

Detección de Emociones de Música Turca

Fase 1

David Alejandro Velázquez Valdez A01632648

Christian Gustavo Martínez Ramírez A01796999

César Gustavo López Zamarripa A00967602



Daniel Vinicio Espinosa Herrera A01796585

Equipo 3 Marcelo Alanis Alvarez A01796009

Agenda



- 1 Introducción y Contexto >
- 2 Análisis de Requerimientos y Propuesta de Valor >
- 3 Manipulación y Preparación de Datos (Limpieza) >
- 4 Exploración y Preprocesamiento Avanzado >
- 5 Versionado y Trazabilidad de Datos (DVC) >
- 6 Construcción, Ajuste y Evaluación de Modelos ML >
- 7 Conclusiones y Reflexión Final >
- 8 Referencias >

Contexto

Tipo de Problema: Clasificación Multiclase Supervisada. Se busca categorizar fragmentos en cuatro emociones discretas:

Joy/Happiness (Alegría/Felicidad)

Sadness/Grief (Tristeza/Pena)

Anger (Enojo)

Peace/Calmness (Paz/Calma)

Definición de MER: El Reconocimiento de Emociones Musicales (MER) busca clasificar automáticamente la emoción transmitida por una pieza musical basándose en sus características acústicas.

Contexto del Dataset: Se utiliza el dataset de Turkish Music Emotion, enfocándose en un contexto musical cultural específico.

El dataset consta de 50 características acústicas de alto nivel pre-extraídas (ej. MFCCs, Tempo, Energía), típicas del dominio de Music Information Retrieval (MIR).



Objetivos de la fase 1

1	2	3	4
Analizar la problemática y documentar los requerimientos (ML Canvas).	Realizar la limpieza y preparación modular de los datos.	Establecer un Baseline Sólido de modelado utilizando algoritmos de Machine Learning tradicionales.	Garantizar la trazabilidad y reproducibilidad del dataset mediante DVC (Data Version Control).

Análisis de requerimientos

Propósito del Modelo

Clasificar la emoción de una pieza musical con alta precisión.

50

Features acústicas extraídas de entrada

Métricas de Éxito:

Principal: F1-Score (Weighted) (debido al posible desbalance).

Secundaria: Accuracy.

Problema de negocio

Mejorar la capacidad de las plataformas de streaming para organizar y recomendar contenido, creando listas de reproducción basadas en el estado de ánimo (Mood-based Playlists).

Un modelo baseline interpretable y eficiente que sirva de punto de partida para fases posteriores.

Resultado deseado

OWNML MACHINE LEARNING CANVAS

Designed for:

Etapa 1 – Detección de Emociones en Música Turca

Designed by:

Equipo 3

Date:

7 de Octubre del 2025

Iteration: 1

PREDICTION TASK	DECISIONS	VALUE PROPOSITION	DATA COLLECTION	DATA SOURCES
<div><p>?</p></div> <p>- Es un tipo de problema de clasificación supervisada multiclase, donde el modelo debe asignar una de cuatro categorías de emoción a un fragmento musical.</p> <p>- La entidad sobre la que se hacen predicciones es cada fragmento musical de 30 segundos, representado por sus características acústicas.</p> <p>- Las etiquetas de salida (outcomes) son cuatro emociones musicales: relax, happy, sad, angry.</p> <p>- Los outcomes se observan inmediatamente después de que el sistema analiza un fragmento de audio. En el dataset, las etiquetas ya están definidas, mientras que en un caso real se obtendrían: al clasificar un nuevo archivo musical, o al integrarse en tiempo real en una plataforma de streaming que evalúa canciones a medida que se reproducen.</p>	<div><p>↻</p></div> <p>Las predicciones se convierten en decisiones al combinar dos parámetros principales:</p> <p>- La emoción predicha (etiqueta de salida).</p> <p>- El nivel de confianza de la predicción (grado de certeza).</p> <p>Esto permite que el sistema brinde recomendaciones relevantes y personalizadas, aplicables tanto en servicios de <i>streaming</i>, como en terapia musical o gestión de catálogos musicales.</p>	<div><p>📁</p></div> <p>Se pueden implementar diversos usos y aplicaciones una vez lanzado el modelo, como lo serían:</p> <p>- Usuarios de apps de música obtienen playlist y recomendaciones basadas en emociones, no solo en géneros.</p> <p>- Terapeutas y apps de bienestar tienen acceso rápido a música clasificada para apoyar la relajación o motivación de pacientes.</p> <p>- Productores de plataforma: organizan grandes catálogos sin clasificar manualmente canción por canción.</p> <p>Problemas que se resuelven:</p> <p>- Clasificación manual de la música</p> <p>- Recomendaciones poco personalizadas</p> <p>- Falta de apoyo automatizado en terapia musical.</p> <p>- Dificultad para curar grandes bibliotecas musicales.</p> <p>Integración flujo de trabajo y interfaces:</p> <p>- Apps de streaming: el modelo clasifica cada canción y ajusta automáticamente la playlist.</p> <p>- Catálogos musicales: al subir la canción, el sistema le asigna una emoción.</p> <p>Interfaces:</p> <p>- Aplicaciones móviles/web de música</p> <p>- Dashboards de terapeutas</p> <p>- Paneles de gestión de catálogos musicales.</p>	<div><p>⬇</p></div> <p>- Se hace uso del dataset de Turkish Music Emotion, con más de 400 fragmentos musicales de 30 segundos ya etiquetados en cuatro emociones (<i>relax, happy, sad, angry</i>).</p> <p>- Los atributos acústicos ya vienen extraídos, por lo que no es necesario un etiquetado manual inicial.</p> <p>Estrategias:</p> <p>- Almacenar y controlar cambios de datasets sin duplicar todo el histórico.</p> <p>- Recalcular solo los nuevos datos en lugar de reprocesar todo el dataset.</p> <p>- Usar subconjuntos representativos para pruebas rápidas, reduciendo cómputo.</p>	<div><p>🗄</p></div> <p>- Los Intires y los outcomes provienen principalmente del dataset.</p> <p>- Se usan dos tablas internas (CSV limpio y CSV modificado) para distintas fases.</p> <p>- Futuras integraciones puedan enriquecerse con APIs para ampliar cobertura y mantener fresca.</p>

