



**Evaluación de impactos monetarios ante afectaciones tecnológicas: un enfoque predictivo
basado en Machine Learning.**

María Salomé Garcés Montero

Informe de prácticas para optar al título de Ingeniera Electrónica

Semestre de Industria

Asesor

Sebastián Isaza Ramírez, PhD

Universidad de Antioquia
Facultad de Ingeniería
Ingeniería Electrónica
Medellín, Antioquia, Colombia
2025

Cita	(Garcés Montero, 2025)
Referencia	Garcés Montero, S. (2025). <i>Evaluación de impactos monetarios ante afectaciones tecnológicas: un enfoque predictivo basado en aprendizaje automático</i> [Informe de práctica]. Universidad de Antioquia, Medellín, Colombia.
Estilo APA 7 (2020)	



Centro de Documentación Ingeniería (CENDOI)

Repositorio Institucional: <http://bibliotecadigital.udea.edu.co>

Universidad de Antioquia - www.udea.edu.co

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

Dedicatoria

A mi mamá, la razón de mi perseverancia y motivación. Tu amor ha sido mi fuente de inspiración; eres el pilar que ha sostenido todo lo que hoy soy. Este proyecto es tan tuyo como mío, porque de no ser por ti, no estaría donde estoy.

A mi viejita linda, quien siempre ha sido un ejemplo de fortaleza, amor y sabiduría. Este logro es, en gran parte, un reflejo de todo el amor que me has dado a lo largo de los años. Gracias por ser mi guía, mi refugio y mi inspiración.

A Laurita, que ha estado a mi lado en cada paso de este camino, acompañándome en los momentos más difíciles y celebrando también cada uno de los logros, por pequeños que fueran. Gracias por creer en mí, incluso cuando ni yo misma lo hacía.

A Cristal y Leonella.

Agradecimientos

Quiero expresar mi más sincero agradecimiento a Bancolombia, por brindarme la oportunidad de desarrollar este proyecto dentro de una de las instituciones más importantes del país. A todo el equipo de SRE, por su apoyo, paciencia y colaboración en cada etapa del proyecto. En especial a Alejo, por su disposición para compartir conocimientos y experiencia.

A mis asesores de proyecto, quienes me guiaron y orientaron a lo largo de este proceso. Sus consejos y recomendaciones fueron esenciales para enfocar correctamente el desarrollo del modelo y para el éxito de este trabajo.

A mi familia, en especial a mi mamá y a mi abuelita, por su apoyo incondicional. Siempre han sido mi fuerza en los momentos más difíciles, aún en la distancia. Gracias por brindarme el amor y la confianza que necesitaba.

A Lau, por su comprensión, apoyo y paciencia. Gracias por estar. Sin ti, este proceso no habría sido lo mismo.

Finalmente, a todos aquellos que, de una u otra manera, contribuyeron al desarrollo de este trabajo. Gracias por ser parte de este importante proyecto.

Tabla de contenido

Resumen	9
Abstract	10
1. Introducción	11
2. Objetivos	12
2.1 Objetivo general	12
2.2 Objetivos específicos.....	12
3. Marco teórico	13
3.1 Machine Learning	13
3.1.1 Tipos de aprendizaje en Machine Learning	13
3.1.2 Tipos de modelos en Machine Learning	14
3.1.3 Hiperparámetros en modelos de Machine Learning	15
3.1.4 Métodos de ajuste de hiperparámetros	15
3.1.5 Métricas de rendimiento en modelos de Machine Learning.....	16
3.2 Landing Zone	17
3.3 Uso de modelos predictivos para la predicción de pérdidas en casos de indisponibilidad.....	18
4. Metodología	19
4.1 Configuración de entornos.....	19
4.2 Recolección y análisis de datos históricos.....	20
4.3 Implementación y selección de modelos predictivos.....	22
4.4 Refinamiento del modelo seleccionado.....	27
4.5 Automatización de la actualización del modelo.....	29
4.6 Despliegue del modelo como API.....	31
4.7 Desarrollo de un dashboard para visualización de predicciones.....	31
5. Análisis de resultados.....	32
5.1 Refinamiento del modelo.....	32
5.2 Dashboard para consulta de resultados.....	34

6. Conclusiones y recomendaciones.....37

Referencias40

Lista de tablas

Tabla 1 Variables seleccionadas para la implementación del modelo	23
Tabla 2 MAPE en la comparación de algoritmos en fechas aleatorias	25
Tabla 3 Desempeño de los algoritmos durante las pruebas basados en el error MAPE	25
Tabla 4 Desempeño de los algoritmos durante las pruebas basados en el error MAE	25
Tabla 5 Desempeño de los algoritmos durante las pruebas basados en el error RMSE	26

Lista de figuras

Figura 1	Histograma de datos históricos de comportamiento de comisiones	21
Figura 2	Comportamiento histórico de transacciones para el año 2023	22
Figura 3	Columnas del dataset inicial de entrenamiento	23
Figura 4	Columnas del dataset de entrenamiento con variables categóricas	27
Figura 5	Stacking en modelos de Machine Learning	28
Figura 6	Comparación valores reales vs predicciones del modelo para octubre de 2024	32
Figura 7	MAPE diario para predicciones del mes de octubre de 2024	33
Figura 8	Portada dashboard en Power BI para consulta de predicciones y métricas	34
Figura 9	Sección de predicciones del dashboard en Power BI	34
Figura 10	Sección de métricas de uso y adopción del dashboard en Power BI	35

Siglas, acrónimos y abreviaturas

BALM	Bancolombia a la mano
SRE	Site Reliability Engineering
VSTI	Vicepresidencia de Servicios de Tecnología
UdeA	Universidad de Antioquia
LZ	Landing Zone
MAPE	Error Absoluto Porcentual Medio
MAE	Error Absoluto Medio
RMSE	Raíz Cuadrada del Error Medio

Resumen

Este proyecto, desarrollado en el equipo de Site Reliability Engineering (SRE) de la Vicepresidencia de Servicios de Tecnología de Bancolombia, aborda el impacto económico de la indisponibilidad de una de las aplicaciones transaccionales clave del banco. Su objetivo principal es desarrollar un modelo predictivo basado en aprendizaje automático que permita estimar las pérdidas económicas potenciales ante eventos de indisponibilidad.

La metodología incluye el análisis de datos históricos de transacciones para predecir las ganancias generadas por la aplicación y calcular las pérdidas económicas potenciales derivadas de eventos críticos. Para ello, se utilizó Python en el desarrollo del modelo, una arquitectura en la Landing Zone de la compañía para su despliegue y gestión, y Power BI para la creación de un dashboard interactivo que facilita la visualización de los resultados y sus métricas.

El modelo implementado proporciona una herramienta para apoyar la toma de decisiones financieras anticipadas y la planificación de contingencias, permitiendo al banco mejorar su capacidad de respuesta y su resiliencia operativa. Este trabajo contribuye significativamente a la optimización de la gestión de riesgos tecnológicos.

Palabras clave: Aprendizaje automático, pérdidas económicas, predictive model, Site Reliability Engineering.

Abstract

This project, developed within the Site Reliability Engineering (SRE) team of Bancolombia's Vice Presidency of Technology Services, addresses the economic impact of the unavailability of one of the bank's key transactional applications. Its main objective is to develop a predictive model based on machine learning to estimate potential economic losses during critical downtime events.

The methodology includes analyzing historical transaction data to predict the application's revenue and calculate economic losses stemming from critical events. Python was used for model development, while the company's Landing Zone hosted the architecture for deployment and management. Additionally, Power BI was utilized to create an interactive dashboard that facilitates the visualization of results and metrics.

The implemented model serves as a tool to support proactive financial decision-making and contingency planning, enabling the bank to enhance its response capacity and operational resilience. This work significantly contributes to optimizing the management of technological risks.

Keywords: Bancolombia, machine learning, economic losses, predictive model, Site Reliability Engineering.

1. Introducción

Bancolombia, reconocida como una de las instituciones financieras más importantes del país, desempeña un papel fundamental en el desarrollo económico de Colombia. Con una amplia cantidad de clientes, el banco ha centrado sus esfuerzos en ofrecer soluciones tecnológicas innovadoras, eficientes y seguras, adaptándose a las crecientes demandas del entorno. Para garantizar la calidad de sus operaciones, la Vicepresidencia de Servicios de Tecnología lidera los desarrollos tecnológicos que soportan sus procesos, asegurando que las plataformas y servicios estén diseñados con los más altos estándares de confiabilidad y rendimiento.

