

# PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

## Escuela de Posgrado



Predicción de la aceptación de pedidos por parte de los repartidores en la industria de entregas a domicilio utilizando machine learning

Trabajo de investigación para obtener el grado académico de Maestro en Informática con mención en Ciencias de la Computación que presenta:

***Jorge Brian Alarcon Flores***

Asesor:

***Dr. César Armando Beltrán Castañón***

Lima, 2024


## Informe de Similitud

Yo, **César Armando BELTRÁN CASTAÑÓN**, docente de la Escuela de Posgrado de la Pontificia Universidad Católica del Perú, asesor de el trabajo de investigación titulado “Predicción de la aceptación de pedidos por parte de los repartidores en la industria de entregas a domicilio utilizando machine learning” de el autor **Jorge Brian ALARCON FLORES**, deajo constancia de lo siguiente:

- El mencionado documento tiene un índice de puntuación de similitud de 11%. Así lo consigna el reporte de similitud emitido por el software *Turnitin* el 18/06/2024.
- He revisado con detalle dicho reporte y el trabajo de investigación, y no se advierte indicios de plagio.
- Las citas a otros autores y sus respectivas referencias cumplen con las pautas académicas.

Lugar y fecha:

San Miguel, 18 de Junio de 2024.

Apellidos y nombres del asesor / de la asesora: <u>BELTRÁN CASTAÑÓN, César Armando</u>	
DNI: 29561260	Firma 
ORCID: 0000-0002-0173-4140	

## Resumen

La industria de entregas a domicilio ha experimentado un auge significativo debido a la creciente demanda de los consumidores que buscan la comodidad de recibir productos y alimentos directamente en sus hogares. El avance de tecnologías y aplicaciones móviles ha impulsado el crecimiento de este mercado, permitiéndole adaptarse a las preferencias cambiantes de los consumidores [10] [19]. Sin embargo, un componente crítico en este proceso son los repartidores, quienes, tras la realización de un pedido por parte del cliente en la plataforma de la empresa, reciben notificaciones que les ofrecen una serie de pedidos sugeridos. Si aceptan, asumen la responsabilidad de recoger y entregar el pedido a los consumidores, así como la ganancia asociada, pero en ocasiones, los repartidores pueden declinar la aceptación de un pedido, lo que potencialmente conlleva a retrasos en la entrega, generando experiencias insatisfactorias para los usuarios. Este aspecto se presenta como un desafío significativo en la optimización de las operaciones de entrega a domicilio, el cual puede abordarse con soluciones de aprendizaje de máquina.

En este artículo se presentan los resultados de la experimentación realizada con diversos modelos de aprendizaje de máquina, aplicándose la técnica de balanceo Smartly OverSampling con SMOTE. Los modelos se aplicaron a un conjunto de datos proporcionado por una institución latinoamericana líder en el sector de entregas a domicilio, reportando el algoritmo LightGBM, los mejores resultados con un AUC de 0.88 y un Average Precision Recall de 0.47.



# ÍNDICE

	<b>Pág</b>
Resumen	iii
Índice	iv
Lista de Tabla	vi
Lista de Figuras	vii
Introducción	8
<b>2. MARCO TEÓRICO</b>	<b>9</b>
2.1. Industria de entregas a domicilio	9
2.2. Machine Learning	9
2.3. Aprendizaje Supervisado	9
2.4. LightGBM	10
<b>3. OBJETIVO</b>	<b>10</b>
<b>4. DATA Y METODOLOGÍA</b>	<b>10</b>
4.1. Descripción del conjunto de datos	10
4.2. Limpieza de data	10
4.3. Partición de la data	11
4.4. Entrenamiento del modelo	11
4.5. Métricas de validación	12
4.6. Estrategia de interpretación de resultados	12
<b>5. RESULTADOS</b>	<b>12</b>
5.1. Exploración inicial	12

5.2.	Modelamiento final	13
5.3.	Interpretación de resultados	14
5.3.1.	Análisis exploratorio	14
5.3.2.	Importancia de variables del modelo	15
<b>6.</b>	<b>DISCUSIÓN</b>	<b>15</b>
<b>7.</b>	<b>CONCLUSIONES</b>	<b>16</b>
	Referencias Bibliográficas	16



## ÍNDICE DE TABLAS

Tabla 1: Descripción de variables originales	10
Tabla 2: Comparación de resultados de modelos iniciales	12



## ÍNDICE DE FIGURAS

Figura 1: Diagrama de flujo de asignación de pedidos a los repartidores	8
Figura 2: Distribución original de variables to_user_distance, tip y total_earning	11
Figura 3: Distribución sin outliers de variables to_user_distance, tip y total_earning	11
Figura 4: Curva ROC modelo Extreme Gradient Boosting inicial	13
Figura 5: Matriz de Confusión modelo Extreme Gradient Boosting inicial sobre la data de prueba	13
Figura 6: Gráfica de distribución de predicciones según categorías del target – modelo Extreme Gradient Boosting inicial	13
Figura 7: Curva ROC modelo LightGBM con SMOTE	13
Figura 8: Matriz de Confusión modelo LightGBM con SMOTE sobre la data de prueba	14
Figura 9: Gráfica de distribución de predicciones según categorías del target – modelo LightGBM con SMOTE	14
Figura 10: Distribución de órdenes aceptadas y rechazadas por día de semana	14
Figura 11: Ratio de aceptación de órdenes por día de semana	14
Figura 12: Gráfica de Shap Values – Modelo LightGBM con SMOTE	15
Figura 13: Gráfica de distribución de probabilidades de aceptaciones de órdenes sin y con propinas incluidas	15

# Predicción de la aceptación de pedidos por parte de los repartidores en la industria de entregas a domicilio utilizando machine learning

Alarcon Flores, Jorge Brian

Pontificia Universidad Católica del Perú, Maestría en informática, Lima, Perú.

E-mail: [brian.alarcon@pucp.edu.pe](mailto:brian.alarcon@pucp.edu.pe)

**Resumen**— La aceptación de pedidos por parte de los repartidores en la industria de entregas a domicilio requiere un análisis y representación muy exhaustiva debido a la importancia que tiene este proceso en la cadena de valor de la industria y en la experiencia del cliente, en este trabajo de investigación se analiza el comportamiento de aceptación de un pedido a través de técnicas supervisadas de machine learning, siendo LightGBM, el algoritmo que presentó los mejores resultados con un AUC de 0.88 y un Average Precision Recall de 0.47.

