Logística Inteligente

"PREDICCIÓN DE RETRASOS EN ENTREGAS DE E-COMMERCE"

MARIO ALCALDE ALVARADO

Problemática de negocio: Predicción de retrasos en entregas de e-commerce

- "En el contexto actual del comercio electrónico, una de las principales preocupaciones logísticas es garantizar que los pedidos lleguen a tiempo a los clientes.
  Retrasos en las entregas afectan directamente la satisfacción del cliente, generan costos operativos adicionales y afectan la reputación de la empresa.

Objetivo del proyecto

- Desarrollar un modelo predictivo capaz de anticipar, con base en las características del pedido (como el tipo de envío, la prioridad del producto, el descuento aplicado y otras variables), si un pedido llegará a tiempo o sufrirá retrasos.

- Esta predicción permitirá a la empresa:

- ✔ **Anticipar problemas logísticos**
- ✔ **Optimizar recursos de transporte**
- ✔ **Mejorar la experiencia del cliente**
- ✔ **Reducir el impacto económico de los retrasos**"

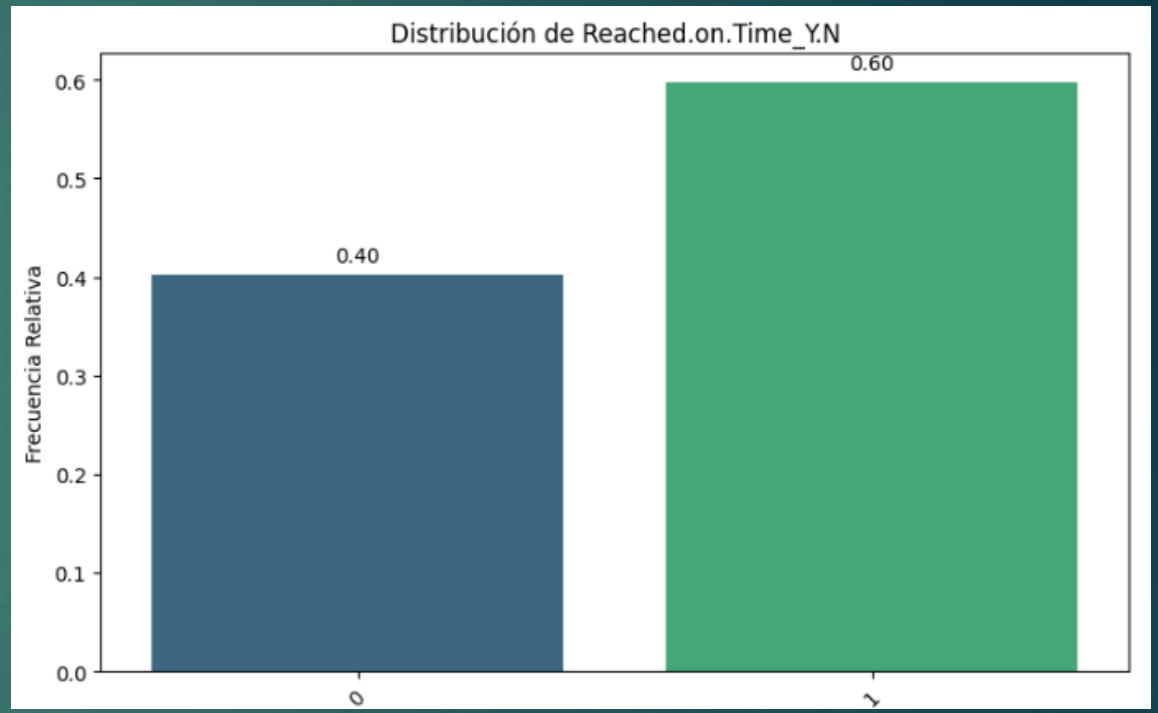
Esta foto de Autor desconocido está bajo licencia CC BY-NC-ND

Descripción de la variable Target

La variable target del proyecto es Reached.on.Time_Y.N , una variable binaria que indica si el pedido fue entregado a tiempo o no.

- 0. Entrega a tiempo
- 1. Entregado con retraso

Se observa un ligero desbalance a favor de la clase 1 (retrasos), lo cual fue considerado durante la elección de métricas y la validación de los modelos. Se utilizaron métricas como **Balanced Accuracy** y **AUC** para evaluar el rendimiento, ya que son más adecuadas en escenarios donde las clases no están perfectamente equilibradas.



Variables más predictivas en el modelo final

Variable	Justificación
Discount_offered	Mostró una diferencia estadísticamente significativa respecto a la target (p-valor extremadamente bajo en Mann-Whitney U). Además, tiene alta correlación negativa con llegar a tiempo (a mayor descuento, mayor riesgo de retraso).
Weight_in_gms	Se presentó diferencias significativas entre las clases, mostrando que el peso influye en la probabilidad de retraso.
Cost_of_the_Product	Mostró relación con la target en las pruebas estadísticas y en la importancia de variables del modelo Random Forest y Gradient Boosting.
Warehouse_block	La ubicación del almacén fue relevante. Ciertos bloques (A, B, etc.) mostraron más retrasos. Esto se vio en el análisis categórico y la codificación one-hot.
Mode_of_Shipment	El tipo de envío (Ship, Flight, Road) fue clave, especialmente en modelos de árboles y Gradient Boosting.
Product_importance	Al ser una variable ordinal, tuvo peso en la discriminación entre pedidos a tiempo y retrasados (especialmente entre low y high).

