

1. Introducción y Contexto del Problema

Con este proyecto buscamos analizar y predecir los tiempos de ciclo y detenciones improductivas de los camiones de acarreo, con el fin de identificar oportunidades de mejora en la eficiencia operativa.

Durante la primera entrega del proyecto recibimos la recomendación de reducir la complejidad del alcance, ya que inicialmente incluíamos tres modelos de regresión y un modelo basado en coordenadas geográficas. Siguiendo esa orientación, en esta versión final decidimos concentrar los esfuerzos en dos modelos de regresión y un modelo de clasificación, excluyendo el componente geográfico. Este ajuste nos permitió simplificar la integración con el tablero de control y enfocar el análisis en las variables más relevantes para la toma de decisiones.

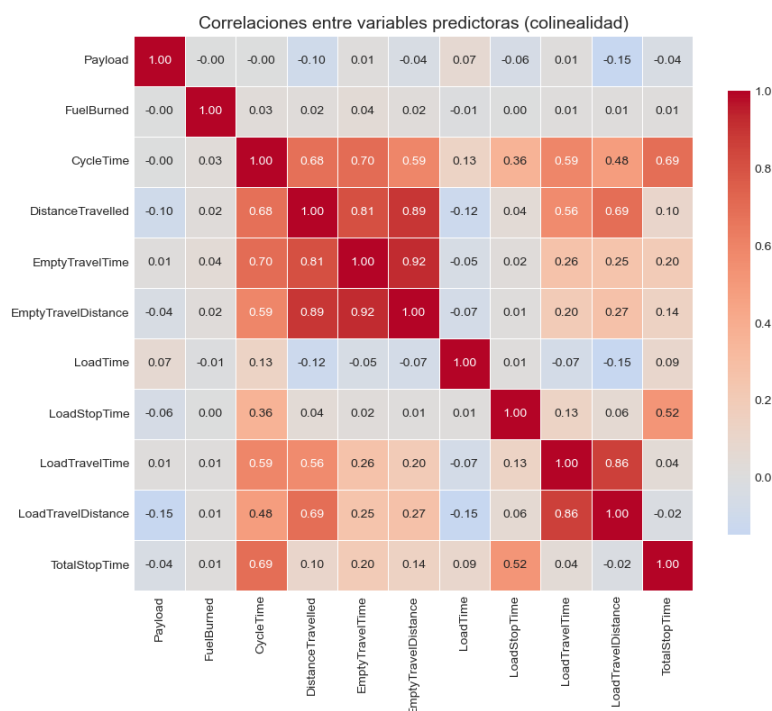
El nuevo alcance se define entonces en torno a tres modelos principales: - **Modelo de regresión 1:** predicción del tiempo detenido en vacío (**EmptyStopTime**). - **Modelo de regresión 2:** predicción del tiempo detenido durante el cargue (**LoadStopTime**). - **Modelo de clasificación:** determinación de la eficiencia del ciclo (**EfficientCycle**), categorizando cada evento como eficiente o ineficiente.

Los datos provienen de la base de VisionLink y corresponden a **cuatro camiones** que cuentan con información completa de telemetría. El tablero está dirigido a **líderes y gerentes de producción**, por lo que el enfoque se centra en facilitar la interpretación de resultados y apoyar la toma de decisiones operativas.

2. Modelado Predictivo

2.1. Preparación de Datos

El preprocesamiento de los datos incluyó la limpieza de valores atípicos y la conversión de variables de tiempo a formato numérico. Se realizaron transformaciones de tipo dummy para las variables categóricas relacionadas con la hora (**Hour_1...Hour_23**) y el turno (**Shift_Night**). Además, se eliminaron las variables con alta colinealidad como DistanceTravelled, EmptyTravelDistance y LoadTravelDistance teniendo en cuenta que su correlación fue mayor a 0.85.



Posteriormente, se realizó una división de los datos en entrenamiento (80%) y prueba (20%), preservando la secuencia cronológica de los eventos. Esto aseguró que los modelos fueran validados de forma realista, respetando la naturaleza temporal de la operación minera.

