# 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 sufrirá retrasos (métrica F1).

- Esta predicción permitirá a la empresa:

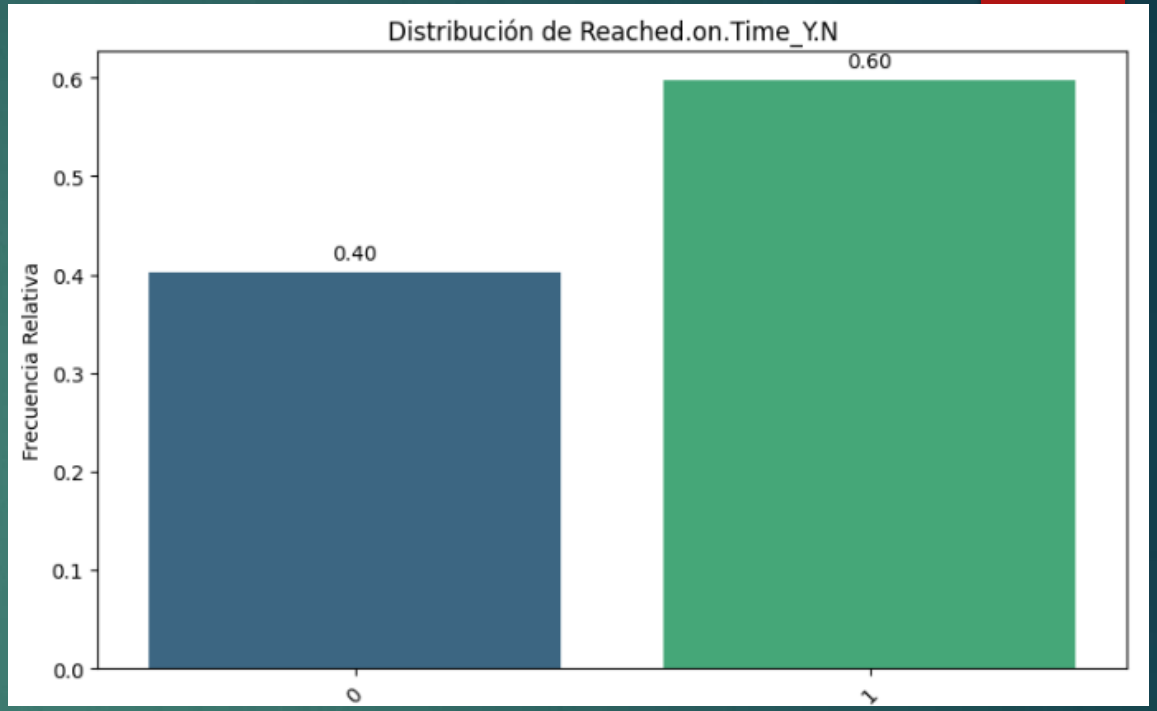
- ✔ **Anticipar problemas logísticos**
- ✔ **Optimizar recursos de transporte**
- ✔ **Mejorar la experiencia del cliente**
- ✔ **Reducir el impacto económico de los retrasos**"

# 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** para evaluar el rendimiento, ya que son más adecuadas en escenarios donde las clases no están perfectamente equilibradas.




Se aplicó **undersampling** porque había **más entregas con retraso que a tiempo**, lo que generaba un desbalance en la variable objetivo. Esta técnica equilibró las clases reduciendo la cantidad de ejemplos de la clase mayoritaria (retrasos), ayudando al modelo a **predecir con mayor precisión las entregas a tiempo**, que son menos frecuentes pero clave para el negocio.

# 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 0.62.**
Este valor refleja que, en promedio, el modelo logró distinguir correctamente ambas clases (entregados a tiempo y retrasados) un **62.6% de las veces**.

```
F1 Score promedio en validación cruzada (5 folds): 0.620
```

# Comparativa con diferentes modelos

- En la comparación de modelos utilizando validación cruzada con F1 Score, Random Forest obtuvo un rendimiento competitivo (CV = 0.646), superando ligeramente a LightGBM y Gradient Boosting, y empatando prácticamente con XGBoost.

