**Sentiment Analysis Dashboard Documentation**

# 1. Project Overview

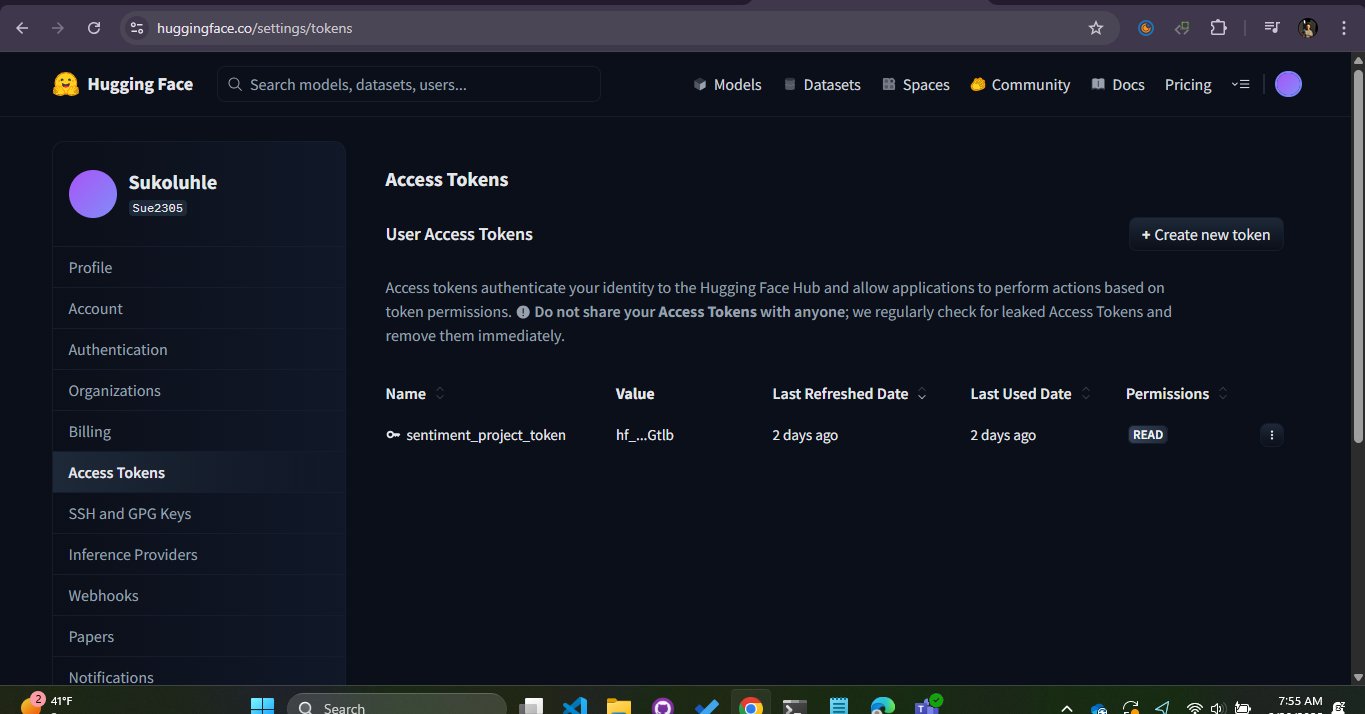
This document describes the Sentiment Analysis Dashboard project. The dashboard is a web-based application that analyzes the emotional tone of textual data such as customer reviews and social media posts. It supports multi-class sentiment classification (positive, negative, neutral), confidence scoring, keyword extraction, batch processing, and visualizations, helping users better understand sentiment drivers in text data.

# 2. Tools Used

* GitHub: Version control and collaboration
* Hugging Face Inference API: Transformer-based sentiment classification
* Python: Core programming language (including Pandas for data processing)
* Stream lit: Interactive and responsive interface
* HTML & CSS: Styling and layout customization
* Command Prompt / Terminal: Package installation and running scripts
* CSV files: Batch input processing

