**1) Objectives (what “good” looks like)**

* **Primary goal:** Isolate *his* voice (not crowd/noise) and produce a **mastered, segment-aligned, richly-annotated** dataset.
* **Secondary goal:** Extract **standard, industry-used acoustic & prosodic features** at multiple tiers (frame, phone, syllable, word, intonation phrase, utterance).
* **Non-goals (now):** ASR/TTS training. (We prepare datasets so those steps later are straightforward.)

**2) Data governance & ethics**

* **Rights/consent:** Confirm usage permissions for speeches/broadcasts. Keep a provenance ledger (source URL, date, venue, mic type if known).
* **PII/security:** Public figure speech is usually fine; still redact incidental bystanders if needed in transcripts.
* **Versioning:** Treat *audio masters* as immutable; all processing outputs are derived versions with full metadata and hashes (MD5/SHA256).

**3) Source ingestion & digitization (the “master”)**

* **File format:** WAV, **48 kHz/24-bit PCM** (or original sample rate if higher). Keep **original uncompressed** master.
* **Resampling:** If needed, use bandlimited, linear-phase, very-high-quality resampler (e.g., SoX rate -v -s).
* **Channel policy:** Mono preferred for analysis; if stereo, retain stereo master and create a mono analysis stem via transparent downmix.

**Deliverables:**  
masters/ (untouched), analysis\_bases/ (mono resampled copies), metadata/sources.csv.

**4) Noise, interference & room control (voice-first signal chain)**

Apply conservatively; prioritize **speech naturalness** over maximum noise removal.

**Recommended order:**

1. **Broadband hum/buzz removal:** Notch/hum at 50/100 Hz (+ harmonics) if present.
2. **Dereverberation:** **WPE** (Weighted Prediction Error) or equivalent dereverb; mild settings to preserve timbre.
3. **Denoising:** Start with statistical **Wiener** or **MMSE-LSA**; evaluate neural denoisers on copies (e.g., RNNoise, Demucs). Keep the least artifact-prone.
4. **Source separation (optional):** If chants/music overlap, test **Conv-TasNet/DPCL++** music/noise-vs-speech separation to reduce bleed.
5. **Spectral repair (optional):** Short dropouts/clicks via interpolation/click removal.
6. **Dynamics:** Gentle **downward expansion** (floor -30 to -40 dBFS) to avoid pumping. No heavy compression at this stage.
7. **Loudness normalization:** For analysis copies, target **-23 LUFS** (EBU R128) or simple RMS norm to a modest peak (e.g., -1 dBFS). Keep pre-normalized versions as well.

**Artifacts watchlist:** musical noise (spectral subtraction), “phasey” tail (over-dereverb), transient smearing. Always A/B against master.

**Deliverables:**  
clean/ + clean\_meta.csv with parameters per file (noise profile, WPE taps, SNR before/after, SI-SDR deltas).

**5) Speech activity, diarization & segmentation**

* **VAD:** Robust VAD (e.g., WebRTC VAD, Silero VAD) → generate **speech-only masks**.
* **Diarization:** **pyannote.audio** pipeline to confirm single-speaker dominance; label any **interjections** (crowd/host).
* **Segment policy:**
  + *Utterance segments* (~4–20 s) bounded by pauses/breaths;
  + *Prosodic phrases* within utterances (ToBI-style boundaries);
  + Maintain **no crossfade**; include natural leading/trailing room if subtle.
* **Alignment scaffolding:** Store **timestamps** and **segment IDs** now; text alignment comes later.

**Deliverables:**  
segments/ (WAV per segment), segments/TextGrid (tier “VAD” & “Speaker”), segments/segments.csv.

**6) Transcription & normalization (Sindhi first)**

* **Orthography:** Sindhi Perso-Arabic script (Nastaʿlīq), **no Roman** in the master transcripts.
* **Normalization rules (document!):**
  + Consistent diacritic policy (usually **no short-vowel diacritics** except for disambiguation examples set aside);
  + Consistent punctuation; normalize numerals; unify common variants/spellings; expand abbreviations; mark code-switching spans (e.g., [URDU], [EN]).
* **Tiered transcripts:**
  + **Verbatim layer** (exact words, fillers, discourse markers);
  + **Normalized layer** (for alignment/ASR);
  + **Translation layer** (gloss in English/Urdu for cross-checkers).

**Deliverables:**  
transcripts/utterance\_id.{txt,json}, validation/orthography\_guidelines.md.

**7) Phonology & G2P for Sindhi (Laari-aware)**

* **Phone inventory:** Build an **IPA-based phone set** covering Sindhi contrasts + **Laari realizations** (e.g., aspiration weakening, retroflex /ɽ/ ~ alveolar tap realizations, cluster simplifications, vowel length/quality shifts).
* **Lexicon:** lexicon.txt with word → phones (primary), and lexicon\_laari.txt with variant mappings observed in his speech.
* **G2P:** Train a small G2P (e.g., Phonetisaurus/G2P-seq2seq) on curated lexicon; hand-review OOVs from his writings/speeches.

