Informe final – Etapa de Transferencia

Curso: Programación para Ciencia de Datos II
Estudiante: Carlos Fernando Velásquez

1. Resumen ejecutivo

En este proyecto analicé los datos de la plataforma de e-commerce Olist. El objetivo fue entender qué factores influyen en el valor total de los pedidos y en los tiempos de entrega.

A lo largo del curso usé herramientas estadísticas y modelos de regresión para encontrar patrones. Uno de los hallazgos más importantes fue que el precio de los productos y el valor del flete explican casi todo el pago total. En cambio, el número de ítems en un pedido tiene un efecto mucho menor.

Después de varias pruebas e iteraciones, apliqué técnicas de regularización que hicieron los modelos más confiables. Finalmente, resumí los resultados en un dashboard interactivo, que facilita ver la información y tomar decisiones.

2. Datos y Calidad

Trabajé con varios datasets de Olist: órdenes, pagos, ítems y productos.

En la primera etapa hice limpieza básica:

- Quité duplicados.
- Revisé valores nulos y los traté según el caso.
- Traduje categorías de productos para tener consistencia.

Aun así, encontré limitaciones. Por ejemplo, los datos no incluyen factores externos que influyen en la logística: clima, festivos o problemas de transporte. Esto hace que algunos retrasos no puedan explicarse solo con la información de Olist.

3. Diseño experimental

Se plantea entonces una hipótesis clara:

- **H0 (hipótesis nula):** no hay diferencia entre el número de ítems y el valor del pedido.
- **H1 (hipótesis alternativa):** sí existe relación entre número de ítems y valor del pedido.

Con la prueba t de Welch encontré que los pedidos de mayor valor sí suelen tener más ítems.

Para evaluar los modelos usé división train/test y, en algunos casos, validación cruzada. Esto me permitió verificar que los resultados no dependieran solo de un subconjunto de datos.

4. Métodos estadísticos computacionales

Probé varios enfoques:

- **Contraste de hipótesis** para confirmar relaciones entre variables.
- **Regresión lineal múltiple** para explicar el valor total en función del precio y el flete.
- **Regresión logística** para clasificar pedidos en grupos (ejemplo: alto vs bajo valor).

Al inicio, los modelos parecían demasiado "perfectos" ($R^2 \approx 0.9999$). Esto fue una señal de sobreajuste. Para controlarlo apliqué **regularización (Ridge, Lasso y ElasticNet)** y probé distintas configuraciones de hiperparámetros.

Gracias a esto logré modelos más equilibrados y realistas.

5. Resultados

Los resultados mejoraron en cada ronda de pruebas:

- En **clasificación**, la regresión logística pasó de un AUC de 0.72 a 0.78 al ajustar la regularización.
- En **regresión**, el modelo Ridge redujo el sobreajuste y alcanzó un R^2 estable de ~0.85.

También usé métricas como **MAE** (error medio absoluto) y **MAPE** (error en porcentaje). Estas métricas son más fáciles de interpretar porque muestran el error en valores o porcentajes, algo que se entiende mejor en un contexto de negocio.

El análisis segmentado reveló que las categorías de producto y el valor del flete son factores claves en el pago final.

6. Interpretación y transferencia al negocio

Los resultados se pueden aplicar en varios frentes:

- Optimizar tarifas de flete en productos grandes o pesados.
- Diseñar promociones que apunten a ítems de alto valor unitario.
- Planear la logística mejor en temporadas altas (Navidad, Black Friday).

En general, el modelo ayuda a la empresa a anticipar retrasos y entender mejor cómo se construye el valor de cada pedido.

7. Conclusiones y próximos pasos

Este proyecto mostró que, con paciencia e iteración, es posible pasar de modelos iniciales muy básicos a modelos estables y confiables.

Las claves del proceso fueron:

- Limpiar y enriquecer los datos.
- Crear nuevas variables.
- Ajustar parámetros y aplicar regularización.

Para siguientes fases, recomendamos:

- Incluir datos externos (clima, festivos, tráfico).
- Probar modelos más avanzados (Random Forest, XGBoost).
- Medir el desempeño con métricas cercanas al negocio, no solo técnicas.