<div><div>IMPACT SIMULATION</div><div></div></div> <div><ul style="list-style-type: none">- Los costos/ganancias dependen de la precisión de las predicciones (personalización, confianza, utilidad terapéutica).- Se usa el dataset original y modificado para simular impacto antes del despliegue.- El despliegue se aprueba si cumple criterios de performance, robustez, eficiencia y trazabilidad.- Hay restricciones de equidad, buscando un rendimiento equilibrado entre todas las emociones.</div>	<div><div>MAKING PREDICTIONS</div><div></div></div> <div><ul style="list-style-type: none">- Las predicciones se pueden realizar en batch (por lotes), cuando se desea clasificar un conjunto grande de canciones, como un catálogo completo. Este proceso no es inmediato y puede tardar desde varios minutos hasta algunas horas, dependiendo de la cantidad de datos.- También pueden hacerse en tiempo real (real time), cuando el usuario sube o reproduce una canción. En este caso se busca rapidez, con un tiempo estimado máximo de 1 minuto, considerando lectura, extracción de características y predicción.- Recursos: el uso de CPUs es suficiente para el despliegue de predicciones. Durante el entrenamiento, si se utilizan modelos más complejos, se pueden aprovechar GPUs para acelerar el proceso.</div>	<div><div>BUILDING MODELS</div><div></div></div> <div><ul style="list-style-type: none">- Se trabaja con un modelo principal de clasificación multiclase, encargado de predecir la emoción de cada fragmento musical (relax, happy, sad, angry).- La actualización del modelo puede hacerse de forma periódica (cada 1 a 3 meses) o cuando las métricas de rendimiento (accuracy, F1-score) comiencen a disminuir.- Tiempos de procesamiento: En batch, pueden destinarse varias horas, ya que no se requiere inmediatez. En tiempo real, el rango esperado es de máximo 1 minuto, considerando preprocesamiento y predicción.- Recursos computacionales: CPUs para inferencia (predicciones). GPUs opcionales durante el entrenamiento para acelerar cálculos en modelos más complejos.</div>	<div><div>FEATURES</div><div></div></div> <div><ul style="list-style-type: none">- En predicción, cada canción se representa como un vector numérico de 50 características.- Los datos crudos pasan por limpieza y normalización.- Opcionalmente, se aplican reducciones de dimensionalidad o nuevas features de APIs externas.</div>
	<div><div>MONITORING</div><div></div></div> <div><ul style="list-style-type: none">-Se monitorean métricas técnicas (accuracy, F1, latencia), métricas de usuario (engagement, satisfacción, aceptación de recomendaciones) y métricas de negocio (costos ahorrados, retención, adopción).-Se revisan en diferentes frecuencias: semanal (técnico), mensual (usuarios) y trimestral (negocio).</div>		

Propuesta de valor

La solución de ML supera a los métodos manuales al capturar patrones sutiles en las 50 features acústicas.

