# Natural Language Processing
# CS6320
# Project Report

THE UNIVERSITY
OF TEXAS AT DALLAS


Akeem Mohammed

Hritik Arasu

Yusuf Ardahan Dogru

## 1-) Contribution Over Base Model

We weren't able to make any contribution to the base model due to not having the required computational resources to train and contribute to a base model. We spent at least 4 days trying to train a base model and maybe contribute to it, but that was not doable due to the model being too large to train. Hence no possibility of contributing.

## 2-) Lessons Learned

Our initial project idea was different, however, we ran into issues in training. The idea was rather unfeasible given our devices and hardware constraints, and we kept running into problems. Therefore, one lesson from this experience is to plan properly before delving into a project, knowing the limits of the possessed resources.

In our current project, we had to deal with some new issues. One such issue was parameter tuning. We had problems working around the right constraint in classifying. The model was very sensitive towards hyperparameters as the model had to detect non-obvious patterns to classify sentiment properly.

Another problem was setting the emotionally appropriate tone. This is a feature that can be further improved in the future. Negative and positive emotions exist on a spectrum; they are very different in terms of how people convey them. However, as of now, our program can’t make clear distinctions between emotions of the same classification. For example, the tone may not be so different between a text of disgust and a text of anger, which are both classified as negative sentiments. As a future improvement, sentiments can be classified amongst themselves and the tone can be matched accordingly, providing a better user experience on the product.

**Test Sentences:**
"I’m really happy with the results." → 80% Positive
“It’s not bad, just not great.” → 53% Negative
“She is reading a book.” → 25% Neutral
“The experience exceeded my expectations.” → 25% Neutral
“The code failed to run.” → 74% Negative

**Rigorous Test Sentences:**
“The food was cold, but the waiter was incredibly kind and accommodating.” → 80% Positive.
“The product fell apart the moment I took it out of the box.” → 25% Neutral
“Oh great, another brilliant idea that crashes my browser.” → 91% Positive
“Honestly, I still don’t know how I feel about it.” → 72% Positive
“I wouldn’t say it’s terrible.” 72% Negative
“Such amazing speed — took only an hour to load.” → 78% Positive
“Beautiful packaging, disappointing content.” → 57% Positive
“It’s not bad.” → 70% Positive

**3-) Findings**

The sentiment analysis web application developed using NLTK demonstrates reasonable performance on straightforward sentences but struggles with more nuanced language. Built to classify input as Positive, Negative, or Neutral and provide a corresponding confidence percentage, the model relies heavily on surface-level lexical cues, which limits its effectiveness in handling ambiguity, sarcasm, or complex linguistic structures.

In standard test cases, the model handled clearly positive and negative sentences fairly well. For instance, the sentence “I’m really happy with the results” was correctly identified as positive with a high confidence of 80%, reflecting the presence of strong sentiment words such as “happy.” Similarly, “The code failed to run” was accurately labeled as negative with 74% confidence, a result consistent with expectations due to the explicit negative action “failed.” However, even within this basic set, the model showed cracks in its reasoning. The phrase “It’s not bad, just not great,” which is a common expression of neutrality, was misclassified as negative. This suggests that NLTK’s sentiment parser does not adequately handle negation or hedged sentiment. Furthermore, “The experience exceeded my expectations,” which carries a strongly positive connotation, was incorrectly labeled as neutral with a low 25% confidence, demonstrating the model's tendency to miss implicit sentiment unless reinforced by emotionally charged words.

When subjected to a more rigorous set of edge cases, the model’s limitations became more pronounced. Mixed sentiment sentences, such as “The food was cold, but the waiter was incredibly kind,” were classified as purely positive with 80% confidence, overlooking the negative component entirely. This illustrates NLTK’s difficulty in balancing multiple conflicting sentiments within a single sentence. In another example, “The product fell apart the moment I took it out of the box” — a clearly negative statement — was labeled neutral, which likely stems from the model's reliance on sentiment-laden vocabulary rather than understanding negative implications in plain description.

The most critical failure mode of the NLTK model was its inability to recognize sarcasm. Sentences such as “Oh great, another idea that crashes my browser” and “Such amazing speed — took only an hour to load” were both incorrectly labeled as positive with high confidence scores, 91% and 78% respectively. This points to a fundamental limitation of rule-based sentiment analysis systems: they interpret words at face value without the ability to infer tone or contradiction from context.

Similarly, the sentence “Honestly, I still don’t know how I feel about it” was misclassified as positive with 72% confidence. While the sentence clearly communicates uncertainty or emotional ambiguity, the presence of words like “feel” may have been enough for the model to lean incorrectly toward a sentiment class.

Negation also posed a persistent challenge. Phrases like “I wouldn’t say it’s terrible” and “It’s not bad” were both misclassified — the former as negative and the latter as positive — when both are more appropriately interpreted as neutral. These examples show that the model does not effectively handle the reversal or softening of sentiment that negation often implies. This is a significant issue, as negated expressions are common in everyday language, especially when users are being tactful or indirect.

In summary, the NLTK model performs reasonably well on sentences with explicit, unambiguous sentiment but consistently misinterprets more complex expressions. Its reliance on a lexical-based, rule-driven approach limits its ability to understand pragmatic elements of language such as sarcasm, mixed tone, and negation. In real-world applications where such linguistic subtleties are common, this model would require significant augmentation — either through hybrid approaches, contextual deep learning models, or custom logic — to deliver reliable and contextually appropriate sentiment analysis.

**Contributions:**

<u>Akeem Mohammed(Frontend Developer)</u>

- Responsible for creating the frontend of the application and making the project interactable.
- Responsible for ensuring that the application is functioning properly in production mode.

<u>Hritik Arasu(Project Lead)</u>

- 80 points - significant exploration beyond baseline (couple issues detected and partially solved)
- 30 points - Innovation or Creativity: Demonstrated unique approaches, such as using different tonations for different sentiments
- 10 points - Exceptional visualization usage in the user experience part

<u>Yusuf Ardahan Dogru(Backend Developer)</u>

- 10 points - Highlighted complexity: Quantized the model and used offloading to fit it into the GPU for training. Caught and fixed memory overflow issues.
- 10 points - Discussion of lessons learned and potential improvements
- 80 points - significant exploration beyond baseline (couple issues detected and partially solved)