2.2. Selección de Variables y Entrenamiento

Para la selección de características se combinaron criterios de correlación estadística y relevancia operativa. Las variables finalmente utilizadas fueron:

Para los modelos de regresión: - Payload, FuelBurned, CycleTime, EmptyTravelTime, LoadTime, LoadTravelTime, Hour_1 a Hour_23, y Shift_Night (Dummies)

Para el modelo de clasificación: - Payload, FuelBurned, DistanceTravelled, EmptyTravelTime, EmptyTravelDistance, LoadTime, LoadTravelTime, LoadTravelDistance, TotalStopTime, Hour_1 a Hour_23, y Shift_Night (Dummies).

2.3. Selección de modelos y resultados

Escogimos algoritmos de RandomForest para los modelos de regresión y un GradientBoosting para el modelo de clasificación. Cada modelo fue implementado con técnicas de aprendizaje supervisado utilizando **scikit-learn**, registrando todos los experimentos en **MLflow** para garantizar la trazabilidad y comparación entre versiones. Además, se realizó una optimización de hiperparámetros mediante **RandomizedSearchCV**, ajustando los parámetros de profundidad, número de estimadores y tamaño de muestra en cada bosque aleatorio.

Hiperparametros utilizados para modelos de regresión:

- **n_estimators**: número de árboles en el bosque. Más árboles reducen la varianza del modelo, aunque incrementan el tiempo de entrenamiento. (**EmptyStopTime**) = 214 y (**LoadStopTime**) = 892
- **max_depth**: profundidad máxima de cada árbol. Controla la complejidad del modelo; profundidades muy altas pueden causar sobreajuste. (**EmptyStopTime**) = 15 y (**LoadStopTime**) = 8
- **min_samples_split**: número mínimo de muestras necesarias para dividir un nodo. Evita divisiones excesivas y ayuda a mejorar la generalización. (**EmptyStopTime**) = 6 y (**LoadStopTime**) = 4
- **min_samples_leaf**: número mínimo de muestras requeridas en las hojas finales de cada árbol. Valores mayores suavizan el modelo. (**EmptyStopTime**) = 1 y (**LoadStopTime**) = 3
- **max_features**: proporción de variables consideradas en cada división. Introduce aleatoriedad y reduce la correlación entre árboles. (**EmptyStopTime**) = 1 y (**LoadStopTime**) = 0.8

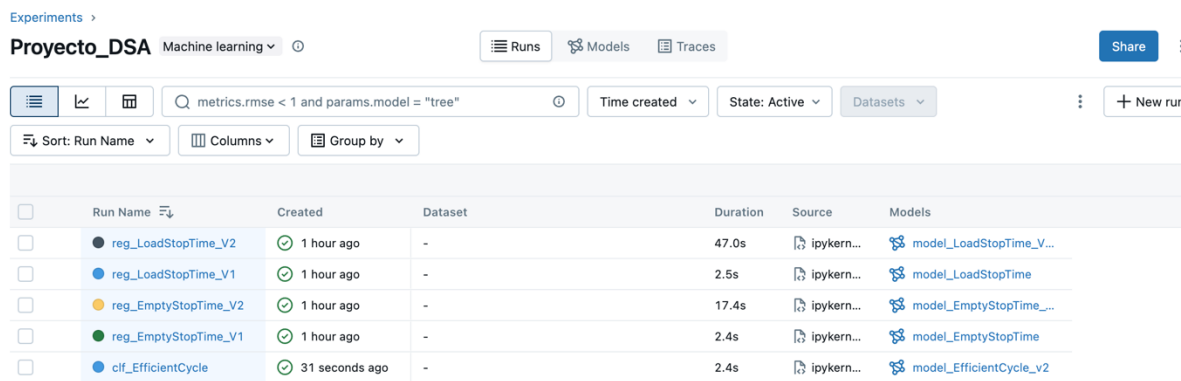
Hiperparametros utilizados para modelos de clasificación:

- **n_estimators**: número de árboles secuenciales. Cada árbol corrige los errores del anterior.
- **learning_rate**: tasa de aprendizaje que controla cuánto influye cada árbol nuevo. Valores bajos aumentan estabilidad pero requieren más árboles.
- **max_depth**: controla la profundidad de los árboles base, afectando directamente la complejidad del modelo.
- **subsample**: fracción de datos usada para entrenar cada árbol; introduce aleatoriedad y mejora la capacidad de generalización.

