

**Maestría en Inteligencia Artificial Aplicada**

**Materia:** Proyecto Integrador

**Profesor Titular:** Dra. Grettel Barceló Alonso / Dr. Luis Eduardo Falcón Morales

**Asesor de Proyecto:** Dr. Carlos Alberto Villaseñor Padilla

Avance 6. Conclusiones clave

**Equipo 10**

David García Robles A01152606

David Nava Jiménez A01168501

José Antonio Hernández Hernández A01381334

**Fecha:** 08 de Junio de 2025

**Optimización de ventas en Nacional Monte de Piedad**

**Avance 6. Conclusiones clave**

Contenido

[**Avance 6. Conclusiones clave**](#_heading=h.dimknyvj2juz)

**1.1 Implementación del modelo (únicamente para el ramo de alhajas)**

**1.2 Margen para mejorar el rendimiento del modelo:**

**1.3 Recomendaciones clave para implementar la solución:**

**1.4 Tareas y procedimientos accionables a los stakeholders**

**2.1 Sugerencias de implementación del modelo**

**2.2 Consideraciones computacionales y técnicas**

**Conclusiones**

**1.1 Implementación del modelo (únicamente para el ramo de alhajas)**

El rendimiento del modelo:

Tras una evaluación de múltiples algoritmos de regresión, se definió como modelo final una estrategia de ensamble heterogénea basada en un VotingRegressor, el cual integra dos estimadores: un XGBoostRegressor y un MLPRegressor. Esta configuración permite capturar tanto relaciones lineales como patrones no lineales complejos presentes en los datos, maximizando así la capacidad predictiva del modelo. Lo consideramos el mejor modelo debido a dos características, desempeño y el más bajo sobre ajuste de todos los modelos aunque el costo computacional es alto en comparación al XGBoostRegressor.

Train R2: 0.819 ± 0.001

Val R2: 0.778 ± 0.003

Train RMSE: 48.286 ± 0.113

Val RMSE: 53.555 ± 0.552

**Test :R^2: 0.7841**

**Test: rmse:52.91**

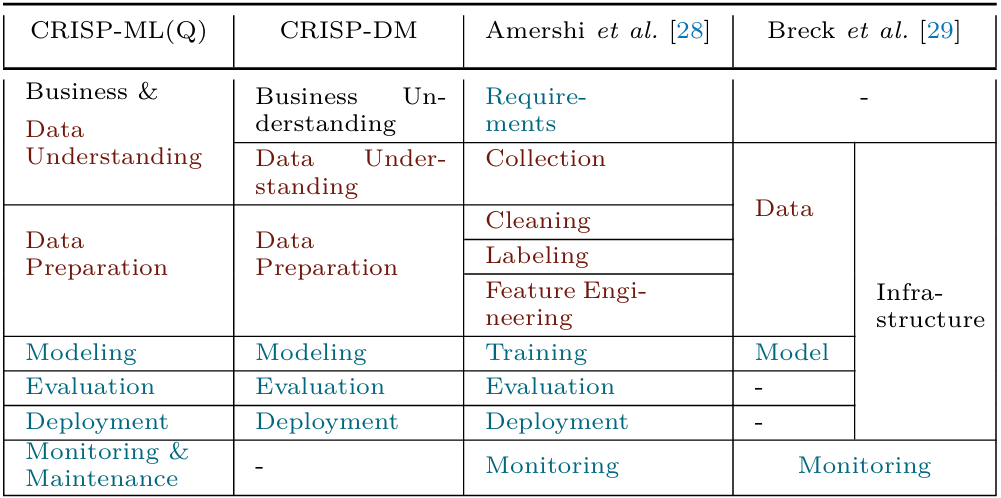
Considerando ambas métricas R^2 y RMSE, la primera de .78 y la segunda de 53 días, consideramos que para el negocio el rendimiento es bueno y será útil para su implementación en producción. Esto derivado de la problemática expuesta que es la lenta rotación de artículos que se otorgan en prenda de créditos. Es importante mencionar que los rangos en días que tardan en desplazarse los artículos va de (0 días -3,819 días), por lo que una desviación de 53 días es efectiva para su aplicación y despliegue.