# API Selection Justification

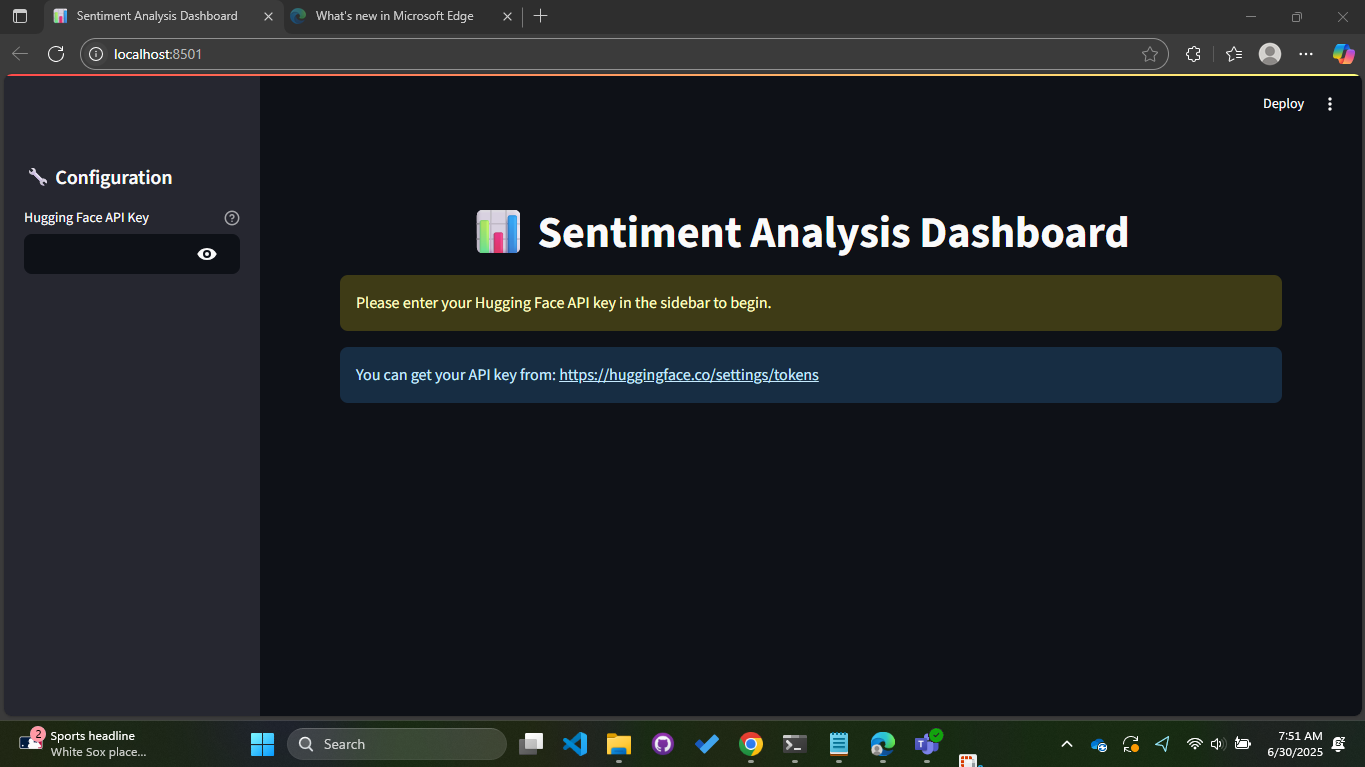
# Hugging Face Access Token Configuration



Hugging Face was selected for its:  
- State-of-the-art transformer-based sentiment models  
- Simple integration via token-based authentication  
- High-quality documentation and active community support  
- Flexibility and scalability  
- Free tier suitable for prototyping  
  
Alternative options, such as AWS Comprehend, were considered but Hugging Face offered easier integration, more transparency, and a stronger open-source ecosystem.