**Palabras clave:** Aprendizaje automático, entregas a domicilio, aprendizaje supervisado.

## 1. Introducción

El problema de la aceptación de pedidos por parte de los repartidores en la industria de entregas a domicilio se refiere a la dificultad que tienen las empresas de entregas en asignar los pedidos de forma eficiente a los repartidores [14]. En muchos casos, los repartidores tienen la opción de aceptar o rechazar un pedido, tal como se observa en la Figura 1, lo que puede generar problemas como la sobrecarga de trabajo para algunos repartidores y la subutilización de recursos para otros. Además, la aceptación de pedidos también puede verse afectada por otros factores, como el tráfico, la ubicación del pedido y la hora del día. Esto hace que sea difícil para las empresas de entregas prever con exactitud cuántos repartidores necesitan para satisfacer la demanda de pedidos y cómo asignar los pedidos de manera justa y equitativa entre los repartidores.

Este problema puede llevar a una experiencia de entrega deficiente para el cliente, ya sea por retrasos en la entrega o por errores en el pedido. También puede generar insatisfacción entre los repartidores, lo que puede llevar a una alta rotación de personal y una disminución en la calidad de servicio.

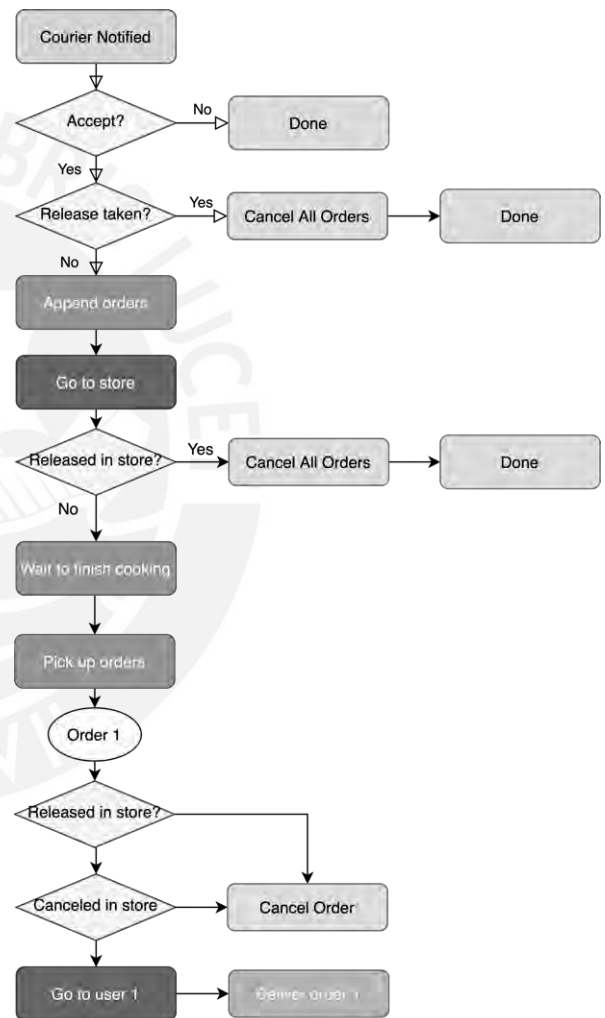


Fig. 1 Diagrama de flujo de asignación de pedidos a los repartidores

En resumen, el problema a resolver en el presente proyecto puede ser expresado con la siguiente pregunta-problema: ¿Cómo predecir la aceptación de pedidos por repartidores en la industria de entregas a domicilio? y ¿Cuáles son los factores que influyen?, a través del desarrollo de un modelo predictivo de machine learning.



## 2. Marco Teórico

### 2.1. Industria de entregas a domicilio

Un sistema de entrega a domicilio implica que los clientes adquieran productos, ya sea comida u otros, mediante una aplicación móvil. Esta aplicación muestra el costo adicional por el servicio de entrega y el tiempo estimado para recibir los productos. Este tipo de servicio está ampliamente extendido en diversos sectores comerciales, como farmacias, supermercados y bodegas, así como en grandes empresas o servicios de mensajería con plazos definidos [5].

Los principales actores involucrados en este servicio son el cliente final, la empresa encargada de la logística de entrega, los repartidores y los establecimientos asociados. En este servicio de entrega a domicilio, la interacción entre estos actores es fundamental para su funcionamiento efectivo. El cliente final inicia el proceso al realizar un pedido a través de la plataforma de la empresa de logística de entrega. Esta empresa, a su vez, coordina la logística y la distribución del pedido, asignándolo a uno de los repartidores disponibles. Los repartidores, quienes son los encargados de llevar los productos al destino final, juegan un papel crucial al garantizar la entrega oportuna y segura ya que tienen la potestad de aceptar o no un pedido asignado. Finalmente, los establecimientos asociados, como restaurantes o tiendas, proporcionan los productos solicitados y colaboran estrechamente con la empresa de logística para garantizar la disponibilidad y calidad de los productos entregados. En conjunto, estos actores trabajan en sincronía para satisfacer las necesidades del cliente final y ofrecer un servicio de entrega eficiente y satisfactorio.

La industria de entregas a domicilio tuvo su auge, tras la pandemia del COVID-19 [1], ya que los patrones de consumo en la mayoría de los países confinados experimentaron cambios significativos en sus hábitos de compra. Se observaron comportamientos como el acaparamiento impulsivo de productos básicos, la adaptación para hacer frente a la situación adversa, la acumulación de demanda en ciertos sectores debido a restricciones y preocupaciones económicas, y un aumento en la adopción de tecnología para actividades diarias como educación, trabajo y salud [16]. Estos cambios reflejaron una reevaluación de las prioridades de consumo, dando mayor importancia a alimentos y medicamentos en este contexto de crisis sanitaria, lo

cual crecimiento económico de la industria de entregas a domicilio [3] [13].