**Deliverables:**  
phonology/phones.txt, phonology/lexicon\*.txt, phonology/g2p\_model/.

**8) Forced alignment (phone/syllable/word)**

* **Tooling:** **Montreal Forced Aligner** (if Sindhi model available) or **Kaldi/ESPnet** custom acoustic model bootstrapped from your clean segments.
* **Strategy:** Start with high-SNR segments to train an initial acoustic model; align the rest iteratively.
* **Outputs:** **Praat TextGrid** tiers: phones, syllables, words, utterance. Keep **alignment confidence** per unit.

**QC:** Random spot-checks + boundary error stats (mean abs boundary error; ≤ 20–30 ms target for clean speech).

**Deliverables:**  
alignments/\*.TextGrid, alignments/qa\_report.json.

**9) Prosody & discourse annotation (Sindhi ToBI-LR)**

Create a **lightweight ToBI-style** scheme adapted to Sindhi/Laari:

**Tiers & labels**

* **Break indices:** 0 (clitic), 1 (word), 2 (minor), 3 (intermediate), 4 (intonation phrase).
* **Pitch accents:** L\*, H\*, L+H\*, H+L\*, L\*+H (choose a compact subset you can label reliably).
* **Boundary tones:** L-, H- (intermediate); L-L%, L-H%, H-L%, H-H% (intonation phrase).
* **Discourse/pragmatics:** [CALL-RESP], [APPLAUSE-COVERED], [CHANT-UNDER], [CODE-SWITCH:URDU/EN], [SLOGAN], [LAARI-INDEX] (where a clear dialectal feature is salient).
* **Rhetoric units:** Mark **Triads** (enumerative 3-item sequences), **Contrast pairs**, **Metaphor anchors** (Indus/water/land motifs).

**Deliverables:**  
prosody/utterance\_id.TextGrid (tiers: tones, breaks, discourse), prosody/handbook.md (examples + audio snippets).

**10) Acoustic & phonetic measurements (targets & methods)**

**Voice quality & phonation**

* **F0** (RAPT/YIN/REAPER): mean, median, range, 5th–95th percentiles, contour shape stats.
* **Jitter / Shimmer** (cycle-based on clean vowels); **H1-H2**, **H1-A3**, **CPP** (cepstral peak prominence).
* **HNR** (Harmonics-to-Noise Ratio) tracks.

**Formants & vowel space**

* **F1–F3** @ vowel midpoints; Bark/Mel transforms; per-vowel ellipses; **VSA** (Vowel Space Area), **FCR** (Formant Centralization Ratio).

**Spectral envelope / energy**

* **Spectral slope** (0–4 kHz), **alpha-ratio**, **LTAS**; **band energy** (250–500–1000–2000–4000 Hz).

**Rhythm & timing**

* **Speech rate** (syll/s), **articulation rate** (exclude pauses), **%V**, **ΔC**, **VarcoV**, **nPVI** (syllable timing).
* **Pause distribution**: counts, duration histograms; **silent vs filled**.

**Prosodic architecture**

* **Accent density** (accents per second/word), **phrase length** stats, **boundary tone distribution**.
* **F0 excursion at accents**, **final lowering**, **post-focus compression** (if applicable).

**Consonant realization (Laari salient)**

* **Retroflex vs dental/apical taps** timing & spectral cues;
* **Aspiration strength** (VOT for plosives, breathy onset energy);
* **Cluster realization** (dr/tr simplifications)—durational patterns.

**Recommended tools:** Praat/Parselmouth, REAPER, librosa/torchaudio, pyWorld, covarep, emuR. Document exact versions & params.

**Deliverables:**  
features/phonetic/ (CSV/Parquet per tier), features/docs/measurements.md.

**11) Standard DSP & ML feature sets (ready for modeling later)**

**Low-level descriptors (frame-level)**

* **MFCC (13/20/40)** + Δ/ΔΔ, **PLP**, **Log-Mel filterbanks** (80 or 128), **PNCC** (robustness to noise), **LPC** (order 12–20).

**Voice quality & prosodic**

* **eGeMAPS/ComParE** sets (openSMILE recipes) for paralinguistics.
* **Pitch, energy, voicing probability** contours; **syllable-synchronized** summaries.

**Articulatory proxies**

* **Formants**, **bandwidths**, **spectral moments** for sibilants; **VOT** distributions (where measurable).

**Glottal analysis**

* **IAIF** or **QUASI-closed phase** to estimate **glottal flow**; derive **Rd**, **Ee**, **OQ**, **NAQ**.

**Deliverables:**  
features/llf/ (HDF5/NPZ), features/prosody/, features/glottal/, plus a **features manifest** mapping back to segments and text tiers.