2.4. Resultados de los modelos seleccionados:

Modelo	Tipo	Variable objetivo	MAE (min)	RMSE (min)
EmptyStopTime	Regresión	Tiempo detenido en vacío	1.33	2.55
LoadStopTime	Regresión	Tiempo detenido en cargue	0.93	2.59

2.5. Experimentos con modelos probados (Mlflow)



2.6. Modelos de LoadStopTime (Mlflow)



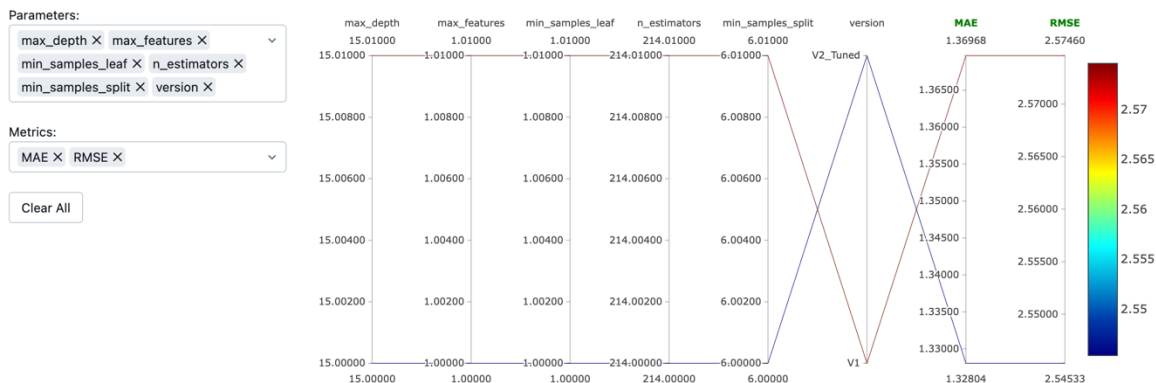
2.7. Modelos de EmptyStopTime (Mlflow)

Proyecto_DSA >

Comparing 2 Runs from 1 Experiment

Visualizations

Parallel Coordinates Plot Scatter Plot Box Plot Contour Plot



2.8. Modelo de Clasificación (Mlflow)

Proyecto_DSA > Runs >

clf_EfficientCycle

Overview Model metrics System metrics Traces Artifacts

Description

No description

Metrics (2)

Search metrics		
Metric	Value	Models
ROC_AUC	0.9974957445481201	model_EfficientCycle_v2
F1	0.9694189602446484	model_EfficientCycle_v2

Parameters (4)

Search parameters	
-------------------	--

About this run

Created at	11/07/2025, 09:22:42 PM
Created by	luiscortes
Experiment ID	476114362142632551
Status	Finished
Run ID	c09aecd83a94aaa93481393567d8666
Duration	2.4s
Source	ipykernel_launcher.py
Registered prompts	—

Datasets

None

Tags

Add tags

El **modelo de clasificación** para la eficiencia del ciclo logró un **F1-score de 0.97** y un **ROC-AUC de 0.997**, evidenciando un excelente poder predictivo. Sin embargo, reconocemos que estos valores pueden indicar un posible **sobreajuste (overfitting)**, dada la alta precisión en los datos de prueba. Este modelo podría beneficiarse de un ajuste más profundo de hiperparámetros y de una mayor variabilidad en los datos de entrenamiento.

Los valores obtenidos de MAE y RMSE indican que los modelos cometen errores promedio muy por debajo del umbral de 6 minutos que define una detención

ineficiente. En términos prácticos, esto significa que las predicciones de los modelos tienen una desviación pequeña respecto al valor real, suficiente para distinguir con alta confianza cuándo una detención está por encima o por debajo del límite operativo crítico. En otras palabras, si el modelo predice que una detención será de 7 u 8 minutos, la probabilidad de que efectivamente supere los 6 minutos es muy alta, mientras que un error de ± 2 minutos no compromete la capacidad de identificar correctamente los eventos anómalos.

3. Desarrollo del Tablero Predictivo

El tablero fue diseñado utilizando **Dash**, integrándose con la API desarrollada en **FastAPI**, que expone los tres modelos mediante endpoints independientes.

El diseño visual se basó en una **paleta gris y roja (Colores de identidad corporativa)**. La interfaz está orientada a usuarios no técnicos (supervisores, jefes y gerentes de producción), priorizando la claridad y simplicidad de uso.