Definición de Modelo Baseline

- Se eligió una **Regresión Logística simple** como punto de partida.

- Esto nos permite tener una referencia de desempeño mínimo para comparar modelos más complejos después.

El modelo baseline, basado en una Regresión Logística sin ajuste de hiperparámetros, nos permitió establecer un punto de partida inicial para medir el rendimiento.
El modelo baseline obtuvo una **Cross-Validation promedio de 0.566.**
Este valor refleja que, en promedio, el modelo logró distinguir correctamente ambas clases (entregados a tiempo y retrasados) un **56.6% de las veces**, teniendo en cuenta el desbalance de la target.

```
Balanced Accuracy promedio en Cross-Validation (5 folds): 0.566
```

Comparativa con diferentes modelos

Tras comparar el rendimiento de varios modelos de clasificación, se concluyó que el **Gradient Boosting** fue el modelo con mejor desempeño, alcanzando la **mayor Cross-Validation (0.620)** frente a las demás alternativas.

También demostró una buena capacidad para manejar tanto variables numéricas como categóricas, así como capturar relaciones no lineales complejas entre las variables predictoras y la target.

Dado este resultado, se decidió seleccionar este modelo como base para la siguiente fase de optimización de hiperparámetros, con el objetivo de mejorar aún más su capacidad predictiva.

```
Regresión Logística: Cross-Validation = 0.566
KNN: Cross-Validation = 0.608
SVM: Cross-Validation = 0.556
```

```
Regresión Logística: Cross-Validation = 0.566
Random Forest: Cross-Validation = 0.617
LightGBM: Cross-Validation = 0.613
XGBoost: Cross-Validation = 0.609
Gradient Boosting: Cross-Validation = 0.620
```

Resultados de modelo final con hiperparametros

Métrica	Resultado
Mejores Hiperparámetros	learning_rate = 0.1, max_depth = 3, n_estimators = 100, subsample = 1.0
Cross-Validation (5 folds)	0.623
Balanced Accuracy en Test Set	0.747
Accuracy (Test Set)	0.710
Recall (Test Set)	0.556
Precision (Test Set)	0.929
F1 Score (Test Set)	0.696
AUC (Test Set)	0.805

Interpretación de mejora

- **Mejora significativa respecto al baseline:**
  Pasamos de una **Cross-Validation de 0.566 (baseline)** a **0.623 en CV** y **0.747 en test**, lo que representa una mejora notable en la capacidad de discriminación del modelo.
- **Generalización adecuada:**
  El resultado de test es mejor que el de CV, lo cual indica que el modelo no está sobreajustado y ha logrado generalizar correctamente a datos no vistos.
- **Buen desempeño en Recall vs Precision:**
  Aunque el **recall (0.556)** todavía muestra margen de mejora (captura de la clase minoritaria), el modelo logra una **muy alta precisión (0.929)**, lo que significa que los casos que predice como retrasados son en su mayoría correctos.

Comparativa modelo Baseline vs Modelo final

Métrica	Baseline (Regresión Logística)	Gradient Boosting Optimizado
Cross-Validation (CV)	0.566	0.623
Cross-Validation (Test)	0.603	0.747
Accuracy	0.628	0.710
Recall	0.672	0.556
Precision	0.735	0.929
F1 Score	0.702	0.696
AUC	0.609	0.805

El proceso de optimización permitió mejorar significativamente el desempeño del modelo, logrando una mayor capacidad de predicción de pedidos con riesgo de retraso y aumentando la precisión en las predicciones.

Acciones de mejora

- **Optimización adicional de hiperparámetros:**
  Realizar una búsqueda más exhaustiva o probar técnicas como **RandomizedSearchCV** o **Bayesian Optimization**.

- **Ingeniería de Features:**
  Explorar nuevas combinaciones o transformaciones de variables, e incorporar variables temporales como día de la semana o mes.

- **Balanceo de clases:**
  Aplicar técnicas de resampling (como **SMOTE** o **undersampling**) para mejorar aún más el **recall** sobre la clase minoritaria (retrasos).

Conclusiones

- **Éxito en el objetivo de negocio:** Se logró desarrollar un modelo de clasificación capaz de predecir con buena precisión los pedidos con riesgo de retraso en el e-commerce.
- **Importancia del preprocesamiento:** La correcta codificación de variables, escalado y transformación numérica mejoraron el rendimiento del modelo.
- **Valor de la selección de features:** Focalizar el modelo en las variables más relevantes mejoró la precisión sin añadir complejidad innecesaria.
- **Necesidad de tratar el desbalance:** Elegimos Balanced Accuracy y AUC como métricas clave por la distribución desigual de la target