

OWNML MACHINE LEARNING CANVAS

Designed for: Turkish Music Model Designed by: Team 24 MLOps Date: Oct, 2025 Iteration: One

PREDICTION TASK



Clasificación supervisada multiclase (single-label) para inferir la emoción predominante en un clip de música turca. La unidad de predicción es cada fragmento de audio ya preprocesado (normalización, duración fija).

Resultados y observación.

El modelo predice una de 4 clases: Feliz, Triste, Enojado, Relajado (salida: probabilidades y clase por argmax). Las etiquetas existen en el dataset; en producción se confirman a posteriori vía anotación/feedback de usuario para cerrar el ciclo de reentrenamiento.

DECISIONS



Cada clip se etiqueta con una emoción (**Feliz/Triste/Enojado/Relajado**) y una confianza. Esa etiqueta alimenta: (1) un recomendador por estado de ánimo (playlists relajantes/energizantes), (2) enriquecimiento del catálogo con metadatos emocionales para búsqueda/ordenación, y (3) dashboards afectivos para análisis (por artista, álbum o sesión de musicoterapia). Para decisiones a nivel pista, agregamos múltiples clips por voto mayoritario o promedio de probabilidades.

Parámetros de decisión (app/pipeline):

Regla principal: argmax si confianza ≥ 0.75 ; si no, Top-2 o "Revisión". Agregación por pista con ventana configurable (p. ej., 30-60 s). Objetivo de latencia < 2 s/clip en línea; procesamiento batch para bibliotecas grandes. Interpretabilidad con SHAP/LIME visible en el dashboard. A/B testing para umbrales y presentación; feedback del usuario registrado como ground truth para reentrenamiento (programado o por drift).

VALUE PROPOSITION



Beneficiarios: plataformas musicales (curaduría/recomendación), musicólogos/investigadores y clínicas de musicoterapia.

Dolores: etiquetado emocional manual e inconsistente; falta de modelos para música turca/no occidental; baja trazabilidad y poca escalabilidad del análisis.

Integración y flujo: API REST/Batch que enriquece catálogos con emoción + confianza y se conecta al recomendador/ETLs; dashboard con distribuciones por emoción y explicaciones (SHAP/LIME). Umbrales configurables, alertas por drift y feedback para retraining.

DATA COLLECTION



Data Collection — Fuente inicial

Dataset modificado/limpio Turkish Music Emotion: 400 clips de 30 s (≈ 100 /clase: Feliz, Triste, Enojado, Relajado). Extraemos MFCC, cromas, tempo, ZCR, RMS a CSV/Parquet; audios/features versionados con DVC (raw→processed).

Actualización continua (frescura/costo)

Ingesta periódica desde Spotify/YouTube APIs con deduplicación; extracción incremental y caché solo para nuevos/cambiados; batch semanal. Costos: storage tiering y retención. Etiquetado humano para baja confianza; retraining por tiempo o drift.

DATA SOURCES



Internas (principal)

Repositorio DVC/S3:

Audio WAV/MP3 (carpetas raw/processed) y features tabulares (CSV/Parquet).

Tablas:

- tracks (id_track, artista, título, fuente, género, año)
- clips (id_clip, id_track, start_s, dur_s, sr, bitrate)
- labels (id_clip, emoción ∈ {Feliz, Triste, Enojado, Relajado}, fuente_etiqueta)
- features (id_clip, mfcc_, chroma_, rolloff, zcr, rms, tempo)
- feedback (id_clip/pista, predicción, confianza, "correcto/incorrecto", timestamp)

Externas (potenciales)

- Spotify Web API: search, tracks/{id}, audio-features, audio-analysis (metadatos, tempo, key, duración).
- YouTube Data API v3: search.list, videos.list (metadatos e IDs; ingesta de audio sujeta a licencias).
- Repos públicos etiquetados (p. ej., corpus de música turca con emociones) para ampliación y benchmark.

Conectores/Extracción: Librosa (MFCC, cromas, espectrales, tempo) y Pandas para consolidar a tablas versionadas con DVC.

