Moringa School AI Capstone Toolkit

Project Title: AI Brand Sentiment Analyzer

Author: Ian Mworia

Executive Summary

The **AI Brand Sentiment Analyzer** is an interactive web app that compares how people feel about different brands or organizations. It simulates realistic online comments, performs **sentiment** and **emotion** analysis using Hugging Face Transformers, and visualizes the results with Streamlit and Plotly.

You can view and interact with the live app here: https://brand-sentiment-analyzer-gndc7pamezhbnbszciygwv.streamlit.app/

Watch a short demo of the project on YouTube: https://youtu.be/I40rKeM5Fx0

This project demonstrates how Generative AI can help beginners learn, experiment, and build meaningful solutions using **open-source**, **zero-cost tools**. It was developed as part of my Moringa School AI Capstone Project.

Technology Summary

Technology	Purpose
Python 3.10+	Core programming language
Streamlit	Interactive web UI
Transformers (Hugging Face)	AI model library for NLP tasks
Torch	Deep learning runtime
Plotly	Interactive visualizations
Pandas	Data handling and tabular display

All tools are **free and open source**, making the project fully reproducible without API costs.

Project Setup and Usage

Installation

```
git clone https://github.com/IanMworia/brand-sentiment-analyzer.git
cd brand-sentiment-analyzer
pip install -r requirements.txt
```

Run the App

```
streamlit run app.py
```

Then open the app at http://localhost:8501.

How It Works

1. The user enters two brand names in the app interface.
2. The app simulates realistic online comments about both brands.
3. Each comment is analyzed using two AI models:
    - **Sentiment model**: Determines if the text is Positive, Negative, or Neutral.

    - **Emotion model**: Detects the emotional tone (e.g., Joy, Sadness, Surprise).
4. The app displays:
    - Comparative sentiment tables and bar charts.
    - Per-brand pie and emotion breakdown charts.
    - Color-coded tables with interactive data display.

Core Code Snippet

```
from transformers import pipeline

sentiment_model = pipeline("sentiment-analysis")
emotion_model = pipeline("text-classification",
                         model="j-hartmann/emotion-english-distilroberta-base",
                         top_k=1)

text = "I recently tried Moringa School and found it amazing."
sentiment = sentiment_model(text)[0]
emotion = emotion_model(text)[0][0]

print(sentiment, emotion)
```

This minimal snippet forms the heart of the analyzer, combining **two NLP pipelines** for layered interpretation of text.

AI Prompt Journal

Throughout the project, Generative AI was used for:

- Brainstorming ideas and refining the project scope.
- Debugging Python and Streamlit code.
- Generating realistic sample text for analysis.
- Improving UX with better prompts and color schemes.
- Writing and refining documentation (including this Toolkit).

Example Prompts

Task	Prompt	Outcome
Idea generation	“Suggest beginner AI project ideas with real-world value.”	Led to the concept of brand sentiment comparison.
Debugging	“Fix Streamlit rerunning on input change.”	Added form submit button.
Enhancement	“Make the sentiment app visually impressive.”	Added dual-model analysis and Plotly charts.

Testing & Iteration

The app was tested iteratively across multiple runs. Each component was validated independently:

- Verified that the sentiment pipeline produces consistent POSITIVE/NEGATIVE results.
- Ensured the emotion model outputs interpretable emotional states.
- Tested form submission, chart rendering, and simulated data consistency.
- Adjusted templates to avoid contradictory text for clearer output.

All issues (e.g., Pylance errors, `Styler` incompatibility) were documented and resolved using AI-assisted debugging.

Evaluation Justification

1. Clarity & Completeness (30%)

The project includes clear setup, explanations, and screenshots in the Toolkit and README. Documentation is concise and replicable.

2. Use of GenAI for Learning (20%)

Generative AI was used throughout the process as a **learning accelerator**, supporting coding, debugging, and documentation writing.

3. Functionality (20%)

The app runs smoothly, providing instant brand comparison and sentiment visualization. Results are clear and user-friendly.

4. Testing & Iteration (20%)

Each iteration was tested, improved, and validated through Streamlit's local environment, ensuring stable performance.

5. Creativity (10%)

Combining sentiment and emotion analysis in an interactive dashboard demonstrates originality and practical thinking.

 Evidence: Working demo, code reproducibility, and full documentation.

Final Submission Checklist

Task	Status
`app.py` runs successfully	
Dual-model sentiment + emotion working	
Form submit button implemented	
Plotly charts render correctly	
Toolkit exported to PDF	
README and repo ready	

Reflection

This project taught me how to use AI not just to generate code but to **think more systematically** — breaking problems into parts and improving through iteration. It also showed that with the right AI tools, building practical solutions doesn't need a budget, only curiosity and persistence.

Watch the reflection and demo on YouTube: https://youtu.be/I40rKeM5Fx0

Future Improvements

- Integrate live web-scraping APIs for real brand mentions (e.g., Reddit, Twitter).
- Add export-to-PDF feature for sentiment reports.
- Include "sentiment over time" graphs.
- Fine-tune models on local data for better accuracy.

Conclusion

The AI Brand Sentiment Analyzer successfully demonstrates the use of **Generative AI for real-world insights**, while remaining beginner-friendly and resource-efficient. It aligns perfectly with Moringa's capstone evaluation goals — **learning through building, with AI as a co-pilot**.

Built with ❤ using Streamlit and Hugging Face.