### 2.2. Machine Learning

El Machine Learning o Aprendizaje Automático, es un subcampo de la inteligencia artificial, que se centra en el desarrollo de algoritmos que permiten a las computadoras mejorar su desempeño en tareas específicas a partir de datos de entrenamiento. Esta disciplina capacita a las máquinas para aprender de la información disponible y tomar decisiones basadas en patrones y conocimientos adquiridos.

Para desarrollar un modelo de aprendizaje automático, es esencial contar con datos históricos que incluyan las características de entrada pertinentes. A partir de esta información, los algoritmos de aprendizaje analizan y extraen conocimientos, pudiendo seguir una metodología supervisada, no supervisada o de refuerzo, dependiendo del enfoque de aprendizaje. El producto final del proceso de entrenamiento es un modelo de machine learning, el cual representa de manera matemática o estadística las relaciones en los datos para realizar predicciones o tomar decisiones basadas en nuevas entradas. Estos modelos se someten a evaluación y validación mediante conjuntos de datos de prueba para asegurar su precisión y su utilidad en escenarios reales.

### 2.3. Aprendizaje Supervisado

El Aprendizaje Supervisado es una metodología del machine learning que se basa en la disponibilidad de un conjunto de datos compuesto por características previamente etiquetados. Estas características consisten en pares de datos de entrada y las correspondientes salidas esperadas. Durante la fase de entrenamiento, el algoritmo analiza estos datos para descubrir patrones y relaciones entre las características de entrada y las salidas asociadas. A través de este proceso de aprendizaje, el modelo ajusta sus parámetros para minimizar la discrepancia entre las predicciones y las etiquetas reales en el conjunto de entrenamiento. Posteriormente, el modelo capacitado puede generalizar su conocimiento para hacer predicciones precisas sobre nuevos datos de entrada sin etiquetar. Si las características etiquetadas forman parte de una variable de naturaleza categórica la metodología utilizada es conocida como Clasificación, en caso sea numérica, la metodología es denominada como Regresión.

## 2.4. LightGBM

LightGBM es un algoritmo de Gradient Boosting Decision Trees (GBDT) de tipo Aprendizaje Supervisado que puede ser utilizado para problemas de Clasificación y Regresión, que se destaca por su eficiencia computacional y su manejo efectivo de grandes conjuntos de datos y un gran número de características. Incorpora dos técnicas innovadoras: Gradient-based One-Side Sampling (GOSS) y Exclusive Feature Bundling (EFB), diseñadas para mejorar la velocidad de procesamiento y la eficiencia en el uso de memoria en comparación con otros algoritmos GBDT, como XGBoost y Stochastic Gradient Boosting. GOSS se enfoca en la selección de muestras unidireccionales basadas en gradientes, mientras que EFB se concentra en la combinación de características exclusivas. LightGBM ha sido objeto de análisis teóricos y experimentales, demostrando consistentemente su superioridad en términos de rendimiento computacional y uso de memoria en una variedad de aplicaciones [12].

## 3. Objetivo

En esta investigación se ha buscado identificar los posibles problemas que se pueden presentar para la predicción de la aceptación de pedidos por parte de los repartidores, se ha comparado entre diferentes algoritmos en vista de ver cuál es el más apropiado para nuestro objeto de estudio, así mismo, se exploró el rendimiento de los más importantes del estado del arte.

## 4. Data y Metodología

### 4.1. Descripción del conjunto de Datos

El conjunto de datos original pertenece a una empresa de la industria de entregas a domicilio líder en Latinoamérica y no es de acceso público, este conjunto de datos consta de 489 216 observaciones (órdenes), realizadas entre el 1 de septiembre al 1 de octubre del 2023, además de contar con un campo binario (taken) que nos indica si el pedido realizado por el cliente fue o no fue aceptado por el repartidor.

Las variables originales consideradas son las mostradas en la Tabla 1 a partir de la cual se realizó un proceso de Ingeniería de las Características para la generación de nuevas variables para el modelo.

En cuanto a la variable objetivo, los casos interesantes no están precisamente en las órdenes tomadas, sino en las órdenes no tomadas. Entonces, decidimos seleccionar el objetivo como no tomado (si el valor no tomado es 1, si no es 0). El número de pedidos no tomados por repartidores es 46 769, que corresponde al 9,56% del total de órdenes.

### 4.2. Limpieza de data

Tabla 1  
Descripción de las variables originales

Variable	Descripción
order_id	Código identificador de la orden
store_id	Código identificador del negocio
to_user_distance	Distancia (Km) entre la tienda y la ubicación del usuario
to_user_elevation	Diferencia en metros entre la tienda y la altitud del usuario (m.s.n.m)
total_earning	Propina al repartidor en dólares
two_hours	Cantidad de dinero que ha ganado en las dos últimas horas el repartidor
notifications	Número de notificaciones que ha recibido un repartidor en la última hora (antes de la notificación)
type_vehicle	Tipo de vehículo con el que trabaja el repartidor (moto, bicicleta, a pie)
weekday	Día de la semana en la que se realizó el pedido
flag_weekend	Flag de orden realizada fin de semana
hour	Hora en la que fue realizada el pedido
minute	Minuto de la hora en la que fue realizada el pedido
second	Segundo del minuto de la hora en la que fue realizada el pedido
temperature	Temperatura en grados centígrados
wind	Velocidad del viento en km/h
humidity	Cantidad de vapor de agua que puede contener el aire depende de su temperatura
pressure	Presión atmosférica en Hectopascuales (hPa)
flag_futbol_seleccion	Flag que indica si existió algún partido de fútbol de la selección del país en el horario que se realizó el pedido
flag_futbol_clubs	Flag que indica si existió algún partido de fútbol de clubes locales en el horario que se realizó el pedido
taken	Aceptación o no del pedido por parte del repartidor

Se realizaron varias técnicas de limpieza de datos, en primer lugar, se identificaron y filtraron los registros con fechas incorrectas, lo cual resultó en la eliminación de un único registro con una fecha malformada que representaba dos fechas diferentes. Posteriormente, se procedió a eliminar los registros duplicados en el conjunto de datos. Tras verificar la ausencia de valores nulos en el conjunto de datos, se confirmó la consistencia numérica en todas las columnas numéricas, garantizando que solo contuvieran valores numéricos y no nulos. Después de estas técnicas de limpieza iniciales, se procedió con los filtrados adicionales, que incluyeron la eliminación de registros con fechas incorrectas y la filtración de casos atípicos, a través de la distancia intercuartílica, para las columnas 'to\_user\_distance', 'tip' y 'total\_earning', tal como se observa en las Figuras 2 y 3. Estas acciones resultaron en la reducción del número total de registros de 489 216 a 488 120.