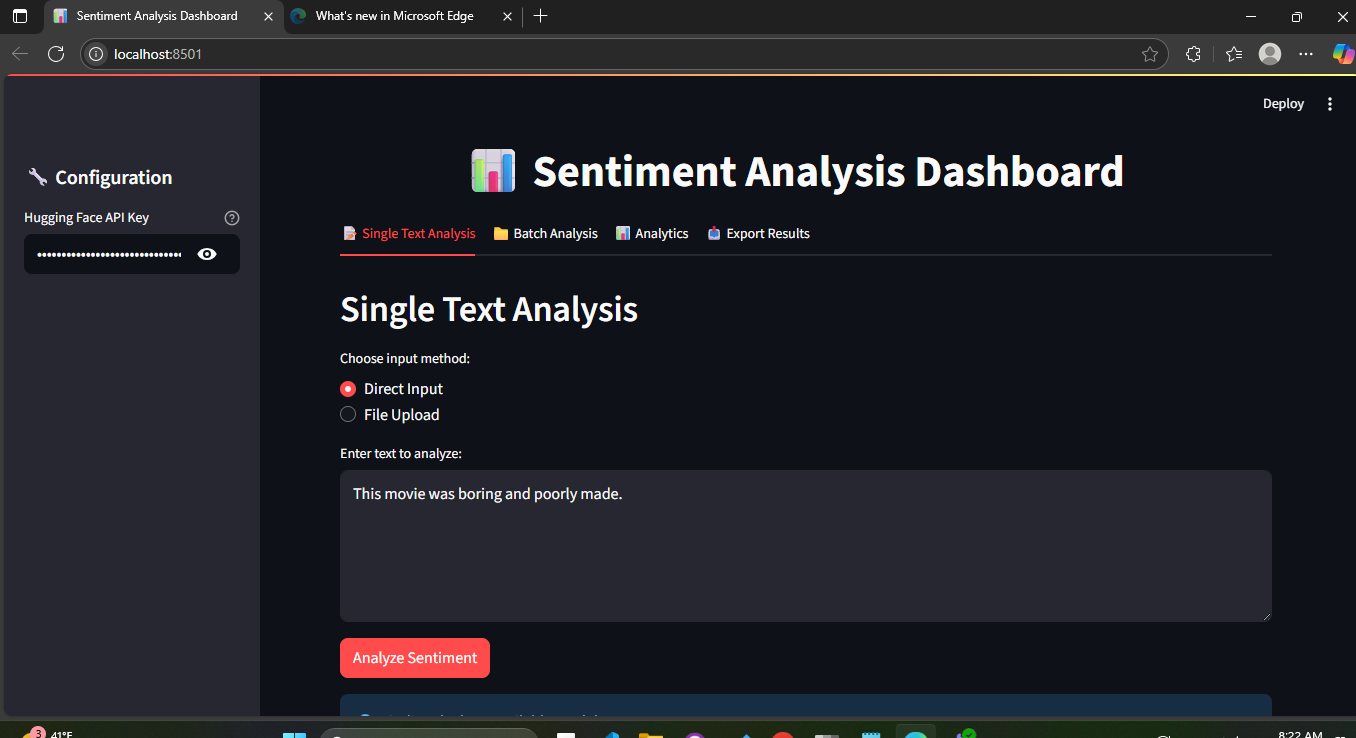
# Features Added

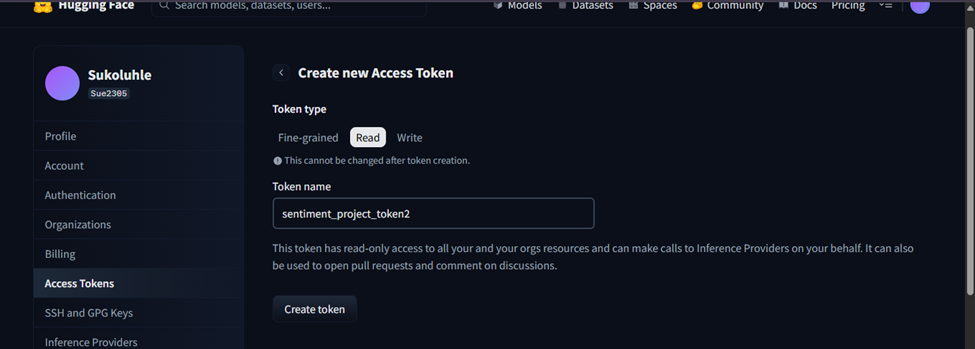
Sentiment Analysis Dashboard Input Screen

