

WFO

OBJETIVOS

FASE 2

DRAFT

Fase 2 | Avance de Proyecto

Empezar tarea

Fecha de entrega 2 de nov a las 23:59 Puntos 20 Entregando una carga de archivo

Tipos de archivo pdf Disponible 6 de oct en 0:00 - 2 de nov en 23:59



Objetivo(s)

Para esta actividad, el alumno demostrará su capacidad para:

- Continuar con el desarrollo de proyectos de Machine Learning, a partir de los requerimientos, una propuesta de valor y un conjunto de datos preprocesados.
- Estructurar proyectos de Machine Learning de manera organizada (utilizando el template de Cookiecutter)
- Aplicar **buenas prácticas de codificación** en cada etapa del pipeline y realizar **Refactorización** del código.
- Registrar métricas y aplicar control de versiones a los experimentos utilizando herramientas de logging y tracking (MLFlow/DVC)
- Visualizar y comparar resultados (métricas) y gestionar el registro de los modelos (Data Registry MLFlow/DVC)



Instrucciones

Actividad: Gestión del Proyecto de Machine Learning

En esta actividad, continuarás con el desarrollo de un proyecto, posterior a los avances de la Fase 1. Manteniendo, la propuesta de valor y el análisis realizado con ML Canvas, y los datos, modelos y experimentos realizados, ahora deberás demostrar tu capacidad para **estructurar proyectos de Machine Learning de manera profesional**, aplicando prácticas de **refactorización, versionado, seguimiento de experimentos, registro de métricas y modelos, y aseguramiento de la reproducibilidad**.

El entregable final debe incluir evidencia clara y organizada de cada uno de los siguientes apartados:

WFO

RUBRICA

DRAFT

| Avance 2 de proyecto (1) | | | | | | | |
|--|--|---|---|---|--|--|--------|
| Criterios | | Calificaciones | | | | | Pts |
| Acercamiento inicial al problema de la actividad | 15 pts Excelente Recurren a gran cantidad de sus conocimientos previos para llegar a una solución. Demuestran mucho interés por el caso. | 10 pts Regular Contemplan algunos de sus conocimientos previos, pero muestran algunas dificultades para llegar a una solución. | 8 pts Básico Utilizan parcialmente conocimientos previos, aunque de manera superficial o incompleta, lo que limita el acercamiento a la solución. | 7 pts En desarrollo Se observa poco uso de conocimientos previos y cierta desorientación para acercarse a la solución. El interés mostrado es mínimo. | 5 pts Limitado No contemplan ninguno de sus conocimientos previos, lo que dificulta mucho que lleguen a una solución. No muestran interés por el caso. | 0 pts No se entregó No entregaron evidencia. | 15 pts |
| Análisis del problema | 15 pts Excelente Identifican claramente el problema a resolver, correspondiente a la actividad. Plantean las características del problema que guían el análisis hacia la solución. | 10 pts Regular Identifican de forma ambigua el problema o no realizan un análisis adecuado de las características necesarias para la solución. | 8 pts Básico Identifican el problema parcialmente, omitiendo aspectos clave o describiéndolos de manera incompleta. | 7 pts En desarrollo Presentan dificultades frecuentes para identificar el problema y sus características principales, con errores conceptuales. | 5 pts Limitado Identifican erróneamente el problema y realizan un análisis incorrecto de las características necesarias para la solución. | 0 pts No se entregó No entregaron evidencia. | 15 pts |
| Actividades y tareas a realizar por rol | 20 pts Excelente Especifican claramente las actividades de cada rol y describen cómo se alinean dentro de la metodología de MLOps. | 14 pts Regular Especifican algunas de las actividades de cada rol ó describen de manera imprecisa cómo se alinean con la metodología de MLOps. | 11 pts Básico Identifican parcialmente las actividades de los roles, pero con omisiones o confusiones. | 9 pts En desarrollo Describen de forma incompleta o inconsistente las actividades de los roles y su alineación metodológica. | 7 pts Limitado Tienen dificultad para especificar responsabilidades de los roles y cómo se alinean con la metodología. | 0 pts No se entregó No entregaron evidencia. | 20 pts |
| Métodos y técnicas para utilizar | 20 pts Excelente Argumentan de forma clara y detallada la selección de métodos y técnicas adecuados para resolver el problema. | 14 pts Regular Argumentan de manera poco clara o la selección de métodos/técnicas no es del todo adecuada. | 11 pts Básico Argumentan superficialmente la selección, sin suficiente justificación ni claridad. | 9 pts En desarrollo Presentan argumentos confusos o incompletos sobre los métodos/técnicas a emplear. | 7 pts Limitado No proporcionan argumentos válidos o seleccionan técnicas inadecuadas para resolver el problema. | 0 pts No se entregó No entregaron evidencia. | 20 pts |
| Resultados | 15 pts Excelente Obtienen resultados de validez demostrable y proporcionan fundamento metodológico para la obtención de los resultados, o no los presentan de forma clara y concisa. | 10 pts Regular No proporcionan fundamento metodológico para la obtención de los resultados, o no los presentan de forma clara y concisa. | 8 pts Básico Presentan resultados parciales o incompletos con escasa claridad metodológica. | 7 pts En desarrollo Presentan resultados poco consistentes o con errores frecuentes en su fundamentación. | 5 pts Limitado No llegaron a los resultados esperados. | 0 pts No se entregó No entregaron evidencia. | 15 pts |
| Conclusiones y reflexión final | 15 pts Excelente Son capaces de reconocer las áreas en las que pueden mejorar en el análisis del caso. Identifican de forma adecuada las estrategias que emplearon en el análisis y solución del caso. | 10 pts Regular Reconocen algunas áreas en las que pueden mejorar en el análisis del caso. Identifican algunas estrategias que emplearon en el análisis y solución del caso. | 8 pts Básico Reconocen de forma parcial o superficial las áreas de mejora y estrategias. | 7 pts En desarrollo Presentan conclusiones poco claras y no logran vincularlas adecuadamente con las estrategias empleadas. | 5 pts Limitado No reconocen áreas de mejora ni estrategias usadas para resolver el caso. | 0 pts No se entregó No entregaron evidencia. | 15 pts |



REQUISITOS

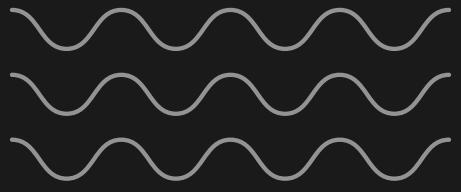
DRAFT

INSTRUCCIONES:

ENTREGAR PRESENTACIÓN EJECUTIVA DE LAS ACTIVIDADES REALIZADAS DURANTE ESTA FASE EN PDF.