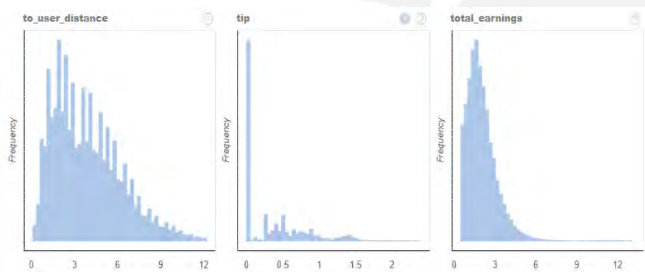


Fig. 2 Distribución original de variables to\_user\_distance, tip y total\_earning

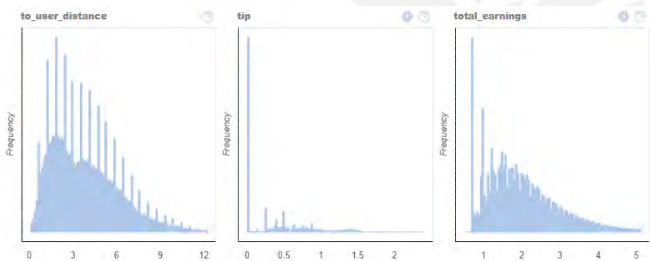


Fig. 3 Distribución sin outliers de variables to\_user\_distance, tip y total\_earning

### 4.3. Partición de la data

Se realizó una partición de los datos utilizando la función `train_test_split` de la biblioteca `scikit-learn`. Los conjuntos de datos de entrada y de salida fueron divididos en conjuntos de entrenamiento y de prueba. La proporción de datos de prueba se estableció en el

24% (97 844 registros) del tamaño total del conjunto de datos original, mientras que para la data de entrenamiento se consideraron 414 902 registros, y se aplicó la estratificación (`stratify`) sobre los datos de salida para mantener la proporción de clases entre los conjuntos de entrenamiento y prueba, lo que es especialmente importante en problemas de clasificación para evitar el desequilibrio de clases [6].

### 4.4. Entrenamiento del modelo

La estrategia de modelamiento de este problema se dividió en dos etapas: Exploración Inicial y Modelamiento Final.

En la etapa de Exploración Inicial, se llevó a cabo una exploración exhaustiva utilizando quince algoritmos diferentes, cada uno con su conjunto estándar de hiperparámetros, y sin aplicar ningún método de balanceo de datos. El propósito fundamental de esta etapa fue evaluar el rendimiento de entrenamiento de los modelos base y determinar cuál de ellos mostraba el potencial más prometedor para abordar el problema con el conjunto de datos dado, siendo LightGBM el algoritmo que presentó mejores resultados [4] [11].

En la fase de Modelamiento Final, se aplicaron dos técnicas al modelo seleccionado durante la fase de Exploración Inicial. En primer lugar, se utilizó la técnica Synthetic Minority Over-sampling Technique (SMOTE) para abordar el desbalance de clases presente en el conjunto de datos [8] [15]. Esta técnica generó datos sintéticos para contrarrestar el bajo porcentaje de registros correspondientes a pedidos no tomados por repartidores, que representaban solo el 9.56% del total. Posteriormente, se empleó la técnica de optimización de hiperparámetros conocida como búsqueda aleatoria (Randomized Search) [2]. Esta estrategia evalúa múltiples combinaciones de forma aleatoria con el objetivo de encontrar la configuración óptima que maximice una métrica de evaluación seleccionada.

El modelo LightGBM se configuró inicialmente con una serie de hiperparámetros claves. Entre estos parámetros se incluyeron “`n_estimators`”, que define el número máximo de estimadores (árboles de decisión) que se construirán en el modelo, y “`max_depth`”, que controla la profundidad máxima de cada árbol en el conjunto. Otros hiperparámetros importantes incluyeron “`colsample_bytree`”, que determina la fracción de columnas consideradas en cada árbol para prevenir el sobreajuste, y “`min_child_samples`” y “`min_child_weight`”, que establecen el número mínimo de muestras y la suma mínima de pesos de muestras

requeridos para formar un nuevo nodo en el árbol, respectivamente. Además, se ajustaron “num\_leaves”, “reg\_alpha”, “reg\_lambda” y “subsample” para optimizar el rendimiento del modelo [7] [20].

#### 4.5. Métricas de validación

Los datos están muy desbalanceados, por lo que si bien podemos tomar la métrica ROC-AUC como una métrica segura para comparar el rendimiento del modelo entre modelos, también es importante visualizar la curva de Precision Recall, que proporciona un mejor diagnóstico para el modelo en comparación con la curva ROC para conjuntos de datos de desbalanceo. Entonces, mientras usamos la métrica ROC-AUC para comparar modelos, también calculamos el Average Precision Recall (Average Precision Score) para comparar cómo se comporta nuestro modelo versus estimar aleatoriamente según la prevalencia de casos no tomados [6].

Durante la evaluación de los modelos, se proporciona visualizaciones de matrices de confusión con un umbral calculado que maximiza la puntuación F1 y con un umbral que apunta a maximizar la precisión, para observar la diferencia. Además, se observó los gráficos de predicciones vs verdaderos en histplots y distplots, entendiendo que el modelo ideal es el que mejor diferencia positivos y negativos independientemente de la frecuencia de ambas clases, por lo que se utilizan distplots (que dependen de la densidad).

#### 4.6. Estrategia de interpretación de resultados

La metodología aplicada para interpretar los resultados del modelo se dividió en dos etapas: análisis exploratorio y evaluación de la importancia de las variables del modelo.