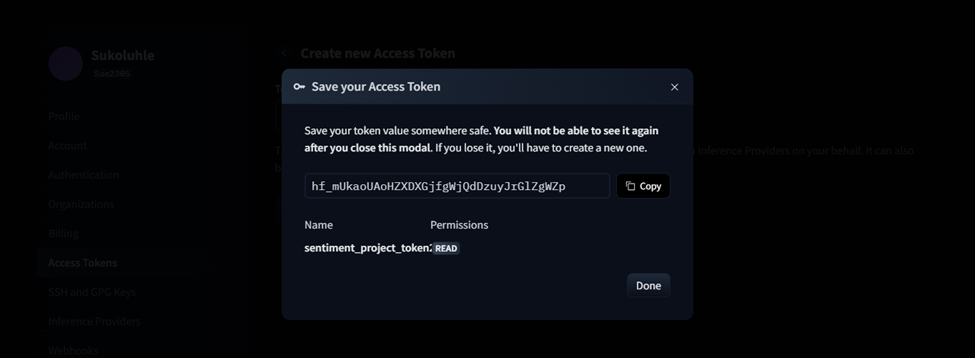


## 4.1 Input Methods

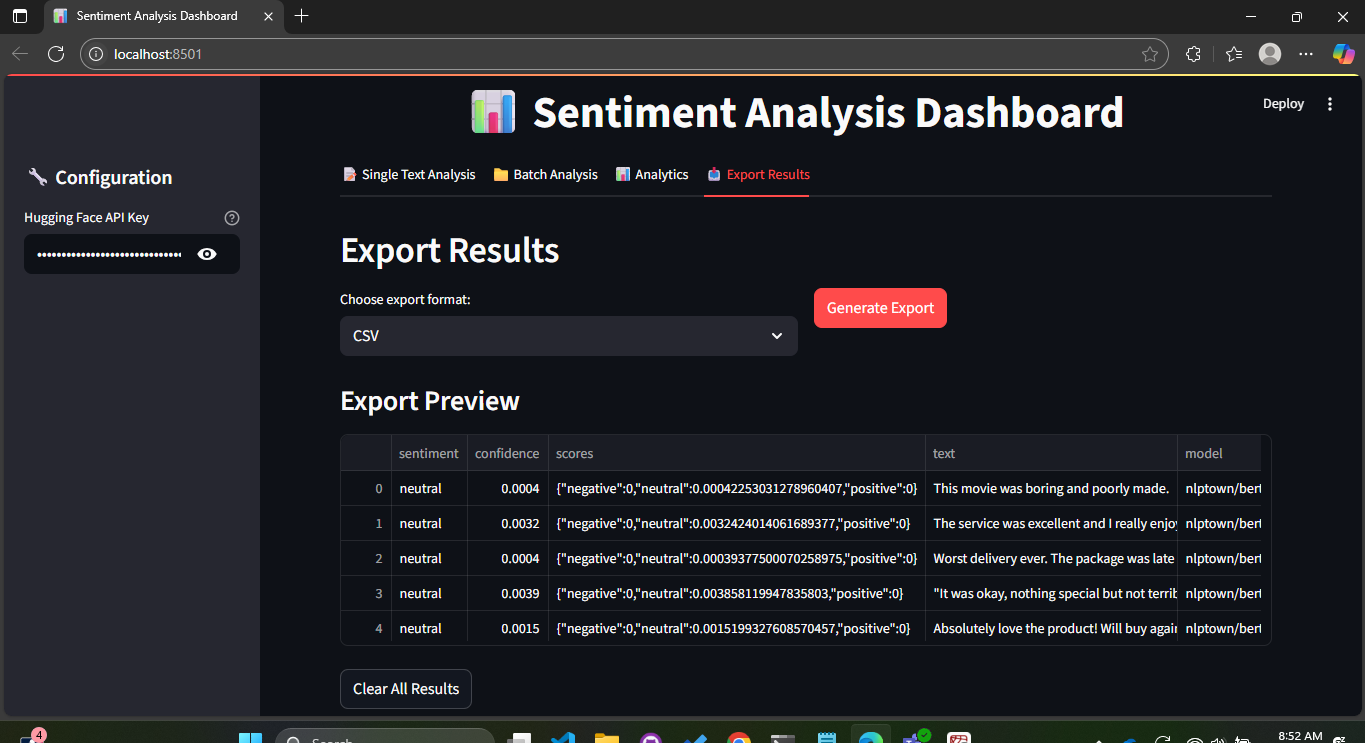
* Upload single text files (.txt)