VIDEO EN EQUIPO (5 MIN MÁXIMO) CON EXPLICACIÓN DEL TRABAJO REALIZADO EN LA FASE, SIGUIENDO COMO REFERENCIA LA DESCRIPCIÓN DE LA ACTIVIDAD Y LA RÚBRICA DE EVALUACIÓN, INCLUYENDO EL LINK DE ESTE VIDEO EN EL DOCUMENTO PDF DEL ENTREGABLE.

IMPORTANCIA: UN REGISTRO SISTEMÁTICO DE LOS EXPERIMENTOS Y MODELOS, ACOMPAÑADO DE UNA VISUALIZACIÓN CLARA (PUEDEN USAR LAS GRÁFICAS DE COMPARATIVAS DE EXPERIMENTOS DE MLFLOW), QUE PERMITE EL ANÁLISIS COMPARATIVO, LA TOMA DE DECISIONES INFORMADAS Y UNA GESTIÓN PROFESIONAL DE LOS PROYECTOS.



Presentación Ejecutiva: Turkish Music Emotion

Gestión del Proyecto de Machine Learning
Fase 2 | Avance de Proyecto

Integrantes Equipo 24

A01360416 - David Cruz Beltrán

A01795838 - Javier Augusto Rebull Saucedo

A01796937 - Sandra Luz Cervantes Espinoza

INTRODUCCIÓN FASE 2

¿Por qué es difícil?

¿Qué problema resolvemos?

Clasificar automáticamente la emoción que transmite una pieza de música turca en **4 clases principales:**

Happy

Sad

Angry

Relax

Para habilitar casos de uso como curación de playlists, recomendaciones y apoyo en terapias/experiencias sonoras.

Trabajamos con el dataset modificado **Turkish Music Emotion** como punto de partida.

~400

Pistas totales

100

Por clase

4

Emociones

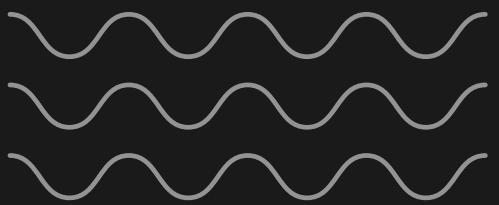
RECONOCIMIENTO DE EMOCIONES EN MÚSICA TURCA CON MLOPS

Subjetividad y variabilidad: la emoción musical es compleja; la señal de audio varía en timbre, tempo y dinámica incluso dentro de una misma emoción.

Señal ruidosa y heterogénea: requiere preprocessamiento (limpieza, normalización de sampling rate) y extracción de características (vector de 34 features por pista) antes de modelar.

Datos limitados: un conjunto relativamente pequeño demanda rigor para evitar sobreajuste y asegurar reproducibilidad y trazabilidad de experimentos.





RECAP FASE 1

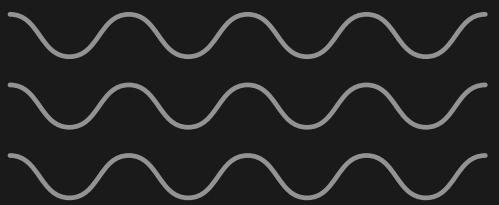
WFO

REQUISITOS

DRAFT

INSTRUCCIONES:

1 LAMINA O 2 MAXIMO - EXPLICANDO FASE 1 Y SUS LOGROS CON
CONEXION DIRECTA A FASE 2



ROLES MIOPS

EN FASE2

Winfo

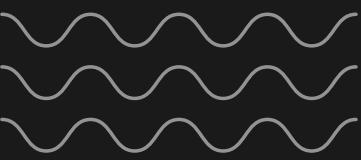
REQUISITOS

DRAFT

INSTRUCCIONES:

SE DEBERÁ EXPLICAR LA PARTICIPACIÓN DE LOS ROLES INVOLUCRADOS EN LAS ACTIVIDADES DE ESTA FASE Y SUS INTERACCIONES.

