Trabajo Fin de Máster

Sistema de alertas para una cadena de suministro

Antón Quintela Ferreiro

Máster en Técnicas Estadísticas

Curso 2023-2024

Propuesta de Trabajo Fin de Máster

Título en galego: Sistema de alertas para unha cadea de subministro
Título en español: Sistema de alertas para una cadena de suministro
English title: Alert System for a Supply Chain
Modalidad: Modalidad B
Autor/a: Antón Quintela Ferreiro, Universidad de La Coruña
Director/a: Ameijeiras Alonso, Jose, Universidad de Santiago de Compostela; Saavedra Nieves, Paula, Universidad de Santiago de Compostela
Tutor/a: Belén María Fernández De Castro, INDITEX
Breve resumen del trabajo: Uno de los principales puntos fuertes de INDITEX es su cadena de suministro. La compañía trabaja con un amplio abanico de proveedores, con fábricas en muy diversas localizaciones, lo que implica la necesidad de coordinar tanto distintas estrategias de fabricación y transporte, como los tiempos de tránsito hasta los centros de distribución. En este contexto, cualquier anomalía en los pedidos esperados por parte de los proveedores tiene un gran impacto en el desempeño de las ventas. El objetivo de este análisis es detectar patrones de comportamiento en las entregas de pedidos que nos permitan anticipar las posibles anomalías, con el fin de generar alertas que permitan tomar acciones y mitigar sus efectos adversos.
Recomendaciones:
Otras observaciones:

Don/doña Ameijeiras Alonso, Jose, Profesor Ayudante de la Universidad de Santiago de Compostela, don/doña Saavedra Nieves, Paula, Profesor Contratado/a de la Universidad de Santiago de Compostela, don/doña Belén María Fernández De Castro, cargo 1 de INDITEX, informan que el Trabajo Fin de Máster titulado

Sistema de alertas para una cadena de suministro

fue realizado bajo su dirección por don/doña Antón Quintela Ferreiro para el Máster en Técnicas Estadísticas. Estimando que el trabajo está terminado, dan su conformidad para su presentación y defensa ante un tribunal.

En A Coruña, a 20 de julio de 2024.

AMEIJEIRAS ALONSO JOSE - 39450236S
Firmado digitalmente por AMEIJEIRAS ALONSO JOSE - 39450236S
Fecha: 2024.07.22 10:09:58 +02'00'

El/la director/a:
Don/doña Ameijeiras Alonso, Jose

SAAVEDRA NIEVES PAULA - 44847364D
Firmado digitalmente por SAAVEDRA NIEVES PAULA - 44847364D
Fecha: 2024.07.22 10:28:50 +02'00'

El/la director/a:
Don/doña Saavedra Nieves, Paula

El/la tutor/a:
Don/doña Belén María Fernández De Castro

QUINTELA FERREIRO ANTON - 54153539Q
Firmado digitalmente por QUINTELA FERREIRO ANTON - 54153539Q
Fecha: 2024.07.21 11:29:52 +02'00'

El/la autor/a:
Don/doña Antón Quintela Ferreiro

Declaración responsable. Para dar cumplimiento a la Ley 3/2022, de 24 de febrero, de convivencia universitaria, referente al plagio en el Trabajo Fin de Máster (Artículo 11, Disposición 2978 del BOE núm. 48 de 2022), **el/la autor/a declara** que el Trabajo Fin de Máster presentado es un documento original en el que se han tenido en cuenta las siguientes consideraciones relativas al uso de material de apoyo desarrollado por otros/as autores/as:

- Todas las fuentes usadas para la elaboración de este trabajo han sido citadas convenientemente (libros, artículos, apuntes de profesorado, páginas web, programas,...)
- Cualquier contenido copiado o traducido textualmente se ha puesto entre comillas, citando su procedencia.
- Se ha hecho constar explícitamente cuando un capítulo, sección, demostración,...sea una adaptación casi literal de alguna fuente existente.

Y, acepta que, si se demostrara lo contrario, se le apliquen las medidas disciplinarias que correspondan.

Agradecimientos

Quisiera expresar mi más sincero agradecimiento a todas las personas que han hecho posible la realización de este Trabajo de Fin de Máster. En primer lugar, a mis tutores académicos, José Ameijeiras Alonso y Paula Saavedra Nieves, por su orientación y apoyo a lo largo de este proceso.

Asimismo, extiendo mi gratitud a mis tutores de empresa en INDITEX: Belén María Fernández de Castro, David Barrientos Guillén y Lucas Longueria Mariño. Ellos me han ayudado a lo largo de todas las prácticas, orientándome y resolviendo mis dudas con dedicación y paciencia. Su apoyo constante fue fundamental para el desarrollo de este trabajo.

Además, agradezco al resto de mis compañeros del departamento de *Business Analytics*: Candela Abejón Fuertes, David Polo Tascón y Alberte Dapena Mora por su colaboración y apoyo diario. Sus conocimientos y experiencia fueron esenciales para la realización de este trabajo.

No puedo dejar de mencionar a aquellos que me proporcionaron formaciones cruciales para comprender los procesos internos de la empresa. A David Vila Álvarez, por su enseñanza en el uso de aplicaciones internas y la creación de informes, a Natalia Vallejo Paredes, por la formación en transporte, y a Saúl Varela Vázquez, por la formación en distribución. Su paciencia y dedicación fueron fundamentales para mi aprendizaje y la correcta ejecución del proyecto.

Finalmente, agradezco a todo el equipo de INDITEX por darme la oportunidad de realizar este TFM en su empresa y por el apoyo brindado durante toda mi estancia.

A todos, muchas gracias.

Índice general

Resumen xi

Prefacio xiii

1. Introducción al problema 1
1.1. Contextualización del problema dentro de la cadena de suministro . . . 1
1.2. Descripción de las Variables del Estudio . . . 3
1.2.1. Análisis exploratorio preliminar . . . 4
1.3. Motivación para el uso de estas tablas . . . 10

2. Preprocesado de los datos y análisis exploratorio 13
2.1. Examen, filtrado y preprocesamiento de datos . . . 13
2.1.1. Filtrado Inicial Común . . . 14
2.1.2. Filtrado de la tabla a nivel de entrega: `fact_pedido` . . . 14
2.1.3. Filtrado de la tabla a nivel de *packing*: `fact_pendiente_transporte` . . . 14
2.1.4. Creación de nuevas variables . . . 15
2.2. Tratamiento de datos faltantes . . . 16
2.2.1. Imputación *Dummy* . . . 18
2.2.2. Imputación por *MICE* . . . 19
2.2.3. Comparativa de los modelos de imputación . . . 22
2.3. Análisis exploratorio . . . 22
2.3.1. Medidas de asociación sobre las posibles variables respuesta . . . 23
2.3.2. Medidas de error sobre la fecha pendiente actual . . . 27

3. Creación de los modelos 33
3.1. Explicación del *LightGBM* . . . 34
3.2. Motivación para la elección del modelo *LightGBM*. . . . 39
3.3. Estructura de los modelo creados . . . 40
3.3.1. Búsqueda de hiperparámetros . . . 40
3.3.2. Selección de variables: Algoritmo genético . . . 42
3.3.3. Búsqueda final de hiperparámetros . . . 45
3.4. Descripción detallada del proceso en los modelos finales . . . 45
3.4.1. Modelo para la variable `dias_hasta_fecha_total_entrado` . . . 45
3.4.2. Modelo para la variable `retraso` . . . 48

4. Medidas de error e interpretación de modelos 49
4.1. Modelo para los Días Hasta la Fecha de Entrada de los Envíos . . . 49
4.1.1. Resultados . . . 49
4.1.2. Interpretabilidad . . . 54
4.2. Modelo para el Retraso de los Envíos . . . 59
4.2.1. Resultados . . . 59

4.2.2. Interpretabilidad 60

5. Conclusiones **65**
5.1. Metodología y Principales Resultados 65
5.2. Discusión de Resultados 66
5.3. Implicaciones Prácticas 66
5.4. Limitaciones del Estudio 66
5.5. Recomendaciones para Futuros Trabajos y Próximos Pasos 67
5.5.1. Recomendaciones para Futuros Trabajos 67
5.5.2. Próximos Pasos 67

A. **69**
A.1. Variables empleadas en los modelos creados 69

B. **77**
B.1. Modelo para el Retraso de los Envíos 77
B.1.1. Resultados 77

Bibliografía **83**

Resumen

Resumen en español

La gestión eficiente de la cadena de suministro ha surgido como un pilar fundamental en la competitividad de las empresas modernas, especialmente en la industria textil, donde la velocidad y la precisión son cruciales. INDITEX ha consolidado su posición líder no solo por su innovación en diseño, sino también por la eficiencia y agilidad de su cadena de suministro. El alcance global de la empresa implica la gestión de una red diversa de proveedores y fábricas distribuidas en múltiples ubicaciones geográficas. La complejidad logística de coordinar la fabricación, el transporte y los tiempos de tránsito hacia los centros de distribución se convierte en un desafío estratégico significativo. En este escenario, cualquier desviación en los patrones de entregas por parte de los proveedores puede tener repercusiones sustanciales en el rendimiento general de las ventas. Por lo tanto, surge la necesidad de anticipar y gestionar proactivamente posibles anomalías en la cadena de suministro. Este trabajo se propone abordar este desafío mediante el desarrollo de un sistema de alertas que, basándose en un extenso análisis exploratorio y empleando modelos predictivos, permita identificar patrones de comportamiento en las entregas de pedidos. La principal meta es proporcionar a INDITEX una herramienta valiosa para anticipar irregularidades en el flujo de suministro, permitiendo la toma de decisiones informadas y la implementación de estrategias preventivas.

English abstract

The efficient management of the supply chain has emerged as a fundamental pillar in the competitiveness of modern companies, particularly in the textile industry, where speed and precision are crucial. INDITEX has solidified its leading position not only through design innovation but also due to the efficiency and agility of its supply chain. The company's global reach involves managing a diverse network of suppliers and factories distributed across multiple geographic locations. The logistical complexity of coordinating manufacturing, transportation, and transit times to distribution centers becomes a significant strategic challenge.

In this scenario, any deviation in delivery patterns by suppliers can have substantial repercussions on overall sales performance. Consequently, there is a need to anticipate and proactively manage potential anomalies in the supply chain. This work aims to address this challenge by developing an alert system that based on exploratory analysis and using predictive models, allows the identification of behavior patterns in deliveries.

The primary goal is to provide INDITEX with a valuable tool to anticipate irregularities in the supply flow, enabling informed decision-making and the implementation of preventive strategies.

Prefacio

En el marco de este estudio, se presenta un prefacio que sirve como preludio a la comprensión detallada del problema que abordaremos y las herramientas fundamentales utilizadas para su resolución. Nos enfrentamos a la complejidad inherente a la gestión de la cadena de suministro, un desafío crítico en la competitividad empresarial, especialmente en el dinámico sector textil. A medida que exploramos la eficiencia y agilidad de la cadena de suministro de INDITEX, identificamos la necesidad apremiante de anticipar y gestionar posibles desviaciones en los patrones de entregas. En este contexto, el prefacio proporciona una visión panorámica del problema, estableciendo las bases para comprender la relevancia de las diferentes variables y herramientas analíticas empleadas en la resolución de esta compleja dinámica logística.

A lo largo de este preámbulo, delinearemos la estructura conceptual que sustenta nuestro enfoque, preparando el terreno para una exploración detallada de las tablas de datos que constituyen los pilares fundamentales en la solución de este desafío logístico. En las tablas empleadas se captura información esencial sobre los tiempos de tránsito, patrones de entrega y otros factores clave. Estas tablas sirven como herramientas fundamentales en el análisis estadístico y la construcción de modelos predictivos que establecen el soporte del sistema de alertas. En este prefacio, se desglosarán tanto el problema central como las tablas utilizadas, proporcionando así un contexto claro para la comprensión del desarrollo y las conclusiones de este estudio.

Antes de adentrarnos en el análisis pormenorizado de las tablas proporcionadas, resulta esencial clarificar conceptos fundamentales que arrojarán luz sobre la complejidad intrínseca del problema que aborda este trabajo de fin de máster. Los envíos de mercancía, estructuradas en entregas identificadas por su número identificador `id_entrega`, contienen artículos únicos salvo variaciones de talla (`id_talla`) y color (`id_color`), respectivamente.

La base de datos empleada tiene tablas que reflejan la evolución de las entregas a través de diversas etapas del proceso logístico. Cada registro, originado en primer lugar por el comprador y posteriormente actualizado por proveedores y personal de transporte, comprende información vital, como la fecha estimada de llegada y las unidades pendientes. Es crucial comprender que, a pesar de que una entrega posea una única fecha pendiente y unidades pendientes, estas unidades no se entregan de forma conjunta. Las entregas se subdividen en *packings*, que son las unidades mínimas de envío (indivisibles), y son configuradas en función del volumen del mismo. Consecuentemente, al analizar la información a nivel de *packing*, obtenemos datos más fiables que al hacerlo a nivel de entrega. Sin embargo, es importante tener en cuenta que la información detallada a nivel de *packing* no está disponible desde el comienzo de la entrega.