Dentro de esta vicepresidencia, el equipo de SRE se ha convertido en un pilar clave al implementar prácticas y herramientas enfocadas en garantizar la estabilidad y disponibilidad de los sistemas. Este enfoque es muy importante en un contexto donde la disponibilidad de las aplicaciones transaccionales es esencial para mantener la confianza del cliente y la continuidad del negocio.

Esta dependencia tecnológica también conlleva riesgos significativos. La indisponibilidad de una aplicación crítica, como las de transacciones, puede no solo generar insatisfacción en los clientes, sino también provocar pérdidas económicas considerables y dañar la reputación de la entidad. Estas situaciones adquieren mayor relevancia durante fechas de alta demanda, como temporadas de pago de nómina, eventos comerciales o festividades, donde el volumen de transacciones aumenta exponencialmente.

Con el objetivo de abordar estos desafíos, este proyecto titulado "Evaluación de Impactos Monetarios ante Afectaciones Tecnológicas: Un Enfoque Predictivo Basado en Machine Learning", desarrollado en el equipo de SRE, busca construir un modelo predictivo que permita al usuario final cuantificar las pérdidas económicas derivadas de la indisponibilidad de una de las principales aplicaciones transaccionales del banco.

El modelo no solo pretende proporcionar una herramienta para estimar las pérdidas, sino también convertirse en un recurso estratégico para la planificación y mitigación de riesgos. A través

de su implementación, Bancolombia podrá fortalecer su capacidad de respuesta ante interrupciones, optimizar la gestión de sus recursos tecnológicos y financieros.

2. Objetivos

2.1 Objetivo general

Desarrollar una aplicación que, mediante un modelo predictivo basado en aprendizaje automático, permita estimar las pérdidas económicas potenciales derivadas de la indisponibilidad de una de las aplicaciones transaccionales de la entidad bancaria, con el fin de apoyar la toma de decisiones estratégicas y la planificación de contingencias.

2.2 Objetivos específicos

- Desarrollar un modelo predictivo en Python que estime las ganancias generadas por la aplicación bancaria en función de la hora y fecha, usando datos históricos sobre las comisiones generadas por las transacciones.
- Implementar una arquitectura para alojar y gestionar el modelo predictivo, garantizando que pueda manejar grandes volúmenes de datos y adaptarse a las necesidades del negocio.
- Establecer un proceso continuo para la actualización del modelo predictivo, incluyendo la recolección de nuevos datos, la reevaluación y ajuste del modelo, con el fin de garantizar que las predicciones se mantengan alineadas con las condiciones actuales.
- Construir un dashboard interactivo y escalable para la visualización de las predicciones, facilitando la interpretación y el análisis de resultados.

3. Marco teórico

3.1. Machine Learning

El Machine Learning (ML) o aprendizaje automático es una subdisciplina dentro de la inteligencia artificial (IA) que permite a las máquinas aprender a partir de datos sin ser explícitamente programadas. A través de diversos algoritmos y modelos, las máquinas pueden identificar patrones y hacer predicciones basadas en estos datos.

El concepto de aprendizaje automático fue propuesto inicialmente por Arthur Samuel en 1959, quien lo definió como "el campo de estudio que da a las computadoras la capacidad de aprender sin ser programadas explícitamente para ello" (Samuel, 1959). A lo largo de los años, el término ha evolucionado y hoy abarca una amplia gama de métodos estadísticos y computacionales para permitir que las máquinas extraigan patrones de los datos y hagan predicciones o decisiones basadas en esos patrones (Alpaydin, 2014).

Los sistemas de aprendizaje automático buscan, principalmente, predecir resultados, clasificar datos, optimizar decisiones y analizar grandes volúmenes de información (Bishop, 2006). A medida que se alimenta a los modelos con más datos, estos pueden mejorar sus capacidades predictivas.

3.1.1. Tipos de Aprendizaje en Machine Learning

Existen tres tipos principales de aprendizaje en Machine Learning: supervisado, no supervisado y por refuerzo. Estos tipos se diferencian por la naturaleza de los datos con los que el modelo se entrena y el tipo de tareas que realizan.

- **Aprendizaje Supervisado:** El aprendizaje supervisado es el enfoque más común en Machine Learning. En este caso, el modelo es entrenado utilizando un conjunto de datos etiquetados, es decir, datos que incluyen tanto las entradas como las salidas deseadas.

El objetivo del modelo es aprender la relación entre las entradas y las salidas, de manera que pueda predecir las salidas para datos no etiquetados (Bishop, 2006).

- **Aprendizaje No Supervisado:** En el aprendizaje no supervisado, el modelo se entrena con datos no etiquetados, es decir, no se conoce la salida esperada para cada entrada. El objetivo principal en este caso es descubrir patrones subyacentes o estructuras en los datos (Hastie et al., 2009).
- **Aprendizaje por Refuerzo:** El aprendizaje por refuerzo es un tipo de aprendizaje en el que un agente interactúa con su entorno y aprende a tomar decisiones a través de recompensas y penalizaciones. El objetivo del agente es maximizar la recompensa acumulada a lo largo del tiempo (Sutton & Barto, 2018).

3.1.2. Tipos de Modelos en Machine Learning

Existen diferentes modelos que se utilizan en el aprendizaje automático. Cada uno tiene aplicaciones específicas según el tipo de problema y los datos disponibles.

- **Árboles de Decisión:** Los árboles de decisión son un modelo supervisado que divide los datos en subconjuntos basados en las características que más contribuyen a reducir la incertidumbre en la predicción. Son fáciles de interpretar y se pueden usar tanto para problemas de clasificación como de regresión (Breiman et al., 1986).
- **Boosting:** El boosting es una técnica que combina varios modelos débiles (por lo general, árboles de decisión) para formar un modelo fuerte. Cada modelo sucesivo en el conjunto es entrenado para corregir los errores cometidos por el modelo anterior, lo que permite mejorar la precisión global (Freund & Schapire, 1997).
- **Redes Neuronales:** Las redes neuronales son modelos de Machine Learning inspirados en la estructura del cerebro humano. Se componen de múltiples capas de neuronas que procesan los datos de forma jerárquica. Son especialmente efectivas para aprender

patrones complejos y no lineales en grandes volúmenes de datos (Goodfellow et al., 2016).

3.1.3. Hiperparámetros en modelos de Machine Learning

En machine learning, los hiperparámetros son configuraciones externas que controlan el proceso de entrenamiento de los modelos. A diferencia de los parámetros aprendidos durante el entrenamiento, los hiperparámetros se definen antes de entrenar el modelo y afectan significativamente su desempeño (Bergstra & Bengio, 2012).

Estos hiperparámetros influyen en la capacidad del modelo para aprender y generalizar, por lo que su selección es muy importante.

3.1.4. Métodos de ajuste de hiperparámetros

El ajuste de hiperparámetros es el proceso de optimizar estos valores para mejorar el rendimiento del modelo. Las principales estrategias incluyen:

- **Búsqueda en cuadrícula (Grid Search):** Evalúa todas las combinaciones posibles de hiperparámetros dentro de un espacio predefinido (Hutter, Hoos, & Leyton-Brown, 2011).
- **Búsqueda aleatoria (Random Search):** Selecciona aleatoriamente combinaciones de hiperparámetros, lo que puede ser más eficiente que la búsqueda exhaustiva (Bergstra & Bengio, 2012).
- **Optimización Bayesiana:** Utiliza modelos probabilísticos para seleccionar de manera más eficiente los hiperparámetros más prometedores (Snoek, Larochelle, & Adams, 2012).
- **Validación cruzada:** Divide el conjunto de datos en subconjuntos y entrena el modelo en diferentes combinaciones para evitar el sobreajuste y obtener una estimación más precisa del rendimiento (Kohavi, 1995).

