Technical Documentation

Rwanda Distance-Based Fare Sentiment Dashboard

Submitted for: Tech Associates Hackathon 2025
Submitted by: Bernard ISHIMWE

Submitted on Friday, 06, May, 2025
Tools Used: Streamlit, Python, TextBlob, Plotly, WordCloud, Pandas, Matplotlib

1. Project Overview

This dashboard analyzes and visualizes public sentiment surrounding Rwanda's transition to a **distance-based fare model** for public transportation. It helps **policymakers and stakeholders** understand how citizens feel, what concerns are most common, and where intervention or clarification may be needed.

2. Goals

- **Track public opinion** over time on the fare reform
- **Identify common concerns** and communication gaps.
- **Provide actionable recommendations** based on sentiment trends.
- Promote **data-driven decisions** using real-time or simulated feedback

3. Main Features

Feature	Description
Overview Tab	Displays a pie chart of overall sentiment distribution (Positive, Neutral, Negative).
Trends Tab	Shows a line graph of sentiment trends over a 30-day period.
WordCloud Tab	Visualizes the most frequent terms for each sentiment type.
Recommendations Tab	Surfaces alerts and suggestions based on sentiment spikes and keyword analysis (e.g, mentions of "confusing").

4. Data Pipeline

a. Input Data:

- Simulated public comments (30-day dataset).
- Each comment includes a date and user-generated feedback text.

b. Preprocessing:

- Comments are shuffled for randomness.
- Sentiment classification is applied using TextBlob.

c. Sentiment Classification Logic:

```
if polarity > 0.1 → Positive
elif polarity < -0.1 → Negative
else → Neutral
```

d. Keyword Trigger:

- The system looks for specific keywords (e.g., “confusing”) to trigger recommendations.

5. Dependencies

- `streamlit`: UI framework
- `pandas`: Data manipulation
- `plotly.express`: Interactive charts
- `wordcloud`: Word cloud generation
- `matplotlib`: Plotting library
- `textblob`: NLP sentiment analysis

Install via:

```
pip install streamlit pandas plotly wordcloud matplotlib textblob
```

6. Code Modules

Section	Purpose
`generate_sample_data()`	Creates a 30-day synthetic dataset with feedback comments
`get_sentiment()`	Applies TextBlob sentiment analysis
`plot_wordcloud()`	Generates WordCloud for selected sentiment
`Streamlit UI`	Organizes tabs, visualizes data, and displays recommendations

7. How to Run

1. Ensure all dependencies are installed.
2. Save the code as `app.py`.
3. Launch with Streamlit:

```
streamlit run app.py
```

8. Limitations & Future Improvements

Current Limitation	Suggested Enhancement
Uses simulated data	Integrate with Twitter/X, Facebook, or surveys using APIs
Simple sentiment analysis	Replace with fine-tuned models (e.g., BERT or VADER)
No geolocation or demographic data	Add user metadata for deeper insights
English-only comments	Add Kinyarwanda/NLP multilingual support

9. Conclusion

This dashboard prototype delivers a functional and extensible tool to support **inclusive policy-making**. It empowers decision-makers to hear public voices clearly, spot problems early, and improve communication in line with real concerns.