# Data Collection and Processing

July 22, 2021

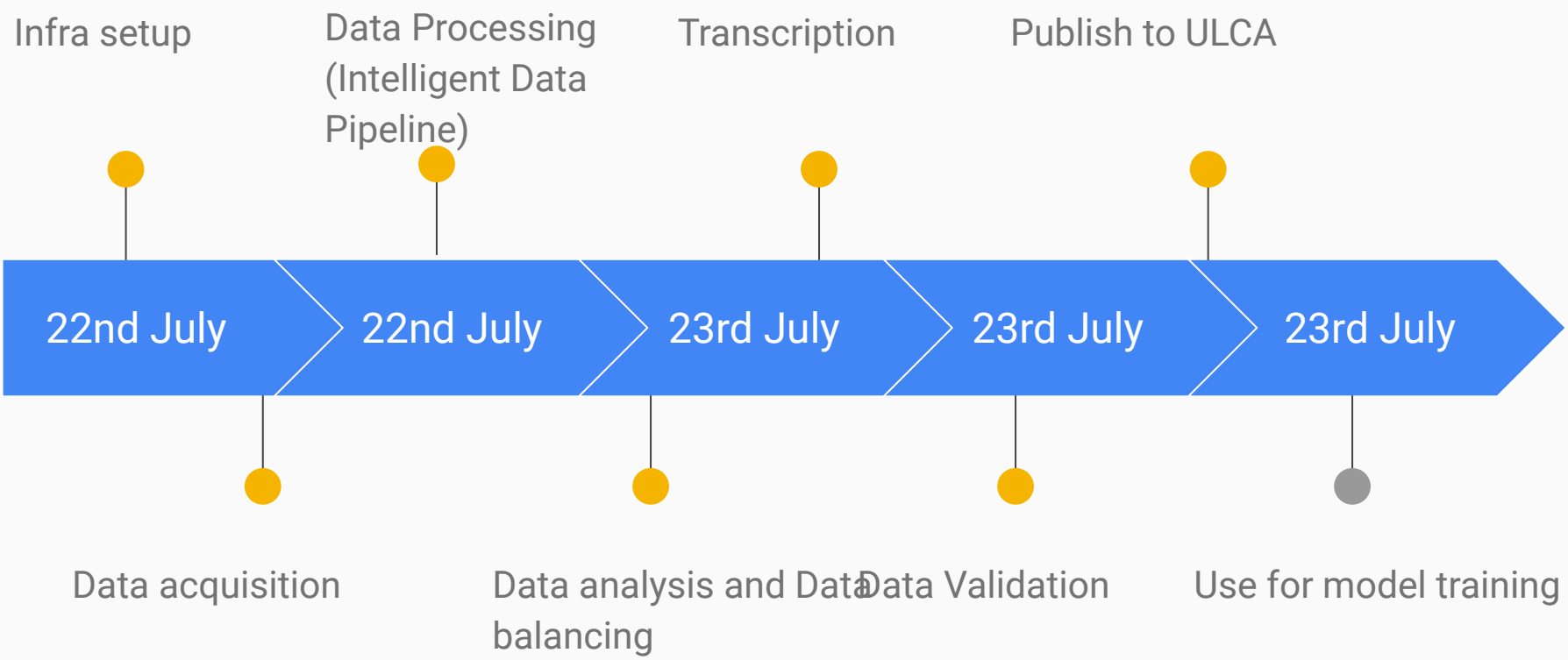
# Model building steps

- Data collection
- Data cleaning and processing
- Model pretraining unsupervised using huge data. (~10K hours)
- Model fine tuning with less data (~500 hours)

# Data collection and processing

- Audio data is in abundance, but it needs systematic and automated approach to collect data
- Data collection and processing needs to be operationalized
- Volume for of data is huge

- Audio is not readable to humans
- Raw audio is not useful for training


Infra setup
Data Processing (Intelligent Data Pipeline)
Transcription
Publish to ULCA
22nd July
22nd July
23rd July
23rd July
23rd July
Data acquisition
Data analysis and Data balancing
Data Validation
Use for model training

# Data Acquisition

- Collect data from multiple sources like youtube, web pages
- Crawl or download specific videos/audios
- Metadata

Documentation

Git repo

# Scrapy

- Framework for scraping and downloading
- Run on infrastructure appropriate for crawling and scraping.
- Run for specific channels, videos
- Or Crawl web using Google search or Bing based on language and keywords

Framework: Scrapy

Infra : Zyte

# Intelligent Data Pipeline

- Convert multiple formats to wav format
- Down sample to sampling rate 16K
- Break into smaller utterances using Voice Activity Detection
- Calculate duration of utterances
- Calculate Signal to Noise Ratio
- Build Catalogue

Documentation

Git repo

# Voice Activity Detection (VAD)

**The typical design of a VAD algorithm is as follows:**

1. There may first be a noise reduction stage, e.g. via spectral subtraction.
2. Then some features or quantities are calculated from a section of the input signal.
3. A classification rule is applied to classify the section as speech or non-speech – often this classification rule finds when a value exceeds a certain threshold.

