Smart Sentiment Analyzer App

1. Objective

The goal of this project is to explore an open-source Large Language Model (LLM) from Hugging Face Hub to perform sentiment analysis on user-provided text.
This helps users understand how AI can automatically classify text as positive, neutral, or negative, and visualize prediction confidence and probability distribution.

2. Instructions & Implementation

Step 1: Environment Setup

Create a Python virtual environment and install the required libraries:

```bash
# Create virtual environment
python3 -m venv venv

# Activate environment (Linux/macOS)
source venv/bin/activate
# Windows
venv\Scripts\activate

# Install dependencies
pip install streamlit transformers torch pandas altair
```

Step 2: Load the Model
Implementations:
- **Model:** cardiffnlp/twitter-roberta-base-sentiment-latest
- **Task:** sentiment-analysis
- **Device:** GPU (if available), otherwise CPU

Prompt Example:

```python
from transformers import pipeline
import torch

device = 0 if torch.cuda.is_available() else -1

classifier = pipeline(
    task="sentiment-analysis",
    model="cardiffnlp/twitter-roberta-base-sentiment-latest",
    device=device
)

# Test prediction
result = classifier("I love this product! It works perfectly. 😄")[0]
print(result)
```

Step 3: Streamlit App for Sentiment Analysis

Features:

- User text input with sample buttons for quick testing
- Color-coded sentiment output with emoji
- Confidence score display
- Probability distribution chart using Altair
- Expandable raw model output

3. Observations

a. Positive and negative texts are classified accurately with high confidence.
b. Neutral texts are sometimes slightly biased due to model pretraining on Twitter data.
c. Visualization improves interpretability of model predictions.
d. Streamlit provides a simple interactive interface for experimentation.