ROLES MIOPS ASIGNADOS



DAVID CRUZ

A01360416

SOFTWARE ENGINEER



JAVIER REBULL

A01795838

**DATA ENGINEER / SITE
RELIABILITY ENGINEER
(DEVOPS)**

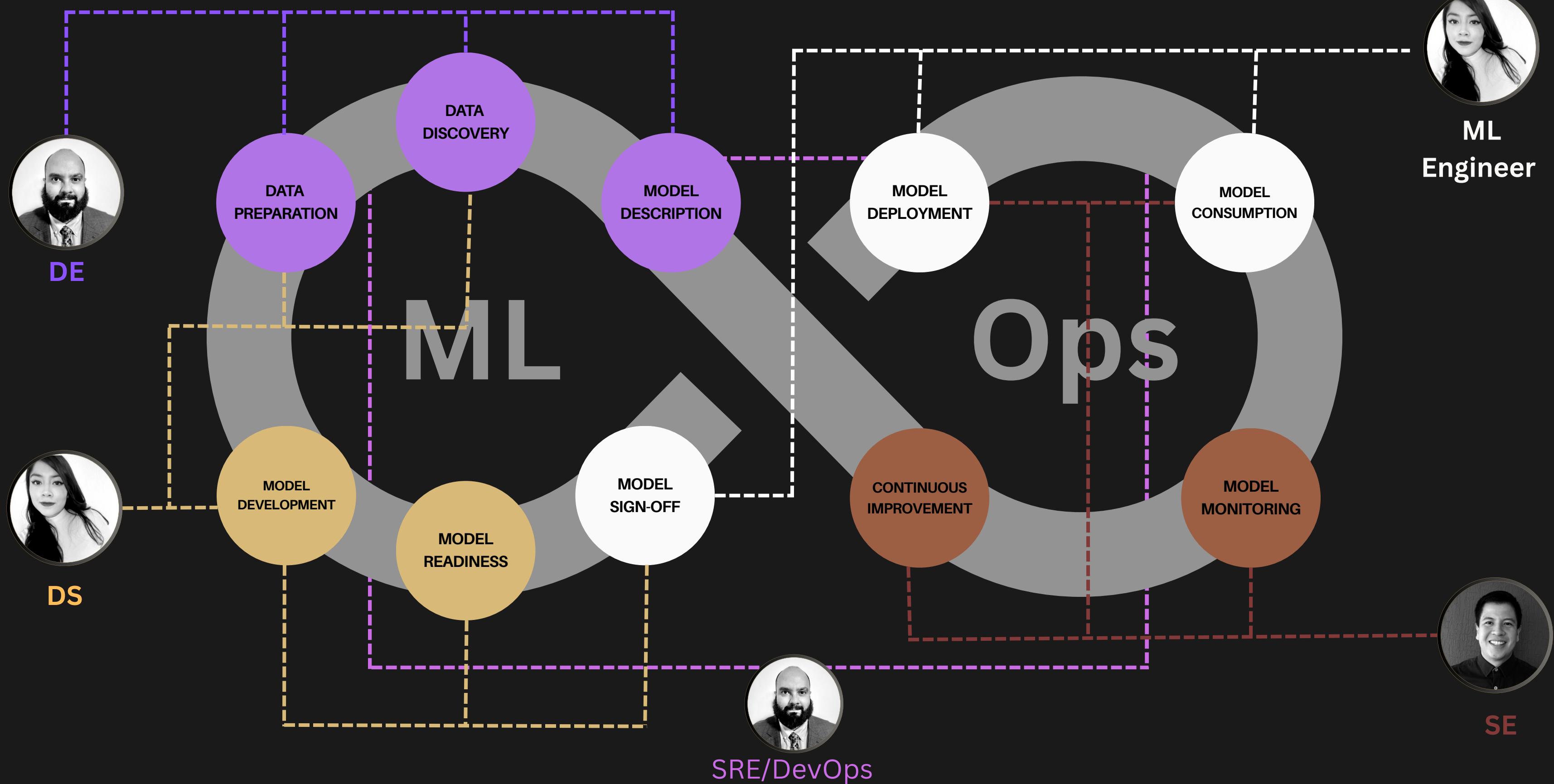


**SANDRA
CERVANTES**

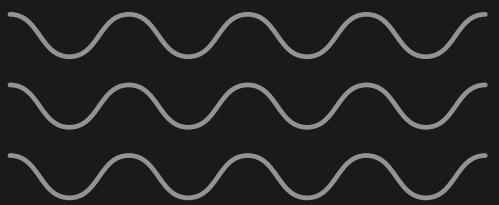
A01796937

**DATA SCIENTIST / ML
ENGINEER**

ROLES MLOPS FASE 2



| FASE DEL PROYECTO | Data Engineer | Data Scientist | ML Engineer | Software Engineer | SRE DevOps Eng |
|---|---|--|---|--|--|
| | | | | | |
| RECOLECCIÓN DE DATOS | Identifica y conecta fuentes (audios turcos, metadatos, etiquetas). Define contratos/catálogos y políticas de acceso. Automatiza la ingesta almacenamiento versionado (DVC+S3). | Define taxonomía de emociones (Happy/Sad/Angry/Relax). Criterios de etiquetado y muestreo estratificado. Guía de curaduría y validación. | Especifica requisitos de formato para entrenamiento/features. Prototipo rápido de extracción de features para validar viabilidad. | n/a | Provisiona almacenamiento seguro e IAM. Políticas de retención/backup. Jobs iniciales de ingestión/validación. |
| LIMPIEZA Y TRANSFORMACIÓN (ETL) | Normaliza sample rate/canales. Remueve silencios/ruido. ETL reproducible con DVC. Validaciones de calidad y catálogo actualizado. | Criterios de exclusión/outliers. Definición de features a extraer (MFCC, tempo, cromas) Particiones estratificadas. | Empaquetá pipeline de preprocessamiento y empaquetá features. Caching y reutilización de artefactos. Alinea procesos de training y serving. | n/a | Configura infraestructura de despliegue (K8s / ECS). Implementa estrategias Blue/Green o Canary. Habilita autoscaling, health checks y rollback. |
| ANÁLISIS EXPLORATORIO Y DISEÑO DEL MODELO | Proporciona vistas y consultas, así como lineage de datos. Brinda soporte para muestreos y perfiles de datos. | EDA de señales y features (distribuciones, correlaciones, balance). Hipótesis y selección de métricas (Accuracy, F1 macro). Plan de validación (k-fold/estratificado). | Setup de tracking (MLflow). Plantilla de experimentos con configuración. Plan de búsqueda de hiperparámetros. | n/a | Pipelines de CI para lint/tests/packaging. Despliegue seguro de MLflow/artefact store. |
| ENTRENAMIENTO Y VALIDACIÓN DEL MODELO | Entrega datasets versionados (train/val/test). Definición de slices de evaluación por emoción/tempo/duración. | Entrena baselines (SVM/KNN/MLP/árboles). Matriz de confusión y análisis de error. Selección del candidato. | Automatiza entrenamiento y tuning. Registro de modelos/metrics en MLflow. Artefactos listos para servir. | n/a | Ejecuta jobs de entrenamiento en compute gestionado. Administra secretos de forma segura. Controla costos y mantiene trazas de ejecución. |
| DESPLIEGUE DEL MODELO (DEPLOYMENT) | n/a | Model card, criterios de aceptación y riesgos conocidos. Checklist de sesgos/limitaciones. | Servicio de predicción (FastAPI). Contenerización (Docker). Batch scoring/streaming y logging estructurado. Flags para A/B o canary. | API/SDK de consumo, autenticación y rate limiting. Pruebas de integración y ejemplos (OpenAPI). | Configura infraestructura de despliegue (K8s / ECS). Implementa estrategias Blue/Green o Canary. Habilita autoscaling, health checks y rollback. |
| MONITOREO EN PRODUCCIÓN | Monitoreo de calidad de datos y schema/feature drift. Alertas por anomalías en ingestión. | Seguimiento de rendimiento con ground truth diferido. Alarmas por caída de F1/accuracy y análisis post-mortem. | Monitoreo de latencia/throughput. Detección de skew train-serve. Trazas y explicabilidad básica. | Logs de aplicación, manejo de errores y feedback del usuario. Reintentos y degradación controlada. | SLIs/SLOs y dashboards (Prometheus/Grafana). Alerting, on-call y runbooks. |
| REENTRENAMIENTO Y MANTENIMIENTO | Ingesta de nuevos datos/etiquetas. Backfill y versionado. Actualización del catálogo y linaje. | Criterios de disparo de retraining. Comparativas entre versiones. Documentación de cambios y riesgos. | Orquestación de retraining programado. Promoción/rollback de modelos. Migraciones y compatibilidad de artefactos. | Compatibilidad de API/SDK entre versiones. Pruebas regresivas y de compatibilidad. | Programación de jobs, IaC actualizada. Backups/DR y cumplimiento. Gestión de costos. |