**1.2 Margen para mejorar el rendimiento del modelo:**

Consideramos que con los datos que nos compartieron por parte del cliente existe probabilidad de mejora, aplicando modelos más complejos como **redes neuronales profundas**. de acuerdo con (Goodfellow et al., 2016) la alta dimensionalidad de un problema es un reto para los modelos tradicionales de machine learning, sin embargo las redes neuronales profundas han sido creadas para este tipo de problemas en específico. Sin embargo el costo computacional es alto en este tipo de modelos, lo que conlleva a una mayor inversión en el despliegue y mantenimiento del mismo.

**1.3 Recomendaciones clave para implementar la solución:**

Es importante considerar que el proyecto se estructuró con base a la metodología CRISP-ML (Q), la cual se muestra a continuación:



*Obtenido de: Studer, S., Bui, T. B., Drescher, C., Hanuschkin, A., Winkler, L., Peters, S., & Mueller, K.-R. (2021). Towards CRISP-ML(Q): A Machine Learning Process Model with Quality Assurance Methodology. Methodology. Preprints 2021, 1.* [*https://arxiv.org/abs/2003.05155*](https://arxiv.org/abs/2003.05155)

El alcance del proyecto se estableció hasta la etapa de evaluación sin considerar el despliegue, monitoreo y mantenimiento del modelo.

La recomendación de acuerdo a las mejores prácticas para desplegar un modelo (Tecnológico de Monterrey, 2024) es la **refactorización del código** desarrollado en los jupyter notebooks, ya que es crucial en los proyectos de machine learning, para mejorar la calidad, mantenimiento y escalabilidad del código.

Normalmente los modelos se crean de manera rápida, enfocando los esfuerzos en la experimentación y pruebas de hipótesis, como fue nuestro caso. Sin embargo, esto puede generar dificultades al momento de administrar, probar y escalar el código para su despliegue. La refactorización transforma el código en un sistema modular, reusable y con componentes eficientes, que genera un estructura más limpia y simplifica el “debbuging” del código. Esto es necesario para establecer mejores prácticas como es el control de versiones, reproducibilidad, pipelines de datos automáticos y bases para un despliegue de modelo más sólido.

Una vez que el modelo se despliega, es necesario **monitorear el desempeño** del mismo (métricas del negocio y métricas propias del modelo, en nuestro caso R^2 y RMSE) , así como **re-entrenarlo si fuera necesario**. Esto derivado a retos que los modelos experimentan a lo largo del tiempo y que se denominan como “data drift” y “concept drift”. El data drift ocurre cuando la distribución inicial de los datos cambia con el tiempo con respecto a los datos de entrenamiento, por lo que el desempeño del modelo disminuye. Por otro lado, “concept drift” se refiere cuando hay cambios en conductas y la realidad de las relaciones entre las variables predictoras y la variable a predecir.

**1.4 Tareas y procedimientos accionables a los stakeholders**

La aplicación de negocio de este modelo de inteligencia artificial es para establecer estrategias comerciales de NMP (Nacional Monte de Piedad) que agilicen la rotación de productos otorgados en prenda. Es decir, algunas de las iniciativas comerciales que podrían aplicar son las siguientes (no son limitativas):