3.1.5. Métricas de rendimiento en modelos de Machine Learning

Es crucial evaluar el rendimiento de los modelos mediante métricas de error. Estas métricas permiten medir la precisión y la capacidad de generalización del modelo en datos no vistos. Las tres métricas empleadas durante el desarrollo de este proyecto corresponden a:

- **Error Absoluto Medio (MAE)**

El MAE mide la media de los errores absolutos entre las predicciones del modelo y los valores reales, sin tener en cuenta la dirección de los errores (ya que se toma el valor absoluto). Su fórmula es:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

- **Raíz del Error Cuadrático Medio (RMSE)**

El RMSE es otra métrica de error ampliamente utilizada, que calcula la raíz cuadrada de la media de los cuadrados de los errores. Es una medida que da más peso a los errores grandes debido a que eleva al cuadrado las diferencias entre las predicciones y los valores reales. Su fórmula es:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

- **Porcentaje Absoluto Medio de Error (MAPE)**

El MAPE mide el error promedio como un porcentaje de los valores reales. Es una métrica útil cuando se quiere obtener una medida de error relativa, ya que no solo mide la magnitud del error, sino también la relación con el valor real. Su fórmula es:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100$$

Donde para los tres casos:

y_i es el valor real

\hat{y}_i es la predicción del modelo

n es el número total de observaciones

3.2. Landing Zone

El término Landing Zone (LZ) se refiere al espacio o entorno configurado en una infraestructura de nube en el que se establece una base para la implementación de aplicaciones, la gestión de recursos y la integración de servicios de manera organizada y segura. En el contexto de la computación en la nube, una LZ no es simplemente una zona de recepción de datos, sino una arquitectura que facilita la automatización de procesos, la gobernanza de recursos y el cumplimiento de políticas de seguridad desde el inicio del despliegue en la nube (Amazon Web Services [AWS], 2023).

Una Landing Zone ofrece un marco estructural que permite a las organizaciones crear un entorno seguro y escalable para operar en la nube, garantizando la conformidad con los requisitos normativos y de seguridad. En este sentido, es una parte fundamental de la estrategia de adopción de la nube de muchas empresas, ya que establece las bases sobre las cuales se gestionarán los recursos y servicios a lo largo de su ciclo de vida (IBM Cloud, 2022).

Las Landing Zones incluyen componentes clave como políticas de acceso, redes seguras, configuraciones de almacenamiento y herramientas para la supervisión de recursos, facilitando tanto el control como la gestión operativa (Google Cloud, 2021). Además, permiten a las organizaciones implementar infraestructuras de nube de manera estandarizada y eficiente, sin comprometer la seguridad ni el control administrativo.

Bancolombia, como parte de su proceso de transformación digital, ha adoptado la LZ para gestionar de manera eficiente su infraestructura en la nube. De esta forma, la entidad ha ubicado sus bases de datos dentro de su Landing Zone, lo que les permite contar con un entorno organizado y estandarizado para el manejo de sus datos sensibles. Al situar las bases de datos en la LZ, la entidad asegura que se implementen políticas estrictas de seguridad, gobernanza y cumplimiento.

3.3. Uso de modelos predictivos para la predicción de pérdidas en casos de indisponibilidad

El análisis predictivo utilizando Machine Learning es una herramienta clave para las empresas que buscan comprender las implicaciones económicas de la interrupción de sus servicios, permitiéndoles tomar decisiones y medidas preventivas.

El artículo de Splunk (2024) destaca cómo las tecnologías avanzadas de Machine Learning están ayudando a las empresas a transformar sus estrategias de gestión de infraestructura. En lugar de reaccionar ante incidentes después de que ocurren, las organizaciones ahora pueden predecir y mitigar las pérdidas económicas asociadas con el downtime mediante el uso de modelos predictivos basados en ML. Las empresas que implementan este tipo de inteligencia artificial no solo reducen los costos de inactividad, sino que también mejoran su resiliencia general al poder identificar y abordar posibles fallos antes de que afecten a los usuarios finales y las operaciones comerciales (Splunk, 2024).

En el sector retail, la disponibilidad de plataformas es fundamental para maximizar las ventas. Un estudio de Smith et al. (2021) aborda la predicción de pérdidas económicas por caídas en plataformas de comercio minorista en línea. Utilizando datos sobre el volumen de ventas y el comportamiento del tráfico web, los investigadores crearon un modelo predictivo que estima las pérdidas económicas debido a la interrupción de servicios. El modelo ayuda a los minoristas a planificar sus operaciones, al proporcionar una estimación precisa de los ingresos que se perderían durante un período de inactividad y, de esta manera, optimizar sus recursos para minimizar estos impactos (Smith et al., 2021).

El estudio de OpenGear (2024), por su parte, resalta la importancia de las tecnologías predictivas en la gestión de redes, señalando que el uso de Machine Learning ha permitido a las empresas reducir el impacto económico derivado de las interrupciones de red. Los modelos de ML optimizan la infraestructura de red y permiten una respuesta automática ante fallos, lo que resulta en una mejor resiliencia organizacional y menores pérdidas económicas.

4. Metodología

Para el desarrollo de este proyecto, se llevaron a cabo las siguientes actividades, cada una diseñada para garantizar la implementación exitosa del modelo predictivo:

4.1 Configuración de entornos

Como paso inicial, se llevaron a cabo varias tareas esenciales para garantizar un entorno adecuado para el desarrollo del proyecto:

- **Instalación de software necesario:** Se instaló Python como la herramienta principal para el desarrollo del modelo predictivo y Visual Studio Code como IDE. Además, se configuraron los controladores específicos que permitieron establecer una conexión con la Landing Zone (LZ) de la compañía.
- **Gestión de permisos:** Se gestionaron y aprobaron los permisos requeridos para el acceso a la LZ, lo que facilitó la integración con las bases de datos y la extracción de datos históricos relevantes para el proyecto. Se concedió acceso a las bases de datos específicas, que contenían datos históricos de transacciones, incluidas variables clave como el tipo de transacción, las comisiones generadas, y la fecha y hora de ejecución.
- **Herramientas de visualización:** Se instaló Power BI, una herramienta clave para el análisis y la presentación de los resultados del modelo predictivo, permitiendo una representación clara de las predicciones.

Estas tareas iniciales fueron fundamentales para establecer un entorno de trabajo funcional y garantizar el acceso a los recursos críticos para el desarrollo del proyecto.

4.2 Recolección y análisis de datos históricos

Durante esta etapa, se implementaron una serie de actividades clave para garantizar una extracción, transformación y análisis de datos:

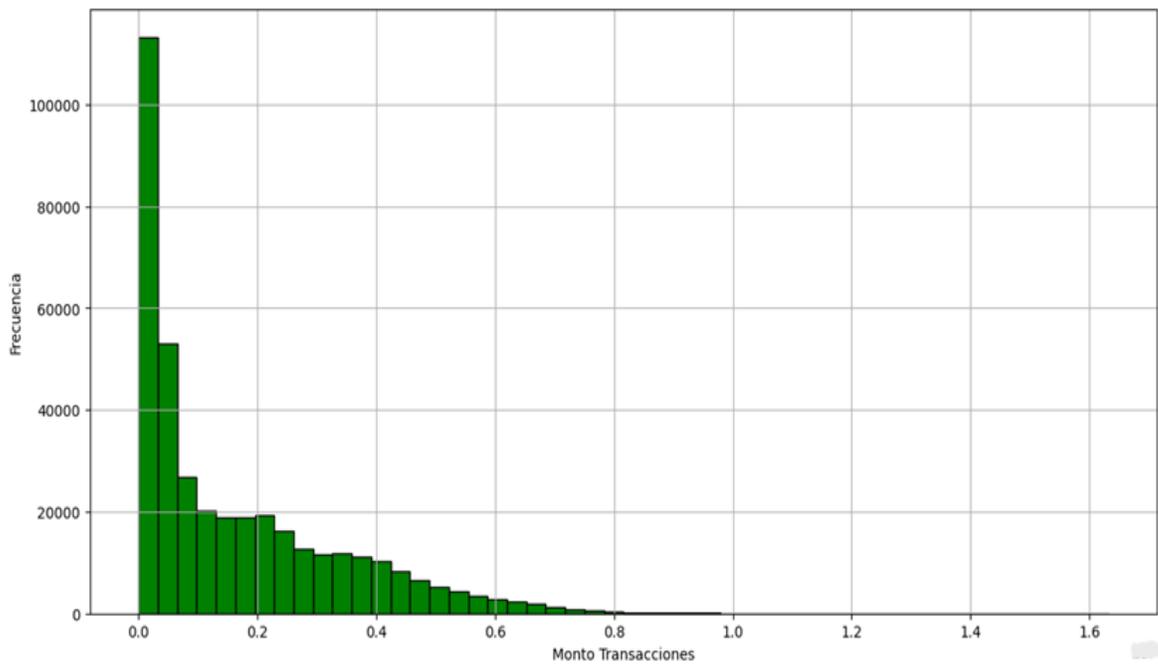
- **Identificación de variables clave:** Se seleccionaron las variables más influyentes relacionadas con las transacciones bancarias, tales como el tipo de transacción, monto, comisiones, y los momentos de ejecución (fecha y hora), con el objetivo de analizar su comportamiento e impacto en las pérdidas monetarias.
- **Construcción de procesos ETL:** Para la extracción y transformación de los datos, se desarrollaron procesos ETL (Extracción, Transformación y Carga) utilizando el motor de consultas SQL **Impala**. Estos procesos permitieron manejar grandes volúmenes de datos, unificando información proveniente de diversas bases de datos dentro de la Landing Zone.
- **Cruce de tablas:** Se llevaron a cabo cruces entre dos tablas dentro del ecosistema de datos para integrar información clave. Por ejemplo, se vincularon registros de transacciones con sus correspondientes comisiones y tiempos de procesamiento. Este trabajo requirió construir consultas avanzadas en SQL sobre Impala, lo que permitió consolidar la información en un formato adecuado para los requerimientos del proyecto.
- **Validación y limpieza de datos:** Se revisaron los datos para identificar valores nulos, duplicados o fuera de rango. Posteriormente, se implementaron estrategias de limpieza para garantizar la calidad y consistencia de los datos procesados.