Valiéndose de esta información, el proyecto se centra en la predicción de la fecha de llegada de las entregas, considerando diversas características como proveedor, etapa, origen, destino... Para el desarrollo del mismo se emplearon cinco tablas de la base de datos, que se detallarán a continuación, y a las que se les ha cambiado el nombre por cuestiones de privacidad. En todas ellas se refleja información similar (unidades y fechas pendientes de los diferentes envíos de mercancía de INDITEX), no obstante,

poseen claves primarias y niveles de detalle diferentes, así como algunas variables distintas. Estas son, en primer lugar, `fact_pendiente`, `fact_entrado` y `fact_pendiente_entrado`, aunque no incidieron directamente en los modelos y el análisis exploratorio, contribuyeron a la comprensión profunda de los datos, y es por ello por lo que se ha querido hacer mención a ellas en este prefacio. En segundo lugar, `fact_pedido` y `fact_pendiente_transporte`, las cuales se emplearon tanto en el análisis exploratorio como en la creación de los modelos, debido a que estas poseían un conjunto de variables más completo. Consecuentemente, dado que estas últimas serán las variables clave del estudio se comentará su estructura: la primera de ellas se construye a nivel de `id_pedido`, `id_entrega`, , `id_talla` e `id_color`. En contraste, la segunda desciende al nivel de `id_paquete` cuando está disponible. Esta distinción confiere una ventaja considerable al proyecto, pues permitirá el empleo de un enfoque híbrido, usando siempre la información más precisa (trataremos esto con detalle en el Capítulo 1).

Seguidamente, procederemos a detallar minuciosamente las variables empleadas en estas dos tablas clave para el análisis exploratorio y los modelos predictivos:

Tabla fact_envio:

- `id_pedido`: Identificador único del pedido (por ejemplo, 1234567). Un pedido puede tener múltiples entregas.
- `id_entrega`: Identificador único de la entrega (por ejemplo, 1264567). Una entrega puede tener múltiples paquetes.
- `id_articulo`: Identificador único del artículo (por ejemplo, 14854567).
- `id_color`: Identificador único del color (por ejemplo, 250).
- `id_talla`: Identificador único de la talla (por ejemplo, 21).
- `id_seccion`: Identificador único de la sección (por ejemplo, mujer, hombre, niño).
- `dia_from_timestamp`: Fecha de inicio de validez del registro.
- `dia_to_timestamp`: Fecha de fin de validez del registro.
- `uds_entradas_acumuladas`: Unidades acumuladas ingresadas.
- `id_pedido_status`: Identificador del estado del pedido.
- `id_entrega_status`: Identificador del estado de la entrega.
- `es_en_transito`: Identificador de si la entrega está en tránsito (0, "No" y 1, "Sí").
- `id_origen_envio`: Identificador único del origen de la entrega.
- `id_proveedor`: Identificador único del proveedor responsable de la entrega.
- `id_tipo_transporte`: Identificador del tipo de transporte (barco, avión, camión).
- `id_tipo_servicio`: Identificador del tipo de entrega (Express, Priority, Regular).
- `id_incoterm`: Identificador de los *International Commercial Terms (INCOTERMS)* que indican las obligaciones de cada una de las partes en la entrega de mercancía.
- `id_destino_envio`: Identificador único del destino de la entrega.
- `id_centro_distribucion`: Identificador único del centro de distribución de llegada.

- es_destino_externo: Indicador de si la entrega se hará en taller o en centro de distribución.
- es_entrega_directa_taller: Indicador de si la entrega va directamente al taller sin pasar por almacén.
- id_taller: Identificador del taller de destino.
- siguiente_fecha_pendiente_transporte: Próxima fecha de entrega pendiente.
- siguiente_uds_pendientes_transporte: Unidades pendientes de entrega.
- fecha_pedido: Fecha de envío del pedido registrada en el sistema interno de la compañía.
- fecha_servicio: Fecha de servicio en el destino.
- fecha_envio: Fecha de envío.
- fecha_handover: Fecha de entrega al transportista.
- fecha_servicio_destino: Fecha de llegada real al centro de distribución.
- id_comprador: Identificador del responsable de la negociación con proveedores.
- es_envio_muestras: Identificador de si el envío es de muestras.
- id_centro_compra: Identificador único del centro de compra.
- id_cadena: Identificador único de la cadena.
- id_campaña: Identificador único de la campaña.
- es_fecha_servicio_dm_confirmada: Identificador de si la aplicación de transportes tiene la entrega confirmada.

Tabla fact_pendiente_transporte:

Esta tabla comparte las siguientes variables con la tabla anterior, por lo que no se volverán a explicar: id_pedido, id_entrega, id_articulo, id_color, id_talla, id_seccion, siguiente_uds_pendientes_transporte, siguiente_fecha_pendiente_transporte, uds_entradas_acumuladas, id_entrega_status, id_origen_envio, id_destino_envio, id_centro_distribucion, es_entrega_directa_taller, id_incoterm, id_tipo_transporte, id_tipo_servicio, id_proveedor, fecha_servicio, fecha_envio, fecha_handover, fecha_servicio_destino, id_centro_compra, id_cadena e id_campaña.

Únicamente se describirán aquellas que sean nuevas:

- id_paquete: Número identificativo del paquete. Esta es la unidad mínima de entrega (indivisible).
- id_tipo_paquete_entrado: Identificador del tipo de entrada del *packing*.
- id_tipo_transporte_envio: Identificador del tipo de transporte.
- es_distribucion_confirmada: Identificador de si el envío ha sido confirmado (1 o 0).
- id_tipo_pedido_muestras: Indicador si el envío es de muestras, normal o mixto.

- `id_envio_status`: Estado del envío, hace referencia a la parte del proceso en la que está el envío.
- `id_taller`: Identificador del taller al que va el envío.
- `fecha_estimada_servicio_destino`: Fecha estimada de llegada por transportes.
- `fecha_servicio_destino_dm`: Fecha estimada de llegada por una aplicación de transportes.
- `fecha_servicio_destino_programada`: Fecha estimada de llegada a partir de la fecha de programación de la información de transporte de otra aplicación.
- `mejor_fecha_envio`: Fecha de la mejor fecha de entrega hasta la fecha para el envío.
- `es_envio_entrado_multiples_fechas`: Indica que la entrada de las unidades de esa entrega se ha producido en días distintos.
- `id_entrega_crossdocking`: Indica que la entrega ha pasado por *cross-docking*.
- `es_contenedor_aduana`: Es un dato de transporte a nivel entrega que indica si el contenida de la entrega está parado en la aduana y sometido a inspección.
- `es_fecha_servicio_programada_confirmada`: Indica si la información de la aplicación de transporte está confirmada.
- `primera_fecha_formalizada`: Fecha de la primera formalización del pedido.
- `primera_fecha_borrador_pedido`: Fecha del primer borrador del pedido.
- `fecha_estimada_taller`: Fecha estimada de llegada a taller.
- `fecha_estimada_taller_transporte`: Fecha programada de llegada a taller según la aplicación de transporte.
- `fecha_asignacion_proveedor`: Fecha en la que se asignó por primera vez el proveedor al pedido.
- `fecha_handover_proveedor`: Fecha indicada por el proveedor en relación a cuando pretende entregar la mercancía al transportista.
- `fecha_handover_forwarder`: Fecha indicada por el *forwarder* en relación a cuando puede recibir la mercancía.
- `fecha_handover_real`: Fecha real en la que se produce la entrega de la mercancía al transportista.
- `fecha_cita_servicio_destino`: Es la fecha estimada de llegada a almacén determinada en base a los sistemas de transporte.
- `fecha_horario_servicio_destino`: Es la fecha programada de llegada a almacén en el sistema de información de transporte.
- `es_entrega_cerrada`: Indicador de si la entrega está cerrada.

Una vez expuestas las variables de las tablas empleadas en este trabajo, procederemos a un resumen del contenido por capítulos. En el primer capítulo, introduciremos el problema, proporcionando su contexto y explorando brevemente las variables a utilizar. El Capítulo 2 se dedicará al procesamiento de los datos, desde el filtrado hasta la creación de nuevas variables, y continuará con el análisis exploratorio, incluyendo medidas de correlación. También se definirán las métricas de error para el algoritmo y se abordará el problema de los datos faltantes, incluyendo una explicación del algoritmo *MICE* utilizado para la imputación de datos faltantes. En el Capítulo 3, describiremos la construcción del modelo, detallando el método de selección de variables, la búsqueda de hiperparámetros y el modelo *LightGBM*.

El Capítulo 4 se centrará en las métricas de error y en la interpretación de los modelos, proporcionando una visión clara de su rendimiento. Finalmente, el Capítulo 5 resumirá las conclusiones principales, destacando los hallazgos clave y proporcionando recomendaciones para futuras investigaciones.

Con esta estructura, buscamos ofrecer una comprensión integral del proceso de análisis y modelado de datos, facilitando así una apreciación profunda de los métodos y resultados presentados.

Capítulo 1

Introducción al problema

Tras el prefacio, que establece las bases para la investigación, este primer capítulo se adentra en la problemática específica que se propone abordar en el Trabajo de Fin de Máster: la predicción de fechas de entrada de las entregas en los centros de distribución y talleres de INDITEX. Este desafío, situado en el núcleo de la eficiencia operativa de la cadena de suministro, se presenta como un paso hacia la mejora de los procesos de distribución, contribuyendo a la competitividad de INDITEX en el sector textil global.

La importancia de anticipar las fechas de entrega radica en su impacto directo sobre la capacidad de la empresa para gestionar con eficacia el inventario, prever la demanda de transporte para el reparto a tienda y, en última instancia, satisfacer la demanda del consumidor de manera oportuna. La complejidad de esta tarea se ve acentuada por las variabilidades inherentes a los procesos logísticos, incluyendo la diversidad de proveedores, las rutas de transporte, y los imprevistos en cada etapa del proceso de entrega.

En este capítulo, se detalla el problema que se busca resolver, situándolo dentro del contexto operativo de INDITEX y destacando su importancia para la estrategia de distribución de la compañía. Además, se establecerá el objetivo principal de esta investigación: desarrollar unos modelos predictivos capaces de anticipar las fechas de entrada de las entregas en los centros de distribución o talleres, apoyándose en un análisis exhaustivo de las tablas de datos descritas en el prefacio. Estos modelos buscarán no solo mejorar la precisión en las predicciones actuales, sino también ofrecer *insights* valiosos para la toma de decisiones estratégicas en la gestión de la cadena de suministro.

Consecuentemente, este capítulo sienta las bases para una comprensión integral del desafío logístico a enfrentar. Así, mientras que el prefacio ofrece una panorámica del contexto del estudio, este primer capítulo introduce de lleno al lector en la problemática específica a abordar, marcando el inicio de nuestra exploración detallada hacia una solución para el sistema de alertas de la cadena de suministro de INDITEX.

1.1. Contextualización del problema dentro de la cadena de suministro

Dentro del universo operativo de INDITEX, el seguimiento de los envíos de productos terminados desde el proveedor hasta el centro de distribución representa una etapa bastante relevante en la cadena de suministro de la empresa. No obstante, esta fase es tan solo una parte de un proceso más amplio y complejo que se inicia mucho antes de que las prendas estén listas para ser enviadas y continúa hasta

que llegan a manos del consumidor final.

El proceso comienza con un trabajo de investigación y análisis de tendencias de moda, donde se capturan las preferencias emergentes de los consumidores y se anticipan las demandas del mercado. Sobre esta base de conocimiento, el equipo de diseño de INDITEX da vida a las colecciones, creando los diseños para la campaña en cuestión. Este proceso creativo culmina en la selección de materiales y proveedores, una etapa donde se establecen las alianzas estratégicas que garantizarán el funcionamiento adecuado de la cadena de suministro.

Una vez definidos los diseños y seleccionados los materiales, se procede a la compra de materias primas (en caso de que sea necesario), seguida de la fabricación de muestras. Estas muestras son esenciales para validar la calidad y el diseño final de cada prenda, sirviendo como referencia para la producción masiva.

Es en este momento donde entra la etapa de envío de producto terminado desde los proveedores hasta los centros de distribución de INDITEX. En dicha etapa se lleva a cabo un seguimiento detallado que asegura que cada prenda llegue en el momento adecuado y en perfecto estado, lista para su próxima fase. Este control es vital, ya que cualquier retraso o problema puede impactar significativamente en toda la cadena de valor.

Una vez que las prendas llegan a alguno de los centros de distribución primarios, se inicia un complejo proceso de gestión de inventario, donde se prepara y organiza el stock para su envío a los distintos destinos. Este es un paso crítico que determina la rapidez con la que los productos pueden ser puestos a disposición de los centros de distribución secundarios y tiendas.

La etapa final del proceso es la distribución a tiendas, un momento clave donde se materializa el esfuerzo de toda la cadena de suministro. Aquí, la logística y la estrategia de distribución juegan un papel fundamental para garantizar que las tiendas reciban los productos adecuados en el momento preciso, permitiendo a INDITEX mantener su promesa de ofrecer moda de calidad al consumidor. La venta al cliente marca el punto final de este meticuloso proceso, donde las prendas finalmente alcanzan su destino y cumplen su propósito.