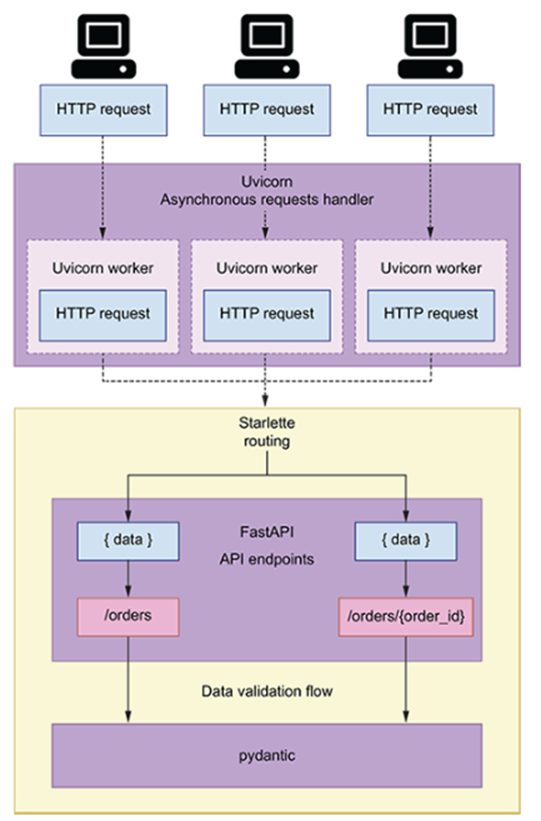
* Con apoyo de un **OCR** escanear y clasificar el tipo de alhaja con las diferentes características que se evaluaron en el modelo. (**modelo adicional a desarrollar**)
* Una vez registrado el artículo, aplicar el modelo desarrollado para predecir los días que estará en inventario.
* Considerando los días y el margen de error del modelo, establecer condiciones comerciales como:
  + En caso de una baja rotación (alto número de días en inventario) proponer una mayor tasa de interés o un menor monto a financiar al sugerido por el valuador.
    - Desarrollar un **programa comercial de marketing o incentivos (descuentos)** para los artículos con una predicción en días, alto de la variable dias\_almoneda.
  + En caso de una alta rotación (bajo número de días en inventario) proponer mejores condiciones comerciales,(e.g) tasa de interés menor y considerando el avalúo ofrecer un monto a financiar similar al solicitado por el cliente.
* Finalmente, consideramos que para optimizar las ventas en NMP, se requiere de un dataset que contenga la interacción de usuarios y clientes con su página de e-commerce (artículos) con el fin de crear un **sistema de recomendación (modelo adicional a desarrollar)** que puede interactuar con el modelo ya desarrollado y dar prioridad a los artículos con menor rotación.

**2.1 Sugerencias de implementación del modelo**

Evaluando que es un modelo ligero, es decir un modelo de regresión consideramos que la implementación y despliegue óptimo puede ser por medio de **FastAPI o Flask** en conjunto con la herramienta de **MLFlow** para monitorear su rendimiento y detectar desviaciones en el modelo como las que comentamos anteriormente.

Qué es FastAPI?

De acuerdo con (Haro, 2022), es un framework para APi’s construido sobre Starlette, el cual es un framework web ASGI (interfaz de servidor asincrónico) de alto rendimiento y ligero, lo que permite implementar los servicios como un conjunto de tareas asincrónicas para mejorar el rendimiento de las aplicaciones. Tiene la siguiente estructura:



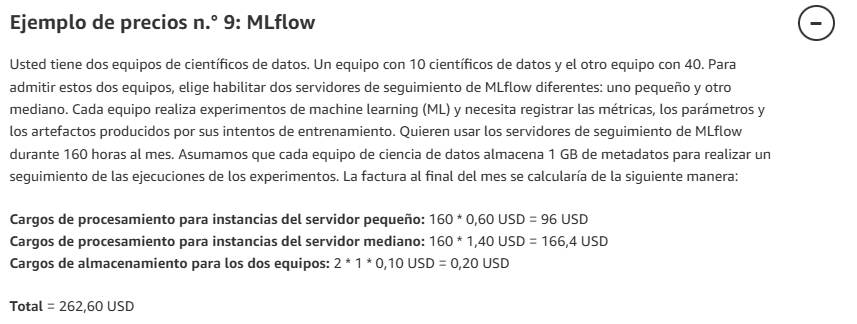
*Obtenido de: Haro, J. (2022). Microservice API's: Using Python, Flask, FastAPI, OpenAPI and more.* [*https://learning.oreilly.com/library/view/microservice-apis/9781617298417/OEBPS/Text/02.htm#heading\_id\_3*](https://learning.oreilly.com/library/view/microservice-apis/9781617298417/OEBPS/Text/02.htm#heading_id_3)

