**Akkaike Technologies Internship Report**

**Project: News Summarization and Text-to-Speech Application**

1. **Project Setup**

**1.1 Installation Requirements**

* Python version -Python 3.11.11
* pip(Python Package Installer)
* Required libraries (please check requirements.txt)
  1. **Steps to Install and Run**
* git clone
* Github Link: <https://github.com/Mrud11/News-Summarization-and-Text-to-Speech-Application>
* cd news-summarization
* Dependencies installation - check requirements.txt file.
* Run the api.py file for backend purpose, (code: python api.py).
* Run the (streamlit) app.py for the frontend purpose (code: streamlit run app.py).

1. **Model Details**

**2.1 News Extraction**

* **Method Used:** BeautifulSoup (Web Scraping).
* **Purpose:** Extracts **title, summary, and article link** for the tata company.

**2.2 Sentiment Analysis**

* **Model Used:** Textblob
* **Purpose:** Classifies the news articles as **Positive, Negative, or Neutral**.

**2.3 Text-to-Speech (TTS)**

* **Model Used:** gTTS (Google Text-to-Speech).
* **Purpose:** Converts the summarized news into **Hindi speech output**.

**2.4 Hugging Face Deployment**

* **UI used**: Streamlit , License used: MIT
* **Purpose:** To showcase it to the users so they can fetch the news and sentiment report

1. **API Development**

The API, built using Flask, fetches news articles and performs sentiment analysis. It enables to retrieve news articles for a given company (tata and reliance) and analyze their sentiment.

**3.1 How to Access the API**

* Endpoint: GET /news?company=<company\_name>
* Example Request: <https://gold-pandas-wait.loca.lt/> (new link comes every time we run this code)
* Response Format:

{

"company": "Tata",

"articles": [

{

"title": "Tata Group companies quietly partner with Elon",

"link": "https://newslink.com",

"sentiment": "Positive"

}

]

}

#### ****Functionality :****

* **Fetch News**: Retrieves news articles from tata\_news\_sentiment.csv.
* Sentiment Analysis: Uses TextBlob to classify sentiment as Positive, Negative, or Neutral based on the news title.

1. **API Usage:**

|  |  |  |
| --- | --- | --- |
| **Libraries** | **Purpose** | **Installation** |
| BeautifulSoup | Web scraping | pip install beautifulsoup4 |
| VADER/ TextBlob | Sentiment analysis | pip install vaderSentiment |
| gTTS | Text-to-Speech conversion | pip install gtts |

## ****5. Assumptions & Limitations****

### ****5.1 Assumptions****

* News articles can be **scraped** without restrictions.
* Sentiment analysis **correctly classifies** article tone.
* TTS **accurately converts Hindi text into speech.**

### ****5.2 Limitations****

* Some websites with JavaScript-rendered content cannot be scraped.
* Sentiment analysis may misinterpret sarcasm or mixed sentiment.
* TTS may struggle with technical terms or brand names.

## ****6. Deployment on Hugging Face Spaces****

### ****6.1 Steps to Deploy****

1. **Create a Hugging Face account.**
2. **Go to Spaces then Create New Space.**
3. **Name the space (e.g., news-summarization).**
4. **Choose "Streamlit" as the SDK.**
5. **Upload the following files:**

* app.py (Streamlit UI)
* api.py (Flask API)
* utils.py (Helper functions)
* requirements.txt (Dependencies)
* README.md (Documentation)

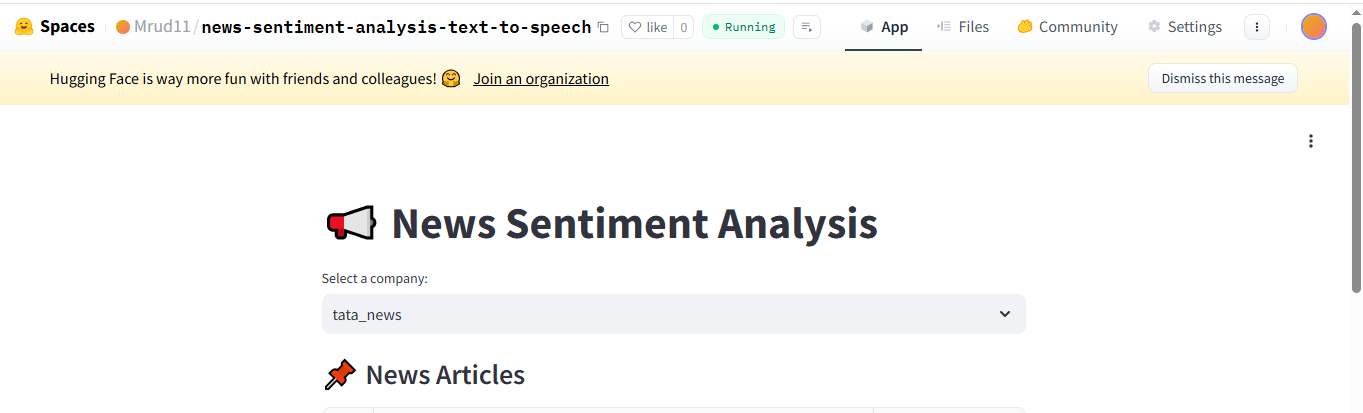
1. **Commit the files to the main branch.**
2. **Wait for the deployment to complete.**
3. **Access the app via the public link** provided by Hugging Face.

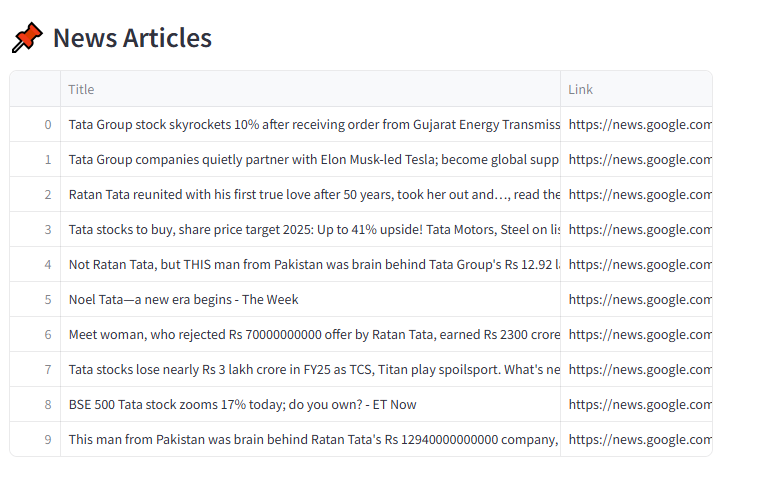
**6.2 Public Deployment Link (my project link)**

**Output Link**: <https://huggingface.co/spaces/Mrud11/news-sentiment-analysis-text-to-speech>

**7. Project File Structure**

* app.py - Frontend (Streamlit UI)
* api.py - Backend API (Flask)
* utils.py - Utility functions (Scraping, Sentiment, TTS)
* requirements.txt - Dependencies
* README.md - Documentation
* data - Folder containing scraped CSV (if required)
  1. **Output Screenshots**









**Output URL Link:** [**https://huggingface.co/spaces/Mrud11/news-sentiment-analysis-text-to-speech**](https://huggingface.co/spaces/Mrud11/news-sentiment-analysis-text-to-speech)

**Additional Notes**

* Ensure the Flask API runs **before** the Streamlit UI.
* If using Hugging Face, update app.py to point to the correct API URL.
* If **deployment fails**, check logs for missing dependencies.