3.1. Componentes Principales

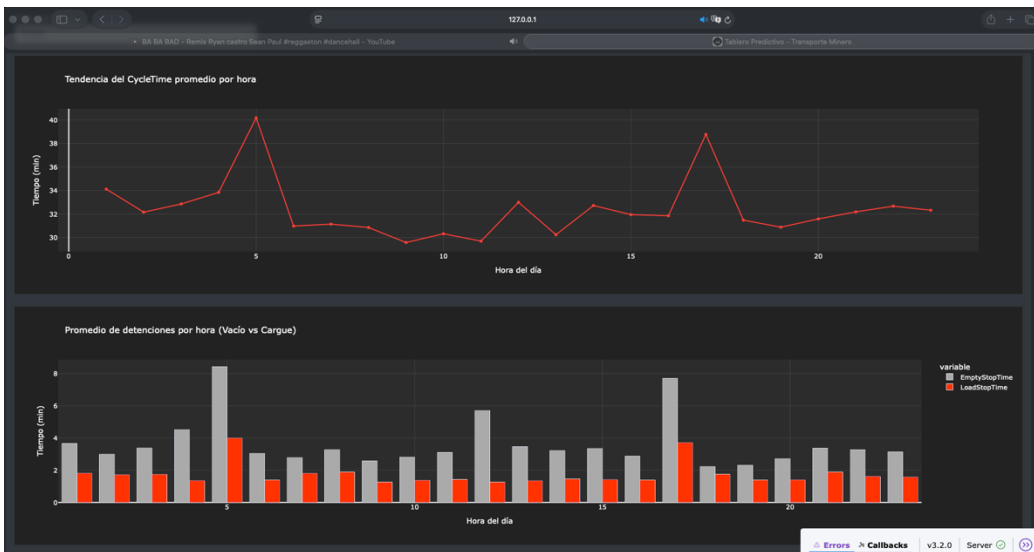
El tablero se compone de cuatro secciones:

1. **Indicadores Clave (KPIs):** muestran el porcentaje de ciclos eficientes, el número de detenciones mayores a 6 minutos en viaje vacío y en viaje cargado, y el total de ciclos analizados.
2. **Gráficas de Tendencia:** incluyen la evolución del tiempo de ciclo promedio por hora y los tiempos de detención promedio en viaje vacío y en viaje cargado, lo que permite identificar franjas horarias críticas.
3. **Gráfica Circular (Pie Chart):** visualiza la proporción de ciclos eficientes frente a ineficientes, brindando una visión general del desempeño operativo.
4. **Sección de Predicciones:** contiene tres formularios interactivos donde los usuarios pueden ingresar variables operativas (carga, combustible, tiempos de viaje y turno) y obtener predicciones automáticas de los modelos.

3.2. Cumplimiento del Alcance y Pregunta de Negocio

El tablero responde directamente a la pregunta de negocio planteada: “**¿Cómo podemos anticipar detenciones improductivas mayores a 6 min y optimizar los tiempos de ciclo de los camiones de acarreo?**”.

Los modelos de regresión permiten predecir con precisión los tiempos de detención, mientras que el modelo de clasificación aporta una visión binaria de eficiencia global. En conjunto, las visualizaciones y los indicadores permiten priorizar acciones correctivas en franjas horarias o turnos específicos, fortaleciendo el control operativo.



Predicciones de Modelos

Modelo 1 - Tiempo detenido en vacío (min)

Payload FuelBurned CycleTime EmptyTravelTime LoadTime LoadTravelTime Hour (1-23) Día

Predecir ciclo vacío

Modelo 2 - Tiempo detenido en carga (min)

Payload FuelBurned CycleTime EmptyTravelTime LoadTime LoadTravelTime Hour (1-23) Día