El análisis exploratorio constituyó la primera etapa, donde se llevó a cabo una exploración detallada de la estructura y características de los datos a partir de gráficos e indicadores estadísticos descriptivos. Durante esta fase, se examinaron patrones y relaciones significativas entre las variables para identificar posibles insights. Esta exploración permitió comprender mejor la distribución de los datos, así como las tendencias y variaciones presentes en ellos.

En la segunda etapa, se evaluó la importancia de las variables del modelo utilizando técnicas avanzadas como los explicadores SHAP (Shapley Additive Explanations) [9]. Estos explicadores proporcionan una medida de la contribución de cada variable a la predicción del modelo, permitiendo identificar qué variables tienen un mayor impacto en la salida del modelo. Esta evaluación permitió determinar qué

características del conjunto de datos eran más relevantes para predecir la aceptación de los pedidos, proporcionando así información valiosa para la toma de decisiones.

## 5. Resultados

### 5.1. Exploración inicial

Se desarrolló una exploración inicial, entrenando el dataset con quince modelos diferentes como Random Forest Regressors, Logistic Regression, K Neighbors Classifier, LightGBM, entre otros, tal como se observa en la Tabla 2. El objetivo de esta etapa era encontrar una línea base de modelo potencial, sobre el cual se apliquen posteriormente técnicas de balanceo y optimización para maximizar la capacidad predictiva.

Tabla 2  
Comparación de resultados de modelos iniciales

Modelo	Accuracy	AUC
Extreme Gradient Boosting	0.6998	0.7189
Light Gradient Boosting Machine	0.699	0.7187
Gradient Boosting Classifier	0.6895	0.6922
Random Forest Classifier	0.6868	0.6901
Extra Trees Classifier	0.6800	0.6750
Ada Boost Classifier	0.6812	0.6609
Quadratic Discriminant Analysis	0.6707	0.6345
Logistic Regression	0.6685	0.6164
Linear Discriminant Analysis	0.6682	0.6162
K Neighbors Classifier	0.6448	0.6137
Naive Bayes	0.6612	0.6101
Decision Tree Classifier	0.6099	0.5669
Dummy Classifier	0.6654	0.5000
SVM - Linear Kernel	0.6654	0
Ridge Classifier	0.6670	0

Debido al alto desequilibrio de clases, roc\_auc es una métrica guía, pero no una métrica de evaluación válida para el modelo; sin embargo, nos da una buena pista sobre cómo empezar a abordar el problema, ya que se espera que los modelos basados en DecisionTree se desempeñen entre los mejores.

Extreme Gradient Boosting fue el mejor modelo en la etapa de Exploración Inicial, con un `auc_roc` de 0.7189 (Tabla 2) y con una curva ROC buena según lo observado en la Figura 4, sin embargo, al observar la matriz de confusión de la Figura 5 y la gráfica de distribución de predicciones del target de la Figura 6 apreciamos que la capacidad de discriminación de este modelo no es adecuada en la clase minoritaria. Además, se conoce que es un modelo computacionalmente costoso, por lo cual usaremos LightGBM, que presentó resultados muy similares en la Exploración Inicial, pero con la ventaja de que es un modelo computacionalmente más liviano y que a menudo desafía a los modelos XGBoost [12] [18].

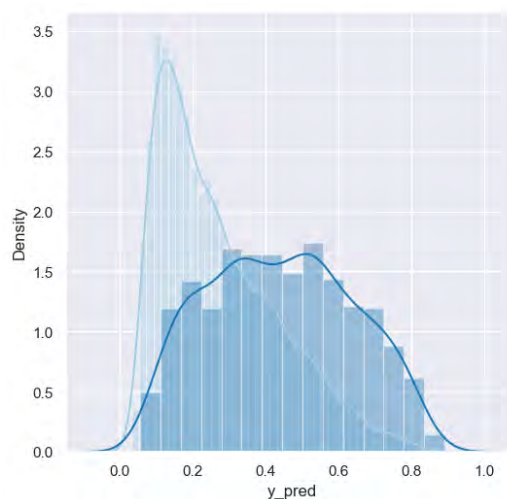


Fig. 6 Gráfica de distribución de predicciones según categorías del target – modelo Extreme Gradient Boosting inicial

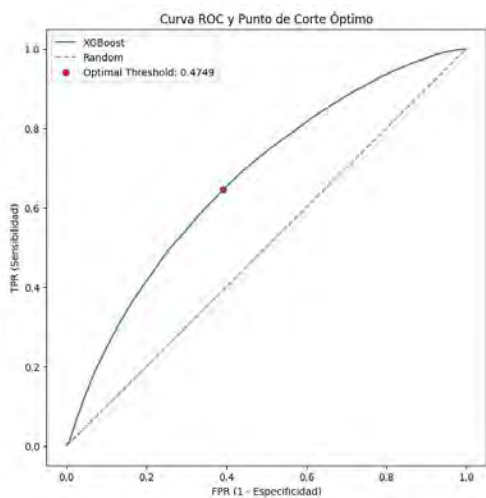


Fig. 4 Curva ROC modelo Extreme Gradient Boosting inicial

## 5.2. Modelamiento Final

Se entrenó un modelo preliminar LightGBM con RandomSearch para hiperparámetros, proporcionando la propiedad `is_unbalance` y proporcionando un `scale_post_weight` directo como parámetro, brindando un rendimiento de 72 AUC y 0,23 de Average Precision Recall, el cual es capaz de diferenciar claramente las órdenes tomadas, pero las órdenes no tomadas no se distinguen fácilmente.