VAD

Python Utility for VAD

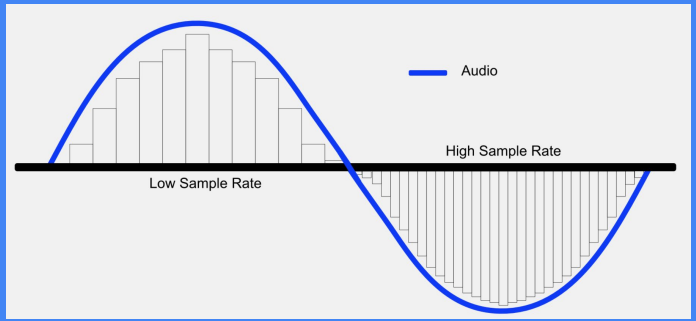
The biggest difficulty in the detection of speech in this environment is the very low signal-to-noise ratios (SNRs) that are encountered. It may be impossible to distinguish between speech and noise using simple level detection techniques when parts of the speech utterance are buried below the noise.

# SNR

Signal-to-noise ratio (SNR) is the measurement used to describe how much desired sound is present in an audio recording, as opposed to unwanted sound (noise).

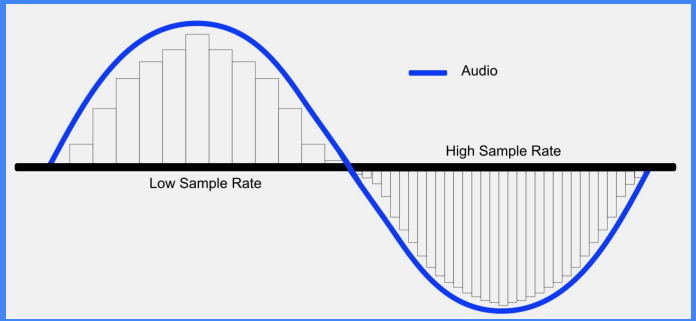
Method used: WADASNR

# Sampling Rate



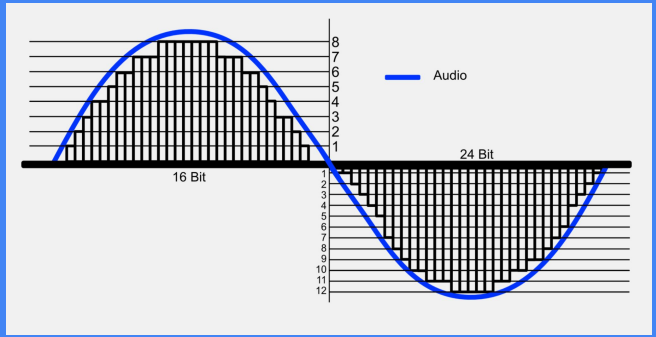

- Since computers “think” in discrete steps, in order to convert analog audio signals to the digital domain, it’s necessary to describe the continuous analog waveform mathematically as a succession of discrete amplitude values.
- In an analog-to-digital converter, this is accomplished by capturing, at a fixed rate, a rapid series of short “snapshots”—samples—of a specified size. Each audio sample contains data that provides the information necessary to accurately reproduce the original analog waveform.
- The sampling rate refers to the number of samples of audio recorded every second. It is measured in samples per second or Hertz (abbreviated as Hz or kHz, with one kHz being 1000 Hz).
- The sampling rate is analogous to the frame rate or FPS (frames per second) measurement for videos.
- For model training 16 kHz is used and we downsample all audio to 16 Khz since human voice pitch is contained within 8 KHz frequency.