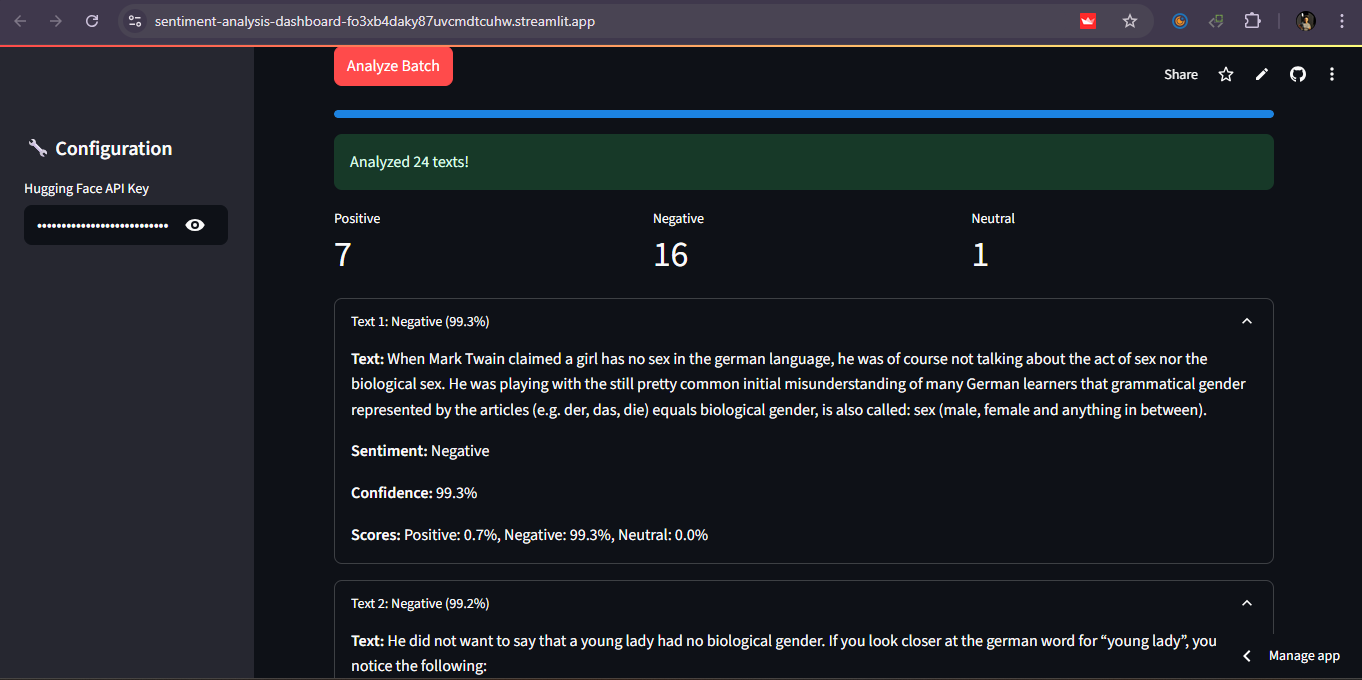




* Upload CSV for batch processing



* Manual multi-text entry



## 4.2 Sentiment Classification

* Positive, Negative, Neutral classes
* Confidence scoring with detailed breakdowns
* Real-time results

## 4.3 Advanced Analytics

* Keyword extraction
* Sentiment distribution charts (pie, histogram)
* Confidence score analysis
* Batch results summary

## 4.4 Visualizations

* Pie charts for sentiment distribution
* Histograms for confidence scores
* Sortable/filterable data tables

## 4.5 Export Options

* CSV
* TXT
* Print-friendly PDF

# 5. Implementation Challenges

* Managing Hugging Face API rate limits and adding retry logic
* Handling different text file encodings
* Designing clean, user-friendly visualizations
* Adding robust error handling for invalid inputs or empty submissions
* Optimizing batch text summarization for large datasets

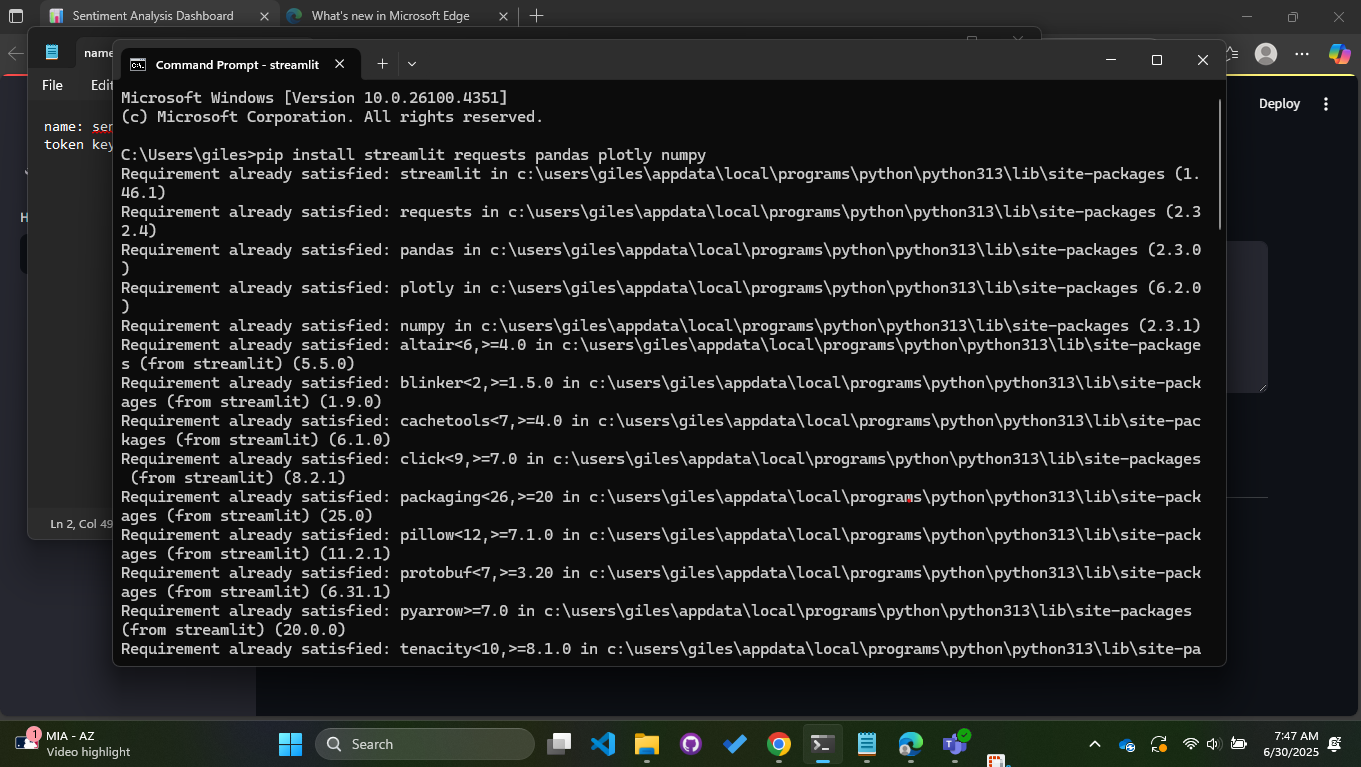
# 6. User Guide

1. Step 1: Launch the App - Start the Stream lit script or open the deployed web application.
2. Step 2: Choose Input Method - Enter text directly, upload a .txt file, upload a CSV, or paste multi-line text.
3. Step 3: Analyze - Click “Analyze” to send the data to the sentiment analysis API.
4. Step 4: View Results - Review sentiment classification, confidence scores, and extracted keywords.
5. Step 5: Export Results - Download as CSV, TXT, or print-friendly PDF.
6. Step 6: Batch Analysis - When using CSV, compare sentiment summaries across multiple records.