Este panorama detallado permite apreciar la etapa de envíos de producto terminado no solo como un componente muy relevante en sí mismo sino también como parte integral de un flujo operativo más extenso, diseñado para mantener la agilidad y eficiencia que caracterizan a INDITEX en el competitivo mercado de la moda.

Transicionando desde la visión panorámica de la totalidad del proceso logístico de INDITEX, nos adentramos específicamente en la fase de envío de producto terminado, un eslabón fundamental que conecta la producción meticulosa con la estratégica distribución de la empresa. Este segmento del proceso representa un punto de inflexión donde la calidad y precisión alcanzadas en las etapas previas se ponen a prueba frente a los desafíos logísticos y las variables externas. Es aquí donde la teoría se encuentra con la práctica, y los planes cuidadosamente trazados se ejecutan para trasladar las prendas desde los centros de producción hasta los nodos centrales de la red de distribución de INDITEX.

Este proceso se inicia con la grabación del pedido en las aplicaciones internas de la compañía, donde el comprador especifica las fechas estimadas de entrega por parte del proveedor y la fecha requerida de servicio en el Centro de Distribución. Este paso es crucial para establecer el cronograma inicial y asegurar la fluidez del resto de la cadena logística. A continuación, se procede con la generación del *booking* por parte del proveedor, una etapa fundamental que implica la reserva del espacio necesario en el medio de transporte elegido, adaptándose a la urgencia y naturaleza del pedido. Para garantizar la eficacia de este proceso, se establecen plazos específicos para el *booking* dependiendo del modo de

transporte: 5 días antes de la entrega al trasportista para el transporte aéreo, 3 días para el terrestre y 10 días para el marítimo. Esta anticipación es vital para coordinar adecuadamente los recursos logísticos y evitar contratiempos.

Paralelamente a la generación del *booking*, se realiza el *packing*, una tarea que, aunque teóricamente precede o se realiza simultáneamente al *booking*, en la práctica a veces ocurre de manera algo posterior. Esta flexibilidad temporal refleja la adaptabilidad del sistema logístico de INDITEX ante las variabilidades del proceso de producción y preparación de las prendas. Una vez establecido el *booking*, el papel del *forwarder* se vuelve esencial. Este agente logístico no solo se encarga de asegurar el espacio de la carga en el transporte sino también de diseñar la ruta óptima de envío. Esta planificación incluye la evaluación de las condiciones meteorológicas, políticas y de infraestructura de transporte para elegir el itinerario que minimice los riesgos y los tiempos de tránsito. El *forwarder* también gestiona la documentación necesaria para el tránsito internacional, incluyendo los trámites aduaneros, lo que garantiza un flujo ininterrumpido de las mercancías a través de las fronteras.

Con la mercancía en tránsito, se realiza el *handover* por parte del proveedor al transportista y, posteriormente, se lleva a cabo el embarque, que fija una ruta y fecha de entrega específica. A la llegada al lugar de destino, el operador logístico establece la fecha de llegada al centro de distribución / taller, facilitando la integración de la nueva mercancía en el sistema de distribución de INDITEX.

El enfoque estratégico de INDITEX en su cadena de suministro se evidencia en la distinción entre el circuito largo y el circuito corto. El circuito largo, con proveedores en países como Bangladesh o China, se caracteriza por costos de producción más bajos pero con mayores tiempos de tránsito, que pueden variar entre 25-30 días por mar y 5-12 días por aire, dependiendo del servicio seleccionado. En contraste, el circuito corto, que aprovecha proveedores en proximidad a los mercados clave como Turquía o Marruecos, permite una respuesta ágil a las tendencias de moda, con tiempos de tránsito terrestre de entre 2 y 9 días. Esta diferenciación estratégica subraya la capacidad de INDITEX para adaptarse dinámicamente a las necesidades del mercado.

Seguidamente, se presentará una descripción general de las variables involucradas en el desarrollo del problema tratado.

1.2. Descripción de las Variables del Estudio

En esta sección se describen las variables del estudio, clasificadas en cinco categorías según su propósito y características: identificación de pedidos, temporalidad y fechas, logística y transporte, gestión y responsabilidades, y contexto comercial. A continuación, se proporciona una descripción detallada de cada categoría:

La información de identificación de pedidos abarca aquellas variables que permiten identificar de forma única cada pedido, entrega y artículo involucrado en el proceso. Estas variables son cruciales para realizar un seguimiento preciso del estado y la logística de las entregas, permitiendo supervisar el movimiento de los productos desde su origen hasta el destino final. Esta información incluye códigos de pedido, números de serie de productos y estados actuales de las entregas.

Las variables relacionadas con información temporal y fechas incluyen aquellas que capturan datos sobre las fechas y los intervalos de tiempo que resultan relevantes para el seguimiento del flujo logístico. Estas fechas incluyen, entre otras, el momento del envío, la llegada al destino, las fechas de servicio y otras marcas temporales que permiten establecer líneas de tiempo y cronogramas en la gestión de las entregas. Las variables de información logística y de transporte se centran en los aspectos prácticos de la entrega de mercancías. Incluyen datos como el lugar de origen y destino, el tipo de transporte

utilizado, las condiciones de entrega y otros factores logísticos. Esta información es vital para optimizar las rutas de entrega y gestionar las condiciones en las que se transportan los productos, asegurando así la eficiencia en la cadena de suministro.

En cuanto a la información de gestión y responsabilidades, esta categoría incluye variables relacionadas con el manejo de pedidos y entregas desde una perspectiva administrativa. Estas variables reflejan las responsabilidades en la negociación con proveedores, los centros de compra y otros actores involucrados en la cadena de suministro. Permiten una adecuada gestión de la relación entre todas las partes interesadas y una mejor planificación de las operaciones.

Finalmente, la información contextual y comercial aporta datos adicionales sobre el contexto en el que se realizan los pedidos y las entregas. Las variables en esta categoría abarcan aspectos como la marca, la campaña a la que pertenece un producto y la sección de productos a la que corresponde. Este contexto proporciona información valiosa para comprender mejor los factores comerciales que pueden afectar el proceso logístico.

Cabe destacar que este trabajo se centrará en el análisis de dos variables respuesta: el retraso, que se define como la diferencia entre la fecha proporcionada por los organismos logísticos y la fecha real, y los días hasta la entrada real, que representan el tiempo hasta el momento en que la mercancía llega efectivamente al destino final. Ambas variables permiten estimar con precisión la fecha prevista de entrada (dada la previsión de los organismos logísticos), pero fueron analizadas por separado debido a sus diferentes interpretaciones.

El análisis del retraso es esencial para entender sus causas en las entregas y determinar las etapas o situaciones del proceso logístico donde se producen estos retrasos. Por otro lado, los días hasta la entrada real nos permiten identificar los factores que afectan la cantidad de días necesarios para que la mercancía esté disponible en el centro de distribución. Ambas interpretaciones son importantes para mejorar la eficiencia logística: la primera ayuda a detectar posibles problemas en el proceso de entrega, mientras que la segunda brinda información valiosa para planificar mejor los envíos de mercancía con antelación.

En el siguiente apartado se llevará a cabo un análisis exploratorio preliminar de estas variables para identificar patrones y tendencias relevantes. Este análisis nos ayudará a comprender mejor las relaciones entre las distintas categorías de variables y los dos tipos de respuesta: retraso y días hasta la entrada real.

1.2.1. Análisis exploratorio preliminar

A continuación, presentamos un análisis de los gráficos de cajas correspondientes a las variables clave en nuestros modelos predictivos, agrupadas en las categorías mencionadas previamente: identificación de pedidos, logística y transporte, gestión y responsabilidades, información temporal y fechas, y contexto comercial. Nos centraremos en su influencia sobre las dos variables respuesta del estudio y separaremos la información a nivel de *packing* de la información a nivel de entrega empleando la variable `nivel_entrega`, la cual toma valor 1 si la información está a nivel de entrega y 0 en caso contrario. Aunque no se abordan todas las variables del conjunto de datos, se destacan las que ofrecen conclusiones más significativas. Por motivos de confidencialidad de INDITEX, los ejes de estos gráficos no se mostrarán, pero el análisis revela varios hallazgos importantes.

Variables de Gestión y Responsabilidades

Las Figuras 1.1 y 1.2 ilustran cómo el comprador de la mercancía influye tanto en el retraso como en los días hasta la entrega total. Las leves diferencias en los retrasos pueden atribuirse a las distintas

formas de interactuar con los proveedores, así como a la especialización de ciertos compradores en determinados proveedores, orígenes o tipos de prendas, que pueden conllevar una mayor probabilidad de retraso.

La Figura 1.2 refuerza esta afirmación, ya que muestra cómo cada comprador obtiene un promedio diferente de días hasta la entrega, probablemente debido a que negocian con proveedores ubicados a distintas distancias.

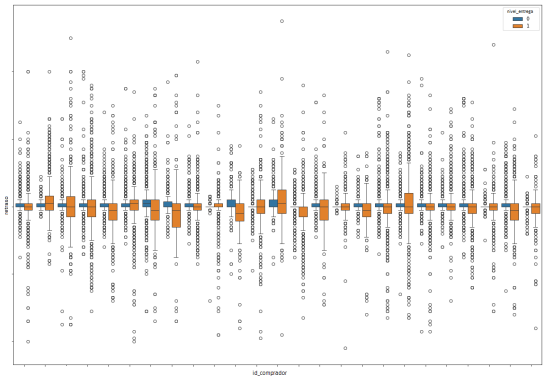

Figura 1.1: Gráfico de cajas del retraso agrupando por comprador

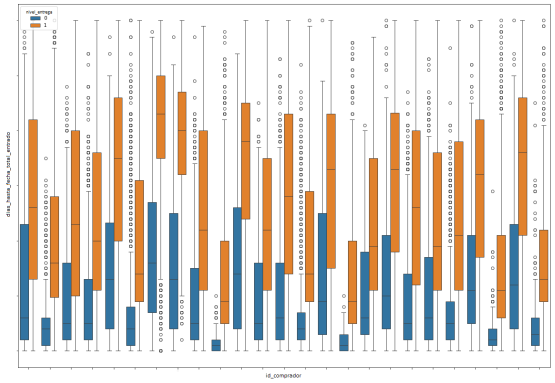

Figura 1.2: Gráfico de cajas de los días hasta la entrega total agrupando por comprador.

Variables de identificación de pedidos

Las Figuras 1.3, 1.5 y 1.7 evidencian cómo los diferentes orígenes, destinos y proveedores influyen en el retraso de los envíos. Asimismo, las Figuras 1.4, 1.6 y 1.8 muestran el impacto de estos factores en el tiempo que tarda la mercancía en llegar al destino final.

Estos gráficos indican que algunos orígenes tienen un trayecto significativamente más largo hacia el destino que otros, y esta diferencia se refleja también en los proveedores que envían desde esos lugares. El tiempo de tránsito de la mercancía es el factor clave que clasifica a los orígenes y proveedores en los circuitos largos y cortos mencionados en la Sección 1.1. Además, estas diferencias se evidencian más claramente cuando se analiza la información a nivel de entrega, ya que hay más días hasta la fecha final del envío y, por ende, mayor incertidumbre.

Es importante destacar que los destinos con menos retrasos pueden ser puertos o infraestructuras mejor gestionadas y dimensionadas, lo que facilita el flujo de mercancías sin interrupciones. Estos destinos deben considerarse más cuidadosamente al realizar un pedido. Otra posibilidad es que estos destinos estén ubicados en el propio país de fabricación de la mercancía, lo que evita complicaciones aduaneras al no salir del país de origen.

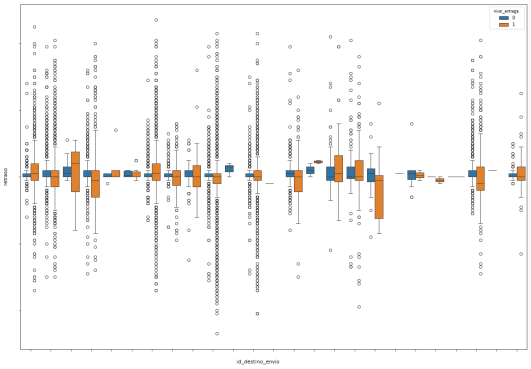

Figura 1.3: Gráfico de cajas del retraso agrupando por destino de envío.

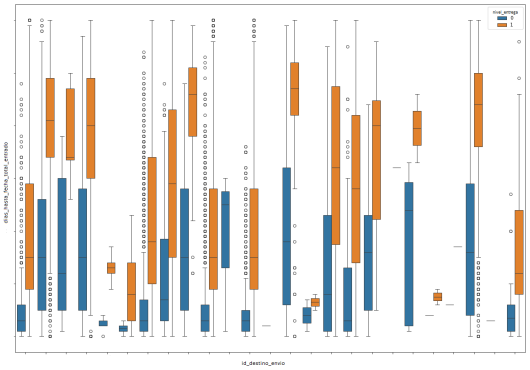

Figura 1.4: Gráfico de cajas de los días hasta la entrega total agrupando por destino de envío.