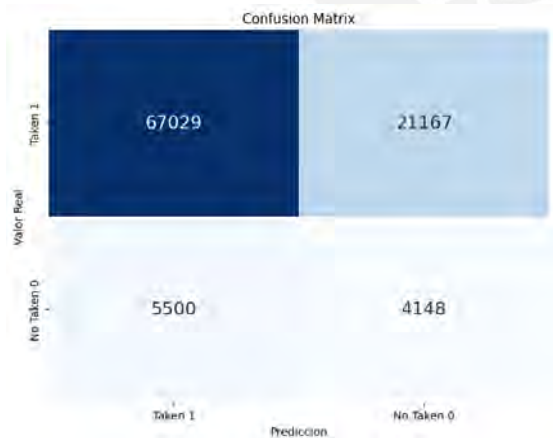


Fig. 5 Matriz de Confusión modelo Extreme Gradient Boosting inicial sobre la data de prueba

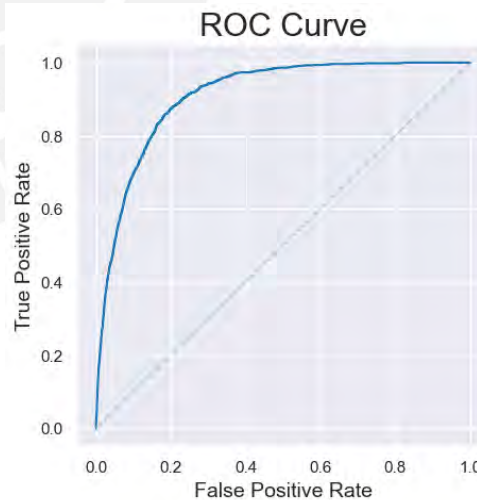


Fig. 7 Curva ROC modelo LightGBM con SMOTE

Es por eso, que se abordó el problema del desbalanceo mediante Smartly OverSampling con SMOTE. Y al hacer esto, logramos aumentar el rendimiento del modelo hasta 88 AUC y 0,47 de Average Precision Recall, lo cual es mucho mejor en



comparación con la prevalencia de la clase y que se refleja en la Figura 7, con una curva ROC bastante buena, lo mismo que se aprecia en la matriz de confusión de la Figura 8 y la gráfica de distribución de predicciones del target de la Figura 9, donde se observa que la capacidad de discriminación de este modelo no es adecuada tanto en las clases minoritaria como mayoritaria.

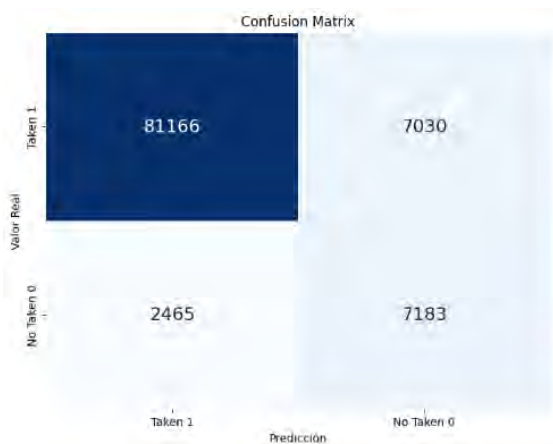


Fig. 8 Matriz de Confusión modelo LightGBM con SMOTE sobre la data de prueba

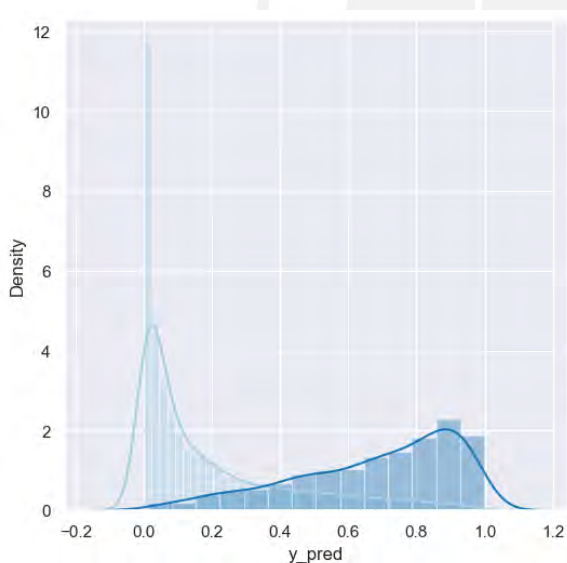


Fig. 9 Gráfica de distribución de predicciones según categorías del target – modelo LigthGBM con SMOTE

### 5.3. Interpretación de los resultados

#### 5.3.1. Análisis Exploratorio

Se observa que las distancias entre el mensajero y el usuario (más predominantemente la longitud y menos predominantemente la elevación) influyen

directamente en las posibilidades de que se realice el pedido: cuanto más largas sean las distancias, mayores serán las posibilidades de que no se realice. Además, los ingresos totales del mensajero también influyen mucho en el pedido que se realiza. Cuando mayor sea el ingreso, mayores serán las posibilidades de que se lo tomen, sin embargo, también viene con una compensación si la distancia es alta. Los pedidos de larga distancia y bajos ingresos (cuando son raros) también tienen menores posibilidades de ser aceptados.

La información temporal sobre cuándo se creó el pedido conduce a la conclusión de que la gran mayoría de pedidos se realizaron los fines de semana, especialmente los viernes. Esto nos dice que la tendencia del mercado de pedidos aumenta los fines de semana, pero el ratio de pedidos tomados disminuye los viernes y sábados, por lo que el número de repartidores disponibles debería aumentar en esos días para abastecer la demanda y aumentar el ratio de pedidos tomados. Podría ser de gran ayuda dar incentivos a los mensajeros para que aumenten su actividad en esos días, debido a que esos días son los que podemos ver mayor área de oportunidad para aumentar el ratio de pedidos tomados, porque solo los viernes el número de pedidos no tomados fue mayor que el número de pedidos tomados el jueves, tal como lo observamos en las Figuras 10 y 11.



Fig. 10 Distribución de órdenes aceptadas y rechazadas por día de semana

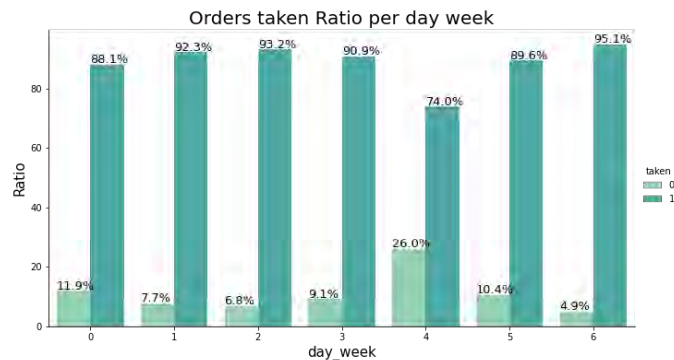


Fig. 11 Ratio de aceptación de órdenes por día de semana

También es muy relevante la hora del día en la que se creó el pedido. El porcentaje más alto de no participantes se presenta muy temprano en la mañana, de 12 a. m. a 5 a. m. (con un máximo a las 5 a. m.) y muy tarde en la noche (de 10 p. m. a 11 p. m.)

