# Bridging NLP and MLOps for Content Insights

Louie Daans, Radna Puriel, Gin Li, Carmen-Nicola Ioniță, Zakariae el Moumni
Team – NLP3

## Introduction

In today's digital media, detecting emotion in audio/video is key to boosting engagement and personalization.

Our app uses advanced NLP and ML to transcribe and identify emotions automatically and accurately.

Built for scalability, it adapts continuously via automated retraining.

Deployed on Azure with containerized microservices and Airflow orchestration for fast, secure, and cost-efficient processing.

## Problem Statement

Many media and entertainment companies face difficulties in understanding the emotional tone of large volumes of audio and video content. Manual annotation is time-consuming, inconsistent, and lacks scalability. There is a need for an automated, cloud-based solution that can process such content and provide emotional insights in real-time or batch.

## Business Value

- Faster content analysis → reduces manual workload
- Actionable insights → improves editorial decisions and audience targeting
- Cost-effective and scalable → using Azure infrastructure
- Reusable components → modular NLP package for other projects
- Applicable in multiple domains → from media to customer service

## System Architecture

This architecture outlines our end-to-end pipeline from user upload to emotion prediction and model retraining, using Azure ML, Airflow, and containerized services.

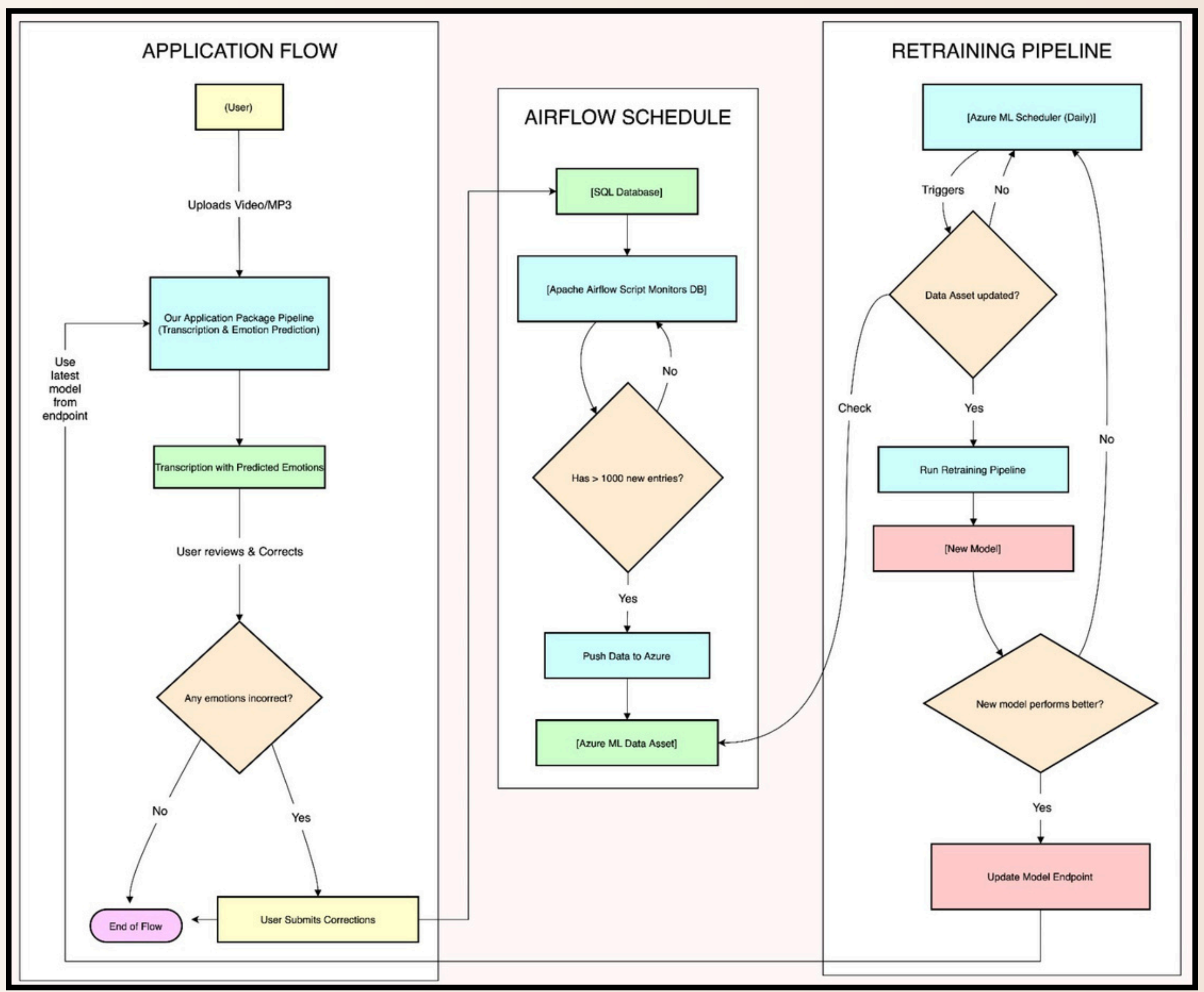


Figure 1: Diagram of the pipeline architecture

## Model

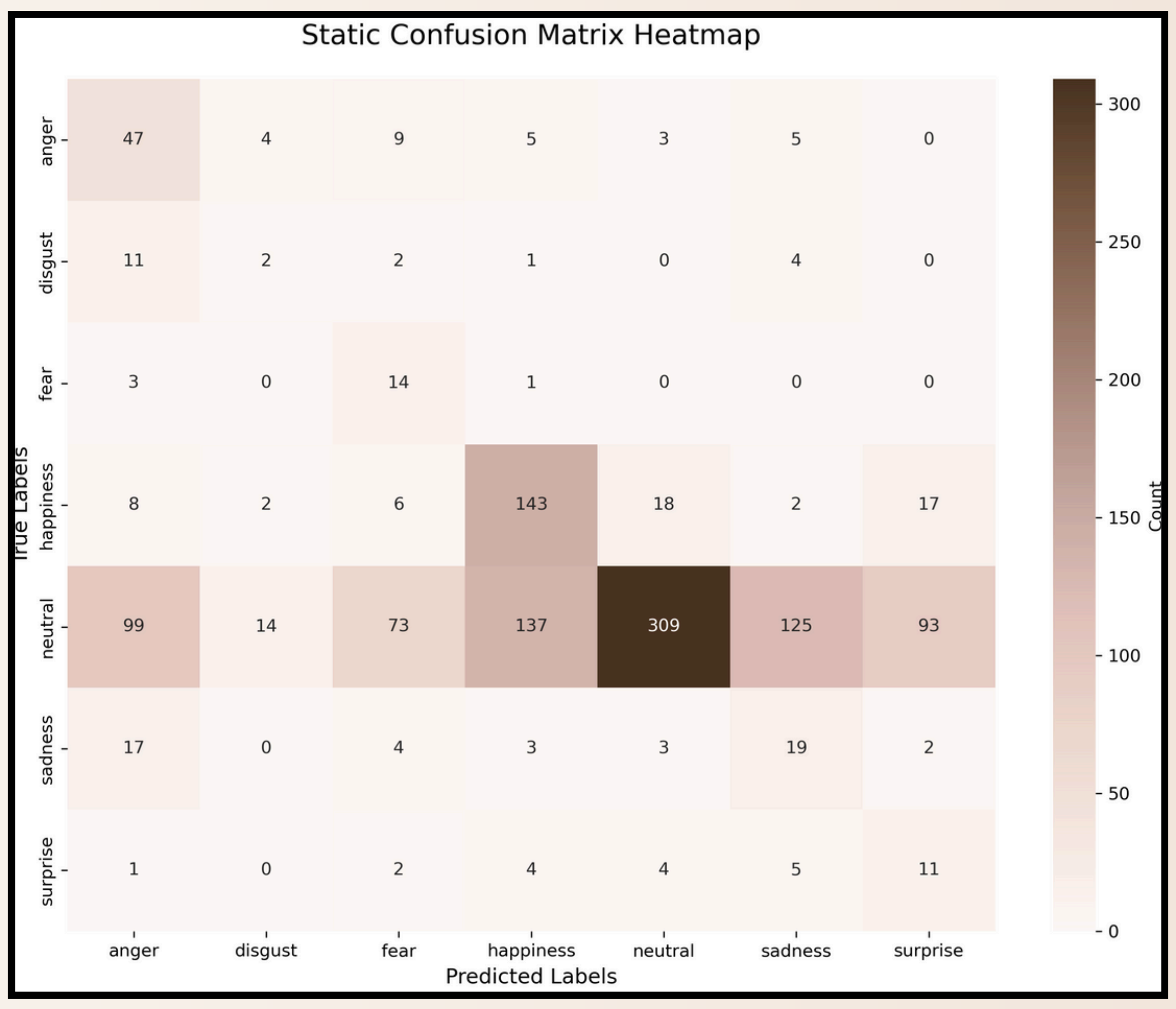
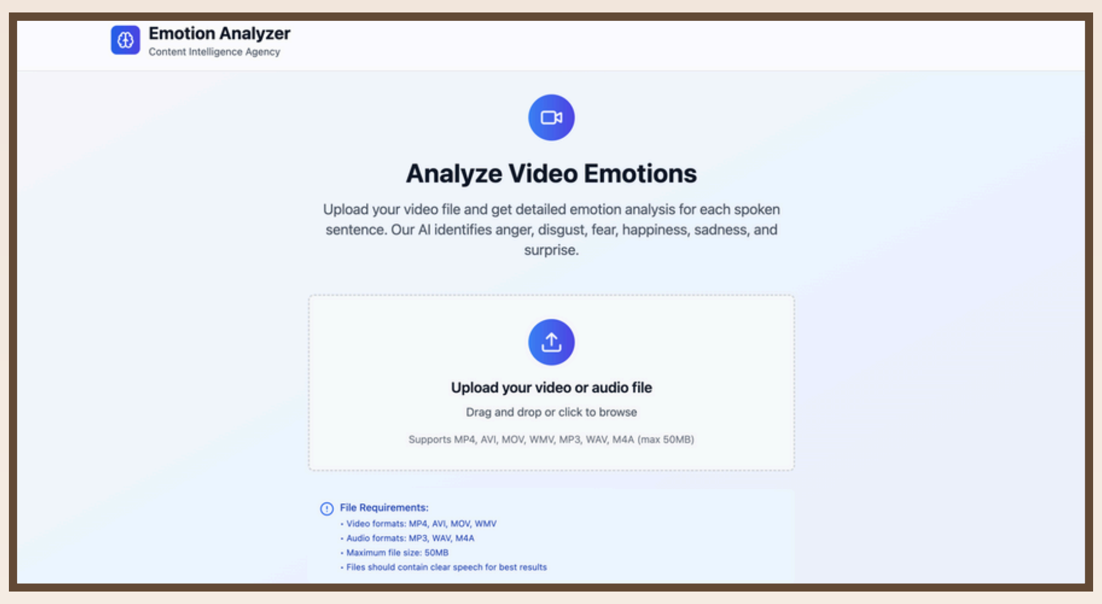