4. Deliverables

- Streamlit App: Interactive interface for real-time sentiment analysis

Fig1: Input Snapshot

- PDF Report: Can include snapshots of user inputs, sentiment results, probability charts, and raw outputs.
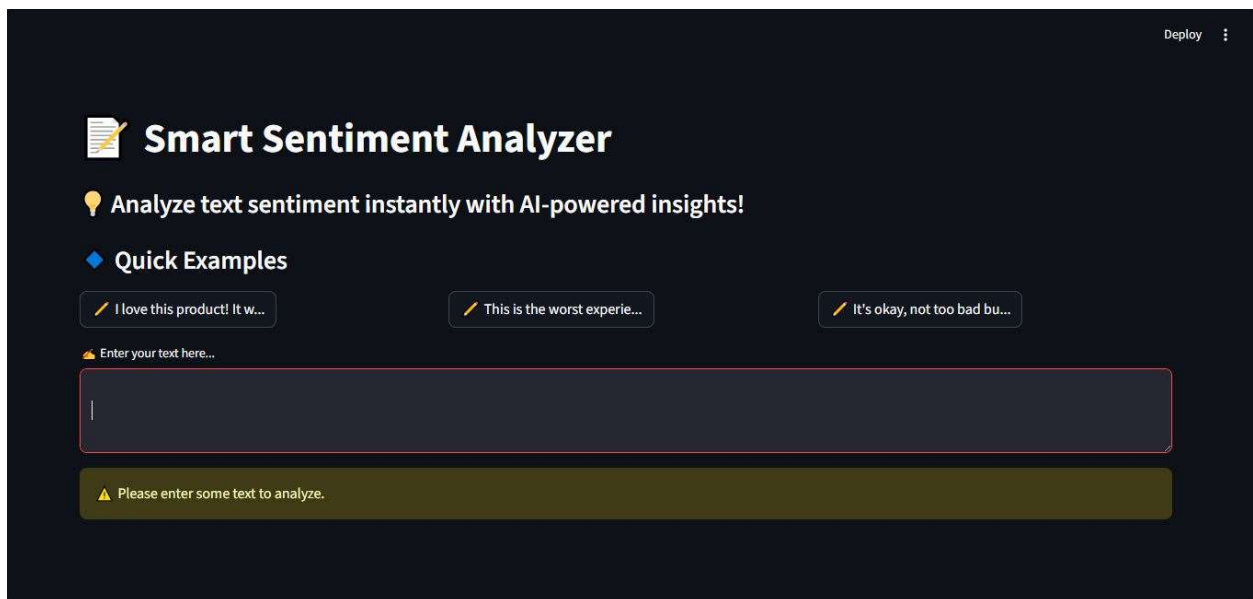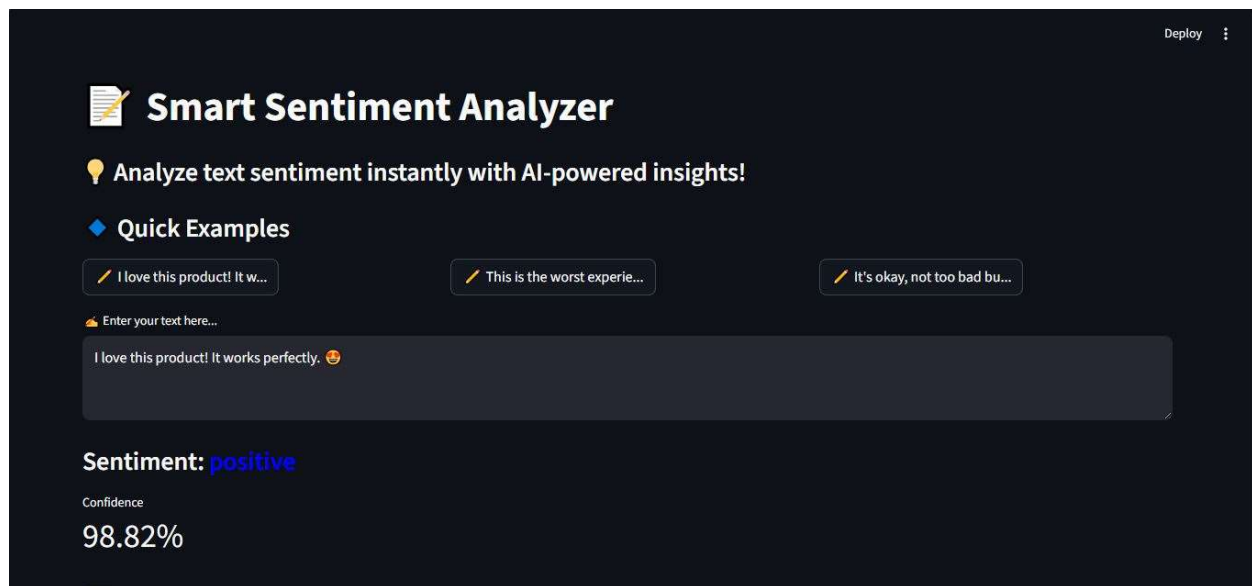
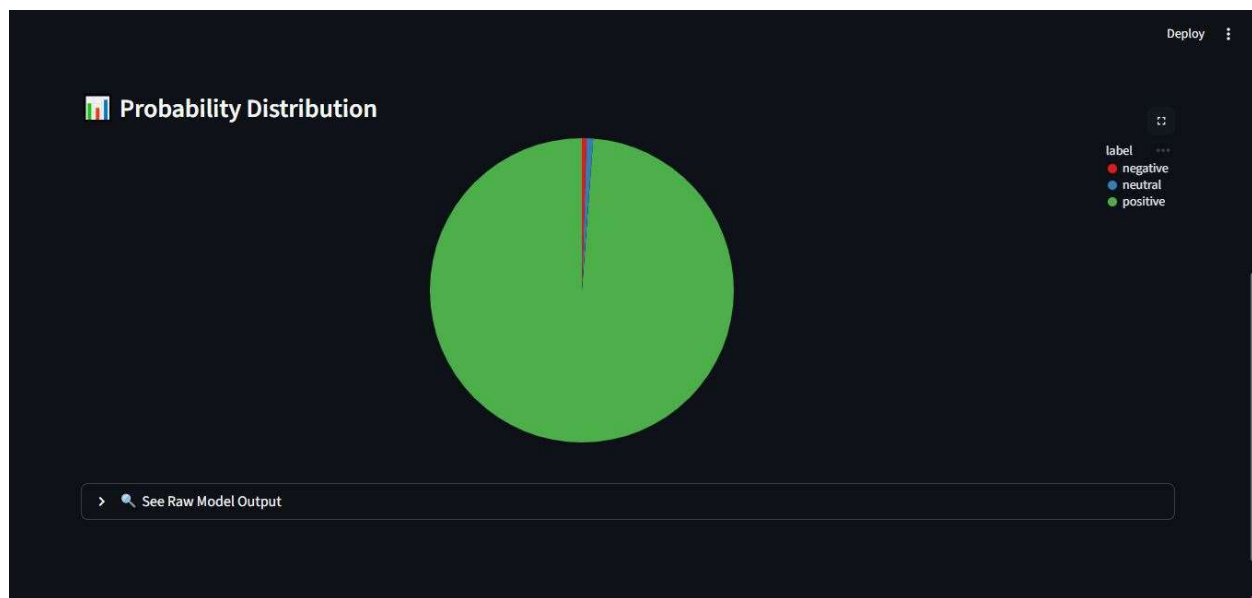## 5. Output