Se eligió **Random Forest** por ser:

- El modelo **más interpretable** entre los de mejor rendimiento.
- **Robusto ante ruido y desbalance** de clases (especialmente con class_weight=balanced).
- Rápido de entrenar y menos sensible a hiperparámetros complejos.
- Esto lo hizo el candidato ideal para realizar optimización y afinar su capacidad predictiva en el problema de entregas a tiempo.

```
Regresión Logística: Cross-Validation = 0.620
KNN: Cross-Validation = 0.626
SVM: Cross-Validation = 0.609
```

```
Random Forest: Cross-Validation = 0.646
LightGBM: Cross-Validation = 0.638
XGBoost: Cross-Validation = 0.647
Gradient Boosting: Cross-Validation = 0.642
```

# Resultados de modelo final con hiperparametros

**Comparativa Modelo Final - RF**

| | Métrica | Train Set | Test Set |
|---|---|---|---|
| 1 | Accuracy | 0.8 | 0.74 |
| 2 | F1 Score | 0.8 | 0.74 |
| 3 | Precision (Clase 1) | 1.0 | 1.0 |
| 4 | Recall (Clase 1) | 0.61 | 0.57 |
| 5 | Precision (Clase 0) | 0.72 | 0.61 |
| 6 | Recall (Clase 0) | 1.0 | 1.0 |
| 7 | Macro Avg F1 | 0.8 | 0.74 |
| 8 | Weighted Avg F1 | 0.8 | 0.74 |

El modelo Random Forest optimizado muestra un buen equilibrio entre precisión y recall en ambos conjuntos. En entrenamiento, alcanza un F1 Score de 0.80, lo que indica un alto rendimiento. En el conjunto de prueba, mantiene un F1 Score sólido de 0.74, demostrando buena capacidad de generalización y estabilidad. Esto confirma su utilidad para identificar de forma efectiva los pedidos que podrían llegar con retraso.

# Comparativa modelo Baseline vs Modelo final

| Métrica | Baseline (Reg. Logística) | Modelo Final (Random Forest) |
|---|---|---|
| F1 Score (Train) | 0.65 | 0.80 |
| F1 Score (Test) | 0.65 | 0.74 |
| Validación Cruzada (F1) | 0.620 | 0.646 |
| Accuracy (Test) | 0.65 | 0.74 |
| Recall clase 1 (Retrasos) | 0.57 | 0.57 |

El modelo final Random Forest optimizado supera al baseline en todas las métricas clave, especialmente en F1 Score, mostrando un mejor equilibrio entre precisión y recall. Esto lo convierte en una mejor herramienta para predecir retrasos en los pedidos.

# Acciones de mejora

| Área de Mejora | Descripción |
| --- | --- |
| Análisis Operativo | Identificar patrones en los pedidos con retraso por zona, producto o logística. |
| Enriquecimiento de Datos | Incluir variables externas como clima, tráfico o carga operativa. |
| Mantenimiento del Modelo | Reentrenar periódicamente con datos nuevos para mantener su efectividad. |
| Estrategias Basadas en Costos | Implementar penalizaciones diferenciadas para mejorar la detección de retrasos. |
| Automatización de Alertas | Generar alertas tempranas ante alto riesgo de entrega tardía. |
| Análisis Temporal | Incorporar estacionalidad para mejorar la precisión del modelo. |

# Aprendizajes del Proyecto

- **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 el undersampling como métricas clave por la distribución desigual de la target.

# Conclusión del Proyecto

**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.

Se desarrolló un modelo predictivo eficaz que permite anticipar entregas con riesgo de retraso en el e-commerce:

- Mejora la planificación logística
- Permite tomar decisiones proactivas
- Aporta valor al negocio mediante una mejor experiencia al cliente

# GRACIAS Y SU ATENCION

“Con decisiones basadas en datos, avanzamos hacia una logística más eficiente y predictiva.”