Figure 2: Confusion Matrix

Our model combines speech transcription, speaker diarization, and emotion classification using state-of-the-art NLP and deep learning techniques. It leverages pretrained transformers and audio feature extraction to accurately detect emotions from spoken content in real time.

## Frontend



Emotion Analyzer
Content Intelligence Agency

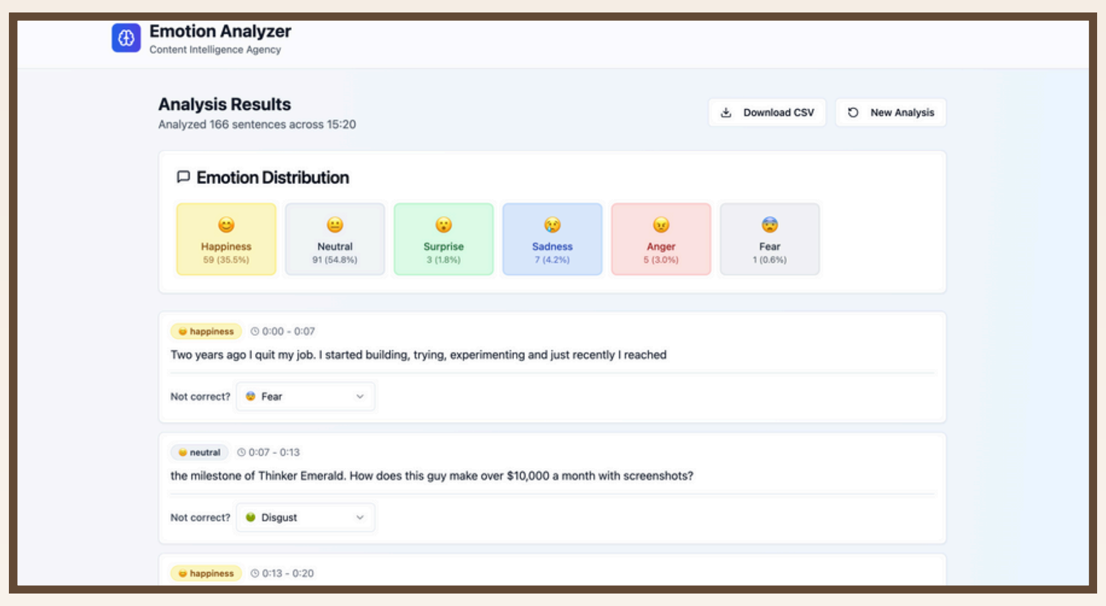
Analyze Video Emotions

Upload your video file and get detailed emotion analysis for each spoken sentence. Our AI identifies anger, disgust, fear, happiness, sadness, and surprise.

Upload your video or audio file
Drag and drop or click to browse
Supports MP4, AVI, MOV, WMV, MP3, WAV, M4A (max 50MB)

File Requirements:
- Video formats: MP4, AVI, MOV, WMV
- Audio formats: MP3, WAV, M4A
- Maximum file size: 50MB
- Files should contain clear speech for best results

Figure 3; Screenshot of the upload page in the frontend



Emotion Analyzer
Content Intelligence Agency

Analysis Results
Analyzed 166 sentences across 15:20

Download CSV | New Analysis

Emotion Distribution

| Happiness | Neutral | Surprise | Sadness | Anger | Fear |
|---|---|---|---|---|---|
| 59 (35.5%) | 91 (54.8%) | 3 (1.8%) | 7 (4.2%) | 5 (3.0%) | 1 (0.6%) |

happiness 0:00 - 0:07
Two years ago I quit my job. I started building, trying, experimenting and just recently I reached
Not correct? Fear

neutral 0:07 - 0:13
the milestone of Thinker Emerald. How does this guy make over $10,000 a month with screenshots?
Not correct? Disgust

happiness 0:13 - 0:20

Figure 4: Screenshot of the results from the frontend

The frontend is a web-based interface designed for ease of use. Users can upload video or audio files, view real-time transcription and emotion detection results, and provide feedback on predictions. It communicates with the backend via an API and is deployed as a static web app on Azure, ensuring accessibility, responsiveness, and scalability.

## Backend / API

- Modular NLP Python Package

Custom-built package for transcription, speaker diarization, and emotion detection.

- RESTful API

Built with FastAPI, exposing endpoints for audio file handling and prediction output.

- Dockerized Backend

Containerized using Docker for consistency across environments and simplified deployment.

- Secure Azure Blob Storage Integration

Uploaded files are safely stored and served from Azure's cloud infrastructure.

- Efficient Model Inference

Optimized emotion detection models ensure quick and accurate feedback.

The backend is a FastAPI service that handles uploads, transcription, and emotion detection. It uses a modular Python package and stores data in Azure Blob Storage. Deployed as an Azure Container App, it's scalable, secure, and integrates seamlessly with the frontend.

## Deployment

- Azure Container Apps

Backend is deployed as a containerized app, scaling automatically based on usage.

- CI/CD Pipeline via GitHub Actions

Continuous deployment ensures latest code is always live and tested.

## Limitations

- Emotion ambiguity: Some emotions are subtle or overlap, making classification challenging.
- Audio quality dependency: Noisy or low-quality inputs reduce accuracy.
- Limited to spoken language: No facial or body cues; emotions from tone only.
- English-only (currently): Multilingual support is a future goal.
- Small domain-specific training data: May reduce performance on niche media content.

The backend is a FastAPI service deployed as an Azure Container App.
It handles file uploads, transcription, and emotion detection using a modular Python package.
Data is stored in Azure Blob Storage, ensuring scalability and security.

Breda University OF APPLIED SCIENCES
Content Intelligence Agency