<div> <div>IMPACT SIMULATION</div> <div> </div> </div> <p>Impact Simulation</p> <p>Costo/Ganancia. Predicción correcta → mayor uso del recomendador y confianza; Incorrecta → degradación de UX y credibilidad ($\approx 10\text{--}15\%$ de caída por cada $+10\%$ de error).</p> <p>Datos para simular. Train/val/test (80/20), k-fold estratificado, piloto en catálogo (batch replay) y muestra con etiquetado humano.</p> <p>Criterios de despliegue. F1_macro ≥ 0.78, Accuracy $\geq 80\%$, latencia < 2 s/clip, confianza calibrada, artefactos en MLflow y datos DVC.</p> <p>Equidad. Métricas por clase y género/tempo; Δ F1 por clase ≤ 10 pts; revisión manual para baja confianza y monitoreo de drift tras el despliegue.</p>	<div> <div>MAKING PREDICTIONS</div> <div> </div> </div> <p>Modos: Batch (offline) y tiempo real (API).</p> <p>Frecuencia: batch al llegar nuevos lotes o en ventana nocturna; online on-demand por clip/pista.</p> <p>SLA/tiempo: < 2 s/clip end-to-end (prepro + features + inferencia + decisión); caché de features y micro-batching cuando aplique.</p> <p>Recursos: CPU optimizada (multihilo) para batch; GPU opcional si el modelo lo requiere; servicio Docker + FastAPI, orquestado (p. ej., K8s/ECS) con autoscaling.</p> <p>Integración: API REST/job batch que retorna probabilidades + clase y registra confianza/telemetría para retraining.</p>		<div> <div>BUILDING MODELS</div> <div> </div> </div> <p>En producción: 1 clasificador multiclase (emociones); modelo sombra para A/B en actualizaciones; calibración opcional.</p> <p>Actualización: Trimestral o por drift/$\geq 10\text{--}15\%$ de nuevos datos etiquetados; promoción controlada en MLflow Registry (datos en DVC).</p> <p>Tiempo (E2E): ventana nocturna (≤ 8 h) para featureado incremental + entrenamiento + análisis (k-fold, F1 macro, matriz de confusión).</p> <p>Recursos: CPU optimizada para Sklearn/featureado; GPU (T4) opcional si se prueba un modelo profundo.</p>	<div> <div>FEATURES</div> <div> </div> </div> <p>Representación (predicción): vector por clip con MFCC (1–13 + $\Delta/\Delta\Delta$), Chroma, Spectral Centroid/Rolloff/Contrast, ZCR, RMS/Energy y Tempo; se agregan por ventanas a estadísticos (media, std, p10/p90) para un único vector por clip.</p> <p>Transformaciones: audio → mono 22.05 kHz y loudness norm; framing (≈ 1 s / 0.5 s hop); z-score (parámetros del train), imputación si aplica; PCA opcional (20–40 comps). Paridad train/serve en el mismo pipeline (librosa/sklearn), caché y versionado DVC (CSV/Parquet).</p>
	<div> <div>MONITORING</div> <div> </div> </div> <ul style="list-style-type: none"> Métricas técnicas (SLIs/SLOs): F1_macro y F1 por clase, Accuracy, latencia p95 < 2 s/clip, tasa de error, calibración (ECE), drift de datos/modelo (PSI/KI, cambio en distribución de clases), cobertura y % de predicciones de baja confianza. Métricas de negocio: uso del recomendador (CTR, skips, tiempo en sesión), satisfacción (CSAT/NPS), tasa de feedback/etiquetado y acuerdos entre anotadores. <p>Revisión & acciones:</p> <ul style="list-style-type: none"> Alertas en tiempo real (latencia, caída de F1, drift): revisión semanal operativa y mensual de negocio en dashboards (Prometheus/Grafana + MLflow). Triggers: retraining/umbral cuando $\Delta\text{F1_macro} > 5\text{--}10$ pts o drift significativo; rollback/canary si empeora; seguimiento de fairness (ΔF1 por clase ≤ 10 pts). 			



Version 1.2. Created by Louis Dorard, Ph.D. Licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/). Please keep this mention and the link to ownml.co when sharing.

[OWNML.CO](https://ownml.co)

[FREE] ONLINE COURSE

Introduction to the Machine Learning Canvas

Get started with the MLC in this short course taught by its author.



Start Here



[Video] Overview of the 10 boxes that make up the MLC

3:31



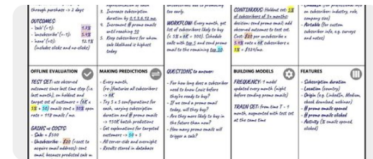
[Video] Why use the MLC?

2:34



[Video] Structure of the MLC

2:49



Get your AI-generated canvas



Prepare for implementation



Start now at ownml.co/intro