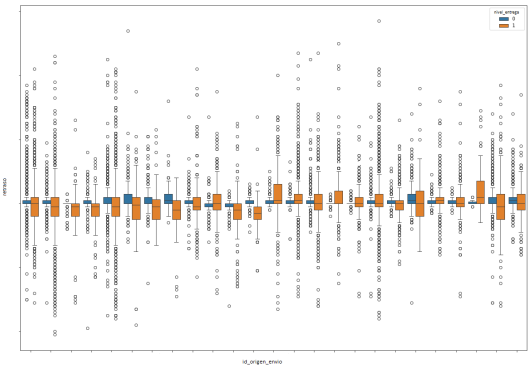

Figura 1.5: Gráfico de cajas del retraso agrupando por origen de envío.

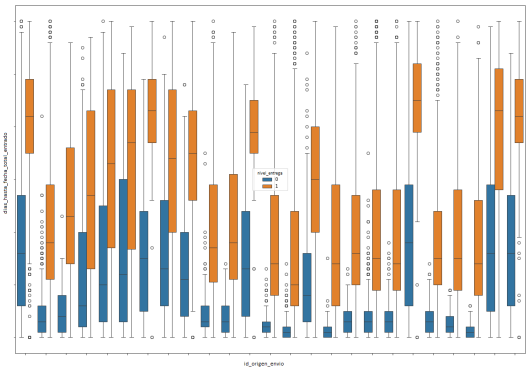

Figura 1.6: Gráfico de cajas de los días hasta la entrega total agrupando por origen de envío.

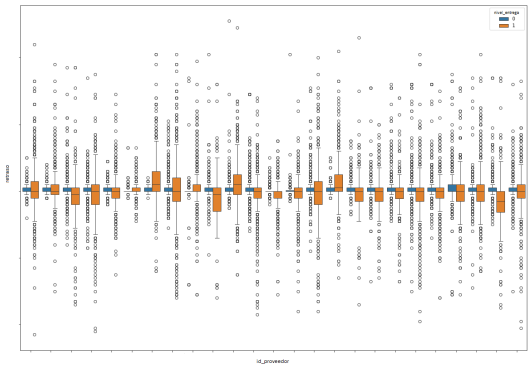

Figura 1.7: Gráfico de cajas del retraso agrupando por proveedor.

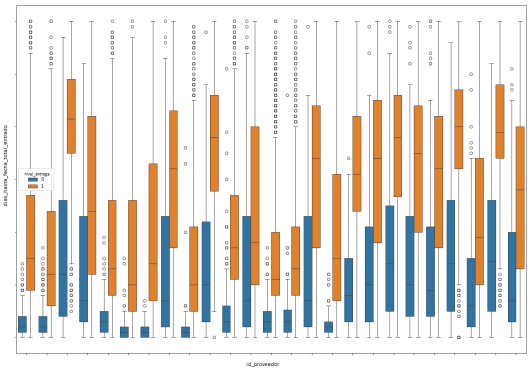

Figura 1.8: Gráfico de cajas de los días hasta la entrega total agrupando por proveedor.

Variables de Logística y Transporte

Las Figuras desde 1.9 hasta 1.20 exploran el impacto de las variables de logística y transporte en el retraso y en los días hasta la entrega total. Específicamente, los *INCOTERMs* y los centros de distribución muestran cómo estos aspectos logísticos afectan los plazos de entrega.

En las Figuras 1.9 y 1.10 se puede ver como dependiendo de qué entidad sea la responsable de la mercancía el retraso sobre el envío varía bastante. Asimismo, también parece claro que ciertos *INCOTERMs* se emplean más a menudo para envíos con más días de tránsito.

Esta correlación sugiere que los INCOTERMS, al definir responsabilidades, costos y riesgos entre compradores y vendedores, impactan de manera significativa en la logística de envío. Por lo tanto, la selección adecuada de estos términos es crucial para optimizar los tiempos de entrega y minimizar posibles retrasos.

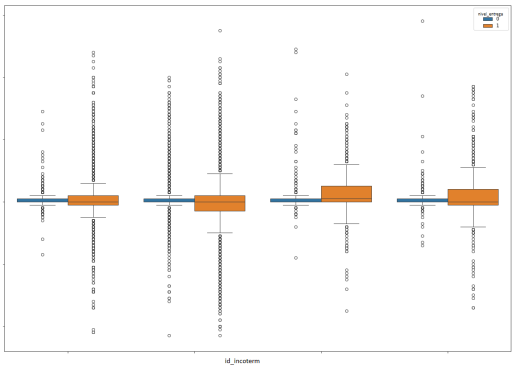

Figura 1.9: Gráfico de cajas del retraso agrupando por el $INCOTERM$.

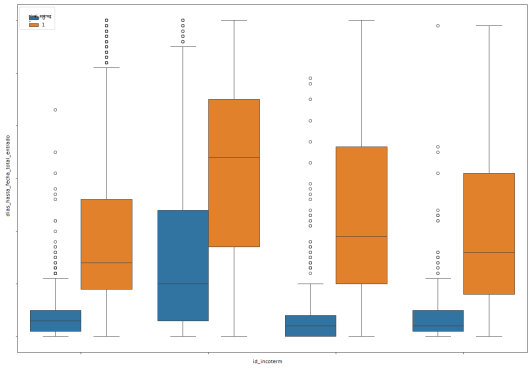

Figura 1.10: Gráfico de cajas de los días hasta la entrega total agrupando por el $INCOTERM$.

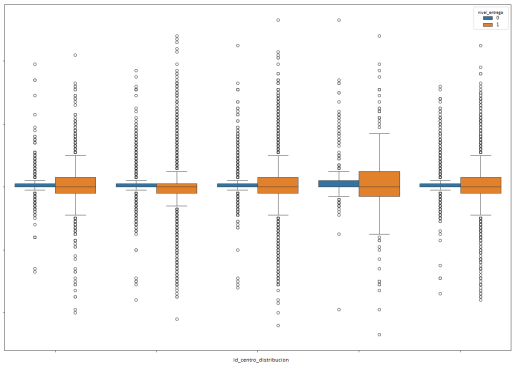

Figura 1.11: Gráfico de cajas del retraso agrupando por centro de distribución.

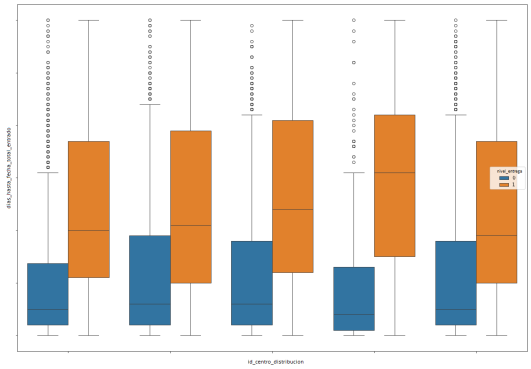

Figura 1.12: Gráfico de cajas de los días hasta la entrega total agrupando por centro de distribución.

El análisis del tipo de servicio, representado en las Figuras 1.13 y 1.14, revela que el nivel de prioridad del envío afecta los tiempos de entrega, tal como cabía esperar, pues los envíos urgentes tendrán

una menor probabilidad de retraso que los envíos regulares.

Además, la Figura 1.14 confirma que uno de los servicios de transporte presenta, en promedio, tiempos hasta la entrega bastante más reducidos que el resto. Tal efecto podría explicarse por la existencia de modalidades de envío rápidas, como el aéreo urgente, que reducen los plazos de tránsito. Este hallazgo subraya cómo ciertos servicios, mediante la priorización de la rapidez, logran minimizar los periodos de entrega a costa de aumentar los costes.

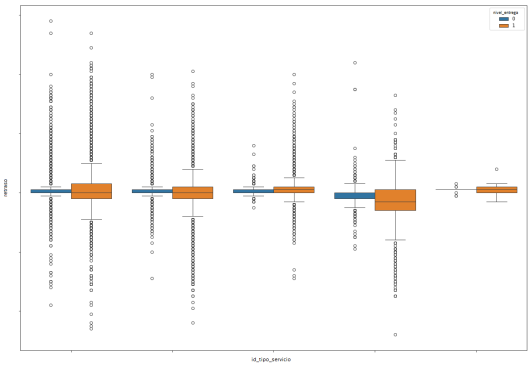

Figura 1.13: Gráfico de cajas del retraso agrupando por tipo de servicio.

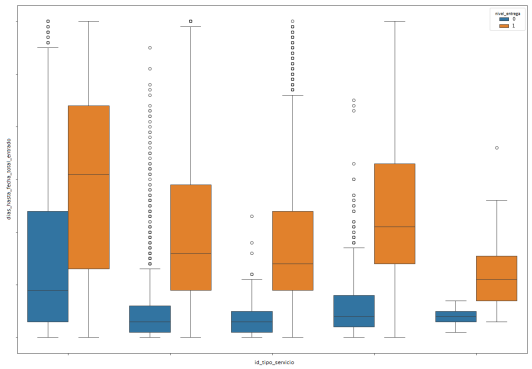

Figura 1.14: Gráfico de cajas de los días hasta la entrega total agrupando por tipo de servicio.

El estatus del envío, mostrado en las Figuras 1.15 y 1.16, es aquel que indica la etapa en la que se encuentra el envío. Es evidente que a medida que el estatus del pedido se acerca a la etapa final el retraso se reduce, tal como se aprecia en las figuras anteriores.

Es por este motivo que al analizar la variable estatus mediante la Figura 1.15, se identifican dos categorías con tiempos de retraso marcadamente superiores en comparación con las otras dos. Esta observación concuerda con la lógica operacional, dado que la variable clasifica las entregas según su etapa en el proceso de envío. Por ende, es esperable que las fases iniciales presenten mayores demoras, reflejo de la complejidad inherente a los primeros pasos del proceso logístico.

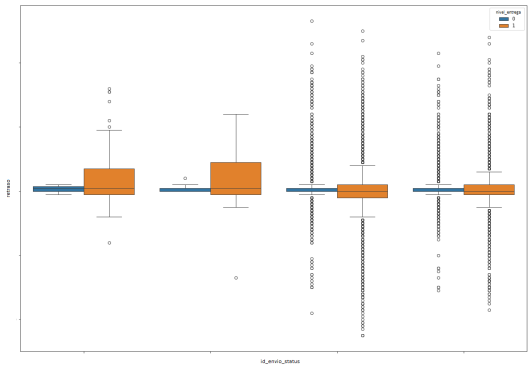

Figura 1.15: Gráfico de cajas del retraso agrupando por estatus del envío.

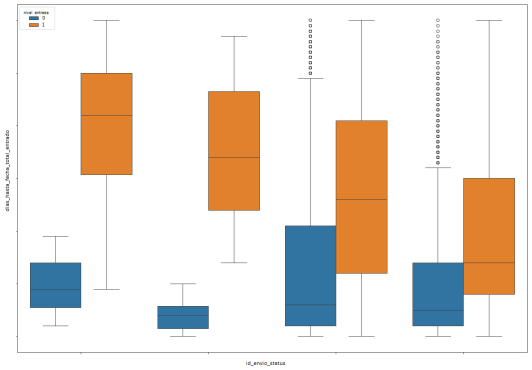

Figura 1.16: Gráfico de cajas de los días hasta la entrega total agrupando por estatus del envío.

La elección del tipo de transporte, como se ilustra en las Figuras 1.17 y 1.18, influye bastante en

el retraso y en la anticipación a la entrega total, pues mientras que el barco requiere una anticipación muy grande (aumentando la incertidumbre sobre la fecha de llegada), el avión requiere una anticipación mínima. El camión, sin embargo se encuentra en una situación intermedia, pero al contrario que el avión o el barco, es más probable que puedan esperar si la producción de la mercancía se retrasa, mientras que el barco y el avión no esperarán por retrasos en la producción.

Estas Figuras demuestran, por tanto, que la elección del transporte no solo afecta la celeridad con la que se procesan los envíos, sino también la predictibilidad y la gestión del tiempo hasta la entrega final. La distinción entre los tipos de transporte, por consiguiente, es un factor determinante en la planificación eficaz y en la reducción de los retrasos en la cadena de suministro.

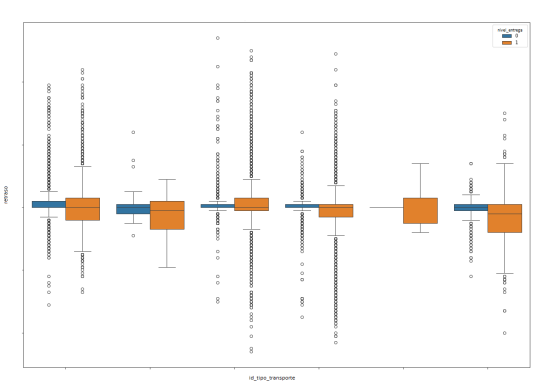

Figura 1.17: Gráfico de cajas del retraso agrupando por tipo de transporte.

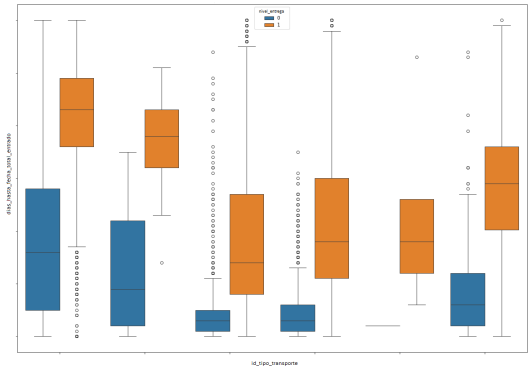

Figura 1.18: Gráfico de cajas de los días hasta la entrega total agrupando por tipo de transporte.

