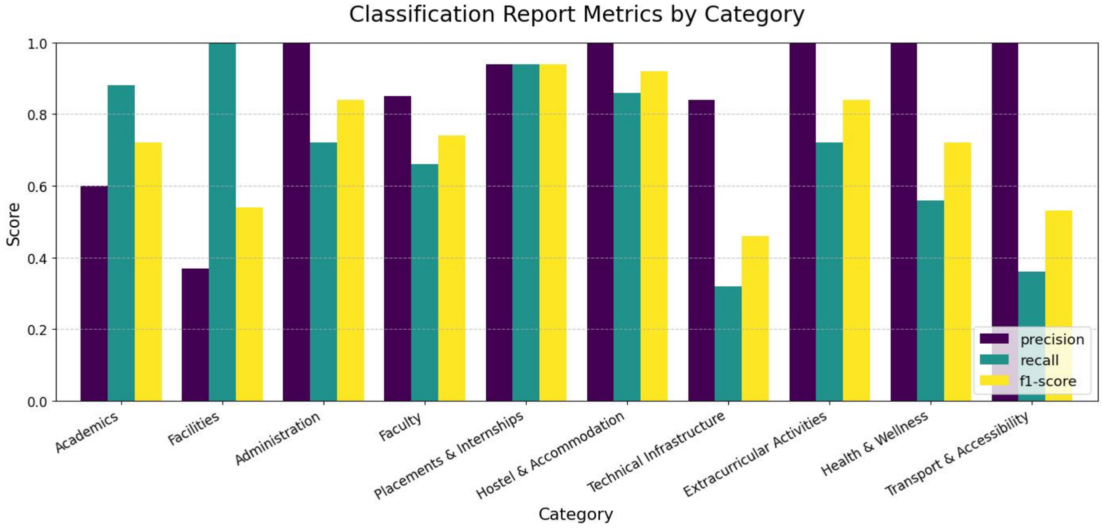
Overall Model Accuracy: 86%

Confusion Matrix for Predicted vs Actual Classifications

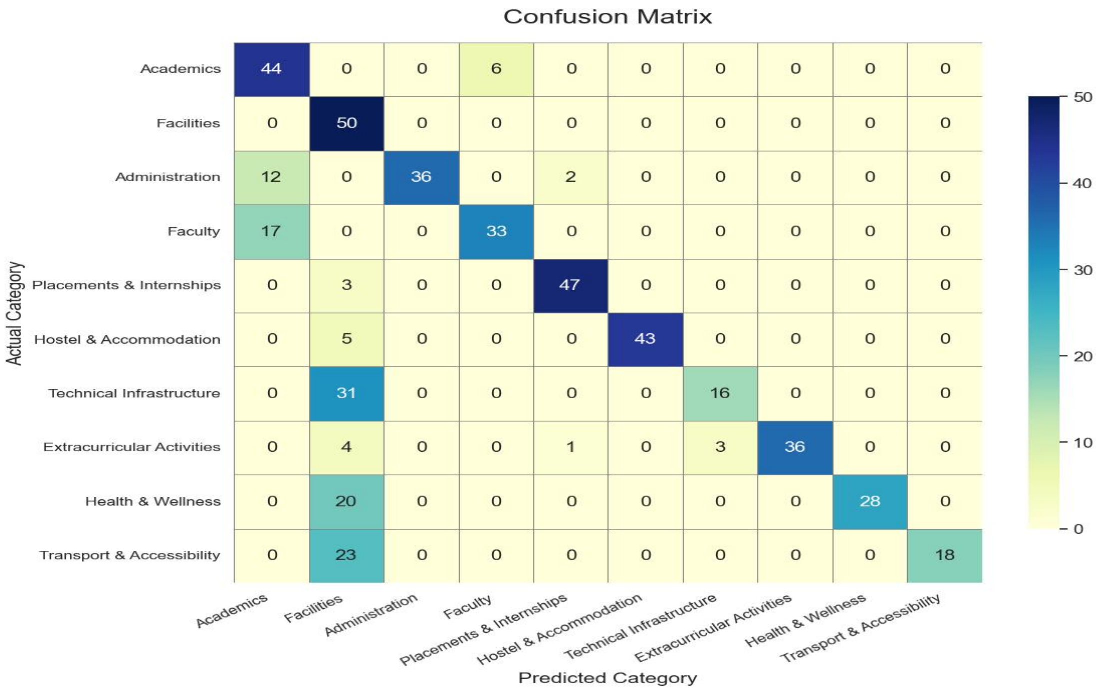
Confusion Matrix:

	Academics	Facilities	Administration	Faculty	Placements & Internships	Hostel & Accommodation	Technical Infrastructure	Extracurricular Activities	Health & Wellness	Transport & Accessibility
Academics	44	0	0	6	0	0	0	0	0	0
Facilities	0	50	0	0	0	0	0	0	0	0
Administration	12	0	36	0	2	0	0	0	0	0
Faculty	17	0	0	33	0	0	0	0	0	0
Placements & Internships	0	3	0	0	47	0	0	0	0	0
Hostel & Accommodation	0	5	0	0	0	43	0	0	0	0
Technical Infrastructure	0	31	0	0	0	0	16	0	0	0
Extracurricular Activities	0	4	0	0	1	0	3	36	0	0
Health & Wellness	0	20	0	0	0	0	0	0	28	0
Transport & Accessibility	0	23	0	0	0	0	0	0	0	18

Classification Report Metrics by Category (Bar Graph)



Confusion Matrix Visualization



Project Links

Github Repository Link
https://github.com/SushenGrover/Student-Feedback-Classifier-GenAI

Google Collab Link
https://colab.research.google.com/drive/1PFfxIsjveHeGLMta3X7ekNWC1MFhrBD7?usp=sharing

Challenges Faced & Solutions

1. Lack of Real-World Labeled Feedback Data

Challenge:
Open-ended student feedback data with labeled categories was not publicly available, which made it difficult to train or test classification models effectively.

Solution:
A synthetic dataset was created by designing seed samples for each category and augmenting them using controlled variations. This approach maintained both diversity and label integrity, enabling effective model evaluation in a simulated yet realistic setting.

2. Ensuring Prompt Consistency for Zero-Shot Learning

Challenge:
Using the FLAN-T5 model for zero-shot classification required careful prompt engineering. Inconsistent or ambiguous prompts led to poor or irrelevant outputs.

Solution:
Structured and templated prompts were designed with explicit category lists and well-defined language. Prompts followed a fixed format to reduce model confusion and increase consistency across predictions.

3. Model Output Variability and Noise

Challenge:
The FLAN-T5 model sometimes returned noisy or loosely related outputs, including typos, paraphrases of labels, or irrelevant responses.

Solution:
Postprocessing logic was introduced to normalize predictions by matching them to the closest valid category. This helped correct off-label or ambiguous outputs and ensured predictions were mapped to one of the 10 predefined classes.

4. Evaluation Complexity Due to Multi-Class Structure

Challenge:
Evaluating performance across 10 categories introduced complexity, especially when dealing with class imbalances or subtle semantic overlaps (e.g., Faculty vs Academics).

Solution:
Comprehensive evaluation using `classification_report` and `confusion_matrix` was implemented. Visual tools like heatmaps made it easier to interpret where the model struggled, guiding further dataset refinement or prompt adjustment.

Conclusion

The College Feedback Classifier project successfully demonstrated how generative AI can be applied to the automated classification of open-ended student feedback. By synthesizing a diverse and balanced dataset of 500 feedback samples across ten critical categories—such as Academics, Faculty, Facilities, and Health & Wellness—we overcame the challenge of data scarcity while ensuring semantic coverage and realism.

We leveraged the capabilities of the Flan-T5 transformer model in a zero-shot learning setup, allowing us to classify feedback without any model retraining. Carefully engineered prompts played a central role in guiding the model to produce accurate category predictions. To improve reliability, we implemented postprocessing logic that corrected noisy outputs and ensured consistency.

Comprehensive evaluation metrics, including precision, recall, F1-score, and confusion matrices, were used to validate model performance. Visualization techniques such as word clouds and heatmaps added depth to our analysis and helped uncover key trends and areas for improvement.

Overall, this project achieved its goal of building an end-to-end AI pipeline for feedback classification using synthetic data, prompt-based inference, and explainable analysis. It also lays the groundwork for future enhancements like fine-tuning models with real data and integrating this system into institutional feedback platforms.

References

1. Hugging Face Transformers Documentation
2. Google FLAN-T5 Model Card
3. Scikit-learn: Machine Learning in Python
4. Seaborn: Statistical Data Visualization

5. Matplotlib: Python Plotting Library
6. WordCloud for Python
7. Prompt Engineering Guide for LLMs
8. Zero-shot Text Classification with Transformers
9. The Illustrated T5 Model

****