

Project Documentation

Sentiment Analysis Dashboard

Project Title: Sentiment Analysis Dashboard

Team/Student Name: Cyber Ninjas

GitHub Repository: <https://github.com/Kgoliath/sentiment-analysis-project>

Deployed App: <https://sentimentanalysisai.streamlit.app/>

Project Overview

This project was developed as part of the *AI in Action Sprint* to demonstrate the use of machine learning in real-world applications. The goal was to build an **interactive dashboard** capable of analyzing sentiment in text data. The system classifies texts into **positive, neutral, or negative sentiment**, provides **confidence scores**, highlights **sentiment-bearing words**, and enables both **individual and batch text analysis**.

The dashboard also supports **comparative analysis** between texts or files, **visualizations of sentiment distribution**, and allows users to **export results** in multiple formats (CSV, JSON, PDF).

Features Implemented

- **Input Options:** Direct text entry, file upload (CSV/TXT), or comparative analysis.
- **Sentiment Classification:** Multi-class labels (Positive, Negative, Neutral).
- **Confidence Scoring:** Displays confidence with levels (High, Medium, Low).
- **Keyword Highlighting:** Detects positive and negative words in the text.
- **Batch Processing:** CSV/TXT analyzed line-by-line with progress tracking.
- **Visualizations:** Sentiment distribution and confidence charts (Plotly).
- **Comparative Analysis:** Side-by-side comparison of texts or files.
- **Explanations:** Shows which keywords influenced sentiment classification.
- **Export Options:** CSV, JSON, and PDF reports with detailed metadata.
- **Error Handling:** Invalid inputs, empty files, and API failures are managed gracefully.

Technical Implementation

- **Frontend:** Streamlit for interactive UI.
- **Backend:** Hugging Face Inference API with the CardiffNLP RoBERTa model.
- **Visualization:** Plotly and Matplotlib for interactive and PDF charts.
- **Data Handling:** Pandas for file processing and exports.
- **Caching:** Streamlit caching for efficient repeated API calls.
- **Deployment Options:** Streamlit Cloud, Hugging Face Spaces, or Heroku.

API Justification

The project uses the Hugging Face cardiffnlp/twitter-roberta-base-sentiment model:

- Optimized for short text and social media posts.
- Provides **three sentiment classes** (Negative, Neutral, Positive), aligning with project requirements.
- Lightweight and fast for real-time API inference.
- Reliable and widely benchmarked in NLP research.


Implementation Challenges

- **API Latency:** Batch file uploads with many texts slowed performance; caching was used to mitigate repeated calls.
- **File Handling:** CSV files needed validation for a text column; TXT files required UTF-8 encoding.
- **Error Handling:** Managed unexpected API errors, invalid files, and empty data.
- **Visualization Complexity:** Designing comparative sentiment distribution and confidence charts required custom Plotly code.

8. Limitations

- **Domain-Specific Bias:** Model trained on Twitter data; results may not generalize to formal text or multilingual data.
- **Internet Dependency:** Requires API connection; cannot run fully offline.
- **Confidence Variability:** Confidence scores vary for long or ambiguous texts.
- **Performance:** Analyzing very large files may still be slow due to API rate limits.

User Guide with Examples

1. **Run the App:**
2. `streamlit run app.py`
3. **Choose Input Method:**
 - *Type Text:* Enter single text in the input box.
 - *Upload File:* Upload CSV/TXT (CSV must include a column named text).
 - *Comparative Analysis:* Compare two texts or two files.
4. **Analyze:** Click the  **Analyze Sentiment** button.
5. **View Results:**
 - *Tab 1:* Analysis Results (tables, confidence, keywords, explanations).
 - *Tab 2:* Visualizations (charts and comparisons).
6. **Export Reports:** At the bottom of the results, export to CSV, JSON, or PDF.

Example:

Input: *"I love this product, it works perfectly!"*

Output: **Positive (Confidence: 92%)**

Conclusion

This project successfully demonstrates the use of NLP and machine learning for sentiment analysis. The interactive dashboard allows for flexible input, visual analysis, and report generation. While limitations exist (domain bias, internet dependency), the system provides valuable insights into customer feedback and can be extended for more advanced sentiment analysis tasks.