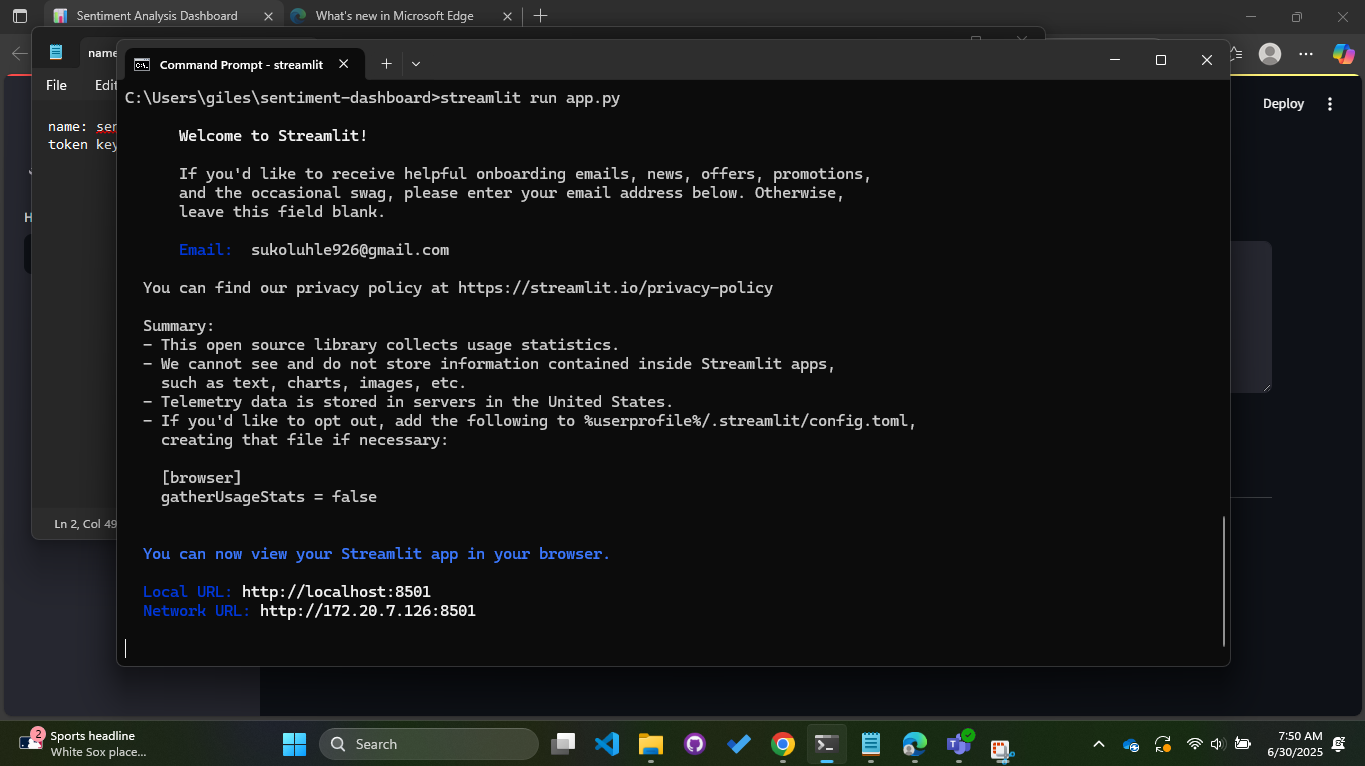
# 7. API Authentication

A Hugging Face access token was created with read permissions for secure authentication.  
Note: Always store tokens in environment variables or a .env file to avoid accidental exposure in public repositories.