Base de Referencia Crucial: El establecimiento de este baseline con ML tradicional (Random Forest, SVM) proporciona un punto de referencia crucial.

Eficiencia de Features Clásicas: Demuestra la eficiencia de las features clásicas antes de migrar a modelos más complejos de Deep Learning (como los basados en Chroma Spectrograms mencionados en la literatura).

Manipulación y preparación de datos

Enfoque Modular: El proceso fue diseñado con funciones modulares (`load_data`, `clean_data`) para asegurar la reproducibilidad y facilitar la integración en un pipeline MLOps.

Carga y Dimensiones: Dataset modificado cargado: (408 filas, 52 columnas).

Valores Nulos: Se confirmó la ausencia de valores nulos en las features numéricas.

Tratamiento de Outliers (Crítico):
Método: Rango Intercuartílico (IQR) aplicado a las 50 features.

Estrategia: Capping (acotamiento de valores) en lugar de eliminación de filas.

Justificación: Preservar el tamaño

de la muestra (408 instancias), evitando la pérdida significativa de datos que podría comprometer el entrenamiento.

Resultado: Reducción de la cuenta de outliers a cero post-procesamiento.

Codificación del Target: Aplicación de LabelEncoder a la columna categórica Class (0, 1, 2, 3).

Análisis Exploratorio de Datos (EDA) Visual:

Balance de Clases: Evaluación del desbalance de las 4 emociones (justifica el uso de F1-Score Weighted).

Correlación entre Features: Mapa de Calor para identificar multicolinealidad (especialmente entre los 12 coeficientes MFCC).

Estandarización de Datos:

Aplicación: Se aplica StandardScaler a todas las features.

Justificación: Necesario para que algoritmos sensibles a la escala (como SVM o KNN) se entrenen de manera óptima y justa, dado que las features tienen diferentes rangos.

Learn more

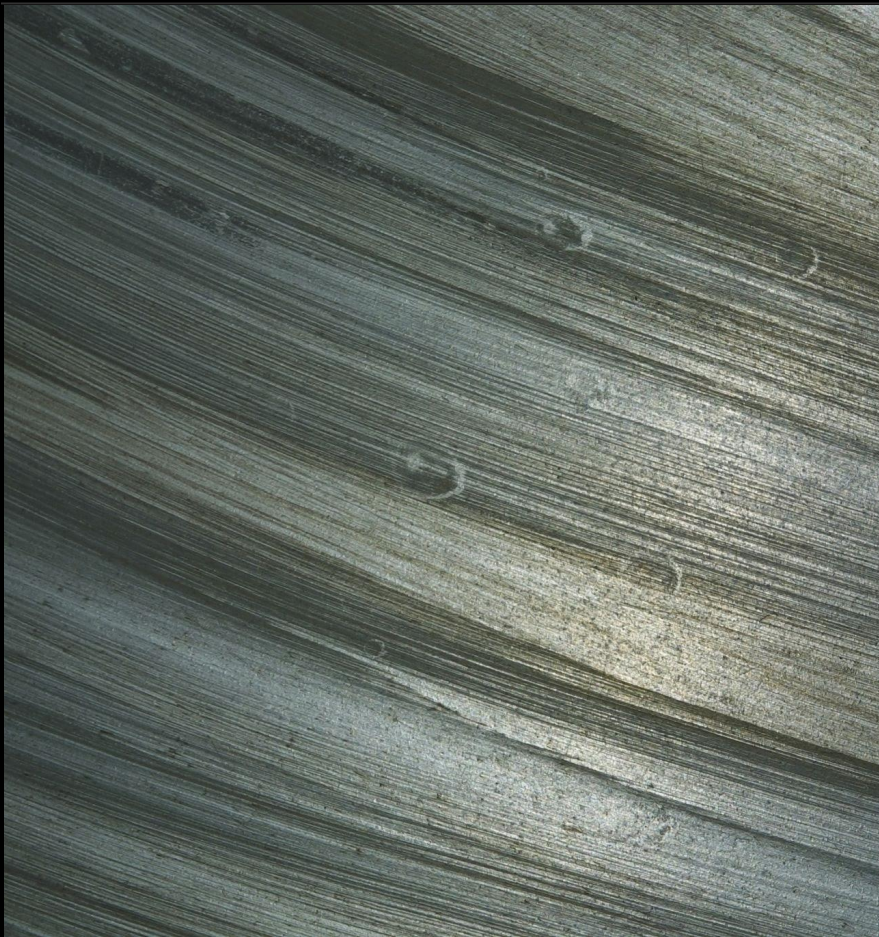


Reducción de Dimensionalidad (Evaluación con PCA):

Justificación: El uso de 50 features puede llevar a redundancia y al "curse of dimensionality".