- **Análisis exploratorio de datos:** Una vez obtenida la tabla de datos consolidada, se realizó un análisis estadístico preliminar. Esto incluyó la generación de histogramas y visualizaciones para identificar patrones y relaciones entre las variables seleccionadas.

Los datos procesados corresponden a un histórico desde enero del año 2019 hasta el mes de septiembre del 2024.

Figura 1

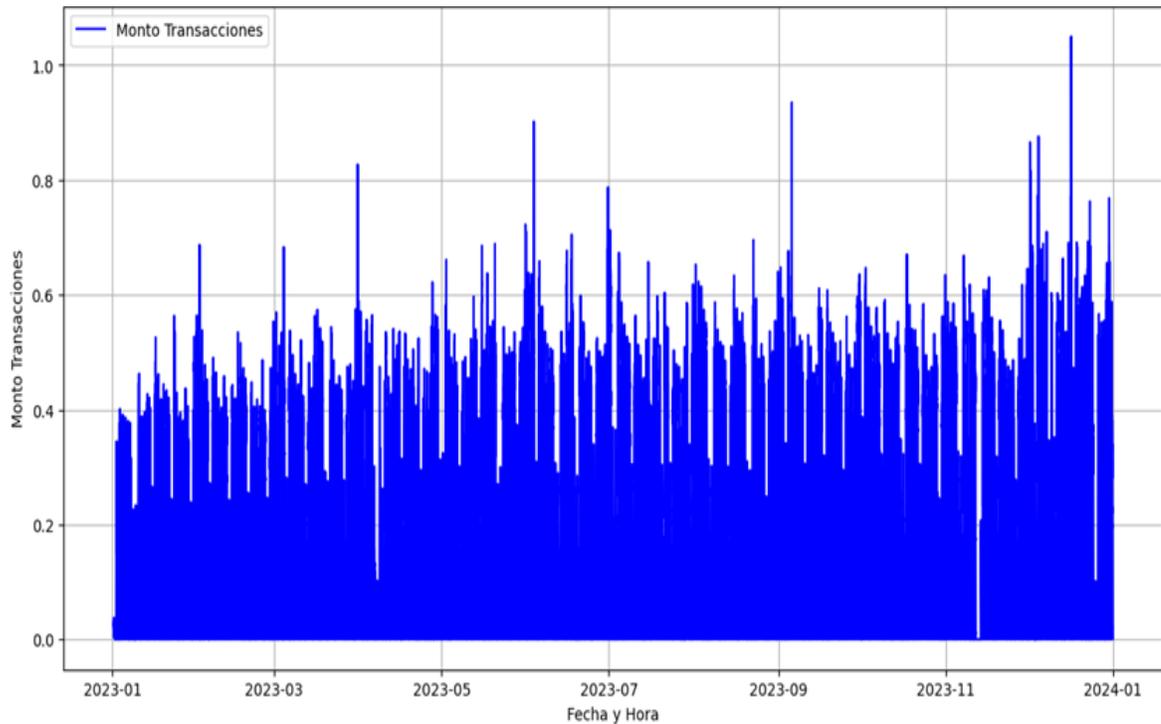
Histograma de datos históricos de comportamiento de comisiones.



Nota. Los valores monetarios en esta gráfica han sido censurados por razones de confidencialidad. Las unidades han sido omitidas deliberadamente.

Figura 2

Comportamiento histórico de transacciones para el año 2023.



Nota. Los valores monetarios en esta gráfica han sido censurados por razones de confidencialidad. Las unidades han sido omitidas deliberadamente.

En ningún momento, para la realización de este proyecto, se manipularon datos sensibles de clientes como números de cédula, números de cuenta, saldos, entre otros.

4.3 Implementación y selección de modelos predictivos

En esta etapa, el enfoque principal fue desarrollar, evaluar y comparar distintos modelos de aprendizaje automático supervisado para la predicción de las ganancias asociadas a las transacciones. Las actividades se llevaron a cabo de la siguiente manera:

- **Preparación del conjunto de datos:** Las variables seleccionadas para la implementación del modelo se presentan en la siguiente tabla:

Tabla 1*Variables seleccionadas para la implementación del modelo*

Nombre de la variable	Descripción	Tipo de variable
year	Año en que se realizan las transacciones.	Input
month	Mes en que se realizan las transacciones.	Input
day	Día en que se realizan las transacciones.	Input
hora	Hora en que se realizan las transacciones.	Input
minuto	Minuto en que se realizan las transacciones.	Input
mnt_trn	Suma del monto total de comisiones que se generan por fecha y hora.	Output

El conjunto de datos procesado se tomó desde enero del 2019 hasta julio del 2024 para entrenamiento y los meses de agosto y septiembre para prueba, siendo estos dos últimos datos completamente aislados de los datos de entrenamiento para evitar cualquier tipo de interferencia con los resultados de validación.

Además, se realizaron transformaciones adicionales como la codificación de variables categóricas para llevar los datos a una línea de tiempo que el modelo pudiera comprender. Para esto se empleó la librería `datetime` de Python, la cual permite dar formato de fecha a los valores dentro de un dataset.

Figura 3*Columnas del dataset inicial de entrenamiento.*

	datetime	mnt_trn	year	month	day	hora	minuto
0	2019-01-01 06:05:00		2019	1	1	6	5
1	2019-01-01 06:10:00	0.0	2019	1	1	6	10
2	2019-01-01 06:15:00	0.0	2019	1	1	6	15
3	2019-01-01 06:20:00	0.0	2019	1	1	6	20
4	2019-01-01 06:25:00	0.0	2019	1	1	6	25

Nota. Los valores monetarios en esta figura han sido censurados por razones de confidencialidad.

-
- **Selección de modelos:** Al tratarse de un problema de regresión, se seleccionaron tres modelos predictivos comúnmente utilizados en este tipo de problemas y en la predicción de comportamientos en el área financiera. La elección de estos tres modelos fue basada en la librería PyCaret de Python, una herramienta que automatiza el flujo de trabajo en Machine Learning. Esta herramienta facilita la selección de los mejores modelos para un conjunto de datos específico a través de un proceso de comparación automatizada. Al usar su función `compare_models()`, la librería entrenó y evaluó múltiples algoritmos de Machine Learning sobre los datos proporcionados, clasificando los modelos según su rendimiento en métricas predefinidas. Lo que arrojó como los tres modelos con mejor desempeño a:
 - **Random Forest:** Modelo basado en árboles de decisión que utiliza múltiples árboles para realizar predicciones robustas, útil para manejar grandes volúmenes de datos y evitar sobreajustes.
 - **XGBoost:** Modelo de boosting eficiente y rápido, diseñado para ofrecer alta precisión al combinar múltiples árboles débiles mediante optimización.
 - **CatBoost:** Este modelo de boosting maneja datos categóricos de manera muy eficiente.

 - **Entrenamiento y ajuste inicial:** Se configuraron parámetros base para cada modelo y se entrenaron utilizando Python y bibliotecas como Scikit-learn, XGBoost y CatBoost.