**Command Prompt Setup**

****

**Command Prompt Script Execution**

****

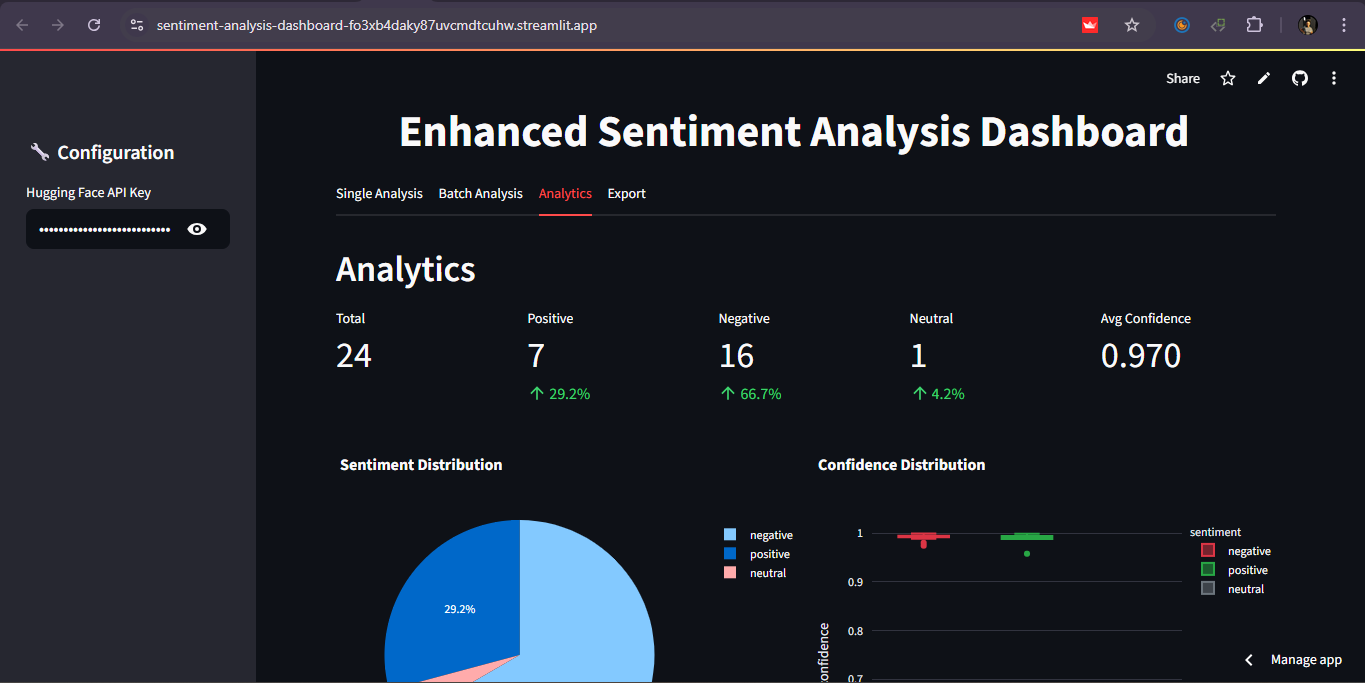
# 8. Model Limitations

* Primarily trained on English language data
* May misinterpret sarcasm, irony, or ambiguous phrases
* Very short text inputs may have lower accuracy
* Confidence score guidelines:  
   - Above 80%: generally reliable  
   - Below 60%: use caution