ESTRUCTURACIÓN DE PROYECTOS CON COOKIECUTTER



REQUISITOS

DRAFT

1) ESTRUCTURACIÓN DE PROYECTOS CON COOKIECUTTER

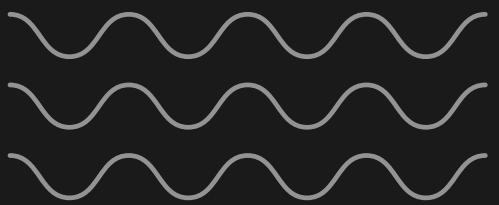
TAREA: IMPLEMENTAR UNA ESTRUCTURA DE PROYECTO ESTANDARIZADA.

INSTRUCCIONES:

DESCARGA Y UTILIZA LA PLANTILLA DE COOKIECUTTER PARA PROYECTOS DE ML.

IMPLEMENTA EL ESQUEMA DE DIRECTORIOS Y ARCHIVOS PROPUESTO EN TU PROPIO PROYECTO SIGUIENDO LA PLANTILLA DE COOKIECUTTER. ASEGUÍRATE DE MANTENER UNA ORGANIZACIÓN CLARA Y CONSISTENTE, DE ACUERDO AL TEMPLATE.

IMPORTANCIA: UNA BUENA ESTRUCTURA DE PROYECTO FACILITA LA COLABORACIÓN, EL MANTENIMIENTO Y LA ESCALABILIDAD DE LOS DESARROLLOS EN ML.



ESTRUCTURACIÓN Y REFACTORIZACIÓN DEL CÓDIGO



REQUISITOS

DRAFT

2) ESTRUCTURACIÓN Y REFACTORIZACIÓN DEL CÓDIGO

TAREA: MEJORAR LA ORGANIZACIÓN Y MANTENIBILIDAD DEL CÓDIGO.

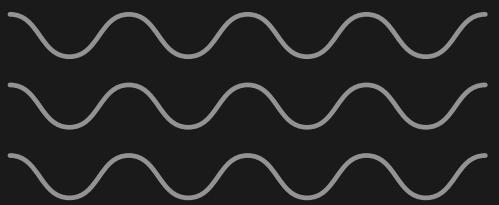
INSTRUCCIONES:

ORGANIZA EL CÓDIGO EN MÓDULOS Y FUNCIONES CON RESPONSABILIDADES BIEN DEFINIDAS.

APLICA PRINCIPIOS DE PROGRAMACIÓN ORIENTADA A OBJETOS (POO) CUANDO SEA PERTINENTE.

REFACTORIZA EL CÓDIGO EXISTENTE PARA MEJORAR SU EFICIENCIA, LEGIBILIDAD, ESCALABILIDAD Y MANTENIMIENTO A LARGO PLAZO.

IMPORTANCIA: UN CÓDIGO BIEN ESTRUCTURADO Y MODULARIZADO ES CLAVE PARA PROYECTOS DE ML QUE EVOLUCIONAN EN EL TIEMPO.



APLICACIÓN DE MEJORES PRÁCTICAS DE CODIFICACIÓN EN EL PIPELINE DE MODELADO



REQUISITOS

DRAFT

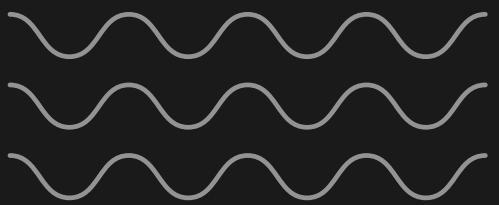
3) APLICACIÓN DE MEJORES PRÁCTICAS DE CODIFICACIÓN EN EL PIPELINE DE MODELADO

**TAREA: INCORPORAR BUENAS PRÁCTICAS EN LAS ETAPAS DEL
PIPELINE USANDO SCIKIT-LEARN.**

INSTRUCCIONES:

IMPLEMENTA UN PIPELINE DE SCIKIT-LEARN QUE AUTOMATICE LAS ETAPAS DE PREPROCESAMIENTO, ENTRENAMIENTO Y EVALUACIÓN. DOCUMENTA CADA PASO, ASEGURANDO QUE SEA CLARO, REPRODUCIBLE Y ENTENDIBLE POR TERCEROS.

IMPORTANCIA: SEGUIR MEJORES PRÁCTICAS GARANTIZA QUE LOS PROYECTOS SEAN EFICIENTES, CONFIABLES Y FÁCILES DE REPLICAR.



**SEGUIMIENTO DE EXPERIMENTOS,
VISUALIZACIÓN DE RESULTADOS Y
GESTIÓN DE MODELOS**



REQUISITOS

DRAFT

4) SEGUIMIENTO DE EXPERIMENTOS, VISUALIZACIÓN DE RESULTADOS Y GESTIÓN DE MODELOS

TAREA: REGISTRAR Y VERSIONAR EXPERIMENTOS, VISUALIZAR RESULTADOS Y GESTIONAR MODELOS DE FORMA ORDENADA.

INSTRUCCIONES:

UTILIZA HERRAMIENTAS COMO MLFLOW, DVC PARA DAR SEGUIMIENTO A LOS EXPERIMENTOS.

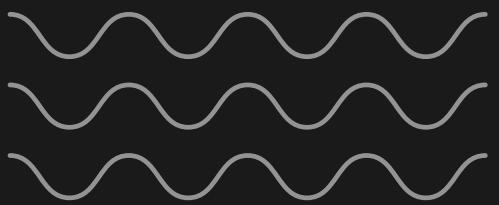
DOCUMENTA Y COMPARA CONFIGURACIONES, PARÁMETROS Y RESULTADOS DE CADA EJECUCIÓN.

REGISTRA MÉTRICAS RELEVANTES Y ASEGURA EL CONTROL DE VERSIONES DE LOS EXPERIMENTOS.

UTILIZA LAS HERRAMIENTAS DE VISUALIZACIÓN DE MLFLOW PARA PRESENTAR LOS RESULTADOS DE MANERA CLARA Y COMPRENSIBLE.