 - **Evaluación de rendimiento y selección del mejor modelo:**

Los modelos fueron evaluados utilizando las siguientes métricas de rendimiento:

 - **Error Porcentual Absoluto Medio (MAPE):** Evalúa el error porcentual entre los valores reales y predichos.
 - **Error Absoluto Medio (MAE):** Mide el promedio de las diferencias absolutas entre predicciones y valores reales.
 - **Raíz del Error Cuadrático Medio (RMSE):** Proporciona una visión más sensible a errores grandes al elevar las diferencias al cuadrado.

A continuación, se observan los resultados durante los 45 casos de evaluación de los tres algoritmos estudiados.

Tabla 2

MAPE en la comparación de algoritmos en fechas aleatorias

Algoritmo	MAPE Mínimo	MAPE Máximo	MAPE Promedio
Random Forest	3.4%	33.75%	18.57%
XG Boost	0.92%	25.48%	13.2%
CatBoost	0.17%	19.75%	9.96%

Tabla 3

Desempeño de los algoritmos durante las pruebas basados en el error MAPE

Algoritmo	Mejor resultado*	% Mejor resultado*
Random Forest	3	6.67%
XG Boost	9	20%
CatBoost	33	73.33%

Tabla 4

Desempeño de los algoritmos durante las pruebas basados en el error MAE

Algoritmo	Mejor resultado*	% Mejor resultado*
Random Forest	3	6.67%
XG Boost	10	22.22%
CatBoost	32	71.11%

Tabla 5*Desempeño de los algoritmos durante las pruebas basados en el error RMSE*

Algoritmo	Mejor resultado*	% Mejor resultado*
Random Forest	3	6.67%
XG Boost	10	22.22%
CatBoost	32	71.11%

*Mejor resultado: Indica el número de veces que el modelo obtuvo el menor error comparado con los otros modelos evaluados.

*% Mejor resultado: Indica de forma porcentual el número de veces que el modelo obtuvo el mejor resultado con respecto al total de veces que se ejecutó la prueba.

Al analizar el MAPE promedio en conjunto con el porcentaje de veces que se obtuvo el mejor resultado, se puede observar una correlación clara entre ambos factores:

- CatBoost tiene tanto el MAPE promedio más bajo (9.96%) como el mayor porcentaje de mejores resultados, lo que sugiere que es el modelo más consistente y preciso en las pruebas evaluadas.
- XG Boost, aunque su MAPE promedio (13.2%) es más alto que el de CatBoost, también tiene un rendimiento bueno, alcanzando el mejor resultado en un alto porcentaje de los casos.
- Random Forest tiene el MAPE promedio más alto (18.57%) y el menor porcentaje de mejores resultados, lo que indica que, en general, este algoritmo es el menos preciso y consistente en comparación con los otros dos.

En cuanto al desempeño global, considerando tanto la precisión como la consistencia, el algoritmo CatBoost ha mostrado los mejores resultados. Dado que el objetivo es obtener el modelo más preciso posible, se decidió continuar trabajando con este algoritmo.

4.4 Refinamiento del modelo seleccionado

Una vez seleccionado el modelo CatBoost como el más eficiente, se procedió a su refinamiento con el objetivo de optimizar su rendimiento y garantizar su robustez y precisión. Las actividades desarrolladas fueron las siguientes:

- **Ajuste de hiperparámetros:** Se implementaron técnicas de ajuste fino mediante búsqueda aleatoria (Random Search) para encontrar la combinación óptima de hiperparámetros, tales como:
 - Learning rate (tasa de aprendizaje)
 - Depth (profundidad de los árboles)
 - L2 regularization (regularización)
 - Número de iteraciones
 - Eval metric (métrica de evaluación)

Estas optimizaciones permitieron mejorar el balance entre sesgo y varianza, evitando problemas de sobreajuste o subajuste.

- **Incorporación de variables categóricas adicionales:** Para aumentar la robustez del modelo y capturar patrones adicionales en las transacciones, se agregaron nuevas variables categóricas, tales como:
 - Fines de semana
 - Festivos
 - Fechas especiales (como Día de la Madre, Black Friday, etc.)
 - Quincenas
 - Fin de año
 - Día de la semana

Esto se hizo por medio de la librería de Python: Holidays, que contiene las fechas festivas de los países. Además, se utilizaron las fechas que el banco tiene definidas como las de mayor transaccionalidad.

Estas variables se insertaron como valores binarios, permitiendo al modelo identificar el impacto que ciertos períodos de tiempo tienen sobre las transacciones y mejorar su capacidad.

Figura 4

Columnas del dataset de entrenamiento con variables categóricas.

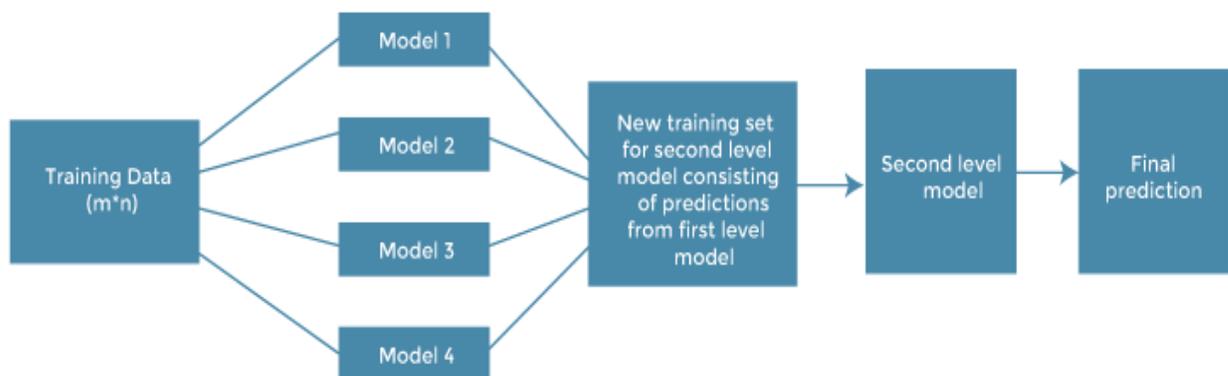
	datetime	mnt_trn	year	month	day	hora	minuto	is_holiday	\
0	2019-01-01 06:05:00		2019	1	1	6	5	0	
1	2019-01-01 06:10:00	0.0	2019	1	1	6	10	0	
2	2019-01-01 06:15:00	0.0	2019	1	1	6	15	0	
3	2019-01-01 06:20:00	0.0	2019	1	1	6	20	0	
4	2019-01-01 06:25:00	0.0	2019	1	1	6	25	0	
	is_weekend	is_quincena	day_of_week	is_special	is_eoty				
0	0	0	1	1	0				
1	0	0	1	1	0				
2	0	0	1	1	0				
3	0	0	1	1	0				
4	0	0	1	1	0				

Nota. Los valores monetarios en esta figura han sido censurados por razones de confidencialidad.

- **Implementación de un modelo de ensamble:** La metodología empleada en esta parte del proceso es la descrita en la siguiente imagen:

Figura 5

Stacking en modelos de Machine Learning



Nota: Fuente <https://www.javatpoint.com/stacking-in-machine-learning>

Se implementó un ensamble de tipo stacking, donde el modelo CatBoost refinado fue combinado con un Random Forest como meta-modelo. Este proceso consistió en: Utilizar las predicciones del modelo base Catboost entrada para el Random Forest. Y posteriormente entrenar el Random Forest con las salidas del modelo anterior para generar estimaciones más precisas. Esta técnica permitió combinar las fortalezas de ambos modelos, reduciendo el error y mejorando la capacidad general de predicción.

- **Pruebas de robustez:** Finalmente, se realizaron pruebas con fechas y horarios aleatorios, dentro del periodo definido como período de validación, así como períodos críticos como fines de semana y festivos, evaluando la capacidad del modelo para adaptarse a diferentes escenarios y mantener su precisión.

En esta etapa, se aplicó nuevamente la métrica MAPE, como en la evaluación anterior.

4.5 Automatización de la actualización del modelo

Este paso fue fundamental para asegurar que el sistema se mantuviera actualizado y eficiente. Para esto se creó un flujo automatizado que permite el reentrenamiento mensual del modelo, asegurando que las predicciones se basen siempre en datos recientes.