Fig2: Output Snapshot1
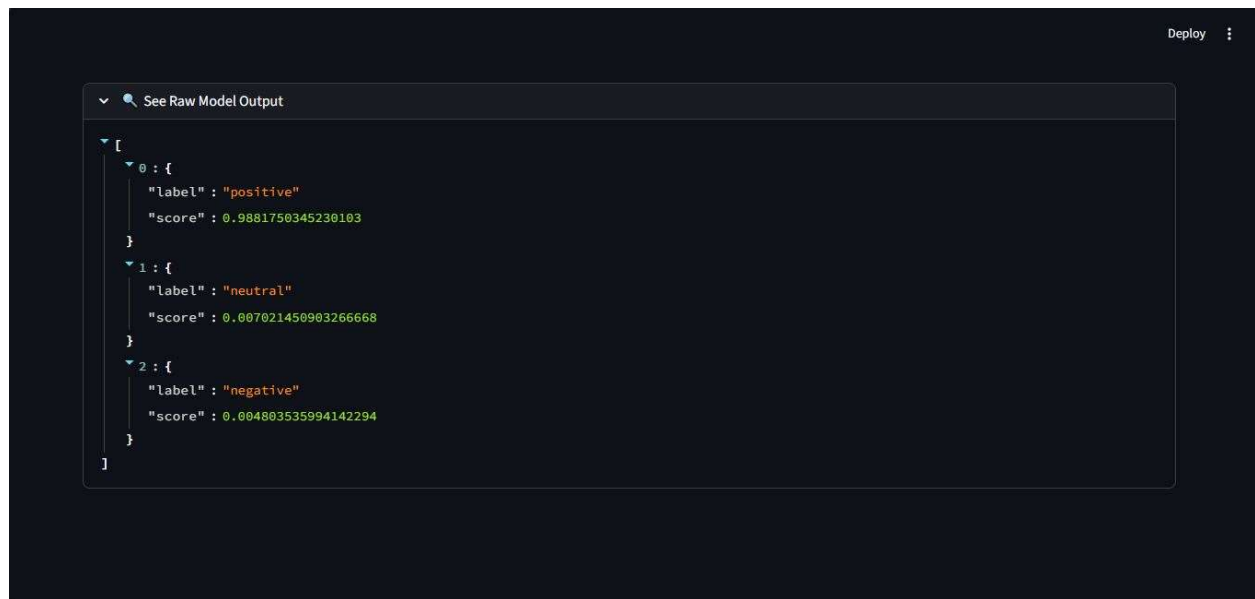
Fig3: Output Snapshot2

Fig4: Output Snapshot3

Deploy ⋮

˅ 🔍 See Raw Model Output

▼ [
    ▼ 0 : {
        "label" : "positive"
        "score" : 0.9881750345230103
    }
    ▼ 1 : {
        "label" : "neutral"
        "score" : 0.007021450903266668
    }
    ▼ 2 : {
        "label" : "negative"
        "score" : 0.004803535994142294
    }
]

Fig4: Output Snapshot3