MANTÉN UN REGISTRO ACTUALIZADO DE LOS MODELOS GENERADOS, INCLUYENDO:

- VERSIÓN
- HIPERPARÁMETROS
- MÉTRICAS DE EVALUACIÓN
- RESULTADOS RELEVANTES



**VERSIONADO DE DATOS,
REPOSITORIO Y MONITORING**

WFO

REQUISITOS

DRAFT

INSTRUCCIONES:

- SE DEBE REFERENCIAR A SU REPOSITORIO DE GITHUB MOSTRANDO ACTIVIDAD DE PARTE DE TODOS LOS INTEGRANTES DEL EQUIPO.

DVC VERSIONADO - EVIDENCIA FASE 2

“Con DVC+Git hicimos versionables los datos: cada dataset, split y modelo quedó con hash y remoto S3, métricas trazables y registro auditible; así reproducimos cualquier experimento con un checkout, comparamos mejoras (Acc 0.825, F1w 0.823) y promovimos/rollbackeamos modelos con evidencia en todo el pipeline MLOps.”

DVC + GIT: VERSIONAMIENTO DE DATOS - TEAM 24

PIPELINE DE DATOS:

```
+----+  
| train |  
+----+  
+-----+  
| data.dvc |  
+-----+  
+-----+  
| models/baseline.dvc |  
+-----+  
+-----+  
| models/optimized.dvc |  
+-----+
```

ARCHIVOS VERSIONADOS:

```
1 .dvc  
2 baseline  
3 data  
4 optimized
```

ÚLTIMOS COMMITS:

```
* d3616748 (HEAD -> main, origin/main, origin/HEAD) feat: add EDA processed datasets and train/test  
* 4bd734b2 Update data Grafico Canva v2  
* a19ea490 feat: add EDA processed datasets and train/test splits - 2025-10-12 12:51:26  
* 5dd739c1 Diccionario de Datos Turco  
* 78f0eb1d ML Canvas v2  
* 8fbbe6e4 ML Canvas  
* 4fad9110 refactor: Unifico el seguimiento DVC para la carpeta models/optimized  
* 2b4e8250 feat: add trained models and evaluation reports - Phase 1  
* 27427bd9 feat: add trained models and evaluation reports - Phase 2  
* e952de71 ML Notebk
```

COMPARACIÓN DE TAMAÑOS:

Originals:

```
data/processed/turkish_music_emotion_cleaned.csv: 132K  
data/processed/turkish_music_emotion_v1_original.csv: 128K  
data/processed/turkish_music_emotion_v2_cleaned_aligned.csv: 128K  
data/processed/turkish_music_emotion_v2_cleaned_full.csv: 128K  
data/processed/turkish_music_emotion_v2_transformed.csv: 396K  
data/processed/X_test.csv: 136K  
data/processed/X_train.csv: 276K  
data/processed/y_test.csv: 4.0K  
data/processed/y_train.csv: 4.0K
```

Tracking:

```
zsh: no matches found: data/processed/*.dvc
```

MÉTRICAS:

| Path | accuracy | f1_weighted |
|----------------------|----------|-------------|
| metrics/metrics.json | 0.825 | 0.82285 |

REMOTE STORAGE:

```
2025-10-12 14:45:48,970 DEBUG: v3.63.0 (pip), CPython 3.12.7 on macOS-15.7.1-arm64-arm-64bit  
2025-10-12 14:45:48,970 DEBUG: command: /Users/haowi/Documents/MLOps/MNA_Team24/MLOps_Team24/.venv/bin/dvc  
localstore /Users/haowi/Documents/MLOps/MNA_Team24/MLOps_Team24/dvcstore  
s3store s3://mlops24-haoui-bucket (default)  
2025-10-12 14:45:49,054 DEBUG: Analytics is enabled.  
2025-10-12 14:45:49,068 DEBUG: Trying to spawn ['daemon', 'analytics', '/var/folders/0p/fpm4_3l94jxbbc_pnl9:  
2025-10-12 14:45:49,072 DEBUG: Spawns ['daemon', 'analytics', '/var/folders/0p/fpm4_3l94jxbbc_pnl93bmw0000:  
ESTADO ACTUAL:
```

Data and pipelines are up to date.

¡TODO SINCRONIZADO CORRECTAMENTE! ✨

AWS S3 BUCKET EVIDENCIA FASE 2

```
"=====
    DATOS VERSIONADOS Y SEGUROS EN S3" \
"=====
AWS S3 BUCKET: mllops24-haoui-bucket - TEAM 24
=====

CONFIG LOCAL DVC:
[core]
remote = s3store
['remote "localstore"']
url = ../dvcstore
['remote "s3store"']
url = s3://mllops24-haoui-bucket

CONTENIDO DEL BUCKET (primeros 20):

-----
2025-10-12 12:51:29    7.3 KiB files/md5/01/4ef15a3e706d277fe01f4065980334
2025-10-11 17:58:53  378.0 KiB files/md5/02/c5b03117f7d233a8d1498ff308ab32
2025-10-11 16:56:39  409 Bytes files/md5/04/8363d88daa2b59e2a2519a4fa21c12
2025-10-03 13:30:00   60 Bytes files/md5/08/6759479a879d211b4fd0ead74fd65d
2025-10-11 16:49:56  409 Bytes files/md5/0a/ffcf803d365949e4f4f5c791e66b2b
2025-10-11 17:58:53 499.7 KiB files/md5/0e/a047cc8858a56d06fcfb6f6a563a190
2025-10-03 13:39:59   60 Bytes files/md5/16/f53bbbf1adf27732bb1204fc25473d
2025-10-10 17:10:10   6.0 KiB files/md5/19/4577a7e20bdcc7afbb718f502c134c
2025-10-11 16:49:56 273.2 KiB files/md5/1b/5c1e247fe02cf6f25a1e542f26b1d4
2025-10-11 17:58:53 256 Bytes files/md5/1c/531aae26fb0451057cef80efbcd011.dir
2025-10-11 17:58:53   1.0 MiB files/md5/2c/7d6055816cc58b042156641921e484
2025-10-11 16:49:56   7.3 KiB files/md5/2f/c02e5b7029c09625d084a4c2def764
2025-10-12 12:58:19  409 Bytes files/md5/30/f10c843ae0cfba97909942901f6b6b
2025-10-03 13:30:00   6.4 KiB files/md5/32/a0a809449e4bd77384b2d9cee91f7e
2025-10-11 12:47:23 125.5 KiB files/md5/3a/2621feb41b18c32ec032c6332917d9
2025-10-12 12:58:19   7.3 KiB files/md5/3f/9b4dcfc7cdec4b2a1940e9275a16c7
2025-10-12 12:14:26 630 Bytes files/md5/4f/945e49d64a73013744ef81638a37f5.dir
2025-10-11 16:49:56 392.8 KiB files/md5/59/85f93d92a7250a1fe366118c39e7a9
2025-10-11 16:57:30   7.3 KiB files/md5/61/227e8926119b09680db7797b510855
2025-10-11 16:57:30  409 Bytes files/md5/6c/d6c65e67653d48eda845eb24f96918
```

