GEN-AI UNIT 1 HANDS-ON

Name: Chandan Chatragadda
SRN: PES2UG23CS141
Sec: C
Date: 28/01/2026

PROBLEM STATEMENT: Category 3; Project 12
Clickbait Title Detector
Goal: Classify if a video title is "Clickbait" or "Informative".
Tech: pipeline('text-classification').

ABSTRACT:
This project developed a tool for classifying video titles as "Clickbait" or "Informative" by leveraging the transformers library's pipeline for text classification. A general sentiment analysis model (distilbert-base-uncased-finetuned-sst-2-english) was utilized, and its 'POSITIVE'/'NEGATIVE' outputs were mapped to 'Clickbait'/'Informative' respectively. The final implementation includes a reusable Python function to classify titles and an interactive interface for user input, demonstrating a functional approach to categorizing video titles based on perceived sentiment.

What I Understood
Clickbait Title Detector Goal
The goal was to classify video titles as "Clickbait" or "Informative" using the transformers pipeline for text classification.

What I Built
Model Selection: A general sentiment analysis model, distilbert-base-uncased-finetuned-sst-2-english, was chosen due to its public accessibility and ability to classify text into 'POSITIVE' or 'NEGATIVE' sentiment categories.

Label Mapping: To adapt the sentiment model for clickbait detection, a custom mapping was established: 'POSITIVE' sentiment was interpreted as 'Clickbait' (as sensational titles often evoke positive, exciting tones), and 'NEGATIVE' sentiment was mapped to 'Informative' (as factual or neutral titles may be perceived as less 'positive').

Text Classification Pipeline: A text-classification pipeline was initialized using the distilbert-base-uncased-finetuned-sst-2-english model. This pipeline processes raw text (video titles) and returns a sentiment label ('POSITIVE' or 'NEGATIVE') along with a confidence score.

classify_titles Function: A Python function classify_titles(titles_list) was developed.
It accepts a list of video title strings.
It uses the pre-loaded classifier pipeline to obtain sentiment predictions for each title.
It applies a label_mapping (POSITIVE -> Clickbait, NEGATIVE -> Informative) to convert the sentiment results into the desired categories.
It returns a list of dictionaries, where each dictionary contains the original title, the mapped prediction ('Clickbait' or 'Informative'), and the confidence score.

Interactive User Input: An interactive code cell was provided, enabling users to enter multiple titles one per line.
Receive immediate classification results for their custom inputs, formatted in the same manner as the example usage.

[69]