# Sampling Rate

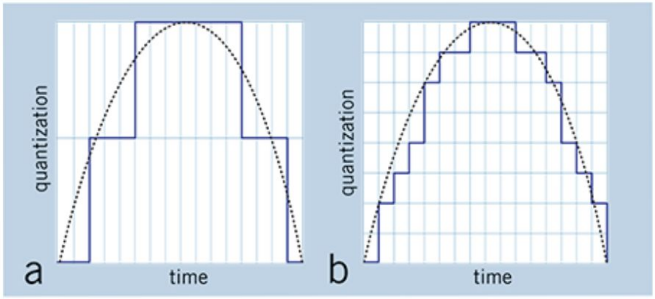



```
Using Sox

For Linux:

>> sudo apt-get install -y sox

For Mac:

>> brew install sox

>> soxi -r some_audio.file -n stats
```

# Sample Depth




- Measured in bits per sample, the sample depth, represents the level of detail, or “quality” each sample has.
- For an audio sample, it simply means that the audio sample can represent a higher range of amplitudes.
- A sample depth of 8 bits means that we have 2^8 = 256 distinct amplitudes and so on for higher sample depths.




- This is analogous to the 8bit or 16bit numbers we might hear about regarding image quality.
- A higher bit depth in a pixel yields a pixel that is more color-accurate, since the pixel has more bits to “describe” the color to be represented on a screen, and the pixel or image overall would look more realistic to how one would see it in real life.
- More technically, the bit depth of a pixel indicates how many distinct colors can be represented in the pixel.

# Bit Rate

- Tying the sampling rate and the sample depth together is the bit rate, which is simply the product of both.

- Since the sampling rate is measured in samples per second and the sample depth is measured in bits per sample, it is therefore measured in (samples per second) x (bits per sample) = bits per second, abbreviated as bps or kbps.

**Bit rate = (bits / sample)  x (samples / second)**

**Bit rate = bits / second**

# Hands on

# Infra Setup

- Infrastructure as code using Terraform
- Documentation

# Infra Setup

**Pre requisites:**

- Install terraform (https://learn.hashicorp.com/tutorials/terraform/install-cli)
- Install cloud sql proxy (https://cloud.google.com/sql/docs/mysql/connect-admin-proxy#linux-64-bit)
- Install gcloud (https://cloud.google.com/sdk/docs/install)
- Install kubectl (https://kubernetes.io/docs/tasks/tools/)
- Install K9s [optional] (https://k9scli.io/)
- GCP Account

**Components**

- Composer
- Database
- Buckets
- Service accounts

# Configuration for data pipelines

**Data collection config:**

```
channel_url_dict = {

"https://www.youtube.com/channel/UC_MJkF5uaauVVOH
UVgK5VUQ": "KANNADA_AUDIOS_TEST"

}
```

**Data processing config:**

```
"snrcatalogue": {

  "KANNADA_AUDIOS_TEST": {

    "count": 5,

    "format": "mp4",

    "language": "kannada"

  }

}
```

# Code Walk through

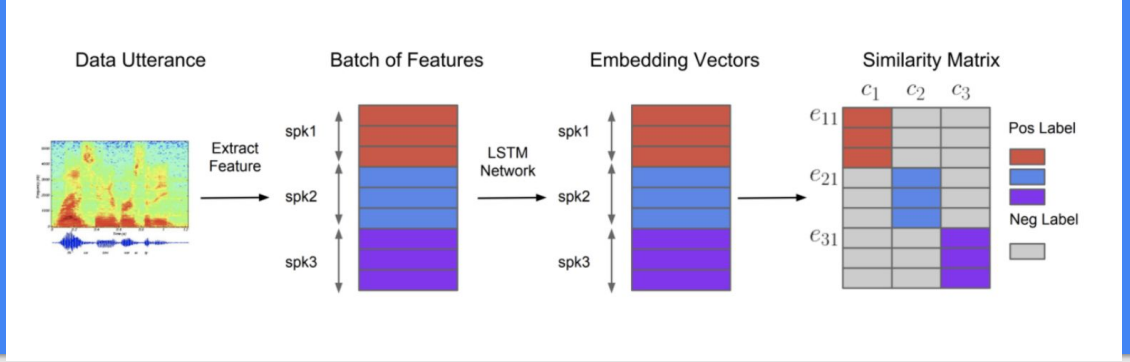
# Data Analysis and Transcription

July 23, 2021

# Data Analysis

- Language Identification
- Gender Identification
- Speaker clustering

# Voice Encoder




Embeddings of audio are generated using Voice encoder model based on Mel Spectrogram.To convert our audio utterances into fixed length summary vectors, we use , Resemblyzer (implementation of Voice Encoder model) for converting our audio utterances into fixed length embeddings.