Análisis: Evaluación de PCA mostrando el gráfico de Varianza Explicada Acumulada para determinar si un subconjunto de componentes (ej. 15-20) puede explicar >90% de la varianza.

Decisión



Versionado y Trazabilidad de Datos (DVC)

Implementación:

Se utiliza DVC para versionar y rastrear al menos dos estados críticos del dataset:

El dataset inicial modificado.

El Dataset Final Limpio y Preprocesado (post-capping, post-escalamiento).

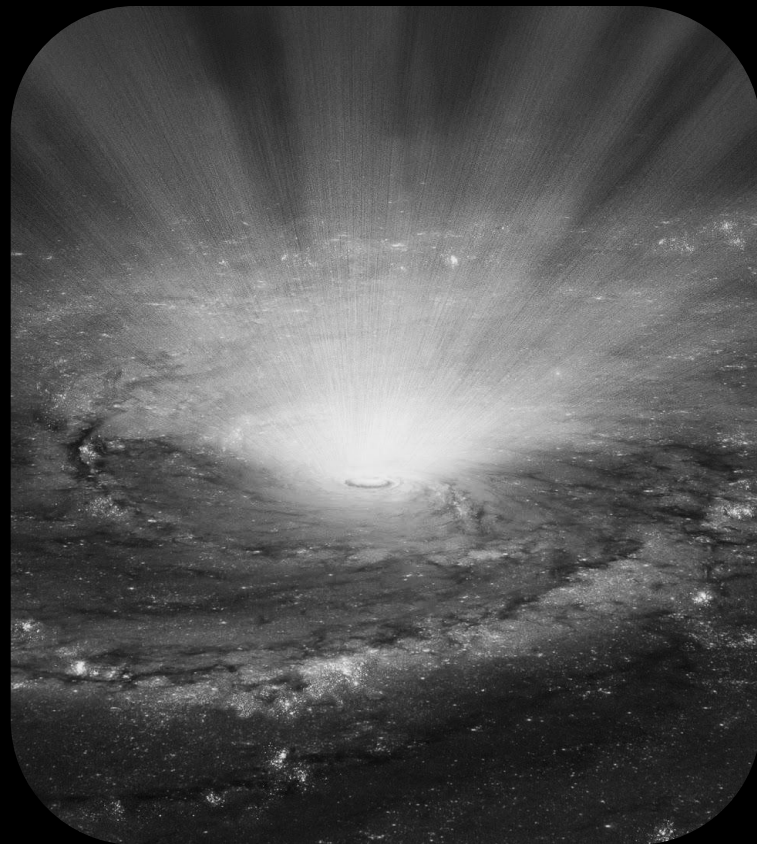
Importancia (MLOps):

La integración de DVC con GitHub asegura la reproducibilidad y trazabilidad.

Cualquier modelo entrenado estará vinculado al hash del dataset exacto que se utilizó, permitiendo replicar el experimento.

Construcción, Ajuste y Evaluación de Modelos ML

- División de Datos: Los datos preprocesados se dividen en conjuntos de Entrenamiento y Prueba (p. ej., 80% / 20%).
- Selección del Baseline:
- Se eligen algoritmos de Clasificación Multiclase como baseline inicial:
- Random Forest Classifier: Por su robustez y manejo de relaciones no lineales.
- Support Vector Machine (SVM): Por su eficacia en espacios de alta dimensión.
- Entrenamiento y Ajuste: Se documenta el uso de técnicas de ajuste de hiperparámetros (ej. Grid Search o Random Search) para optimizar el rendimiento.
- Métricas: Principal: F1-Score (Weighted). Secundaria: Accuracy.



Resultados del Baseline (Pendiente de Llenar)

Llenar datos

Gestión de Roles y MLOps

1



Data Engineer/Analyst:

Ingesta de datos, limpieza, Capping de Outliers, versión inicial del dataset con DVC.

2



Data Scientist:

EDA visual, Estandarización/PCA, selección de algoritmos, training y evaluación del baseline.

3



MLOps Engineer:

Modularidad del código, integración de DVC y GitHub, y documentación de interacciones.

4



LLM Engineer:

falta

Alineación MLOps (Evidencia):

- Modularización del código en funciones para un pipeline reproducible.
- Uso de DVC para garantizar la trazabilidad del dataset.
- Repositorio de GitHub para la colaboración activa.

Gracias