**12) Dataset structure (reproducible & scalable)**

dataset\_root/

masters/

clean/

segments/

transcripts/

alignments/

prosody/

features/

metadata/

docs/

**Key metadata files**

* metadata/segments.csv (id, source\_id, t\_start, t\_end, SNR, LUFS, diar\_conf, notes)
* metadata/text.csv (id, script\_text, normalized\_text, transliteration\_opt, language\_spans)
* metadata/speaker.json (biographical, dialect tags = {"laari": true, "thatta": true})
* metadata/processing\_log.jsonl (each step with parameters & hashes)

**13) Quality control & benchmarks**

**Signal quality**

* **SNR** (ITU-T P.56 style), **PESQ**, **STOI**, **SI-SDR** vs master. Keep artifact scorecards for each denoising profile.

**Alignment quality**

* **Boundary error** (ms) against hand-labeled gold subsets; target ≤ 20–30 ms mean absolute.
* **Phone/word alignment confidence** distribution; manual review of outliers.

**Annotation reliability**

* Inter-annotator agreement on **tone labels, break indices, dialect flags** (Cohen’s κ).
* Weekly adjudication sessions; update the guidebook.

**Versioning**

* **Semantic version the corpus**: v0.1 (10h pilot) → v0.2 (cleaned) → v1.0 (full). Keep changelogs.

**14) Tooling stack (battle-tested & standard)**

* **Audio I/O:** SoX, FFmpeg.
* **DSP:** librosa, torchaudio, pyRoomAcoustics (WPE), rnnoise/demucs (optional path), scipy.signal.
* **Diarization/VAD:** pyannote.audio, WebRTC VAD, Silero VAD.
* **Alignment:** Montreal Forced Aligner (if Sindhi model—else Kaldi/ESPnet custom).
* **Annotation:** Praat (TextGrid), ELAN (EAF).
* **Prosody/phonetics:** Praat/Parselmouth, REAPER, covarep, emuR.
* **Feature extraction:** openSMILE (eGeMAPS/ComParE), librosa/torchaudio, pyWorld.
* **Data ops:** Python + Pandas/Polars, Parquet, DVC or Git-LFS for storage/versioning.

**15) Pilot plan (10 hours → template for scale)**

1. **Select 10 h** spanning: rallies (crowd), TV studio (clean), seminar (medium reverb), interview (code-switching).
2. Run **two denoise chains** (classical vs neural) on a 1 h subset; pick the safer chain by blind listening + SI-SDR/PESQ deltas.
3. VAD + diarization; hand-audit 30 mins.
4. Transcribe → normalize → build **seed lexicon** (2–3k frequent words).
5. Train quick **alignment model** (or MFA with a close acoustic model) → align the 10 h → hand-check 1 h stratified.
6. Label **prosody on 30 mins** (ToBI-LR) to test scheme reliability.
7. Extract **core features** (MFCC/logmel/eGeMAPS, F0/energy) → produce **pilot analytics** (his F0 range, speech rate, phrase length histo, accent density).
8. Write **v0.1 report** with decisions, pitfalls, and parameter locks.

**16) Laari-specific watchpoints (so we truly capture *his* voice)**

* **Retroflex vs alveolar taps**: Make sure phone set and alignment distinguish **/ɾ/ vs /ɽ/**; annotate tokens where Laari realization surfaces.
* **Aspiration weakening**: Measure **VOT** in voiceless/voiced plosives across contexts; keep per-environment stats (word-initial, post-sonorant).
* **Cluster simplification**: Track **dr/tr → d/t** realizations—mark in lexicon variants; quantify rates by context and speech style (rally vs studio).
* **Prosodic cadence**: Capture **final holds**, **wide F0 excursions** at slogans, and **pause-led call-and-response** timing.

**17) Documentation & reproducibility**

* A single **docs/pipeline.md** with **exact commands**, seeds, and parameters.
* A **Makefile/invoke/Make-Flow** that runs end-to-end (ingest → clean → VAD → diarize → segment → align → features → QC).
* **Unit tests** for parsers (TextGrid <-> CSV), feature shapes, timestamp integrity.

**18) What you hand off to DSP & ML teams (clear interfaces)**

* **Clean segments** + **TextGrid tiers** (phones/syllables/words/phrases/tones).
* **Feature bundles**:
  + LLF.h5 (MFCC/logmel/PLP + Δ/ΔΔ @ 10 ms hop)
  + Prosody.csv (F0/energy/voicing + phrase/word summaries)
  + Phonetic.csv (formants, jitter, shimmer, CPP, HNR, VOT, spectral moments)
  + Glottal.csv (IAIF/QUASI params)
* **Metadata** linking each row to audio/time spans and text.
* **Guidebooks**: orthography & normalization, ToBI-LR, phonology/lexicon, processing parameters.
* **QC reports** with plots & thresholds.

**A few practical tips (hard-won)**

* Keep **two parallel “clean” sets**: *conservative* and *aggressive* denoise. Many ML front-ends prefer the conservative one.
* Always keep **a tiny dev set** of raw vs processed for **regression tests** when you tweak DSP.
* Start **formant tracking** on handpicked, steady-state vowels first; tune LPC orders and pre-emphasis per mic/venue, then batch-run.
* For **pitch**, compare **REAPER vs YAAPT**; lock one and note its failure modes (falsetto, chant overlap).
* Use **checksum gating**: any change in upstream audio invalidates downstream artifacts automatically (DVC can help).