Voice-Based Cognitive Decline Pattern Detection

Objective:

Build a basic proof-of-concept pipeline that uses raw voice data samples to detect cognitive stress or decline indicators using NLP + audio feature extraction.

Problem Statement:

Task is to process 5–10 anonymized voice clips (Find simulated/realistic samples) and extract patterns that might indicate early cognitive impairment.

Data Collection:

I collected data from common voice website. I selected 6 random audios for sampling.

Link- https://commonvoice.mozilla.org/en/datasets

Convert Audio to Text:

I used Whisper model because it is one of the most powerful and open-source speech-to-text models. It understands natural, imperfect speech and it deal well with fillers and hesitations which is relevant for cognitive impairment analysis.

It trained on 680000 hours of multilingual and noisy audio and it is easy to use.

Result from whisper model:

```
/usr/local/lib/python3.11/dist-packages/whisper/transcribe.py:132: UserWarning: FP16 is not supported on CPU; using FP32 instead
  warnings.warn("FP16 is not supported on CPU; using FP32 instead")
In this position he learned mathematics, Greek, Italian, Spanish, and several oriental languages.
/usr/local/lib/python3.11/dist-packages/whisper/transcribe.py:132: UserWarning: FP16 is not supported on CPU; using FP32 instead
  warnings.warn("FP16 is not supported on CPU; using FP32 instead")
She herself defended her verse as Holy Eurotica.
/usr/local/lib/python3.11/dist-packages/whisper/transcribe.py:132: UserWarning: FP16 is not supported on CPU; using FP32 instead
  warnings.warn("FP16 is not supported on CPU; using FP32 instead")
They followed the street car lines to areas south of the raccoon river.
/usr/local/lib/python3.11/dist-packages/whisper/transcribe.py:132: UserWarning: FP16 is not supported on CPU; using FP32 instead
  warnings.warn("FP16 is not supported on CPU; using FP32 instead")
The key allows customers to buy plus bus for the Crawley and Brighton areas.
/usr/local/lib/python3.11/dist-packages/whisper/transcribe.py:132: UserWarning: FP16 is not supported on CPU; using FP32 instead
  warnings.warn("FP16 is not supported on CPU; using FP32 instead")
My books and my stories.
/usr/local/lib/python3.11/dist-packages/whisper/transcribe.py:132: UserWarning: FP16 is not supported on CPU; using FP32 instead
  warnings.warn("FP16 is not supported on CPU: using FP32 instead")
Mayor Max requested cuts be made in Christopher initially refused.
```

Features Extraction:

I wrote code for features extraction. First, I found total duration of clip by using librosa library. Then I converted audio to text with the help of whisper model. After it, I measured speech rate- slower speech rate can be an early cognitive indicator. Then I counted filler words like "um" or "uh" by using re library. I measured number and duration of pauses and captured intonation variability. After it, I encoded timbre and spectral shape of speech which is useful for clustering or anomaly detection. And last, I simulated a word recall task.

Feature	Why It's Important
hesitation_count	Hesitations increase with uncertainty, word-finding difficulty
pause_count / avg_pause_duration	Reflects cognitive load, processing delays
speech_rate	Slower speech is a common sign of cognitive decline
pitch_std	Monotone voice may indicate neurological changes
word_recall_errors	Simulates episodic memory or lexical access issues
mfccs	Good for ML models, but not human-interpretable on their own

Unsupervised Anomaly Detection:

I used IsolationForest model because in the model, no labels needed and great for outlier detection in high dimensional data. It focuses on identifying samples that differ significantly from the rest which makes it perfect for flag patients at risk based on unusual combinations of speech features.

PCA Visualization:

PCA reduced complex audio features into 2D for easy visualization. We can clearly spot the outlier- the light colored point in the bottom right – which IsolationForest has flagged.

PCA of Audio Features with Anomaly Detection
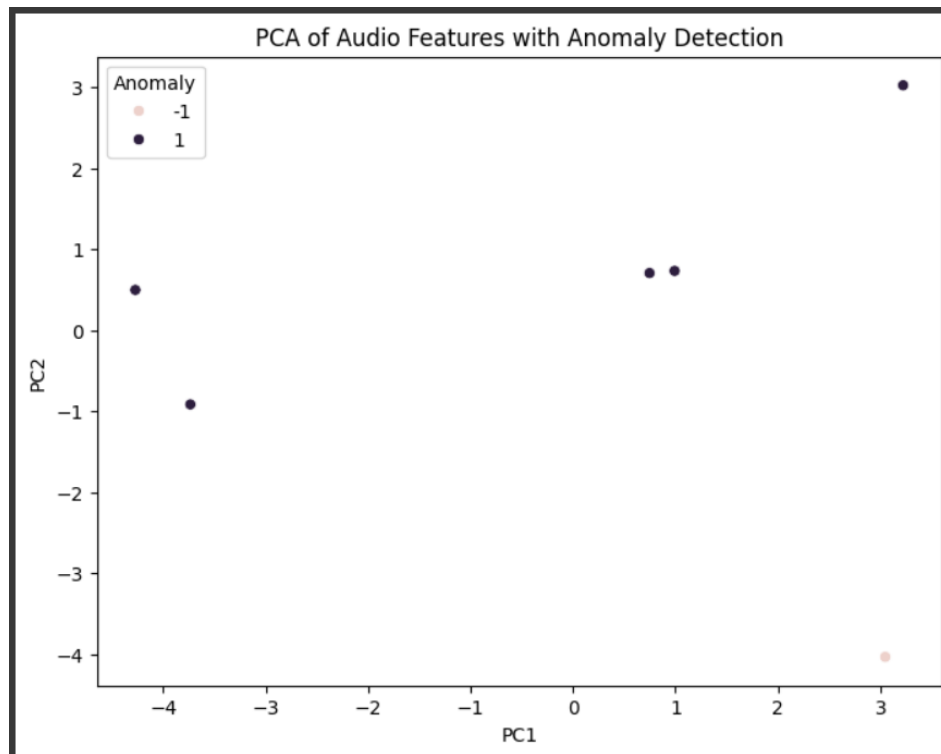
Next Steps to make is Clinically Robust:

1. Expand Dataset
2. Add Linguistic/NLP Features
3. Multi- modal Integration
4. Collaborate with Clinicians
5. Risk Stratification
6. Bias and Fairness Checks