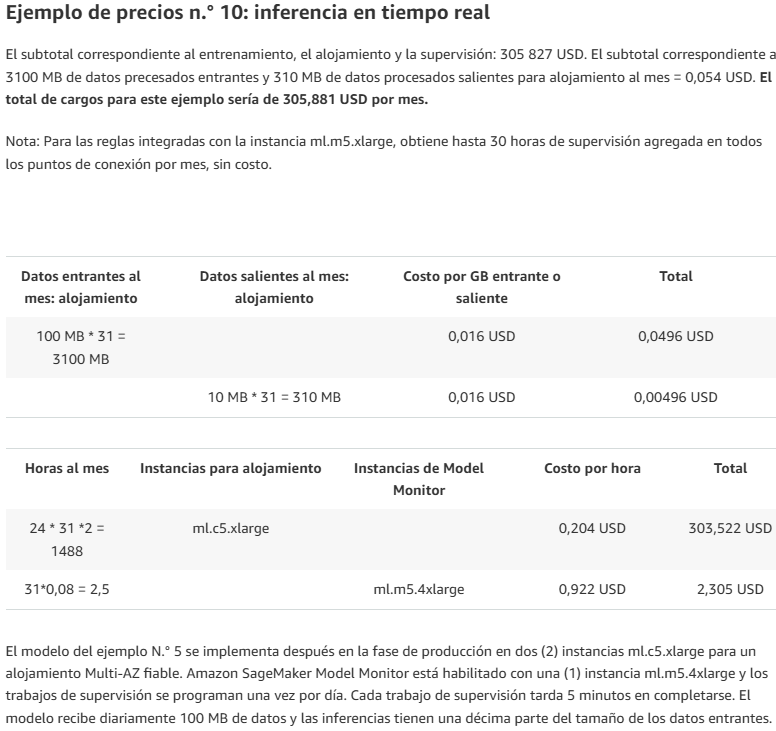
Como ventaja de esta solución es que puede correr en entornos locales por medio del protocolo HTTP, el OPEX es muy bajo a comparación de una solución en la nube. Se recomienda que el despliegue del modelo sea por microservicios, los cuales se pueden virtualizar desde contenedores (docker) o máquinas virtuales locales.

Ventajas de una solución on-premise:

* Los gastos asociados son de los desarrolladores internos y científicos de datos de NMP.
* Arquitectura diseñada de acuerdo a las necesidades de NMP.

El despliegue del código también se puede realizar de manera monolítica, es decir todo el bloque de código. Existen soluciones en la nube como SageMaker de AWS, Azure ML, GCP y Vertex AI. En este contexto, las aplicaciones implementadas de esta forma se ocupan cuando el código base es pequeño y no se espera desarrollar más sobre él. Su implementación es sencilla y el acceso a los datos, de igual manera.





*Obtenido de: AWS. (2024). Precios de Amazon SageMaker AI. AWS.* [*https://aws.amazon.com/es/sagemaker-ai/pricing/*](https://aws.amazon.com/es/sagemaker-ai/pricing/)

Si se decide una solución en la nube, las ventajas son las siguientes:

* No hay costes por adelantado.
* No es necesario comprar infraestructuras costosas
* Escalabilidad. Se puede pagar más para obtener más recursos cuando se requieran.
* Se puede dejar de pagar por recursos que no se requieran.
* Alta seguridad y confiabilidad.
* Alta disponibilidad. Garantizan la máxima disponibilidad independientemente de interrupción o eventos que puedan producirse.