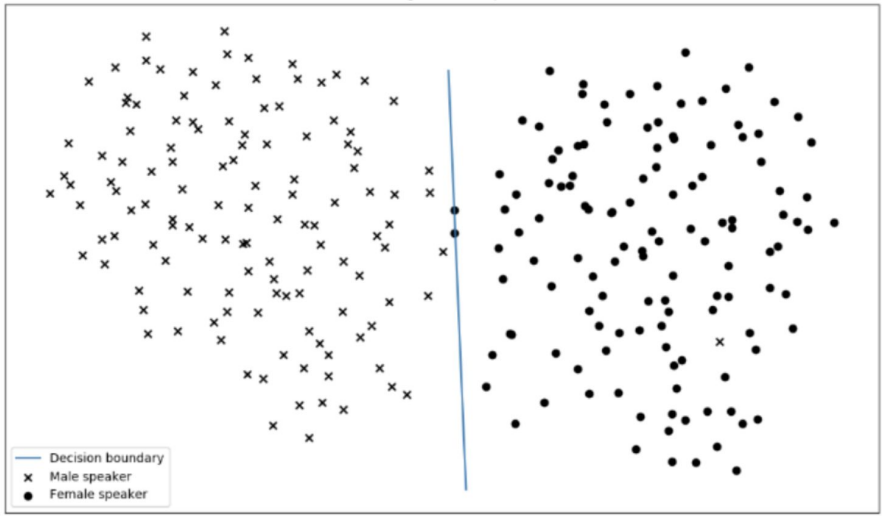
Voice Encoder is a speaker-discriminative model trained on a text-independent speaker verification task. Since these embeddings are able to summarise the characteristics of the voice spoken, they have been used for Gender Classification and Speaker clustering

# Gender Identification

- Documentation

# Gender Identification

Once embeddings are created, Support Vector Machines algorithm is used for classification




**Train data**: Male: 17.4 hours - female: 16.2 hours

**Language Representation**: Hindi, Tamil, Telugu, Kannada

.

**Test data**¶: Male: 3.6 hours - Female: 3.6 hours

**Language Representation:** Hindi, Tamil, Telugu, Kannada, *Marathi and Bengali*.

# Speaker Clustering

- Documentation

# Speaker Clustering

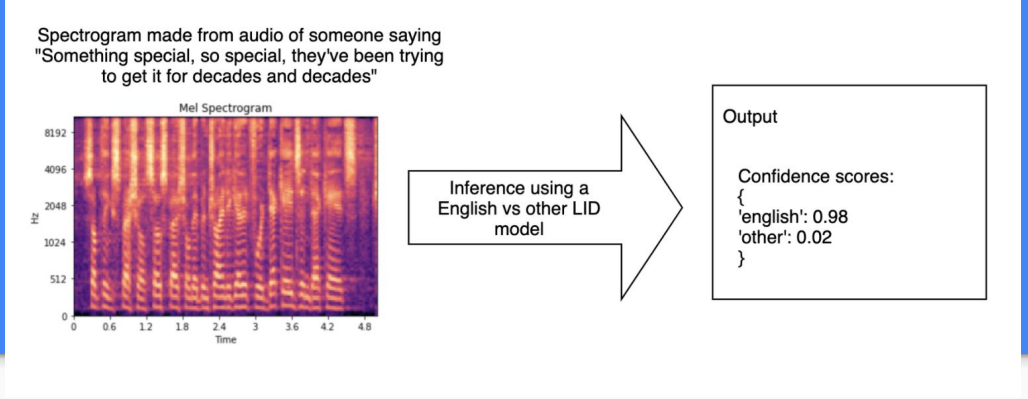
Once embeddings are created, clustering is performed using Hierarchical Density-Based Spatial Clustering of Applications with Noise (**HDBSCAN**) as our core clustering algorithm. This step also classifies some points as noise points - meaning they couldn't be used up in any clusters formed .We keep a record of all these noise points for fitting later.

More Details

# Language Identification

- Documentation

# Language Identification




We extract spectrograms from audio utterances and use them to train a CNN based classifier: ResNet18, which is 18 layers deep. We don't use the pretrained version, and treat Language Identification as an image classification problem.

Users can decide upon the number of classes they want to train on. We follow a one vs other approach where main language is the one we want to classify against.

More Details

**English vs others classification model**

**Train data**¶: 30-40 hours data for each of the classes

**Test data :** 14 hours of data balanced between the classes.

Confidence score threshold is chosen for higher recall for Pre-training data and for higher precision for Fine-tuning data.

Completely relying on language confidence score is not recommended for Fine tuning datasets

# Data Balancing

Number of Speakers

Gender ratio

Language

Duration filter

SNR filter