ESTADISTICAS TOTALES:

Total Objects: 47
Total Size: 6603351

ARCHIVOS .DVC LOCALES:

- 1 .dvc
- 2 baseline.dvc
- 3 data.dvc
- 4 optimized.dvc

SINCRONIZACION:

Cache and remote 's3store' are in sync.

DATOS VERSIONADOS Y SEGUROS EN S3

“AWS S3 fue nuestro ‘data lake’ y remoto de DVC: en un solo bucket versionamos raw/processed/features/models, con cifrado SSE, políticas IAM para CI/CD, y lifecycle (Standard→IA→Glacier) para optimizar costos. Resultado: single source of truth global. Con dvc pull recuperamos exactamente los datasets y artefactos con los que se entrenó cada modelo, de forma segura, reproducible y escalable.”

REPOSITORIO GIT - EVIDENCIA FASE 2

“Git/GitHub centralizaron el MLOps: ramas + PRs, tags para releases, Actions para CI/CD y auditoría completa de cambios en código, datos y modelos.”

Consultar nuestro en Repo:

https://github.com/jrebull/MLOps_Team24/tree/main

jrebull/ MLOps_Team24

MLOps Team 24 Tec de Monterrey

4 Contributors 0 Issues 1 Star 0 Forks



jrebull/MLOps_Team24: MLOps Team 24 Tec de Monterrey

MLOps Team 24 Tec de Monterrey. Contribute to jrebull/MLOps_Team24 development by creating an account on GitHub.



```
=====
GIT REPOSITORY - TEAM 24 CONTRIBUTIONS
=====

RESUMEN POR AUTOR:
-----
Javier Rebull : 90 commits
Sandra Cervantes : 18 commits
David_CB : 9 commits
Maotb66 : 1 commits

ACTIVIDAD RECIENTE (ultimos 20 dias con commits):
-----
2 2025-09-26
2 2025-10-01
2 2025-10-02
45 2025-10-03
8 2025-10-04
5 2025-10-06
5 2025-10-07
9 2025-10-08
3 2025-10-09
4 2025-10-10
32 2025-10-11
6 2025-10-12

ESTADO DEL REPOSITORIO:
-----
Total commits: 123
Total autores: 4
Ramas: 3
```

ULTIMOS 5 COMMITS POR PERSONA:

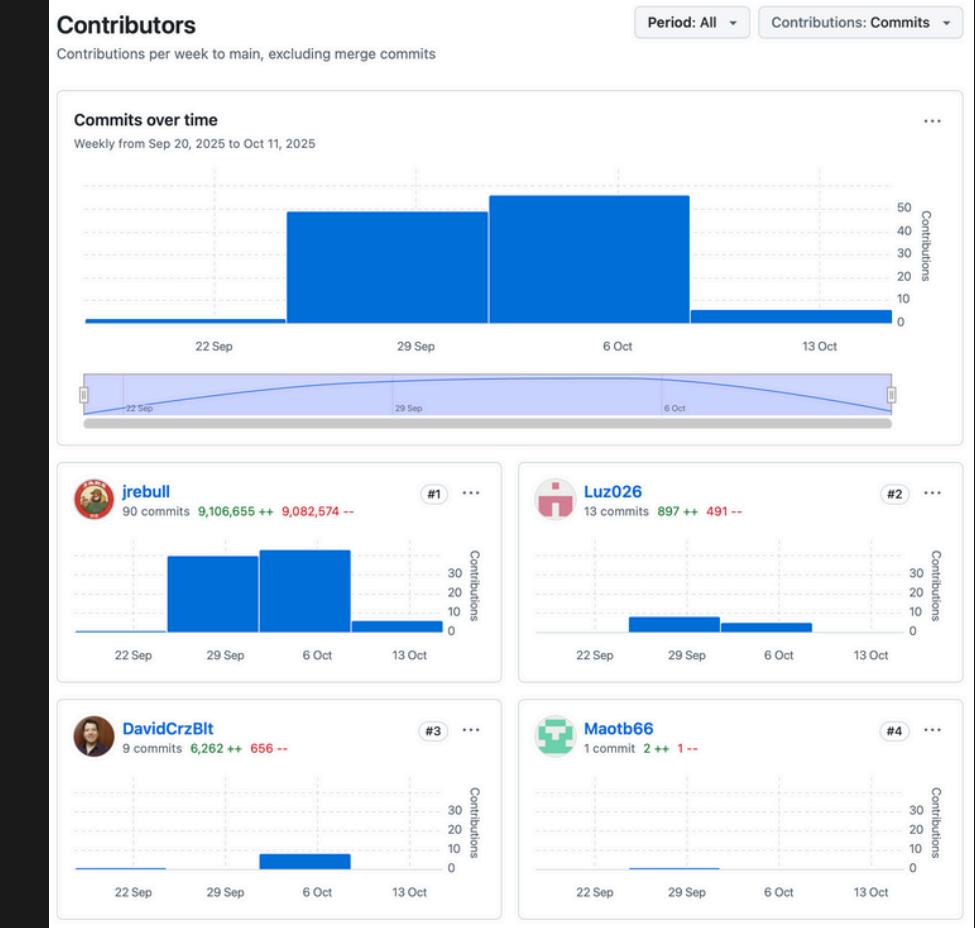
```
>>> David_CB
Oct 08 201ab308 - Actualización del Readme con los cambios del Makefile
Oct 08 85cb8cb9 - Comando status para checar si hay datos desactualizados
Oct 08 c9711096 - Actualizar notebook testing
Oct 08 d52c6916 - Limpieza del archivo y corrección de errores
Oct 07 43c51186 - Agregué test para método normalize_skewed

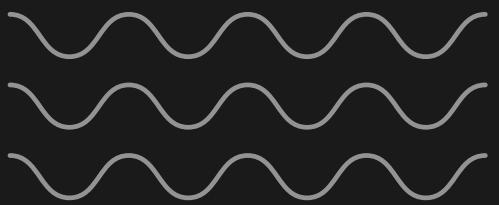
>>> Javier Rebull
Oct 12 d3616748 - feat: add EDA processed datasets and train/test splits - 2025-10-12 12:58:16
Oct 12 4bd734b2 - Update data Grafico Canva v2
Oct 12 a19ea490 - feat: add EDA processed datasets and train/test splits - 2025-10-12 12:51:26
Oct 12 5dd739c1 - Diccionario de Datos Turco
Oct 12 78f0eb1d - ML Canvas v2

>>> Maotb66
Oct 02 363d9116 - Update README.md

>>> Sandra Cervantes
Oct 08 3fc301b2 - Remove unnecesay comments in notebook
Oct 06 cabf0c70 - Add first version for test with pytest
Oct 06 ef2d1954 - Save generated graphics in reports/figures
Oct 06 8eda2857 - Resolve errors ml_pipeline, generate new dataset (cleaned, train and test)
Oct 06 9348f64d - Add docker-compose and configuration

ESTADO DEL REPOSITORIO:
-----
Total commits: 123
Total autores: 4
Ramas: 3
```