Para la integración del proceso con la infraestructura existente en Bancolombia, se decidió adaptar el proyecto al *Orquestador 2.0*, una plataforma interna diseñada desde la compañía que gestiona el ciclo de vida de los modelos de Machine Learning en producción y proporciona la arquitectura para mantenerlos alojados. Esta decisión implicó una serie de cambios técnicos que debían alinearse con las buenas prácticas definidas para el uso de este orquestador.

- **Modificaciones técnicas de consultas SQL:** Uno de los primeros pasos fue realizar ajustes en las consultas SQL utilizadas para extraer los datos históricos necesarios para el entrenamiento del modelo. Las consultas existentes debían ser optimizadas para garantizar

la eficiencia y el rendimiento dentro del entorno del orquestador. Esto implicó revisar las consultas para asegurarse de que pudieran ser ejecutadas de manera eficiente en el entorno de producción sin afectar el rendimiento de las bases de datos críticas. Se incorporaron mejoras en la estructuración de las consultas, la parametrización de las tablas y el uso de índices adecuados para reducir los tiempos de respuesta y evitar posibles bloqueos en la base de datos. Las consultas fueron correctamente probadas por el equipo.

- **Adaptación al Orquestador 2.0 y buenas prácticas de Python:** Una vez optimizadas las consultas, se procedió a adaptar el modelo a las especificaciones del *Orquestador 2.0*, lo que requería seguir una serie de buenas prácticas definidas por el equipo de analítica de Bancolombia, dueño del proceso. Estas prácticas incluyen la utilización de plantillas predefinidas para la automatización de cada parte del proceso, el uso de programación orientada a objetos (POO) dentro de los Scripts y la configuración adecuada de los pipelines de Machine Learning. La ejecución del proceso, desde la creación de tablas, la inserción de datos, hasta el reentrenamiento fue debidamente probada por el equipo.

Los procesos de buenas prácticas tanto de SQL como de Python debieron ser sometidos a una inspección rigurosa ejecutada mediante un Pipeline de validación entregado por el área de analítica del banco.

- **Fases de Pruebas: Desarrollo, QA y Despliegue en Producción:** El proceso de automatización fue probado en diversas fases antes de su despliegue final. En primer lugar, se realizó una fase de pruebas en ambiente de desarrollo. Durante esta fase, se verificó que el modelo se reentrenara correctamente y que las consultas SQL se ejecutaran de manera eficiente. Además, se revisaron los logs y se realizaron pruebas de estrés para asegurarse de que el proceso de reentrenamiento no interfiriera con otras tareas críticas en el sistema.

Una vez superada esta fase, se procedió a realizar las pruebas en el ambiente QA, donde se validaron aspectos como la integración del modelo con la infraestructura del Orquestador 2.0, la correcta ejecución de las tareas programadas y la fiabilidad del flujo de trabajo automatizado.

Finalmente, tras validar el proceso en QA, el flujo automatizado fue desplegado en el entorno de producción.

Este proceso nos entrega como resultado el modelo reentrenado en un archivo con extensión .pkl que se aloja en la zona de resultados de la LZ.

4.6 Despliegue del modelo como API

Una vez completado el proceso de automatización del flujo de reentrenamiento del modelo, programado de forma mensual, el proyecto fue entregado al equipo DevOps del banco. Este equipo se encargó de llevar a cabo el proceso denominado *Productización*, el cual consiste en convertir el modelo con extensión .pkl que quedó alojado en la LZ en un producto final que pueda ser utilizado en un entorno de producción. Como parte de este proceso, se generó una API que permite que el modelo sea consumido de manera eficiente por otros sistemas y procesos dentro de la infraestructura del banco. De este modo, la integración del modelo de Machine Learning en los flujos operativos del banco se hace posible, garantizando que las predicciones y los datos actualizados estén disponibles de forma continua y accesible para otros equipos o aplicaciones, sin necesidad de intervención manual.

4.7 Desarrollo de un dashboard para visualización de resultados

Se desarrolló y desplegó un dashboard en Power BI que permite visualizar las predicciones del modelo. Este dashboard incluye filtros por fecha y hora, facilitando el análisis de los pronósticos, además de una sección que muestra las métricas de uso y adopción de la herramienta, como consultas totales, uso durante el día y equipos que más hacen uso de ella. Este sistema consume los datos desde la API para realizar la predicción con los valores que los usuarios ingresan en los filtros, y además toma los resultados alojados en la Landing Zone, referentes al uso y adopción de la herramienta.

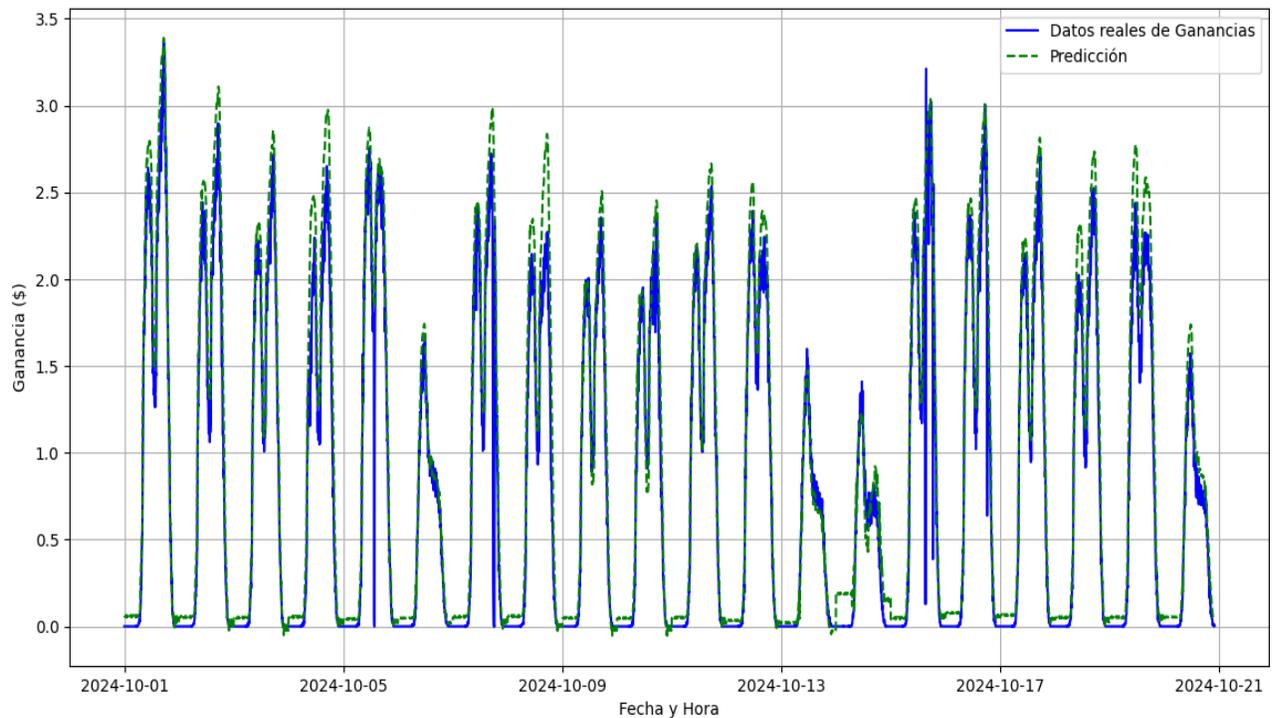
5. Análisis de resultados

5.1. Refinamiento del modelo

Tras realizar el procedimiento de refinamiento del modelo, descrito en la sección 4.4, *Refinamiento del modelo seleccionado*, se obtuvo el siguiente comportamiento en la gráfica comparativa entre datos estimados y datos reales para el mes de octubre de 2024.

Figura 6

Comparación valores reales vs predicciones del modelo para el mes de octubre de 2024



Nota. Los valores monetarios en esta gráfica han sido censurados por razones de confidencialidad. Las unidades han sido omitidas deliberadamente.