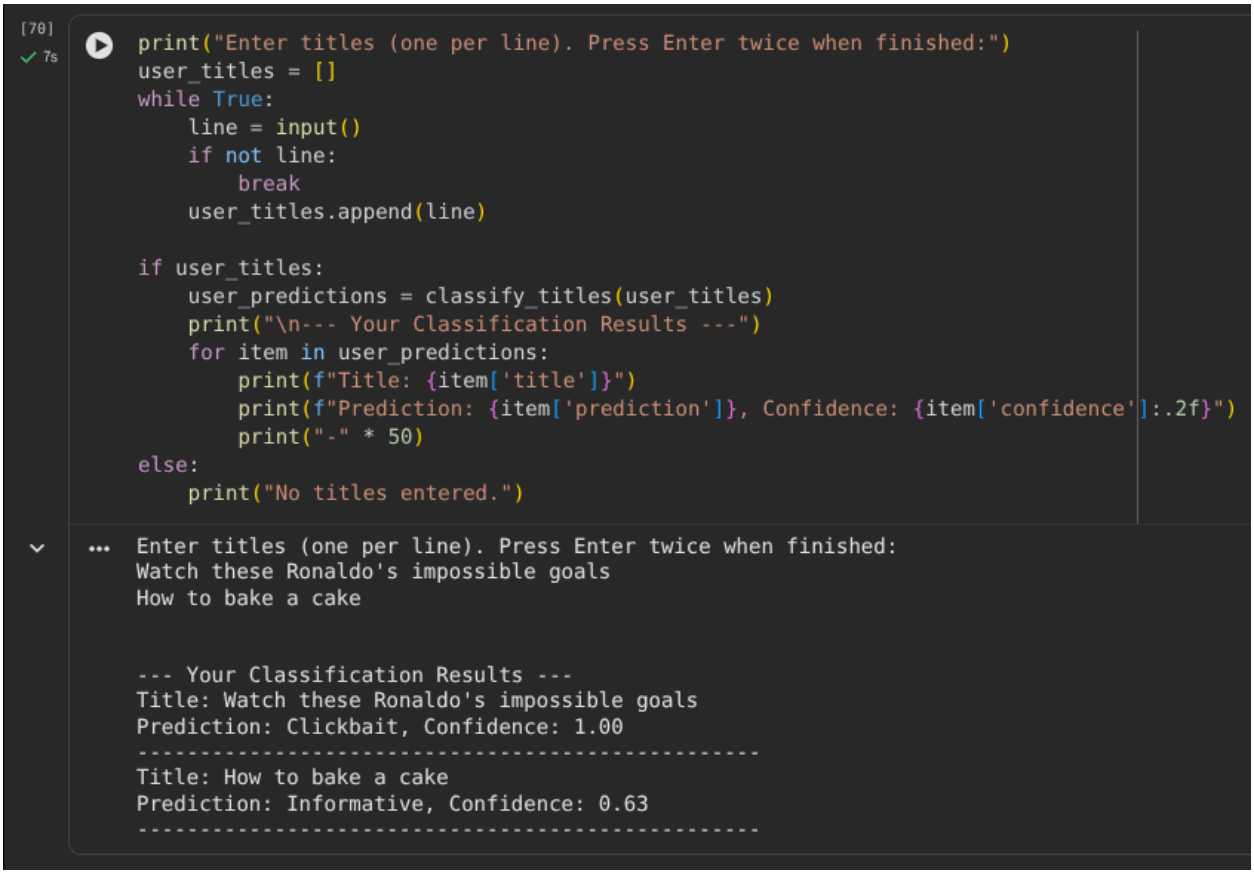
```python
titles = [
    "Watch These Impossible tricks!",
    "How to Train a Neural Network Using PyTorch",
    "This One Trick Will Change Your Life Forever",
    "Lecture 5: Linear Regression and Gradient Descent"
]

results = classifier(titles)

label_mapping = {
    'POSITIVE': 'Clickbait',
    'NEGATIVE': 'Informative'
}

for title, result in zip(titles, results):
    predicted_label = label_mapping.get(result['label'], result['label']) # Fallback
    print(f"Title: {title}")
    print(f"Prediction: {predicted_label}, Confidence: {result['score']:.2f}")
    print("-" * 50)
```

```
Title: Watch These Impossible tricks!
Prediction: Clickbait, Confidence: 1.00
--------------------------------------------------
Title: How to Train a Neural Network Using PyTorch
Prediction: Informative, Confidence: 0.78
--------------------------------------------------
Title: This One Trick Will Change Your Life Forever
Prediction: Clickbait, Confidence: 1.00
--------------------------------------------------
Title: Lecture 5: Linear Regression and Gradient Descent
Prediction: Informative, Confidence: 0.78
--------------------------------------------------
```



```
[70]
✓ 7s
print("Enter titles (one per line). Press Enter twice when finished:")
user_titles = []
while True:
    line = input()
    if not line:
        break
    user_titles.append(line)

if user_titles:
    user_predictions = classify_titles(user_titles)
    print("\n--- Your Classification Results ---")
    for item in user_predictions:
        print(f"Title: {item['title']}")
        print(f"Prediction: {item['prediction']}, Confidence: {item['confidence']:.2f}")
        print("-" * 50)
else:
    print("No titles entered.")

Enter titles (one per line). Press Enter twice when finished:
Watch these Ronaldo's impossible goals
How to bake a cake


--- Your Classification Results ---
Title: Watch these Ronaldo's impossible goals
Prediction: Clickbait, Confidence: 1.00
--------------------------------------------------
Title: How to bake a cake
Prediction: Informative, Confidence: 0.63
--------------------------------------------------
```

GITHUB LINK:
https://github.com/chandan365c/UE23CS342BA4_GenAI/tree/main/UNIT1_Project