### 5.3.2. Importancia de variables del modelo

Se analizó la importancia de los inputs del modelo con explicadores SHAP como se puede observar en la Figura 12. A continuación, se detalla un análisis para cada variable y cómo contribuye a la probabilidad de salida general.

- **hour:** Variable más importante, franjas horarias entre comidas con mayor probabilidad de aceptación por parte de los repartidores, motivo menor cantidad de pedidos a elegir.
- **distance\_to\_store:** también es relevante (relación inversa), cuanto más cerca se encuentre el repartidor al negocio mayor probabilidad de aceptación.
- **total\_earning:** Cuanto mayor sea la ganancia, mayores serán las posibilidades de que el pedido se acepte, tiene relación con la propina (tip).
- **weekday:** más cerca del fin de semana, las posibilidades de que un pedido no sea tomado aumentan, excepto el domingo donde los pedidos tomados aumentan nuevamente. Esta idea se corrobora con el análisis realizado en el análisis exploratorio.
- **day\_month:** Los días iniciales del mes aumentan las posibilidades de que un pedido no sea tomado, en su mayor parte cerca del final del mes es donde los pedidos tomados aumentan más.
- **humidity, temperature:** condiciones climatológicas desfavorables como lluvias, altas temperatura, disminuyen la probabilidad de aceptación.

Las horas y el día del mes muestran la misma relación que se observa en el análisis exploratorio, las horas medias durante el día no impactan fuertemente, las mayores posibilidades de que el modelo lo interprete

como no tomadores es temprano en la mañana, y menores posibilidades de que sea muy tarde en la noche, en el caso de día y mes. Los finales de mes se caracterizan normalmente por una mayor cantidad de pedidos recibidos.

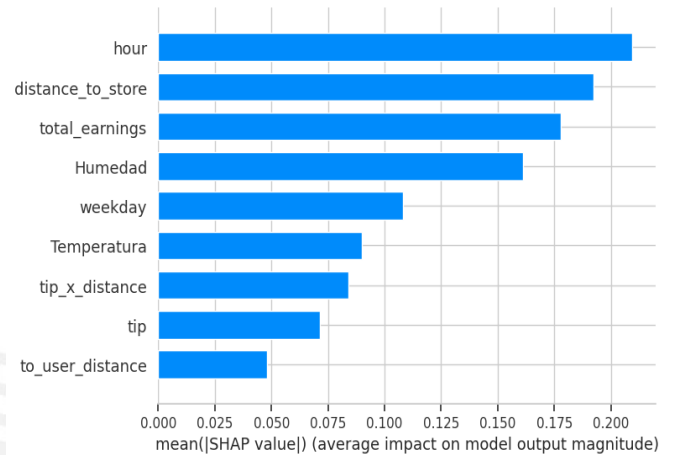


Fig. 12 Gráfica de Shap Values – Modelo LightGBM con SMOTE

Las ganancias son de gran importancia como se observa en la Figura 13, por lo que una gran parte de cómo explotar estos insights es tomar medidas en este lado. Las sugerencias podrían ir desde aumentar ligeramente el costo de envío de estos pedidos (ya que el resto de la información, desde que ingresa a la tienda se puede obtener y así estimar), hasta explicar al usuario que debido a las características de su pedido existe una gran posibilidad de que tomará algún tiempo y sugerirá una propina que se enviará al mensajero para aumentar sus posibilidades de que los mensajeros dentro del rango acepten su pedido.

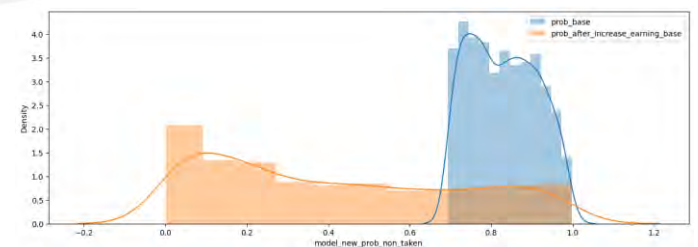


Fig. 13 Gráfica de distribución de probabilidades de aceptaciones de órdenes sin y con propinas incluidas

## 6. Discusión

Los resultados obtenidos a través del modelo LightGBM, potenciado por las técnica de balanceo de datos SMOTE y de optimización de hiperparámetros

Randomized Search, revelan un rendimiento alentador en la predicción de la aceptación de pedidos por parte de repartidores en la industria de entregas a domicilio. Con un AUC de 0.88 y un Average Precision Recall de 0.47, el modelo demuestra una capacidad sustancial para distinguir entre órdenes tomadas y no tomadas.

La aplicación de SMOTE para abordar el desbalanceo de clases ha demostrado ser crucial. Esta técnica ha mejorado significativamente la capacidad predictiva del modelo, especialmente en la detección de casos de órdenes no tomadas. Esta sensibilidad mejorada es esencial para una toma de decisiones efectiva en la asignación de pedidos. Comparando con modelos iniciales, la elección estratégica de LightGBM y SMOTE ha mostrado una superioridad significativa en términos de métricas de evaluación como AUC y Average Precision Recall. La capacidad de LightGBM para manejar grandes conjuntos de datos y su eficacia en problemas de clasificación desbalanceada respaldan esta elección.

Al explorar la importancia de las variables, se destaca la hora del día como el factor más influyente. Franjas horarias estratégicas entre comidas presentan mayores probabilidades de aceptación, indicando oportunidades para implementar incentivos específicos y mejorar la asignación eficiente de pedidos. La inclusión de variables climáticas, como temperatura y humedad, resalta la influencia directa de las condiciones climáticas en las decisiones de los repartidores. Condiciones adversas, como lluvias o altas temperaturas, disminuyen la probabilidad de aceptación. Estrategias que consideren estas variables podrían incluir incentivos adicionales durante condiciones climáticas desfavorables para mejorar la aceptación de pedidos.