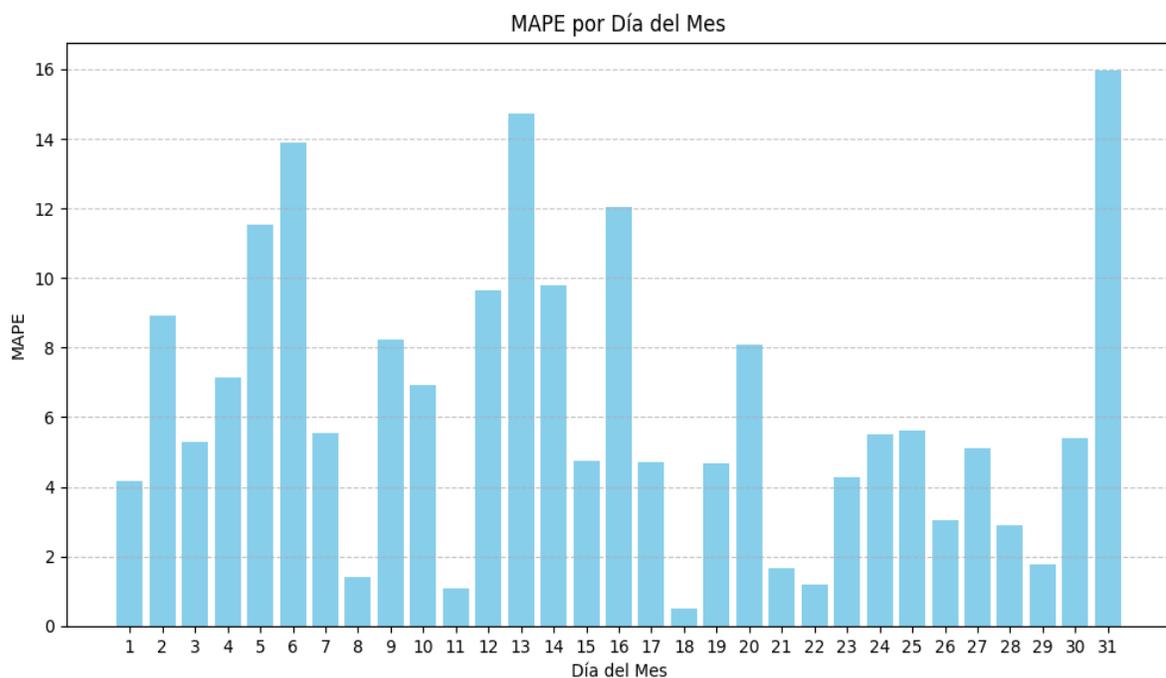
La gráfica compara el comportamiento real de las transacciones con las predicciones realizadas por el modelo. En general, el modelo ha mostrado un buen desempeño, replicando correctamente las fluctuaciones y tendencias observadas en los datos históricos. Sin embargo,

algunos picos o caídas pueden presentar pequeñas desviaciones, lo cual es común en cualquier modelo de predicción, especialmente cuando se enfrenta a datos con alta variabilidad.

Además, al graficar los MAPE correspondientes a este mismo mes de prueba, se puede observar que en ningún día el error supera el valor establecido como el límite máximo aceptable para este del sistema. Como se puede observar en la siguiente figura:

Figura 7

MAPE diario para predicciones del mes de octubre de 2024



Los resultados obtenidos en la figura anterior mostraron que, en todos los casos evaluados para este mes de prueba el error MAPE fue inferior al **25%**, lo que indica que los modelos alcanzaron una precisión superior al **75%**. Esto sugiere que el modelo está ofreciendo predicciones con un margen de error controlado.

5.2. Dashboard para consulta de resultados

El dashboard de Power BI proporciona a los usuarios una herramienta visual y dinámica para consultar las predicciones generadas por el modelo, así como para evaluar su uso y adopción a lo largo del tiempo. A través de las diferentes secciones del dashboard, los usuarios pueden obtener información detallada sobre las predicciones y los indicadores clave relacionados con la adopción y el uso del sistema.

Figura 8

Portada dashboard en Power BI para consulta de predicciones y métricas



Figura 9
Sección de predicciones del dashboard en Power BI



Nota. Los valores monetarios en esta gráfica han sido censurados por razones de confidencialidad. Las unidades han sido omitidas deliberadamente.

En esta sección del dashboard, los usuarios pueden seleccionar un rango específico de fechas y horas para generar predicciones relacionadas con el comportamiento de las transacciones. El gráfico resultante muestra las transacciones esperadas durante el período indicado, acompañado por el cálculo del monto total de ganancias estimadas. Además, se incluye una tabla de datos que presenta el registro acumulado de los montos, actualizada cada 5 minutos, lo que permite un análisis detallado de la evolución de las transacciones en tiempo real.

Con base en las predicciones de ganancias generadas por el modelo, los usuarios finales pueden estimar las pérdidas comparando las ganancias proyectadas con los valores reales obtenidos durante el mismo período. En caso de una indisponibilidad total de la aplicación, el monto de las ganancias predichas representará directamente el total de las pérdidas económicas a las que se enfrentarían, proporcionando una herramienta fundamental para evaluar los impactos de dichas afectaciones y planificar estrategias de mitigación.

Figura 10

Sección de métricas de uso y adopción del dashboard en Power BI



Esta sección del dashboard presenta cinco gráficos principales que permiten evaluar en detalle el uso y la adopción del modelo a lo largo del tiempo, así como su integración en las diferentes áreas del banco:

- **Total de consultas:** Muestra el número total de consultas realizadas al modelo desde su implementación hasta la fecha. Este gráfico proporciona una visión global del nivel de interacción con el sistema, permitiendo identificar tendencias generales en el uso.
- **Consultas por día:** Presenta la cantidad de consultas realizadas diariamente, lo que facilita observar variaciones en el uso del modelo a lo largo del día. Este análisis es útil para identificar horas pico de demanda.
- **Áreas que han utilizado el modelo:** Muestra la proporción de áreas que han interactuado con el modelo en comparación con el total de áreas del banco. Este gráfico es clave para

evaluar la adopción del modelo a nivel organizacional, permitiendo identificar las áreas que aún no lo han implementado o que lo usan con poca frecuencia.

- Número de ejecuciones por mes: Este gráfico visualiza la cantidad de ejecuciones del modelo realizadas cada mes. Permite observar la evolución de su uso en el tiempo y detectar posibles incrementos o descensos en la frecuencia de uso.
- Número de ejecuciones por área: Muestra el número de ejecuciones del modelo realizadas en cada área del banco, lo que ayuda a identificar qué departamentos están aprovechando más el modelo y cuáles podrían necesitar más apoyo o capacitación para fomentar su adopción.

6. Conclusiones y recomendaciones

El modelo predictivo desarrollado cumplió con el objetivo de estimar las ganancias proyectadas generadas por las transacciones, utilizando datos históricos sobre las comisiones. Con un MAPE inferior al 25%, el modelo demostró una alta precisión en sus predicciones, cumpliendo con los estándares de precisión establecidos.

El modelo logró capturar de manera efectiva la relación entre las variables de hora y fecha de las transacciones y las ganancias esperadas, permitiendo obtener estimaciones confiables. Estas predicciones sirven como base para que el usuario final evalúe las pérdidas económicas, comparando las ganancias proyectadas con los valores reales obtenidos durante el período correspondiente. En escenarios de indisponibilidad total de la aplicación, las ganancias proyectadas representarían el monto de las pérdidas económicas. De esta manera, el modelo se convierte en una herramienta estratégica para la toma de decisiones, permitiendo a la entidad bancaria identificar afectaciones económicas, planificar estrategias y mitigar riesgos operativos.

Se hizo uso del proceso y la arquitectura que el banco ya poseía para alojar este tipo de proyectos de aprendizaje automático, asegurando que el modelo pudiera manejar grandes volúmenes de datos sin comprometer el rendimiento. La integración con la Landing Zone de la entidad bancaria ha permitido que el modelo funcione de manera eficiente y pueda actualizarse con facilidad a medida que se generan nuevos datos.

Se estableció un proceso mensual para la actualización y ajuste del modelo. Incluyendo en él, la recolección de nuevos datos y el reentrenamiento del modelo, lo que asegura que las predicciones se mantengan alineadas con las condiciones actuales del negocio. Aunque se ha logrado una alta precisión en las predicciones actuales, es necesario seguir evaluando y ajustando el modelo en función de los nuevos patrones de datos que puedan surgir.

El dashboard interactivo fue exitosamente implementado, permitiendo visualizar las predicciones de manera clara y accesible para los usuarios. Este dashboard proporciona una herramienta flexible para la interpretación de los resultados del modelo y su uso, facilitando la toma de decisiones basadas en las predicciones.

