Checkpoint 1: Chordonomicon Project

Problem Definition

Main Question. How does harmony shape the sound of popular music, and can harmonic "fingerprints" (chord progressions, key usage, modulation patterns) be used to classify songs by decade and genre?

Decision/Action Informed.

- For researchers and musicians: insights into how harmony defines genres and how those harmonic fingerprints evolve across time.
- For streaming platforms: evaluate whether harmony alone can improve genre tagging and recommendation engines.
- For portfolio/recruiters: demonstrate a full data science workflow (EDA, modeling, KPIs, interpretation).

Stakeholders.

- Researchers and theorists: care about interpretable insights into harmonic change.
- Musicians: want to see how harmonic conventions differ by style and era.
- Streaming platforms: interested in practical classification and prediction.
- Non-technical audiences: benefit from clear visuals and storytelling.

Unit of Analysis. Song-level representation, with features such as chord n-grams, chord intervals, key/key confidence distributions, and modulation frequency.

Scope and Boundaries.

- Time horizon: 20th–21st century (per dataset).
- Geographic/population scope: primarily Western popular music (dataset bias).
- Features included: chords, chord sequences, keys/modes, modulation patterns.
- Features excluded: lyrics, timbre, instrumentation, production quality, cultural/marketing factors.

Anti-goals.

- Not a full recommendation engine.
- Not analyzing non-Western music.
- Not modeling lyrical, timbral, or production-based features.
- Not claiming harmony alone explains popularity.

Data Gathering

Source Identification.

- Primary: Chordonomicon dataset ($\approx$ 666,000 chord-annotated songs).
- Supplementary: Spotify API (release year, genre labels, popularity score).
- Other sources (optional): Million Song Dataset, Billboard archives for metadata.
- Tools considered: flat file storage with pandas; no databases (e.g., sqlite3, SQLAlchemy, pymongo) expected at this stage.

Acquisition Strategy.

- Chordonomicon: one-time dataset download.
- Spotify API: scripted acquisition with rate-limit handling (e.g. `spotipy`).
- Save raw snapshots in `data/raw/` for reproducibility.
- Explicit choice: one-time acquisition for reproducibility; no automated pipeline planned.

Provenance.

- Record dataset version, download URL, and date.
- Log Spotify API queries (timestamps, song IDs).

Ethical/Legal Considerations.

- Chordonomicon: confirm licensing, open-source use.
- Spotify API: respect developer terms of service; no scraping beyond documented endpoints.
- No personally identifiable data used.

Data Assessment

Volume and Coverage. $\sim 666{,}000$ songs, sufficient for EDA and modeling.
Granularity. Chords are annotated at the section level (verses, choruses), which can be aggregated into song-level features.
Bias and Representativeness. Western-pop bias, sparse coverage of niche/independent artists, exclusion of songs with 3 chords or less, temporal bias toward recent decades. Possible class imbalance: certain decades/genres are much more represented than others. Oversampling or weighting may be needed.

Assessing Learnability

Signal vs. Noise. Harmony plausibly encodes genre and era information.
Data Sufficiency. Enough examples per class (decade/genre). Time horizon long enough to capture temporal trends.
Feature-Target Alignment. All chordal features are observable at release time (no leakage).
Back-of-the-Envelope Baselines.

- Dummy classifier (predict most common decade/genre).
- Logistic regression and random forest with minimal preprocessing.
- Cross-validation to confirm predictive signal beyond trivial baselines.
- Baseline KPIs will be recorded in a summary table comparing Dummy, Logistic Regression, and Random Forest across CV folds.

Domain Sanity Check. Music theory supports the hypothesis: harmony is genre/era-specific, so classification is realistic.

KPI Definition

Primary KPI. Classification accuracy (percentage of correctly classified songs).
Secondary KPIs.

- F1 score (balances precision/recall).
- Confusion matrix (to visualize which genres/decades are often confused).
- Interpretability metrics: feature importances (random forest), SHAP values, or logistic regression coefficients.

Baseline Definition.

- Dummy classifier (predict majority class).
- Logistic regression (linear baseline).
- Random forest (tree-based baseline).
- Evaluate all with cross-validation (e.g. KFold).

Deliverables

- `README.md`: project description, problem statement, links to notebooks.
- `data_inventory.md`: document data sources (Chordonomicon, Spotify API).
- Data acquisition scripts in `src/data/`.
- Raw data snapshots in `data/raw/`.
- Baseline notebook: `notebooks/baseline.ipynb`.
- KPI definition file: `kpis.md`.
- Environment file: `requirements.txt` or `environment.yml`.
- (Optional) Provenance log: `logs/` for script-generated timestamps, queries, and file hashes.