El dashboard que construimos al final es una herramienta práctica que resume todo el análisis y permite a otros interactuar con los resultados.

Estrategia de mejora:

Mejoré el modelo de forma iterativa y controlada. En cada ronda apliqué un solo cambio (una feature o un hiperparámetro), entrené y validé con el mismo criterio, y registré los resultados. Esto me permitió aislar el impacto de cada decisión y evitar conclusiones falsas. Cuando una variante no mejoró, la descarté; cuando sí lo hizo, la incorporé al pipeline.

El proceso dejó claro que el precio de los ítems y el flete son los factores principales que explican el pago total. El número de ítems influye, pero menos. La prueba de Welch respalda que los pedidos de alto pago suelen tener más ítems. Con esto, ahora podemos proponer acciones concretas: ajustar tarifas de flete en productos voluminosos, diseñar promociones en ítems de mayor valor unitario y planear la logística en temporadas de alta demanda.

Con estas rondas de mejora, el modelo pasó de una versión inicial moderada a un modelo regularizado y validado con mejor desempeño y mayor interpretabilidad. La evidencia (métricas, validación y análisis de errores) respalda que el modelo es más fiable y que la comprensión del problema es suficiente para proponer soluciones que impacten al negocio.

8. Mejora iterativa del modelo

El modelo no se construyó en un solo intento. Se trabajó en ciclos cortos de mejora donde se aplicó un cambio a la vez (feature nueva, ajuste de hiperparámetro o regularización), se entrenó de nuevo, se validó con el mismo conjunto de prueba y se registraron las métricas. Esto permitió aislar el impacto de cada decisión y evitar conclusiones falsas. Cuando una variante no mejoró, se descartó; cuando sí lo hizo, se incorporó al pipeline.

A continuación se muestra la tabla con las principales iteraciones del modelo:

Run	Cambio aplicado	Justificación	Métrica base	Métrica nueva	Δ (mejora)	Decisión
r001	Logística sin regularización	Línea base	AUC 0.72	—	—	Baseline
r002	Logística con L2 (C=1.0)	Reducir sobreajuste, mejorar discriminación	AUC 0.72	0.78	+0.06	Mantener
r003	Logística con L2 más fuerte (C=0.5)	Probar mayor penalización	AUC 0.78	0.77	-0.01	Volver a C=1.0
r004	Feature avg_price_per_item	Capturar valor unitario	Acc 0.74	0.76	+0.02	Mantener
r005	Regresión Ridge en lugar de lineal	Evitar sobreajuste en R^2	R^2 0.9999	0.85	Más realista	Mantener

La fiabilidad del modelo se sustenta en métricas consistentes en validación. En clasificación, el AUC mejoró de ~0.72 a ~0.78 y la exactitud de ~0.70 a ~0.76 con regularización L2 (C=1.0). En regresión, se pasó de un R^2 casi perfecto (~0.9999), que era señal de sobreajuste, a un R^2 estable de ~0.85 usando Ridge, junto con errores más realistas en el conjunto de prueba.

El proceso dejó claro que el precio de los ítems y el flete son los factores principales que explican el pago total. El número de ítems influye, pero en menor medida. Con esto ahora es posible proponer acciones concretas: ajustar tarifas de flete en productos voluminosos, diseñar promociones en ítems de alto valor unitario y planear la logística en temporadas de alta demanda.

En conclusión, gracias a la mejora iterativa el modelo pasó de ser una versión inicial moderada a un modelo regularizado y validado, con mejor desempeño y mayor interpretabilidad. La evidencia recogida respalda que el modelo es más fiable y que la comprensión del problema es suficiente para proponer soluciones reales al negocio.

9. Reproducibilidad

El proyecto está preparado para ser replicado:

- El dashboard se ejecuta en Python usando Dash, Plotly y Pandas.
- Los datasets están en formato CSV dentro de la carpeta *data*.
- Se recomienda trabajar en un entorno virtual (trabajé con venv) para mantener la compatibilidad de versiones.
- La carpeta final incluye los notebooks, el dashboard y un archivo de texto con los enlaces al repositorio en GitHub y a la versión en Binder (para correr el proyecto en la nube).