La variable de entrega directa al taller presenta diferencias significativas tanto en los retrasos como en los días hasta la entrega total, como se observa en las Figuras 1.19 y 1.20. Se evidencia que cuando la mercancía va al taller, el retraso es ligeramente mayor, independientemente de si la información se analiza a nivel de *packing* o entrega, y aumenta la variabilidad. Esto puede indicar que la información sobre la llegada de envíos al taller es menos fiable.

También se observan diferencias en los días hasta la entrega, lo que podría deberse a que las prendas que tienen un trayecto más largo requieren un tratamiento en el taller que otras no necesitan. Alternativamente, es posible que ciertos orígenes devuelvan las prendas en diferentes estados, lo que puede requerir o no un paso por el taller.

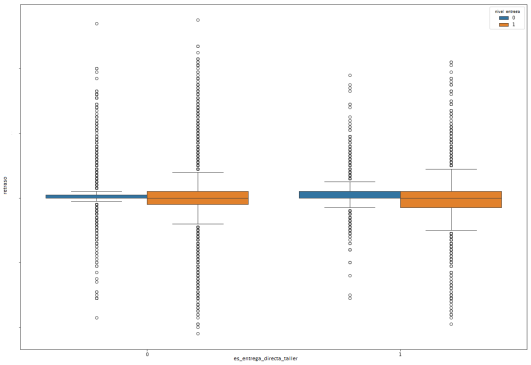

Figura 1.19: Gráfico de cajas del retraso agrupando por entrega directa al taller.

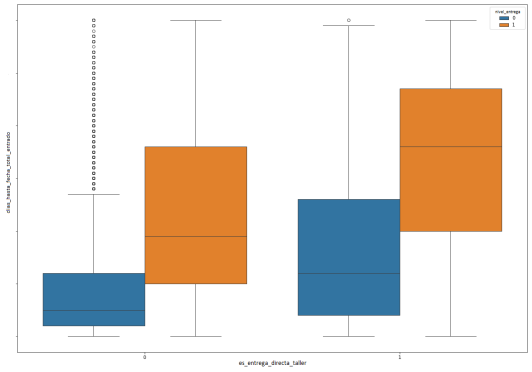

Figura 1.20: Gráfico de cajas de los días hasta la entrega total agrupando por entrega directa al taller.

En esta sección se ha realizado un análisis exploratorio de las variables clave del estudio, agrupadas en distintas categorías. A través de gráficos de cajas, hemos identificado los factores que influyen en los retrasos de los envíos y en los días hasta la entrega total. También hemos resaltado cómo el rol de ciertas variables afectan los plazos de entrega y a la variabilidad en los tiempos de tránsito.

En la siguiente sección, presentaremos en detalle las tablas de datos utilizadas en este estudio y discutiremos las razones que motivaron su selección, destacando su relevancia para el análisis y su capacidad para proporcionar información significativa sobre la cadena de suministro y la logística de los envíos.

1.3. Motivación para el uso de estas tablas

En el desarrollo de este Trabajo de Fin de Máster, la selección y análisis de las fuentes de datos juegan un papel crucial. Al comienzo de este proceso, se ha realizado un examen exhaustivo de varias tablas de datos, específicamente las tablas `fact_pendiente`, `fact_entrado`, y `fact_pendiente_entrado` (mencionadas en el prefacio). Esta última, que representa la unión de los datos de pendiente y entrado, inicialmente parecía ser la opción más conveniente debido a su estructura integrada. Sin embargo, pronto se puso de manifiesto que carecía de información relevante para llevar a cabo el estudio, como los detalles sobre el comprador responsable del envío, las fechas de embarque y de *handover*, entre otros aspectos fundamentales.

Ante la limitación observada, se optó inicialmente por utilizar la tabla `fact_pedido`. Sin embargo, al analizar en detalle su estructura y el mecanismo de registro de datos, se evidenció la necesidad de una granularidad más fina que la ofrecida por el nivel de entrega. El problema radica en que, para una entrega que incluye múltiples *packings*, solo se registra una única fecha pendiente y un conjunto de unidades pendientes. Esto puede dar lugar a problemas, pues parece indicar que todas las unidades pendientes se entregarán en esa fecha, cuando en realidad los *packings* pueden entregarse en fechas distintas.

Para capturar con precisión el momento en que la mercancía efectivamente llega, es imprescindible que la clave primaria de la tabla se defina en el nivel más detallado posible. Se identificó que el nivel de *packing* proporciona la granularidad requerida, funcionando como la unidad básica de registro de llegadas de mercancía y asegurando una relación uno a uno con estas llegadas. Este aspecto, aunque pueda parecer trivial, es muy importante: aproximadamente el 15 % de las entregas incluyen varios *packings*, y es fundamental poder registrar distintas fechas de llegada para cada uno. De no ajustarse la granularidad al nivel de *packing*, y mantenerla en el nivel de entrega, se registraría únicamente una

fecha pendiente, lo que podría generar discrepancias importantes si los *packings* se entregan en fechas considerablemente diferentes.

Para superar las deficiencias de la tabla `fact_pedido` y alcanzar el nivel de detalle requerido, se recurrió a la tabla `fact_pending_transport`, que proporciona información a nivel de *packing*. Aunque esta tabla también presentaba limitaciones, como la ausencia de datos sobre el comprador responsable del envío, se implementó una solución mediante la unión de `fact_pedido` y `fact_pending_transport`. Esta operación de unión enriqueció `fact_pending_transport` con la información relevante de `fact_pedido`, resultando en la tabla definitiva para el análisis (pendiente de ser sometida al preprocesado).

Además, se agruparon los registros de las diferentes tablas por sus identificadores únicos de entrega, artículo y *packing*, (esta última variable solo se empleó en la tabla que lo permite), sumando las unidades pendientes y entregadas a lo largo de los diferentes colores y tallas. Este enfoque permitió eliminar redundancias, considerando que diferentes tallas y colores de un artículo, en caso de ir en la misma entrega (o *packing*), comparten la misma fecha de entrega.

Es importante destacar que, aunque la tabla `fact_pending_transport` ofrece datos a nivel de *packing*, esta información solo se encuentra disponible cuando el proveedor genera el *packing* (ver Figura 1.21), lo que ocurre aproximadamente una semana antes de la entrega en la mayoría de los casos (se hará un breve análisis más adelante). En ausencia de datos a nivel de *packing*, la información a nivel de entrega resulta ser la más fiable. Por lo tanto, se optó por adoptar una metodología híbrida entre *packing* y entrega para seleccionar los datos más fiables para el modelo, priorizando la información a nivel de *packing* cuando está disponible, y recurriendo a los datos a nivel de entrega en su ausencia. Este enfoque permite capturar la dinámica y complejidad de los procesos de entrega con mayor precisión, lo cual es fundamental para la calidad y relevancia de los resultados de este trabajo.

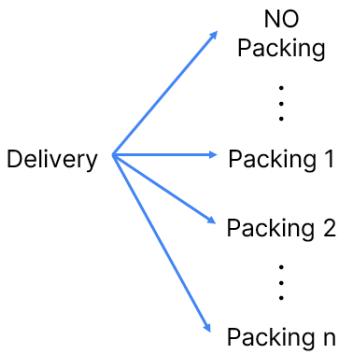

Figura 1.21: Esquema de la configuración de una entrega a lo largo del tiempo en la tabla de `fact_pending_transport`. Comienza sin asignarse información a *packing* y a medida que pasa el tiempo se van delimitando cuantas unidades corresponden en cada *packing* y cual es su fecha estimada de entrega.

En esta sección, se han explicado las motivaciones detrás de cada conjunto de datos, destacando su importancia para abordar los objetivos del estudio.

Capítulo 2

Preprocesado de los datos y análisis exploratorio

En este capítulo, se aborda una de las fases más críticas del trabajo: el preprocesado y el exploratorio. Se tratará en primer lugar el preprocesado de datos, una etapa fundamental para asegurar la calidad y la fiabilidad del conjunto de datos. Este proceso incluyó el examen y filtrado de los datos existentes, donde se identificaron y corrigieron errores, se eliminaron las inconsistencias, y se trataron los valores atípicos. Además, se destaca la importancia de la creación de nuevas variables a partir de los datos existentes, enriqueciendo así el conjunto de datos para un análisis más profundo.

Posteriormente, se profundizó en la exploración y tratamiento de los datos faltantes, un desafío omnipresente en el manejo de datos. Para abordar esta cuestión, se introdujeron dos algoritmos con el objetivo de comparar las diferencias en su desempeño: un algoritmo "*dummy*", y un enfoque más sofisticado que utiliza el algoritmo Multiple Imputation Chained Equations (*MICE*) con *LightGBM*. La comparativa entre estos métodos ofrece una perspectiva valiosa sobre cómo manejar de manera efectiva y eficiente los datos faltantes, resaltando las ventajas y limitaciones de cada enfoque.

El capítulo concluye con una sección dedicada a la exploración exhaustiva de los datos. Esta parte tiene como objetivo ofrecer una comprensión detallada de las características fundamentales del conjunto de datos, empleando técnicas de visualización y análisis estadístico para identificar patrones, relaciones y posibles anomalías. Este análisis exploratorio es esencial para sentar las bases de cualquier análisis posterior, facilitando la toma de decisiones basada en datos y el descubrimiento de información relevante.

A lo largo de este capítulo, se demostrará cómo el preprocesado y la exploración de datos son pasos indispensables que preparan el terreno para análisis futuros, subrayando la importancia de estos procesos para garantizar la integridad y la utilidad de los datos en proyectos de ciencia de datos como el que nos ocupa.

2.1. Examen, filtrado y preprocesamiento de datos

El proceso de filtrado y preprocesamiento de los datos se centró en los conjuntos de datos `fact_pedido` y `fact_pending_transport`, buscando obtener subconjuntos representativos para el análisis. Este proceso se realizó a través de varios pasos detallados a continuación, aplicando criterios específicos para filtrar y preparar los datos adecuadamente.

2.1.1. Filtrado Inicial Común

Al comienzo, aplicamos un conjunto de filtros comunes a ambos conjuntos de datos. Primero, eliminamos los envíos de muestras, utilizando las variables `es_envio_muestras` e `id_tipo_pedido_muestras`. Luego, seleccionamos exclusivamente los envíos de ropa destinados a Zara, empleando las variables `id_cadena` e `id_centro_compra`. A continuación, mantuvimos la última actualización diaria de cada envío para simplificar el seguimiento, lo cual logramos mediante la creación de la variable `dia_from_timestamp`. Finalmente, filtramos por fecha, quedándonos tan solo con los registros cuyo `dia_from_timestamp` estuviera entre el 1 de enero de 2021 y el 1 de junio de 2023, con el fin de evitar períodos como el de la pandemia de COVID-19, que podrían sesgar los datos de estudio.

2.1.2. Filtrado de la tabla a nivel de entrega: `fact_pedido`

La limpieza del conjunto de datos `fact_pedido` implicó pasos detallados para asegurar la precisión de los datos.

Primero, calculamos las unidades recibidas diariamente, basándonos en la diferencia de las unidades acumuladas de un día para otro. Durante este proceso, identificamos anomalías atribuibles a correcciones de errores o devoluciones de mercancía defectuosa, ya que a veces parecía que las unidades salían en lugar de entrar. Por ello, creamos variables separadas para las unidades recibidas y devueltas: `uds_entradas_ac_dia`, `uds_entradas_acumuladas`, `uds_devueltas_ac_dia` y `uds_devueltas_ac`.

A continuación, se eliminaron los registros a nivel de entrega cuya fecha pendiente de transporte (`siguiente_fecha_pendiente_transporte`), fuese el 1 de enero de 1900. Esta es la fecha que se establecía si el pedido ya había sido entregado o si en el momento en el que se generó ese dato aún no se conocía la fecha pendiente. También eliminamos aquellas entregas que nunca generaron un *booking*, usando la variable `id_entrega_status`, que indica el estatus de la entrega, ya que esto indica que el envío fue cancelado y nunca se llegó a enviar la mercancía.

Adicionalmente, eliminamos entregas con un único registro, ya que probablemente se trataba de un pedido que nunca se completó o de una entrega no registrada adecuadamente. Finalmente, descartamos las entregas donde el `dia_from_timestamp` siempre precedía a la `siguiente_fecha_pendiente_transporte`, para eliminar datos no válidos. Esto se debe a que si el dato se registra en una fecha posterior a la fecha en la que el pedido tendría que haber llegado no nos aporta información útil.

2.1.3. Filtrado de la tabla a nivel de *packing*: `fact_pendiente_transporte`

La preparación del conjunto de datos `fact_pendiente_transporte` se realizó mediante los siguientes pasos.