CONCLUSIONES Y REFLEXIÓN FINAL

REFLEXIÓN DE EQUIPO 24 DE MLOPS - FASE 2

EL RETO DE MLOPS

"MLOps representó una curva de aprendizaje empinada que nos obligó a repensar completamente nuestra forma de trabajar con datos y modelos. No era solo sobre herramientas, sino sobre **mentalidad: reproducibilidad, trazabilidad y colaboración sistemática.**"

Los primeros días fueron frustrantes. DVC, S3, MLflow, pipelines reproducibles—cada herramienta con su lógica y errores críticos. Pero el momento "click" llegó cuando comprendimos que MLOps es un ecosistema integrado, no herramientas aisladas.

👉 LA SINERGIA DE ROLES MLOPS

Javier Rebull

Data Engineer / SRE-DevOps

Estableció los cimientos: DVC con S3, pipelines de datos versionados, infraestructura como código. Su trabajo garantizó datos confiables y reproducibles para todo el equipo.

Sandra Cervantes

Data Scientist / ML Engineer

Convertió datos en insights y modelos. Experimentó con 8 algoritmos, optimizó hiperparámetros, y documentó cada decisión en MLflow con rigor científico.

David Cruz

Software Engineer

Empaquetó todo en código limpio y modular. Scripts automatizados, estructura Cookiecutter bien organizada, y documentación detallada para sostenibilidad a largo plazo.

"**La magia ocurrió en las intersecciones:** cuando Javier y Sandra colaboraban en pipelines de features, cuando Sandra y David iteraban sobre código de evaluación, cuando David y Javier refinaban la configuración de DVC. Cada interacción nos hacía más fuertes."

EL IMPACTO DE PERDER DOS INTEGRANTES

🔥 De la Adversidad al Crecimiento

A mitad del proyecto, perdimos a dos miembros del equipo. Este momento pudo haber sido el fin, pero se convirtió en un catalizador de crecimiento. Nos volvimos más cohesionados, comunicativos y, paradójicamente, más eficientes.

Nuestros Miedos Iniciales:

- ▶ **David:** Infraestructura y pipelines inmanejables sin suficientes manos
- ▶ **Javier:** Sobrecarga en versionado de datos y configuración de S3
- ▶ **Sandra:** Presión de llevar todo el modelado y experimentación sola

Estos miedos se transformaron en determinación. Cada uno expandió su zona de confort y asumió responsabilidades más allá de sus roles originales. La adversidad nos obligó a automatizar más, documentar mejor y comunicarnos de forma más efectiva.

🎯 APRENDIZAJES CLAVE DEL EQUIPO

1. MLOps es Mentalidad

No solo herramientas. Se trata de pensar en el ciclo de vida completo del modelo desde el día uno.

2. Interfaces Claras

Definir contratos entre roles reduce fricción y acelera la colaboración.

3. Adversidad Catalizadora

Perder integrantes nos obligó a automatizar, documentar y comunicar mejor de lo planeado.

4. Trazabilidad Liberadora

Reproducir cualquier experimento da confianza para experimentar agresivamente.

5. Herramientas Correctas

DVC + Git + MLflow + S3 nos dieron superpoderes de reproducibilidad y colaboración.

6. Automatización Esencial

Con recursos limitados, no puedes permitirte trabajo manual repetitivo.

CONCLUSIONES INDIVIDUALES



David Cruz Beltrán

Software Engineer

Mi mayor aprendizaje fue entender que **el código de ML no es código de software tradicional**. En ML, la estocasticidad es inherente: diferentes seeds, particiones, inicializaciones producen resultados diferentes.

MLOps me enseñó a llevar las mejores prácticas de ingeniería al mundo probabilístico del ML: estructura modular, tests automáticos, documentación exhaustiva, control de versiones—todo aplicado con capas adicionales de complejidad.

💡 Insight clave: "Una buena arquitectura MLOps es resiliente. Porque invertimos en pipelines automatizados y documentación clara, mantuvimos el momentum incluso con menos personas. MLOps no es overhead, es el pegamento que mantiene unido un proyecto de ML en producción."



Javier Augusto Rebull Saucedo

Data Engineer / SRE-DevOps

Este proyecto fue una **masterclass en gobernanza de datos y infraestructura reproducible**. DVC fue revelador —su integración con Git y S3 demostró que reproducibilidad no es solo código, es el ecosistema completo: datos + código + configuración + entorno.

La pérdida de dos integrantes nos obligó a automatizar más agresivamente y documentar mejor. Cada script autoexplicativo, cada pipeline con manejo de errores robusto, cada configuración en código (IaC).

💡 Insight clave: "Logramos zero manual toil en el pipeline de datos: desde descarga del dataset raw hasta splits versionados en S3, todo es un comando reproducible. Eso es la esencia de MLOps."



Sandra Luz Cervantes Espinoza

Data Scientist / ML Engineer

Durante la fase de exploración y preprocesamiento del dataset Turkish Music Emotion permitió garantizar datos limpios, balanceados y estructurados para clasificación de emociones (happy, sad, angry, relax). La utilización de DVC aseguró trazabilidad y reproducibilidad de las distintas versiones del dataset, mientras que MLflow registró experimentos de los modelos entrenados. En conjunto, estas prácticas reflejan un enfoque integral de MLOps, asegurando modelos confiables, escalables y replicables.