Estos resultados no solo tienen implicaciones prácticas en la optimización de la plataforma de entregas, sino que también sugieren estrategias específicas de incentivos basadas en el análisis detallado de variables temporales y climáticas. Considerar estos hallazgos en la toma de decisiones puede mejorar significativamente la satisfacción del cliente y la eficiencia operativa en la asignación de pedidos.

## 7. Conclusiones

El presente trabajo ha tenido como objeto de estudio identificar los posibles problemas que se pueden presentar para la predicción de la aceptación de pedidos por parte de los repartidores, se ha comparado entre

diferentes algoritmos en vista de ver cuál es el más apropiado para nuestro objeto de estudio, siendo LightGBM el cual ha demostrado ser una elección robusta para la clasificación en este contexto. Su capacidad para manejar grandes volúmenes de datos y lidiar con desbalances resalta su idoneidad para problemas similares.

En cuanto a trabajos futuros, se recomienda una exploración más profunda de relaciones entre variables, así como un análisis más detallado de tiendas problemáticas para comprender las causas subyacentes de las tasas bajas de aceptación. Esto podría implicar revisar las calificaciones de las tiendas o examinar posibles problemas operativos, ya que podría haber una causa subyacente para el bajo porcentaje de pedidos recibidos: alteración de precedentes, mal trato a los repartidores, ubicaciones peligrosas. Con todo, en este caso, simplemente creando una bandera para algunas de estas tiendas (cuyo recuento de pedidos figura dentro de los deciles 8,9,10) y con porcentaje de pedidos tomados  $< 0,85$  (porcentaje bajo de pedidos tomados), puede ayudar al modelo en los casos en que aplique.

La inclusión de información adicional sobre los clientes podría proporcionar una visión más detallada de los factores que influyen en la aceptación de pedidos. Esto podría incluir datos demográficos, preferencias de compra y ubicaciones frecuentes.

La estrategia de incentivos basada en ingresos y condiciones climáticas puede ser refinada y optimizada continuamente. La experimentación con diferentes estructuras de incentivos y su impacto en las tasas de aceptación podría ser un área de mejora constante.

En conclusión, este estudio sienta las bases para mejoras continuas y adaptaciones a medida que la industria evoluciona. La combinación de técnicas avanzadas y análisis detallados ofrece un enfoque sólido para abordar los desafíos operativos en la entrega a domicilio, proporcionando insights valiosos para la toma de decisiones estratégicas en el ámbito logístico.

## Referencias Bibliográficas

1. Al Amin, M., Arefin, M.S., Alam, M.R., Ahammad, T., & Hoque, M.R. (2021). Using mobile food delivery applications during COVID-19 pandemic: An extended model of planned behavior. *Journal of Food Products Marketing*, 27(2), 105-126.
2. Bergstra, J., Bengio, Y. (2012). Random search for hyper-parameter optimization. *J. Mach. Learn.*



Res. 13, 281–305.

3. Bonfanti, A., Vigolo, V., Yfantidou, G., & Gutuleac, R. (2023). Customer experience management strategies in upscale restaurants: Lessons from the Covid-19 pandemic. *International Journal of Hospitality Management*, 109, 103416.
4. Chen, L. (2008). A model of consumer acceptance of mobile payment. *International Journal of Mobile Communications*, 6(1), 32.
5. Cho, M., Bonn, M.A., & Li, J. (Justin). (2019). Differences in perceptions about food delivery apps between single-person and multi-person households. *International Journal of Hospitality Management*, 77, 108-116.
6. Duda, R., Hart, P., & Stork, D. (2001). *Pattern Classification*. Wiley-Interscience.
7. F. Hutter, J. Lücke, L. Schmidt-Thieme (2015). Beyond manual tuning of hyperparameters, *DISKI* 29 (4) 329–337.
8. Hamdy, A., El-Laithy, A. (2019). SMOTE and Feature Selection for More Effective Bug Severity Prediction. *Int. J. Softw. Eng. Knowl. Eng.* 29, 897–919
9. Hutter, F., Kotthoff, L., & Vanschoren, J. (2019). *Automated Machine Learning Methods, Systems, Challenges*. Cham: Springer International Publishing.
10. IMARC. (2020). Global Online Food Delivery Market to Reach US\$ 164.5 Billion by 2024, Stimulated by Development of User-Friendly Applications. <https://www.imarcgroup.com/global-online-food-delivery-market>.
11. Kabari, L.G., & Onwuka, U.C. (2019). Comparison of bagging and voting ensemble machine learning algorithm as a classifier. *International Journals of Advanced Research in Computer Science and Software Engineering*, 9(3), 19-23.
12. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T. (2017). LightGBM: a highly efficient gradient boosting decision tree. 31st Conference on Neural Information Processing Systems.
13. Kumar, S., & Shah, A. (2021). Revisiting food delivery apps during COVID-19 pandemic. Investigating the role of emotions. *Journal of Retailing and Consumer Services*, 62, 102595.
14. Lee, E.-Y., Lee, S.-B., & Jeon, Y.J.J. (2017). Factors influencing the behavioral intention to use food delivery apps. *Social Behavior and Personality: An International Journal*, 45(9), 1461-1473.
15. Li, Q., & Mao, Y. (2014). A review of boosting methods for imbalanced data classification. *Pattern Analysis and Applications*, 17(4), 679-693.
16. Mehroliya, S., Alagarsamy, S., & Solaikutty, V.M. (2021). Customers response to online food delivery services during COVID-19 outbreak using binary logistic regression. *International Journal of Consumer Studies*, 45(3), 396-408.
17. Patil, R. V., Kale, A., Pawar, D., & Patil, T. (2017). Wireless Customizable Food Ordering System for a Restaurant Using Apriori and K-means Algorithm. *Imperial Journal of Interdisciplinary Research*, 3(4).
18. Speiser, J.L., Miller, M.E., Tooze, J., & Ip, E. (2019). A comparison of random forest variable selection methods for classification prediction modeling. *Expert Systems with Applications*, 134, 93-101.
19. Statista. (2020). Services Report 2019 – Online Food Delivery. <https://www.statista.com/study/40457/food-delivery/>.
20. Yu, T., & Zhu, H. (2020). Hyper-Parameter Optimization: A Review of Algorithms and Applications