Primero, realizamos una unión con la tabla de `fact_pedido` para sincronizar las entregas filtradas, obteniendo así el conjunto `fact_pendiente_transporte_final`. Esto también nos permitió añadir la columna `id_comprador` de `fact_pedido` que hacía referencia al comprador y que la tabla a nivel de *packing* no tenía. Posteriormente, introdujimos las variables `n_id_color` y `n_id_talla` para contabilizar los diferentes colores y tallas sin perder información al agrupar a nivel de entrega o *packing*. Esto es esencial, ya que las entradas de mercancía dependen de la entrega en la que llegan, no del color o la talla del artículo. Por lo tanto, no tiene sentido tener registros duplicados (una fecha pendiente para cada color y talla de los artículos que vienen en el mismo envío). Sin embargo, con el objetivo

de preservar la información sobre cuántos colores y tallas tenía un determinado artículo, se genera la tabla de esta forma.

Seguidamente, eliminamos entregas con registros atípicos, como aquellas en las que los *packings* lleván a cabo entregas en más de un día o cuya distribución nunca fue confirmada (`es_distribucion_confirmada` = 0), ya que los *packings* se deben entregar en un solo día. También eliminamos las entregas en las que nunca se generó un *packing*, y aquellas que presentan una diferencia de más del 5 % en unidades entregadas con `fact_pedido`, ya que a esta tabla se le han aplicado correcciones para ajustarse mejor a la realidad. Finalmente, para adecuarnos a la estructura de datos requerida por los modelos, eliminamos registros posteriores a la última entrega de unidades, ya que no son útiles para predecir la llegada de las mismas.

A continuación, para abordar el desafío que presentan ciertas entregas, donde, después de haber creado los *packings*, se observa que algunos registros posteriores se siguen poniendo a nivel de entrega (aunque ya existan datos específicos de *packing*), se implementó una solución que reduce sustancialmente este problema. Esta consistió en analizar las últimas unidades pendientes en el nivel de entrega justo antes de la creación del primer *packing*. Posteriormente, decidimos conservar únicamente aquellos registros en el nivel de entrega si la suma de las unidades pendientes a nivel de *packing* hasta dicha fecha era inferior al 95 % de estas últimas unidades pendientes a nivel de entrega. Esto se hizo bajo la premisa de que si la suma no alcanzaba el 95 %, indicaría que aún quedaban *packings* por reflejar, lo cual sería información relevante y necesaria para el proceso, por lo que se debería mantener la información a nivel de entrega.

2.1.4. Creación de nuevas variables

En este apartado, explicamos en detalle el desarrollo de nuevas variables críticas para el análisis y modelado de los datos. Estas variables están diseñadas para capturar aspectos específicos del comportamiento de las entregas, las tendencias de compra, y la planificación de la carga de trabajo en los centros de distribución, así como ajustes en las fechas y unidades pendientes.

En primer lugar introdujimos `es_base_venta` para identificar artículos con compras superiores a 50,000 unidades, señalando así una fuerte inversión de la compañía y un posible seguimiento intensivo del envío. Además, creamos las variables `ID_BF` e `ID_CHINA_NY` para marcar si el envío coincide con fechas cercanas al Black Friday o al Año Nuevo Chino, respectivamente, momentos que pueden alterar los patrones de entrega.

Para identificar posibles problemas en la entrega, establecimos `id_entrega_peligrosa`, que marca registros que, a 3 días o menos de su fecha pendiente, carecen de información de *packing*. Además, con el fin de anticipar la carga de trabajo en los centros de distribución por sección, desarrollamos `porcentaje_pendiente_dhp_cd_seccion`, basándonos en el porcentaje de unidades pendientes respecto a los máximos y mínimos históricos. También calculamos `porcentaje_mediano_entrado_cd_seccion`, que proporciona una perspectiva sobre la carga esperada en los centros de distribución en la fecha de entrega al calcular la mediana del porcentaje de unidades recibidas frente a las unidades pendientes. Creamos variables para días hasta eventos clave, como `dias_hasta_fecha_envio` y `dias_hasta_siguiente_fecha_pendiente_transporte`, para adaptar las fechas al formato tabular requerido por los modelos.

Además, utilizamos variables híbridas (que bajan a nivel de packing si este está disponible), para ajustar y medir la variación en las entregas. Las variables `correccion_fecha`, `maximo_retraso_fecha` y `maximo_adelanto_fecha` capturan los ajustes en las fechas de entrega y las variaciones máximas observadas, lo que proporciona información sobre la estabilidad y fiabilidad de las estimaciones de

entrega. Por otro lado, las variables `uds_pendientes_correccion`, `uds_pendientes_max_incremento` y `uds_pendientes_max_decremento` registran las variaciones en las unidades pendientes y los cambios máximos, proporcionando una visión detallada sobre las variaciones en las unidades pendientes a lo largo del envío. Asimismo, también creamos las variable `nivel_entrega` que es 1 si el registro está a nivel de entrega y 0 en caso contrario (información a nivel de packing).

Por último, calculamos `retraso` y `handover_retraso` para medir los retrasos en la entrega y en el proceso de *handover*, lo que ofrece medidas críticas de la eficiencia y puntualidad de las entregas. La variable `retraso` se utilizará como variable de respuesta en algunos de los modelos creados, tal como se comentó en el Capítulo 1, mientras que `handover_retraso` hace referencia al retraso que existe en el proceso de *handover*, es decir, la diferencia entre la fecha en que el proveedor se compromete a entregar la mercancía y la fecha real de entrega al *forwarder*.

Este conjunto detallado de variables aporta una comprensión profunda de los flujos de entrega, las expectativas de carga de trabajo, y las dinámicas de ajuste en las fechas y unidades pendientes. Al analizar estas variables, los modelos predictivos pueden ofrecer predicciones más informadas y precisas, mejorando sustancialmente la gestión logística y la planificación estratégica.

2.2. Tratamiento de datos faltantes

En el análisis de los datos, nos enfrentamos al desafío de manejar valores faltantes, los cuales se abordaron de manera diferenciada según su naturaleza. Este proceso se divide en dos categorías principales: datos de fechas y el resto de variables.

Las variables de fecha, como `fecha_handover_real` o `fecha_servicio_destino_dm`, presentan características únicas. Estas fechas no están disponibles hasta que se ha completado el *handover* o la mercancía ha llegado al puerto, respectivamente. Sin embargo, dado que están altamente correlacionadas con `dias_hasta_fecha_total_entrado`, que es la variable que pretendemos predecir, decidimos no imputarlas mediante algoritmos para evitar redundancias en la predicción. Además, los valores nulos en estas variables aportan información relevante al modelo porque indican etapas aún no completadas del proceso de entrega. Esto justifica la preferencia por modelos capaces de manejar estos valores nulos directamente, aunque muchos modelos estadísticos tradicionales no están diseñados para trabajar con datos incompletos de este tipo, lo que representa un desafío. Igualmente, cabe destacar que para la variable `dias_hasta_siguiente_fecha_pendiente_transporte` algunos registros si han tenido que ser imputados (ya que esta fecha si se conocía pero no estaba adecuadamente registrada), por lo que se creo la variables `es_fecha_transporte_imputada` que indica si en ese registro se ha imputado la fecha pendiente o no.

En cuanto a las variables que no son de fecha, identificamos valores faltantes en `id_origen_envio`, `id_destino_envio`, `id_tipo_servicio`, e `id_taller`. Dado que `id_taller` presentaba un 94 % de datos faltantes, se decidió eliminar esta variable del análisis. Las otras variables mostraron porcentajes menores de datos faltantes de 2,2 % en el caso del origen, de 14,5 % en el caso del destino, y del 15,5 % en el caso del tipo de servicio de transporte. Estos datos faltantes se pueden deber principalmente a dos posibles motivos: grabado de información erróneo, o que el pedido se haya llevado a cabo con un origen, destino o servicio que actualmente no esté disponible o con el que no se trabaje.

Al continuar con el análisis de los orígenes, destinos y tipos de servicio de transporte se observó que en un pequeño porcentaje de las entregas, alrededor del 2 %, estos valores podían variar durante la misma. Es decir, que una entrega podía decidir inicialmente entregarse a un destino o con un tipo de servicio de transporte y luego tener que cambiar la idea inicial por diferentes imprevistos. Es por ello

que para establecer una relación entre el resultado de la entrega (ya sea retraso, adelanto o llegada a tiempo) y sus causas potenciales, se optó por asignar el valor más reciente observado en estos campos a la totalidad de la entrega. Esta metodología garantiza una correlación directa entre las causas y efectos dentro de los datos de entrenamiento, facilitando así un análisis más preciso.

Asimismo, se realizaron análisis para determinar si existía alguna relación de causalidad entre los valores faltantes entre variables. Según se muestra en la Figura 2.2, concluímos que no existe una relación de causalidad directa. Esto se debe a que el hecho de que falte el valor de una variable en un registro no implica directamente que falte el valor de otra variable en ese mismo registro (aunque pueda influir en la probabilidad de que suceda). Sin embargo, se notó una correlación interesante entre la ausencia del `id_tipo_servicio` y la de `id_origen_envio`, como se detalla en la Figura 2.1. Aunque no se identificó una causalidad directa, la presencia de correlación sugiere una relación entre estas variables que no puede ser menospreciada, y que se puede aprovechar a la hora de llevar a cabo el algoritmo de imputación. Cabe destacar que la correlación mencionada es el estadístico V de Crámer, el cual se emplea habitualmente para medir la asociación entre dos variables categóricas, y se calcula como:

$$V = \sqrt{\frac{\chi^2}{n \cdot (\min\{r, c\} - 1)}},$$

donde χ^2 es el estadístico de sobra conocido, n es el volumen de los datos y r y c son el número de filas y de columnas de la tabla de contingencia respectivamente. Este coeficiente, V, mide la asociación con valores entre 0 y 1. Un valor cercano a 0 indica que no hay asociación, mientras que un valor cercano a 1 indica una fuerte asociación (véase [20]).

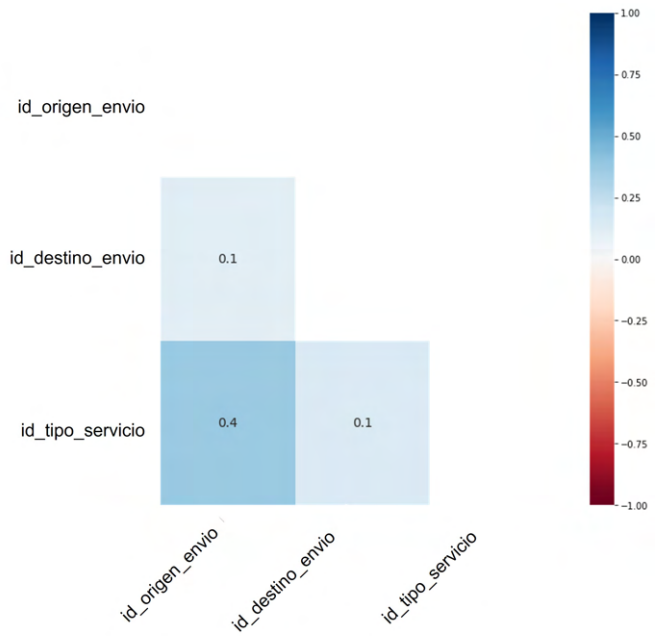

Figura 2.1: Gráfico de calor de las variables con datos faltantes que identifica el estadístico V de Crámer entre los valores faltantes de las variables origen, destino y tipo de servicio de transporte.

Tras este análisis y dado que la eliminación de datos faltantes se consideró una solución subóptima, decidimos imputar estos valores empleando dos métodos diferentes con el fin de compararlos y escoger el más adecuado. Además de imputar los valores en las variables correspondientes, se crearon

nuevas variables que indican si `id_origen_envio`, `id_destino_envio` o `id_tipo_servicio` han sido imputados, tanto en el propio registro (`nombre_variable_imputed_register`), como en algún registro de la entrega (`nombre_variable_imputed_delivery`).

En las siguientes subsecciones se abordarán en detalle ambos métodos de imputación. El primero, denominado por su sencillez *dummy*, es un enfoque simple y directo que asigna un valor razonable a los datos faltantes. Por otro lado, el segundo método, conocido como *MICE* (Multiple Imputation by Chained Equations) junto con *LightGBM*, es una técnica mucho más compleja que utiliza modelos avanzados para predecir los valores faltantes en función de las relaciones entre las características del conjunto de datos. En las siguientes subsecciones se discutirá en profundidad la aplicación de cada método, así como una comparativa de sus resultados, mostrando las claras diferencias en la complejidad y el rendimiento entre ambos.