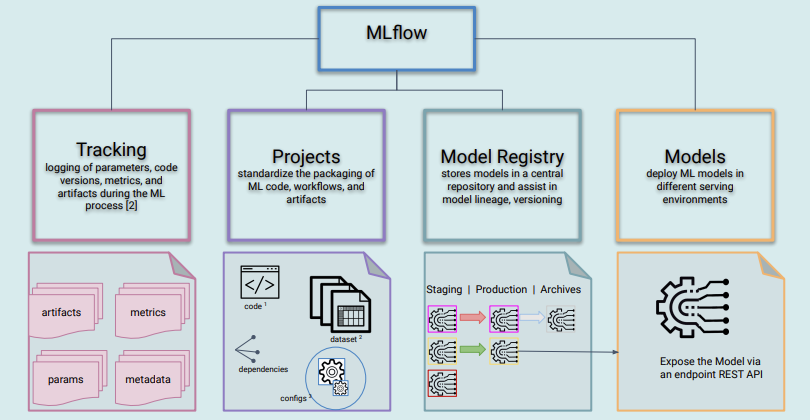
¿Qué es MLflow?

MLFlow es una plataforma de código abierto diseñada para gestionar el ciclo de vida completo de modelos de machine learning. Su objetivo principal es facilitar el desarrollo, la experimentación, el seguimiento, el registro y el monitoreo de modelos en producción de forma estructurada y reproducible.

Para asegurar la eficiencia, trazabilidad y control continuo del modelo predictivo en producción, proponemos la adopción de un enfoque de MLOps basado en MLFlow. Este enfoque abarca todas las etapas del ciclo de vida del modelo, desde la ingestión y preprocesamiento de datos, hasta el entrenamiento, evaluación, despliegue y monitoreo en tiempo real.

Beneficios Clave para NMP

* Despliegue controlado del modelo en un entorno API para uso en tiempo real.
* Detección temporada de desviaciones en el comportamiento del modelo, gracias al monitoreo continuo.
* Historial y trazabilidad de todos los experimentos realizados, facilitando auditorías y mejoras.
* Interoperabilidad futura con nuevos modelos, como sistemas de recomendación o clasificación automática de productos vía OCR.



*Obtenido de: Berberi, L. (2023).MLflow and its usage* [*https://www.researchgate.net/publication/376029507\_MLflow\_and\_its\_usage*](https://learning.oreilly.com/library/view/microservice-apis/9781617298417/OEBPS/Text/02.htm#heading_id_3)

**2.2 Consideraciones computacionales y técnicas**

1. **Infraestructura y requerimientos de hardware**

Durante el desarrollo y evaluación de los modelos, se identificó que los algoritmos más robustos, como XGBoost y MLPRegressor, presentan una alta demanda computacional, especia en tareas de validación cruzada y búsqueda de hiperparametros. En este sentido, se recomiendan los siguientes requerimientos mínimos para ejecutar el pipeline completo de manera eficiente:

* Procesador: CPU de múltiples núcleos (mínimo 8 cores).
* Memoria RAM: Al menos 16 GB.
* GPU (Opcional): Para explorar modelos de redes neuronales más profundas en el futuro.
* Almacenamiento: Al menos 20 GB disponibles para manejo de datasets, modelos entrenados y respaldos.

1. **Herramientas clave utilizadas**

El set tecnológico aplicado fue construido con herramientas ampliamente utilizadas en proyectos de ciencia de datos y machine learning, incluyendo:

* Python 3.9+
* Jupyter Notebooks
* Pandas/Numpy para manipulación de datos.
* Scikit-learn para regresores base (LinearRegression, ElasticNet, Desicion Tree, MLPRegressor), validación cruzada y ensambles ( VotinRegressor).
* XGBoost para modelos de boosting de alto rendimiento.
* Matplotlib y Seaborn para visualizaciones.
* t-SNE para reducción de dimensionalidad y exploración visual.
* RandomizedSearchCV para optimización de hiperparametros.

