

GenAI Project Unit 1 Submission 2

Name: Daksh Yadav

SRN: PES2UG23CS926

Section: C

Problem Statement

Customer Feedback Analyzer

Goal: Analyze 100s of product reviews to see if people are happy or angry.

Tech: pipeline('sentiment-analysis') (Positive/Negative classification).

Build an NLP application that analyzes customer reviews and classifies them as Positive or Negative using a pre-trained transformer model.

Abstract

This project implements a Customer Feedback Analyzer using Hugging Face Transformers. The system takes customer review text as input and uses a sentiment analysis model to determine whether the feedback is positive or negative. The model is loaded using the pipeline API and tested on multiple sample reviews and this project demonstrates the use of pre-trained NLP models for real-world text classification tasks.

Short Documentation

I understood that sentiment analysis is an NLP task used to determine emotional tone in the text. In this project, I used a pre-trained transformer model from Hugging Face to classify customer reviews. The pipeline API was used to simplify model loading and inference. The system accepts text input and outputs the sentiment label with a confidence score between 0 and 1. This project helped me understand how transformer models can be applied to practical scenarios like the customer feedback analysis.

Screenshots

Code Screenshot

The screenshot shows a Google Colab notebook titled "Customer_Feedback_Analyzer.ipynb". The code cell [8] contains the command `!pip install transformers`. The code cell [9] imports the `transformers` library and initializes a classifier with `classifier = pipeline("sentiment-analysis")`. The code cell [10] demonstrates testing with one sentence: `result = classifier("The product quality is very good and I love it")` followed by `print(result)`. The code cell [11] shows testing with multiple reviews, defining a list of reviews and a loop to print each review with its classification result. The status bar at the bottom indicates "NZ - IND In 1 hour" and shows system icons.

```
Customer_Feedback_Analyzer.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
[8] ✓ pip install transformers
[9] ✓ from transformers import pipeline
[10] ✓ classifier = pipeline("sentiment-analysis")
[11] ✓ Testing with one sentence
[12] ✓ result = classifier("The product quality is very good and I love it")
    print(result)
[13] ✓ Testing with multiple reviews
[14] ✓ reviews = [
    "This phone is amazing",
    "Battery life is terrible",
    "I love this product",
    "Worst service ever",
    "Good value for money"
]
for r in reviews:
    print(r, ":", classifier(r))
```

Output Screenshots

The screenshot shows the output of the code in the previous image. The heading "Testing with one sentence" is expanded. The code cell [14] shows the input: `result = classifier("The product quality is very good and I love it")` and `print(result)`. The output cell [14] shows the result: `[{'label': 'POSITIVE', 'score': 0.9998841285705566}]`.

```
result = classifier("The product quality is very good and I love it")
print(result)

[{'label': 'POSITIVE', 'score': 0.9998841285705566}]
```

Testing with multiple reviews

[15]

```
reviews = [
    "This phone is amazing",
    "Battery life is terrible",
    "I love this product",
    "Worst service ever",
    "Good value for money"
]

for r in reviews:
    print(r, "->", classifier(r))

...
This phone is amazing -> [{"label': 'POSITIVE', 'score': 0.9998575448989868}]
Battery life is terrible -> [{"label': 'NEGATIVE', 'score': 0.9995101690292358}]
I love this product -> [{"label': 'POSITIVE', 'score': 0.9998788833618164}]
Worst service ever -> [{"label': 'NEGATIVE', 'score': 0.9997915625572205}]
Good value for money -> [{"label': 'POSITIVE', 'score': 0.9998501539230347}]
```

The screenshot shows a Google Colab notebook titled 'Customer_Feedback_Analyzer.ipynb'. The interface includes a toolbar with file operations, a search bar, and a status bar at the bottom showing network, RAM, disk usage, and system information.

The notebook content is organized into sections:

- Testing with one sentence**:
[14] result = classifier("The product quality is very good and I love it")
print(result)
[{"label": "POSITIVE", "score": 0.9998841285705566}]
- Testing with multiple reviews**:
[15] reviews = [
 "This phone is amazing",
 "Battery life is terrible",
 "I love this product",
 "Worst service ever",
 "Good value for money"
]

for r in reviews:
 print(r, "->", classifier(r))

...
This phone is amazing -> [{"label': 'POSITIVE', 'score': 0.9998575448989868}]
Battery life is terrible -> [{"label': 'NEGATIVE', 'score': 0.9995101690292358}]
I love this product -> [{"label': 'POSITIVE', 'score': 0.9998788833618164}]
Worst service ever -> [{"label': 'NEGATIVE', 'score': 0.9997915625572205}]
Good value for money -> [{"label': 'POSITIVE', 'score': 0.9998501539230347}]