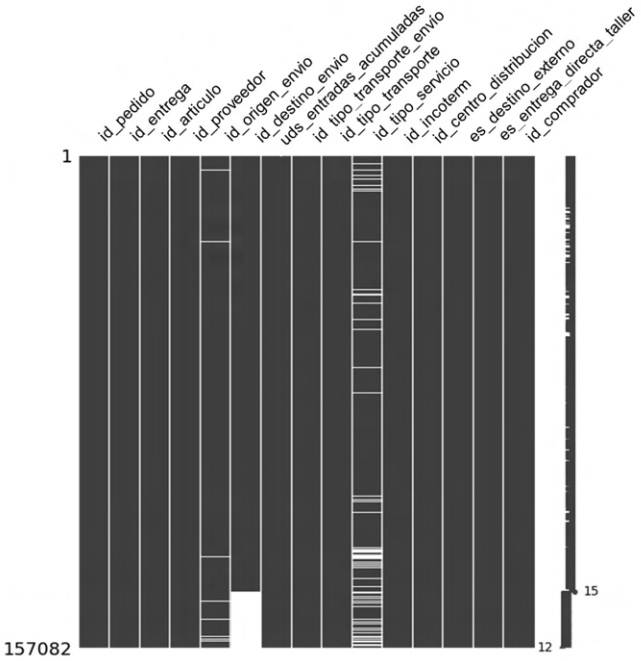

Figura 2.2: Gráfico de registros faltantes. El gráfico representa cada observación del conjunto de datos para una cierta variable con una línea horizontal. Las líneas horizontales se agrupan en filas que representan las diferentes variables de una misma observación. Los valores con dato están en gris, mientras que los valores faltantes se muestran en blanco.

2.2.1. Imputación *Dummy*

En primer lugar, se presenta el modelo *dummy*, el cual asigna el origen, destino o servicio de transporte más común del proveedor encargado de enviar la mercancía. Este enfoque sencillo tiene el objetivo de proporcionar un valor predeterminado o común para los datos faltantes, utilizando patrones generales observados en el conjunto de datos. Sin embargo, enfrenta limitaciones cuando todos los registros de un proveedor específico carecen de estos datos, imposibilitando la imputación de manera efectiva. En estos casos, el modelo *dummy* se ve incapaz de predecir el valor adecuado, ya que no cuenta con información suficiente para hacerlo.

Para abordar esta situación, dejamos los valores como faltantes, de modo que no influyan en el cálculo de la precisión del algoritmo en el análisis subsiguiente. Esta decisión es importante para evitar introducir sesgos significativos que puedan distorsionar los resultados del modelo final, no obstante, es importante tener en cuenta esta limitación.

2.2.2. Imputación por *MICE*

En esta sebsección presentamos el algoritmo de imputación múltiple por ecuaciones encadenadas (*MICE*, por sus siglas en inglés), un enfoque robusto y flexible para manejar datos faltantes en análisis estadísticos (véase [23]). Este método, que en este caso utiliza el modelo *LightGBM* para imputar los datos faltantes realizando múltiples regresiones de forma iterativa, se ha consolidado como una herramienta muy útil en la imputación de datos. Esto se debe a que provecha al máximo los patrones presentes en el conjunto de datos y genera múltiples imputaciones para cada dato faltante, lo que permite inferencias estadísticas más precisas y confiables, teniendo en cuenta la incertidumbre asociada a los datos faltantes.

En esta sección, detallaremos el funcionamiento del algoritmo *MICE* y resaltaremos sus ventajas y desventajas. Los detalles específicos del modelo *LightGBM*, utilizado en *MICE* para realizar las regresiones, se encuentran en la Sección 3.1.

El algoritmo *MICE* consta de las siguientes tres etapas:

1. **Inicialización:** La inicialización es el punto de partida del *MICE*, donde se imputan los valores faltantes con estimaciones iniciales. Este paso se puede realizar de diversas maneras, como imputar con la media, mediana o mediante valores escogidos aleatoriamente del conjunto de datos. La elección del método de inicialización puede influir en la convergencia del algoritmo, pero el impacto sobre las inferencias finales es generalmente limitado debido a las iteraciones subsiguientes que refinan estas imputaciones iniciales [23]. En el caso del paquete empleado (`miceforest` de python) la iteración inicial se lleva a cabo empleando valores escogidos aleatoriamente y con reemplazamiento del conjunto de datos (véase [24]), lo cual es muy conveniente en el caso que nos ocupa, pues la media o la mediana de las variables categóricas que queremos imputar no se puede obtener.

2. **Imputación Secuencial:** En este paso, *MICE* imputa secuencialmente cada variable con datos faltantes, utilizando las demás variables en el conjunto de datos como predictores. Para cada variable objetivo, se ajusta un modelo de regresión *LightGBM*, basándose en los valores actuales (tanto observados como imputados en pasos anteriores) de otras variables. Las imputaciones se generan a partir de este modelo, sustituyendo los valores faltantes con las predicciones obtenidas. Este proceso se aplica de manera iterativa a todas las variables con datos faltantes, siguiendo un orden aleatorio en cada iteración [23]. En el caso del *MICE* empleado, los modelos con los que se generaron las imputaciones eran los siquientes:

   - Origen:

$$\begin{aligned}\text{id\_origen\_envio} \sim & \text{id\_proveedor} + \text{id\_destino\_envio} + \text{id\_tipo\_transporte} + \\ & \text{id\_tipo\_servicio} + \text{id\_incoterm} + \text{id\_centro\_distribucion} + \\ & \text{es\_destino\_externo} + \text{es\_entrega\_directa\_taller} + \\ & \text{id\_comprador}\end{aligned}$$

- Destino:

$$\begin{aligned}\text{id\_destino\_envio} \sim & \text{id\_proveedor} + \text{id\_origen\_envio} + \text{id\_tipo\_transporte} + \\ & \text{id\_tipo\_servicio} + \text{id\_incoterm} + \text{id\_centro\_distribucion} + \\ & \text{es\_destino\_externo} + \text{es\_entrega\_directa\_taller} + \\ & \text{id\_comprador}\end{aligned}$$

- Servicio de transporte:

$$\begin{aligned}\text{id\_tipo\_servicio} \sim & \text{id\_proveedor} + \text{id\_origen\_envio} + \text{id\_destino\_envio} + \\ & \text{id\_tipo\_transporte} + \text{id\_incoterm} + \text{id\_centro\_distribucion} + \\ & \text{es\_destino\_externo} + \text{es\_entrega\_directa\_taller} + \\ & \text{id\_comprador}\end{aligned}$$

3. **Iteración:** El ciclo de imputación se repite varias veces, actualizando las imputaciones en cada ronda con la información más reciente de las iteraciones anteriores. Este proceso iterativo ayuda a asegurar la convergencia hacia un conjunto de imputaciones coherente y razonable. La convergencia se evalúa a menudo mediante el monitoreo de cambios en las imputaciones o en los parámetros del modelo entre iteraciones sucesivas, deteniéndose el proceso cuando estos cambios son suficientemente pequeños o cuando se alcance el límite de iteraciones previamente establecido [23]. En este trabajo se llevaron a cabo 25 iteraciones para asegurar una convergencia adecuada del algoritmo (pues el paquete `miceforest` no incluía la condición de parada por convergencia).

En el caso de este trabajo el *LightGBM* empleado para hacer las regresiones dentro del *MICE* llevó a cabo una búsqueda de hiperparámetros por 10 *fold cross validation* en 1000 puntos por *random search*. De esta forma aseguramos que las regresiones realizadas obtengan un mejor resultado.

Ventajas de *MICE*

Una de las principales ventajas de *MICE* es su capacidad para manejar diferentes tipos de datos y patrones de faltantes. Al ajustar modelos *LightGBM* específicos para cada variable, se pueden tratar adecuadamente tanto variables continuas como categóricas, respetando sus distribuciones y relaciones inherentes. Además, *MICE* permite una estimación más precisa de la incertidumbre alrededor de las imputaciones al generar múltiples conjuntos de datos completos, lo que facilita análisis subsiguientes que reflejan adecuadamente la variabilidad resultante de los datos faltantes [23].

Inconvenientes de *MICE*

Pese a sus ventajas, *MICE* no está exento de desafíos. La selección de modelos de regresión apropiados para cada variable con datos faltantes es crucial y puede ser compleja, especialmente en conjuntos de datos con muchas variables o relaciones no lineales complejas. Además, el proceso iterativo puede ser computacionalmente demandante para conjuntos de datos grandes o con altas proporciones de faltantes. Aunque la convergencia generalmente se alcanza, en situaciones con patrones de datos faltantes especialmente complicados o relaciones altamente no lineales, el algoritmo puede requerir un número elevado de iteraciones para estabilizarse.

Explicación del modelo de imputación

En esta sección, se intentará revelar cuáles son las características más determinantes para nuestro modelo de imputación, lo que proporcionará insights valiosos no solo para su posible mejora en un futuro, sino también para el entendimiento de las relaciones entre las diferentes variables explicativas. Asimismo, el gráfico de importancia de variables que se presenta en la Figura 2.3 se enfoca específicamente en mostrar la importancia de las variables en el contexto de un modelo *LightGBM*, donde esta importancia se calcula a partir del número de veces que una variable es utilizada en las divisiones de los árboles de decisión, así como la mejora que aporta al desempeño del modelo cada vez que se utiliza (ver [1]).

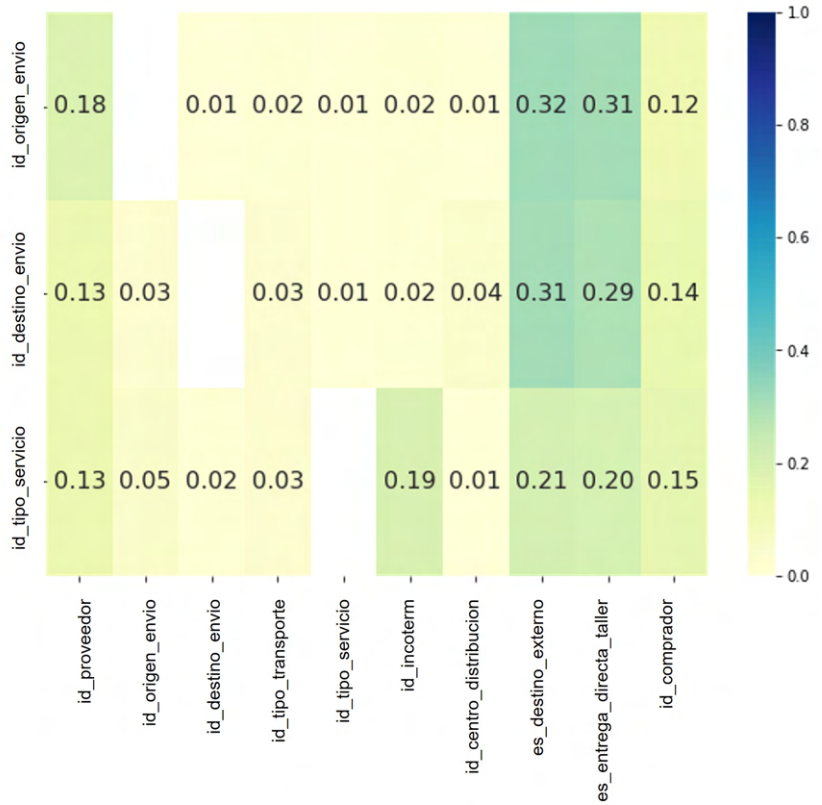

Figura 2.3: Gráfico de importancia de las variables en la imputación de valores faltantes del algoritmo *MICE*.

A continuación, analizaremos detalladamente la relevancia de las variables en los distintos modelos de imputación de datos faltantes. Empezaremos con la variable `id_origen_envio`, donde destacan como factores predominantes el proveedor, y la presencia de indicadores que señalan si el envío requiere pasar por el taller y si lo hace directamente. Esto se justifica porque los proveedores suelen limitar los orígenes específicos de los productos (por proximidad geográfica), y ciertos artículos de determinados orígenes necesitan un proceso adicional antes de ser enviados a las tiendas, lo cual hace que transiten con mayor frecuencia por el taller. El comprador también tiene un papel relevante, aunque en menor medida en comparación con los factores previamente mencionados, posiblemente debido a preferencias por ciertos orígenes sobre otros.

En el caso de `id_destino_envio`, el proveedor sigue siendo un factor de influencia, pero el comprador adquiere mayor relevancia, probablemente porque tienden a especializarse en secciones específicas (mujer, caballero, niño) que están predominantemente asignadas a un centro de distribución particular, facilitando la predicción de los destinos de envío. Sin embargo, las variables más críticas son si el artículo pasa por el taller y si se envía directamente a este. Esto se debe a que, de enviarse a un taller en lugar de a un centro de distribución, quizás es más razonable enviarlo a un puerto o aeropuerto de destino diferente.

Por último, al evaluar la variable `id_tipo_servicio`, encontramos que las variables de que el envío pase por el taller y de si lo hace directamente son nuevamente variables clave. No obstante, en este escenario, el *INCOTERM* adquiere una gran importancia, lo cual es esperable al ser un término directamente relacionado con el transporte. El comprador, debido a que establece ciertas fechas y puede solicitar envíos de mayor o menor urgencia, y el proveedor, quien, en caso de encargarse del transporte, decide la modalidad del mismo (siempre que cumpla con los plazos acordados), también son factores significativos.

2.2.3. Comparativa de los modelos de imputación

Para evaluar la eficacia de estos métodos de imputación, se utilizó un conjunto de prueba. Este conjunto se no es más que una parte de los datos de entrenamiento que originalmente no presentaban valores faltantes, a los cuales se les insertaron valores faltantes de forma aleatoria en proporciones equivalentes a las observadas en el conjunto de datos completo de entrenamiento. Este enfoque nos permitió comparar de manera efectiva los métodos de imputación utilizados, los cuales se presentarán a continuación.