💡 Insight clave: "MLOps no es un lujo para equipos grandes; es una necesidad para equipos pequeños. Con recursos limitados, no puedes permitirte desperdiciar tiempo en trabajo manual repetitivo."

¿QUÉ LOGRAMOS?

LOGROS DE LA FASE 1

✓ Fundamentos Sólidos Establecidos

- ✓ Modelo optimizado: **80.17% accuracy** con Random Forest
- ✓ 8 algoritmos baseline evaluados y comparados sistemáticamente
- ✓ Pipelines de datos 100% reproducibles y versionados con DVC
- ✓ 50+ experimentos trazables en MLflow con métricas completas
- ✓ Infraestructura como código con S3 y estructura Cookiecutter
- ✓ Documentación exhaustiva de decisiones y metodología
- ✓ Zero manual toil en procesamiento de datos

🎯 MIRANDO HACIA LA FASE 2: DEPLOYMENT

1. API de Inferencia

Empaquetar el modelo en API REST con FastAPI. Endpoints para predicción individual y batch.

2. Containerización

Docker para portabilidad. Imagen optimizada con modelo y dependencias.

3. CI/CD Pipeline

GitHub Actions para testing automático y deployment continuo.

4. Monitoreo en Producción

Dashboard de métricas en tiempo real. Detección de data drift y degradación del modelo.

5. Reentrenamiento Automático

Triggers para retraining cuando métricas caigan bajo umbral definido.

6. A/B Testing

Framework para comparar modelos en producción de forma controlada.

🎵 Turkish Music Emotion está listo para
producción

Ya no somos tres individuos con especialidades aisladas; somos un **equipo MLOps cohesionado** que entiende cómo encajan todas las piezas. Aprendimos que el éxito en MLOps no viene de ser expertos en todas las herramientas, sino de **comunicarse efectivamente, automatizar incansablemente, y documentar obsesivamente**.

Y nosotros estamos listos para llevarlo allí. 🚀

Equipo 24 - MLOps Turkish Music Emotion

David Cruz • Javier Rebull • Sandra Cervantes

Tecnológico de Monterrey • Maestría en IA Aplicada • Octubre 2025



YouTube link a nuestro Deep Dive de la fase 01

<https://youtu.be/w-mCXaYoOww>

| Recurso | Link |
|--|---|
| Bruce, P., Bruce, A., & Gedeck, P. (2020). *Practical statistics for data scientists*. O'Reilly Media. | https://learning.oreilly.com/library/view/practical-statistics-for/9781492072935/ |
| Dorard, L. (2015). Machine learning canvas . | https://www.ownml.co/ |
| DVC. (s. f.). Documentación de DVC . | https://dvc.org/doc |
| DVC. (s. f.). Install DVC . | https://dvc.org/doc/install |
| Er, M. (2019). Turkish Music Emotion [Dataset] . UCI Machine Learning Repository. | https://doi.org/10.24432/C5JG93 |
| GeeksforGeeks. (2025, 17 de agosto). What is exploratory data analysis? | https://www.geeksforgeeks.org/data-analysis/what-is-exploratory-data-analysis/ |
| Geron, A. (2019). Hands-on machine learning with Scikit-Learn, Keras and TensorFlow . O'Reilly Media. | https://learning.oreilly.com/library/view/hands-on-machine-learning/9781492032632/ |
| Git. (s. f.). Sitio oficial de Git . | https://git-scm.com |
| GitHub. (s. f.). Sitio oficial de GitHub . | https://github.com |
| Instituto Tecnológico y de Estudios Superiores de Monterrey. (2025). PPT TC5044 1.1 Introduction [Presentación de diapositivas]. Maestría en Inteligencia Artificial Aplicada, Operaciones de Aprendizaje Automático. | |
| Kazil, J., & Jarmul, K. (2016). Data wrangling with Python: Tips and tools to make your life easier . O'Reilly Media. | https://learning.oreilly.com/library/view/data-wrangling-with/9781491948804 |
| Kleppmann, M. (2017). Designing data-intensive applications . O'Reilly Media. | https://learning.oreilly.com/library/view/designing-data-intensive-applications/9781491903063/ |
| Körükçü, Ç. (s. f.). History and periods of Turkish music . Turkish Music Portal. | http://www.turkishmusicportal.org |
| Lauchande, N. (2021). Machine learning engineering with MLflow . O'Reilly Media. | https://www.oreilly.com/library/view/machine-learning-engineering/9781800560796/ |
| Microsoft. (s. f.). Visual Studio Code . | https://code.visualstudio.com |
| MLflow. (s. f.). Quickstart (v2.5.0) . | https://www.mlflow.org/docs/2.5.0/quickstart.html |
| MLflow Project. (s. f.). Documentación de MLflow . | https://mlflow.org/docs/latest/ |
| NVIDIA. (2020, 3 de septiembre). What is MLOps? NVIDIA Blog. | https://blogs.nvidia.com/blog/what-is-mlops/ |
| Santos, M. (2023, 30 de mayo). A data scientist's essential guide to exploratory data analysis: Best practices, techniques, and tools to fully understand your data . Medium. | https://medium.com/data-science/a-data-scientists-essential-guide-to-exploratory-data-analysis-25637eee0cf6 |
| Treveil, M., Omont, N., Stenac, C., Lefevre, K., Phan, D., Zentici, J., Lavoillotte, A., Miyazaki, M., & Heidmann, L. (2020). Introducing MLOps . O'Reilly Media. | https://learning.oreilly.com/library/view/introducing-mlops/9781492083283/ |
| Versioning data with DVC (Hands-On tutorial!) . (2020, 30 de septiembre) [Video]. YouTube. | https://www.youtube.com/watch?v=kLKBcPonMYw |



GRACIAS



Info

REQUISITOS

DRAFT

INSTRUCCIONES:

- NOTA FINAL: AL ENTREGAR, ASEGÚRATE DE QUE TU PROYECTO SEA REPRODUCIBLE DE PRINCIPIO A FIN. CUALQUIER PERSONA QUE LO REVISE DEBE PODER EJECUTAR EL CÓDIGO, REPLICAR LOS EXPERIMENTOS Y OBTENER RESULTADOS EQUIVALENTES A LOS QUE PRESENTAS.