**Conclusiones**

El desarrollo de este proyecto permitió establecer una solución basada en inteligencia artificial que predice los días que una alhaja permanecerá en inventario en Nacional Monte de Piedad (NMP). A través de un riguroso proceso de experimentación y evaluación, se identificó como modelo óptimo una estrategia de ensamble heterogénea con VotingRegressor, que combina XGBoostRegressor y MLPRegressor, logrando un equilibrio entre precisión predictiva (R2 = 0.78) y bajo sobreajuste (RMSE = 53 días).

Durante las fases iniciales, se identificó el riesgo de fuga de información por variables altamente correlacionadas, lo cual fue subsanado para garantizar la validez del modelo en escenarios reales. Esto permitió construir una solución más robusta y representativa del entorno operativo de NMP. Además, el análisis de permutaciones evidenció que variables como el canal de cumplimiento, ubicación de la sucursal y características específicas del artículo como el kilataje son altamente influyentes en la predicción.

En términos de operación, el modelo demostró ser eficaz para anticipar comportamientos de inventario y puede ser una herramienta clave en la definición de estrategias comerciales: desde el ajuste de tasas de interés y montos de financiamiento, hasta la ejecución de campañas específicas para productos de baja rotación.

Desde una perspectiva técnica y de implementación, se recomienda adoptar un enfoque MLOps para asegurar la trazabilidad, control de versiones y mantenimiento continuo del modelo en producción. En particular, se sugiere el uso de MLflow, una plataforma open-source que permite gestionar todo el ciclo de vida del modelo de forma estructurada: desde el entrenamiento y evaluación hasta el despliegue y monitoreo.

MLflow facilitará:

* La gestión de experimentos, parámetros y métricas en un entorno centralizado.
* El despliegue de modelos como microservicios a través de FastAPI o Flask, permitiendo una integración sencilla en flujos operativos.
* El monitoreo en tiempo real de métricas como R2 o RMSE para detectar posibles desviaciones del modelo.
* La interoperabilidad con modelos futuros, como un sistema de recomendación o clasificadores automáticos vía OCR.

Finalmente, este proyecto sienta las bases para un ecosistema de analítica avanzada en NMP, donde el uso de modelos híbridos, buenas prácticas de ingeniería de características y herramientas modernas de MLOps permiten transformar datos en decisiones accionables y con impacto operativo real.

**Referencias**

* Goodfellow, I., Bengio, Y., & Courville, A. (2016). Machine Learning Basics. In *Deep Learning* (pp. 83-122). The MIT Press. <https://research.ebsco.com/c/oefy3m/ebook-viewer/epub/znpbphupuz>
* Tecnológico de Monterrey. (2024, Marzo 10). *Clase MLOps -MNA*.
* Studer, S., Bui, T. B., Drescher, C., Hanuschkin, A., Winkler, L., Peters, S., & Mueller, K.-R. (2021). Towards CRISP-ML(Q): A Machine Learning Process Model with Quality Assurance Methodology. *Methodology. Preprints 2021*, *1*. <https://arxiv.org/abs/2003.05155>
* Ridho,R & Arifudin, R. (2023). Stock Return Prediction Using Voting Regressor Ensemble Learning. *Recursive Journal of Informatics* (pp. 59-66). <https://www.researchgate.net/publication/374306042_Stock_Return_Prediction_Using_Voting_Regressor_Ensemble_Learning/link/6518186b1e2386049deb9e94/download?_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6InB1YmxpY2F0aW9uIiwicGFnZSI6InB1YmxpY2F0aW9uIn19>
* Haro, J. (2022). *Microservice API's: Using Python, Flask, FastAPI, OpenAPI and more*. <https://learning.oreilly.com/library/view/microservice-apis/9781617298417/OEBPS/Text/02.htm#heading_id_3>
* Berberi, L. (2023).MLflow and its usage. *Introduction to MLFlow.* ( pp- 04-05). <https://www.researchgate.net/publication/376029507_MLflow_and_its_usage>
* AWS. (2024). *Precios de Amazon SageMaker AI*. AWS. https://aws.amazon.com/es/sagemaker-ai/pricing/