Aunque el modelo ha demostrado ser confiable, se recomienda explorar la inclusión de otros factores que puedan influir en el comportamiento de las transacciones, tales como factores externos (como cambios en las políticas del banco, eventos económicos imprevistos, etc.) o comportamientos del cliente que puedan surgir en el tiempo. La incorporación de estas variables podría mejorar la precisión de las predicciones, especialmente en escenarios atípicos.

Actualmente, el modelo se enfoca en estimar las ganancias proyectadas generadas por el canal de Bancolombia a la mano, lo que permite a los usuarios finales evaluar las pérdidas derivadas de la indisponibilidad de sus aplicaciones transaccionales. Sin embargo, se recomienda explorar la posibilidad de ampliar el alcance del modelo para incluir otros canales y tipos de transacciones que estos posean.

Departamento de Ingeniería Electrónica y de Telecomunicaciones

Evaluación de impactos monetarios ante afectaciones tecnológicas: un enfoque predictivo basado en Machine Learning



UNIVERSIDAD DE ANTIOQUIA

Facultad de Ingeniería

PRACTICANTE: María Salomé Garcés Montero

PROGRAMA: Ingeniería Electrónica

ASESOR: Sebastián Isaza Ramírez

Semestre de la práctica: 2024-2



Introducción

Contexto

En un entorno con altos riesgos tecnológicos, la indisponibilidad de aplicaciones transaccionales puede generar pérdidas económicas y dañar la reputación de la entidad. Este riesgo se intensifica en épocas de alta demanda, como en pagos de nómina o eventos comerciales.

Con este proyecto se busca desarrollar un modelo predictivo utilizando Machine Learning, que sirva para cuantificar las pérdidas económicas derivadas de la caída de una de aplicaciones clave de Bancolombia. Además de ayudar a estimar los impactos, el modelo se convertirá en una herramienta estratégica para la planificación y gestión de riesgos, mejorando la capacidad de respuesta y optimización de recursos de la entidad.



Metodología

- 1. Recolección y análisis de datos históricos:** Se recopilan datos históricos sobre transacciones y comisiones generadas por cada una, identificando patrones y correlaciones clave para el modelo predictivo.
- 2. Implementación y selección de modelos predictivos:** Se exploran y prueban tres diferentes modelos de Machine Learning para encontrar el más adecuado para predecir las pérdidas. Tras varias pruebas se elige el modelo CatBoost por su precisión.
- 3. Refinamiento del modelo seleccionado:** Se ajustan los hiperparámetros y la estructura del modelo elegido para mejorar su precisión y capacidad predictiva. Creando un ensamble con Catboost
- 4. Automatización del reentrenamiento:** El modelo refinado se automatiza y se prepara para su integración en los sistemas del banco, garantizando su eficiencia en tiempo real.

5. Despliegue del modelo como API: El modelo se implementa como una API, permitiendo su acceso y utilización desde diferentes aplicaciones y plataformas internas de Bancolombia.

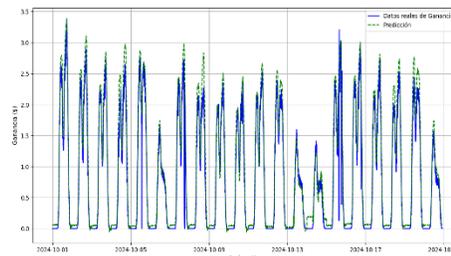
6. Desarrollo de un dashboard para visualización de predicciones: Se crea un dashboard interactivo en Power BI para visualizar las predicciones y resultados del modelo, facilitando la toma de decisiones estratégicas en tiempo real.



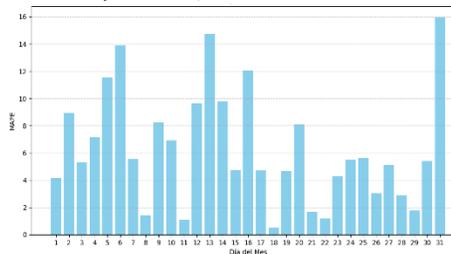
Resultados

Después de realizado el refinamiento del modelo se obtuvieron los siguientes resultados:

Predicciones vs datos reales. Octubre 2024



Porcentaje de error por día del mes. Octubre 2024



Dashboard para la predicción de comisiones.



Objetivos

- Desarrollar un modelo predictivo en Python que estime las ganancias generadas por la aplicación bancaria en función de la hora y fecha, usando datos históricos sobre las comisiones generadas por las transacciones.
- Implementar una arquitectura para alojar y gestionar el modelo predictivo, garantizando que pueda manejar grandes volúmenes de datos y adaptarse a las necesidades del negocio.
- Establecer un proceso continuo para la actualización del modelo predictivo, incluyendo la recolección de nuevos datos, la reevaluación y ajuste del modelo, con el fin de garantizar que las predicciones se mantengan alineadas con las condiciones actuales.
- Construir un dashboard interactivo y escalable para la visualización de las predicciones, facilitando la interpretación y el análisis de resultados.

Conclusiones

- El modelo predictivo estimó con éxito las ganancias económicas por la indisponibilidad de la aplicación transaccional, logrando un MAPE inferior al 25%, lo que cumple con los estándares de precisión establecidos.
- El modelo se integró de manera efectiva con la infraestructura de Bancolombia, lo que permitió manejar grandes volúmenes de datos y facilitar futuras actualizaciones.
- Se estableció un proceso mensual de actualización y ajuste del modelo, asegurando que las predicciones sigan alineadas con los datos actuales del negocio y adaptándose a nuevos patrones.
- El dashboard implementado permite visualizar las predicciones de manera clara y accesible.

DATOS DE CONTACTO DEL AUTOR:



3123359998



+57 3123359998



msalome.garces@udea.edu.co



salomegarces



http://ca.linkedin.com/in/salomegarcesmontero

Referencias

- Alpaydin, E. (2014). *Introducción al aprendizaje automático* (2.^a ed.). MIT Press. <https://mitpress.mit.edu/books/introduction-machine-learning>.
- Amazon Web Services. (2023). *Landing Zone: Implementación de entornos en la nube segura y escalable*. <https://aws.amazon.com/landing-zone>
- IBM Cloud. (2022). *La zona de aterrizaje: Cómo diseñar y gestionar tu infraestructura en la nube*. <https://www.ibm.com/cloud/landing-zone>
- Google Cloud. (2021). *Guía de Landing Zone en Google Cloud*. Recuperado de <https://cloud.google.com/landing-zone>
- Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer. Springer. <https://www.springer.com/gp/book/9780387310732>
- Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1986). *Classification and regression trees*. Wadsworth & Brooks/Cole. <https://link.springer.com/book/10.1007/978-1-4612-6239-9>
- Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *European Conference on Computational Learning Theory*, 23-37. ResearchGate. https://www.researchgate.net/publication/220705290_A_Decision-Theoretic_Generalization_of_On-Line_Learning_and_an_Application_to_Boosting
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. <https://mitpress.mit.edu/books/deep-learning>
- Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3), 210-229
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2.^a ed.). Springer. <https://link.springer.com/book/10.1007/978-0-387-84858-7>
- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (2.^a ed.). MIT Press. <https://mitpress.mit.edu/books/reinforcement-learning>
- Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13, 281-305. <http://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf>
- Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2011). Sequential model-based optimization for general algorithm configuration. In *LION 5: Proceedings of the 5th International*

-
- Conference on Learning and Intelligent Optimization* (pp. 507-523). Springer. https://link.springer.com/chapter/10.1007/978-3-642-25566-3_48
- Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems* (pp. 2951-2959). <https://papers.nips.cc/paper/2012/hash/36c8f85c3bbf1e8d3daed3358318fbc0-Abstract.html>
- Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International Joint Conference on Artificial Intelligence* (Vol. 14, pp. 1137-1143). <https://www.ijcai.org/Proceedings/95/Papers/0027.pdf>
- Splunk. (2024, diciembre 5). *Splunk report shows downtime costs global 2000 companies \$400 billion annually*. Splunk. https://www.splunk.com/en_us/newsroom/press-releases/2024/conf24-splunk-report-shows-downtime-costs-global-2000-companies-400-billion-annually.html
- Smith, A., Jones, L., & Patel, R. (2021). *Predicting business downtime costs: Machine learning models for estimating lost revenue*. *Journal of Retailing and Consumer Services*. <https://www.journals.elsevier.com/journal-of-retailing-and-consumer-services>
- OpenGear. (2024). *Measuring the true cost of network outages*. OpenGear. <https://opengear.com/white-paper/measuring-the-true-cost-of-network-outages>