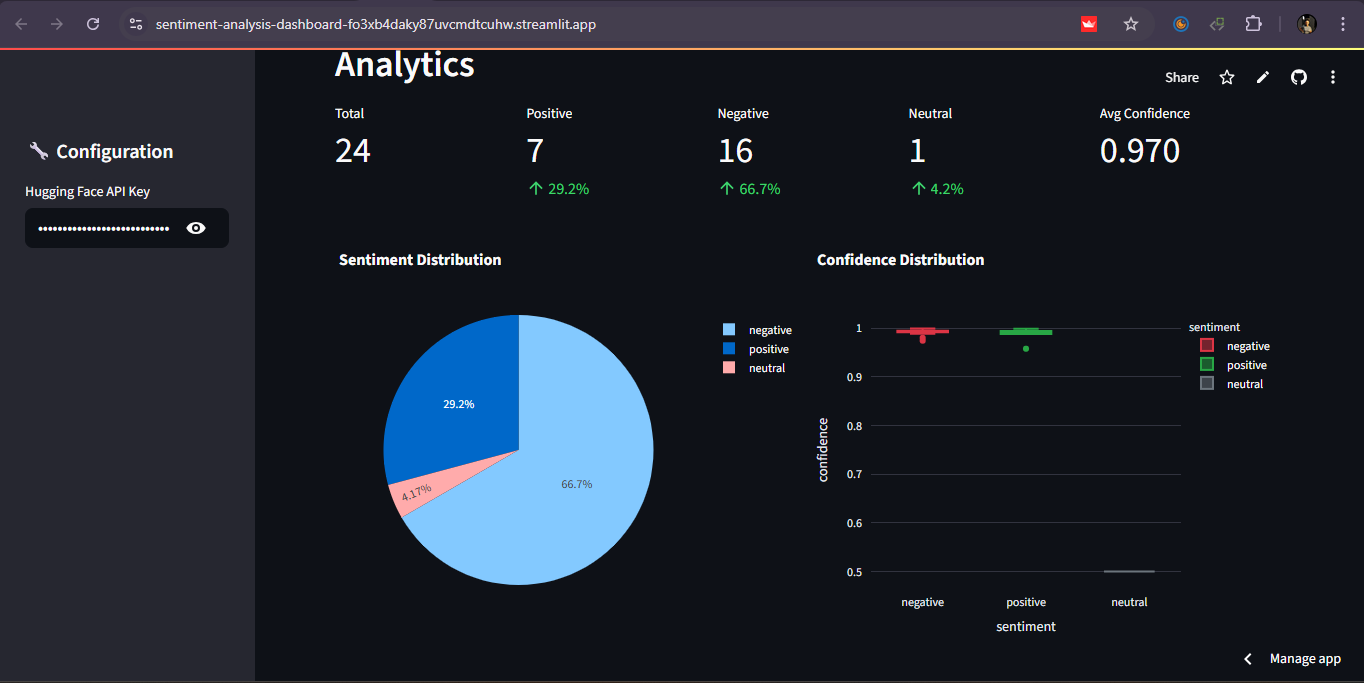
# 9. Deployment Notes

The application supports deployment on Stream lit Cloud or local Docker.  
Setup instructions are documented in the GitHub repository.

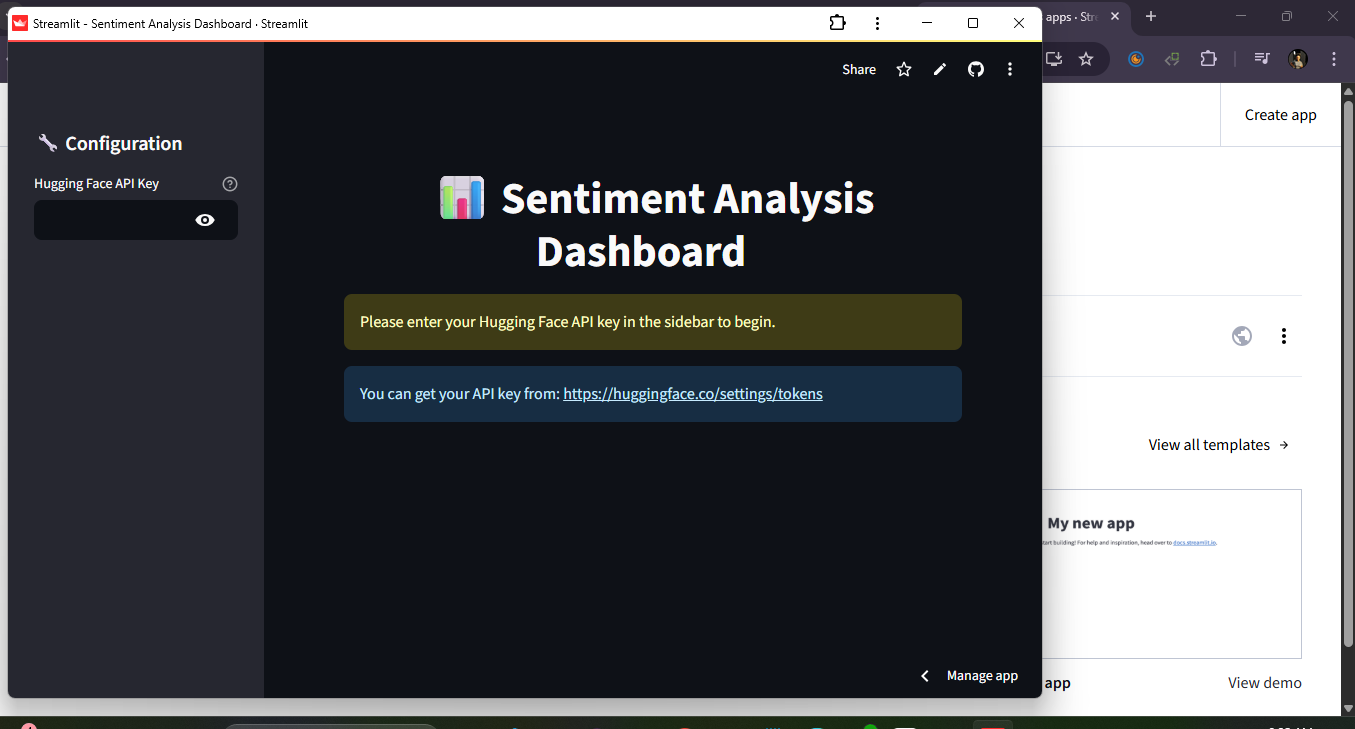
**Analysis Report Sample 2**



**Analysis Report Sample 3**



**Streamlit App Deployment Screenshot**



# 10. References & Acknowledgement

* Hugging Face Inference API
* Streamlit documentation
* IBM Python for Data Science