Predecir ciclo cargado

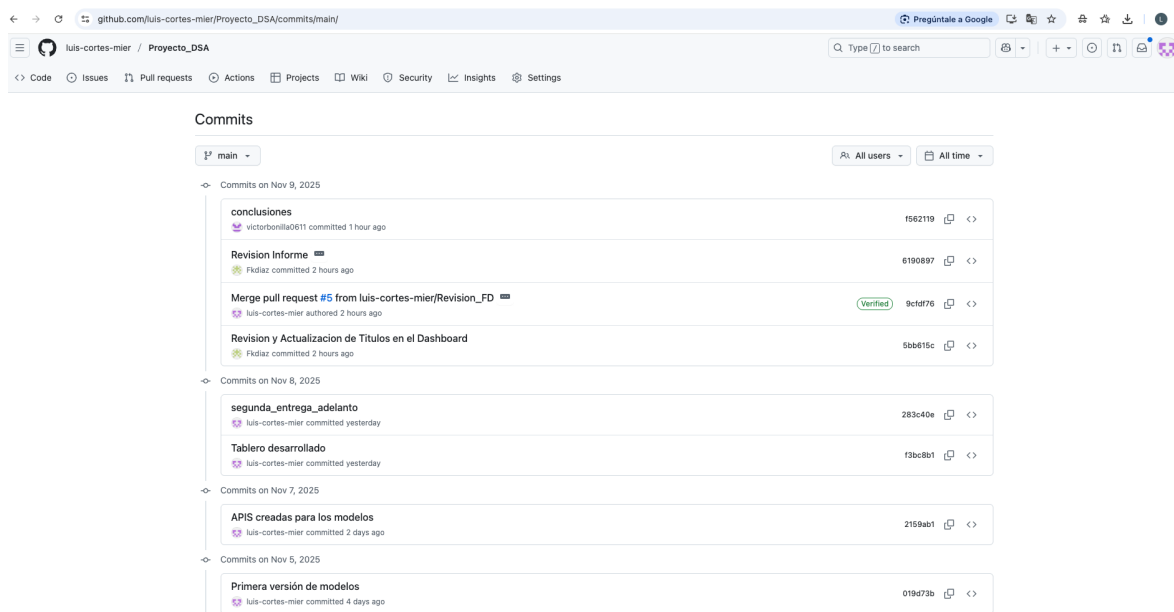
Modelo 3 - Clasificación de Ciclo Eficiente / Ineficiente

Payload FuelBurned DistanceTravelled EmptyTravelTime EmptyTravelDistance LoadTime LoadTravelTime LoadTravelDistance TotalStopTime Hour (1-23) Día

Ejecutar predicción de eficiencia

Errors Callbacks v3.2.0 Server

Trabajo colaborativo



Para esta segunda entrega decidimos mejorar nuestra forma de trabajo y organización. Una de las principales acciones fue crear ramas independientes a la principal (main) para que cada uno pudiera avanzar en su parte del proyecto sin afectar el trabajo de los demás. Esto nos ayudó a mantener un mejor control de versiones, probar cambios con más confianza y luego integrar todo de manera ordenada.

Nuestro equipo está conformado por tres integrantes, y aunque cada uno tiene un rol principal definido, todos participamos activamente en las diferentes fases del proyecto. Desde el análisis de datos y el entrenamiento de los modelos, hasta la construcción del tablero y la documentación final, siempre nos hemos apoyado mutuamente para mantener coherencia y lograr los objetivos propuestos.

- Luis Cortes ha estado a cargo de todo el entorno de MLOps. Su trabajo se ha centrado en que el proyecto sea colaborativo, organizado y fácil de reproducir. Para lograrlo, configuró herramientas como Git, GitHub y DVC, que nos han permitido versionar tanto el código como los datos y tener un flujo de trabajo más profesional.
- Víctor Bonilla ha liderado la parte de modelado predictivo, probando diferentes enfoques y ajustando los modelos de regresión y clasificación hasta obtener los mejores resultados. Su enfoque ha sido garantizar que las soluciones sean técnicamente sólidas y realmente aporten valor a la pregunta de negocio planteada.

- Frank Díaz ha sido el responsable principal del diseño y desarrollo del tablero interactivo, donde se integran los resultados de los modelos y se presentan de manera visual. Gracias a su trabajo, logramos una interfaz sencilla e intuitiva pensada para que los líderes y gerentes de producción puedan interpretar fácilmente los resultados y tomar decisiones basadas en datos.

Más allá de estos roles, hemos trabajado siempre de forma conjunta, revisando el trabajo de los demás, compartiendo ideas y resolviendo los problemas en equipo. Esta colaboración constante nos ha permitido mantener una línea coherente en todo el proyecto y entregar una solución completa, funcional y bien estructurada.

El repositorio remoto en github es: https://github.com/luis-cortes-mier/Proyecto_DSA.git