Los resultados de precisión para ambos modelos se presentan en la Tabla 2.1, mostrando una mejora considerable con el uso de *MICE LightGBM* en comparación con el modelo *dummy* en 2 de las 3 variables estudiadas, y una mejora mas modesta en la variable restante.

Modelo	Origen	Destino	Servicio de transporte
Dummy	78,19 %	63,84 %	60,84 %
MICE LightGBM	80,45 %	86,23 %	76,27 %

Tabla 2.1: Comparación del rendimiento de la precisión de los modelos en diferentes categorías.

Este análisis destaca la importancia de elegir el método de imputación adecuado para cada tipo de variable faltante, equilibrando entre la precisión del algoritmo y su gasto computacional.

2.3. Análisis exploratorio

En esta sección, nos sumergiremos en el análisis exploratorio de datos (EDA) de nuestro proyecto, lo cual es fundamental para comprender a fondo el conjunto de datos con el que estamos trabajando. Esta

fase es crucial, ya que nos permite obtener una visión preliminar y profunda de las tendencias, patrones y anomalías presentes en nuestros datos, a través de la visualización gráfica. Presentaremos una serie de gráficos meticulosamente seleccionados que nos ayudarán a desvelar la estructura subyacente de los datos, las relaciones entre variables y posibles áreas de interés para análisis posteriores. Este proceso no solo enriquece nuestra comprensión inicial, sino que también guía nuestras decisiones metodológicas y analíticas en las fases subsiguientes del proyecto.

2.3.1. Medidas de asociación sobre las posibles variables respuesta

En esta subsección, nos enfocaremos en el desarrollo de medidas de asociación para identificar cómo distintas variables explicativas interactúan con las variables respuesta seleccionadas para los modelos.

En primer lugar, utilizando el paquete de Python *dython* (ver [18]), se analizó la matriz de asociaciones que se muestra en la Figura 2.4. Esta matriz se calculó separando los pares de variables según su tipo. Para pares de variables numéricas, se utilizó la correlación de Pearson cuya fórmula es:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}},$$

donde x_i e y_i son la observación i-ésima de las variables X e Y respectivamente y n es el número de observaciones. El coeficiente, r, mide la dirección y la fuerza de una relación lineal entre dos variables numéricas, con un valor entre -1 y 1. Un valor cercano a 1 indica una fuerte correlación positiva, mientras que un valor cercano a -1 indica una fuerte correlación negativa (véase [20]).

Para medir la relación entre una variable numérica y una categórica, se empleó el ratio de correlación. La fórmula es:

$$\eta = \frac{\sqrt{\sum_{j=1}^{k} n_j(\bar{x}_j - \bar{x})^2}}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}},$$

donde x_i es la observación i-ésima de la variable continua X, n es el número total de observaciones, k es el número de categorías de la variable categórica, n_j el número de observaciones asociadas a la categoría j-ésima y $\bar{x}_j$ es la media de x para las observaciones con la categoría j-ésima. Aquí, η indica cuánta varianza de la variable numérica puede explicarse por la categórica. Los valores oscilan entre 0 y 1, donde un valor cercano a 0 implica que la variable categórica no explica la varianza, mientras que un valor cercano a 1 implica una explicación completa de la varianza (véase [20]).

De esta forma, se obtuvieron medidas de asociación sobre las variables más interesantes del conjunto de datos. En vista de la Figura 2.4 se puede ver como las variables más relacionadas con `dias_hasta_fecha_total_entrado` son el resto de variables que hacen referencia a días hasta alguna fecha relevante, como `dias_hasta_mejor_fecha_envio` o `dias_hasta_fecha_entrega`. Esto es razonable, pues los días hasta un cierto evento de la entrega repercuten directamente en los días hasta cualquiera de los eventos posteriores. Sin embargo, la relación de las variables con `retraso` no es tan clara, pues el máximo valor que toman las medidas de asociación es 0,18. Aquellas que obtienen valores más altos en la matriz son: `id_tipo_servicio`, `id_tipo_transporte` e `id_proveedor`, lo cual es bastante lógico, pues hacen referencia al servicio de transporte contratado, el medio de transporte y el proveedor responsable de la mercancía respectivamente.

Asimismo, también se llevaron a cabo otras medidas de asociación como la de información mutua que se pueden ver en las Figuras 2.5 y 2.6 y diferentes gráficos de cajas y diagramas de dispersión sobre las nuevas variables creadas, como los de las Figuras de la 2.7 a la 2.10, los cuales se comentarán a continuación, no sin antes explicar en detalle qué es la información mutua y cómo se calcula.

La información mutua es un concepto fundamental en la teoría de la información que mide la dependencia mutua entre dos variables aleatorias (véase [3]). Evalúa cuánto se reduce la incertidumbre

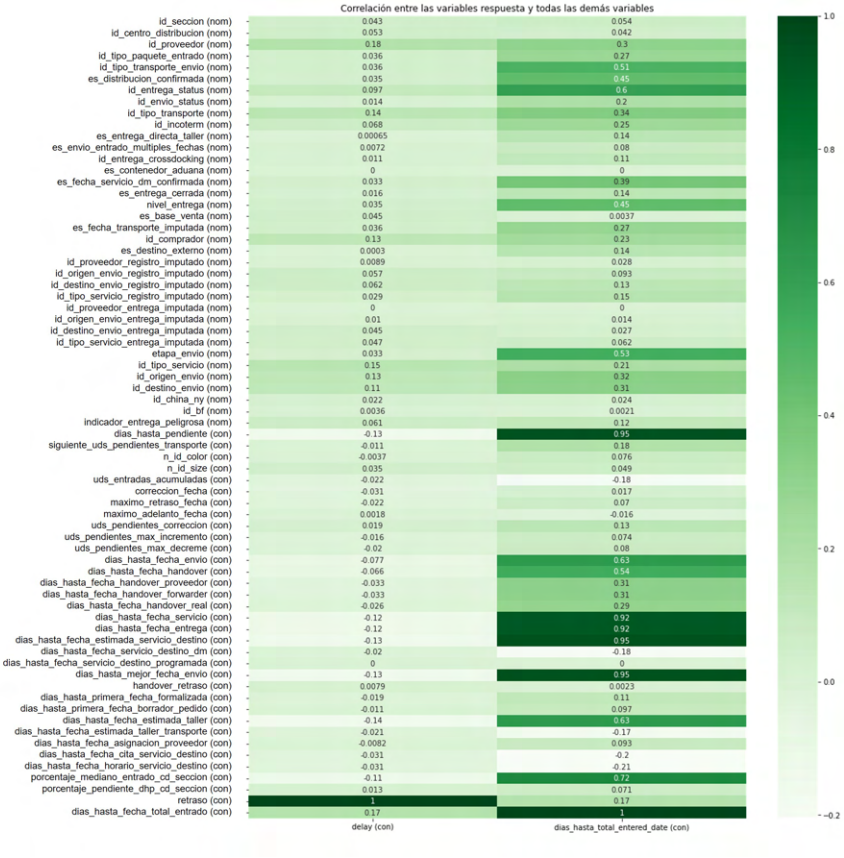

Figura 2.4: Matriz de medidas de asociación entre las diferentes variables de los conjuntos de datos y las variables respuesta `retraso` y `dias_hasta_fecha_total_entrado`.

sobre una variable cuando se conoce el valor de la otra, lo que la convierte en una herramienta valiosa para medir relaciones no lineales, a diferencia de métricas como la correlación de Pearson, que solo mide relaciones lineales.

Matemáticamente, la información mutua $I(X;Y)$ entre dos variables aleatorias X e Y se define como:

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \left(\frac{p(x,y)}{p(x)p(y)} \right),$$

donde $p(x,y)$ es la probabilidad conjunta de $X = x$ e $Y = y$, y $p(x)$ y $p(y)$ son las probabilidades marginales de X e Y respectivamente. La información mutua cuantifica la información que proporciona el conocimiento del valor de una variable sobre otra. Si $I(X;Y) = 0$, las variables son independientes y conocer una no aporta información sobre la otra. Valores mayores indican una mayor dependencia.

Para calcular la información mutua, se estiman las distribuciones de probabilidad $p(x)$, $p(y)$ y $p(x,y)$ a partir de los datos, y se sustituyen en la fórmula para calcular la suma. Esto permite su uso en diversas aplicaciones, como la selección de características en modelos complejos o también llamados "de caja negra". Seguidamente comentaremos la aplicación de estas medidas a nuestro conjunto de datos.

Al analizar las gráficas de información mutua, se puede observar que la variable de unidades pendientes, `siguiente_uds_pendientes_transporte`, se presenta como la más correlacionada con `retraso` y `dias_hasta_fecha_total_entrado` en las Figuras 2.5 y 2.6, respectivamente. Esta observación es coherente, puesto que los envíos con un mayor número de unidades pendientes suelen requerir una gestión más exhaustiva y una planificación proactiva, factores que favorecen la eficiencia en la reducción de `retraso`. Por otro lado, la correlación con `dias_hasta_fecha_total_entrado` se explica por la disminución progresiva del número de unidades pendientes a medida que se efectúan las entregas, evidenciando una conexión directa entre ambas variables.

Siguiendo con el análisis, se identifican varias variables relacionadas con el conteo regresivo hacia fechas clave del proceso de envío, como `dias_hasta_siguiente_fecha_pendiente_transporte`, `dias_hasta_mejor_fecha_envio`, y `dias_hasta_fecha_entrega`. Estas variables ilustran la importancia de los días restantes hasta un evento significativo dentro del ciclo de envío, afectando directamente tanto al `retraso` como a los `dias_hasta_fecha_total_entrado`.

Con un nivel de información mutua más bajo, se sitúan variables asociadas con agentes del proceso de envío como `id_proveedor`, `id_tipo_transporte_envio`, y `id_comprador`. Estas relaciones indican que la identidad del proveedor, el medio de transporte seleccionado y la gestión por parte del comprador están relacionadas con la puntualidad del proceso de entrega y con la fecha en que la mercancía llega a su destino final.

Es relevante señalar que, debido a la conexión intrínseca entre las variables de respuesta, no sorprende encontrar similitudes marcadas entre los gráficos de información mutua. Estos patrones reflejan la interdependencia de los tiempos de entrega y los retrasos dentro del proceso logístico de distribución.

Por otra parte, llevaremos a cabo un análisis muy similar al de la Sección 1.2.1, donde estudiaremos los gráficos de cajas de las variables creadas en al comienzo de este capítulo: `id_bf` e `id_china_ny`, que hacían referencia a si la entrega estaba cercana a los períodos del Black Friday o el Año Nuevo Chino.

El análisis de las Figuras 2.7, 2.8, 2.9 y 2.10 revela cómo la proximidad temporal con eventos globales como el Black Friday y el Año Nuevo Chino influye en la logística y el retraso en los envíos.

Las observaciones derivadas de las Figuras mencionadas anteriormente revelan una sorprendente uniformidad en los efectos de estas variables sobre los retrasos y los tiempos hasta la entrega total.

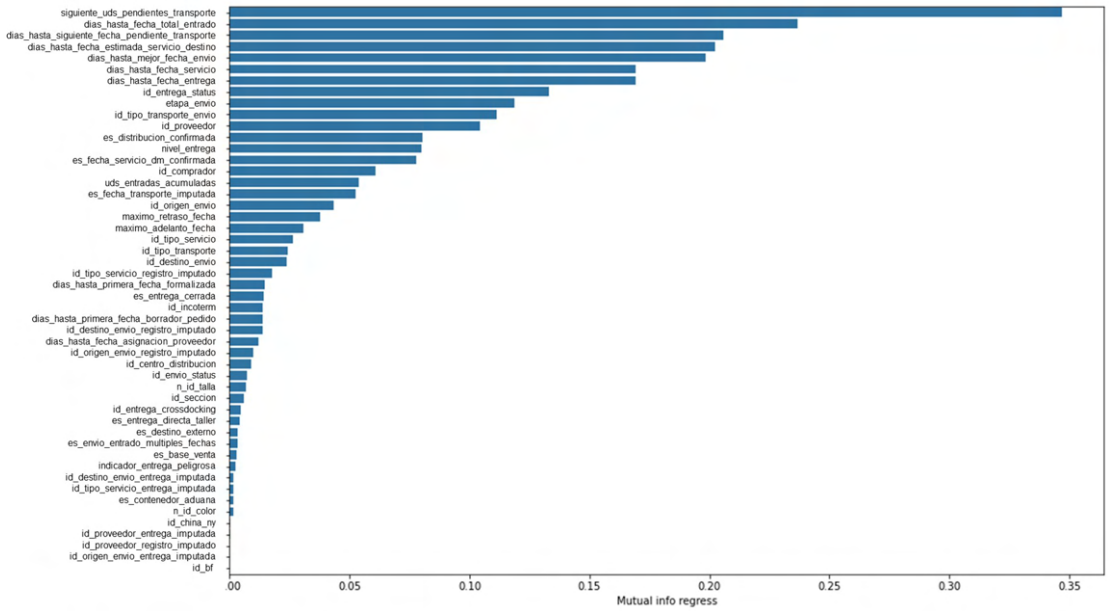

Figura 2.5: Gráficos de barras de la información mutua entre las diferentes variables y la variable `retraso`.

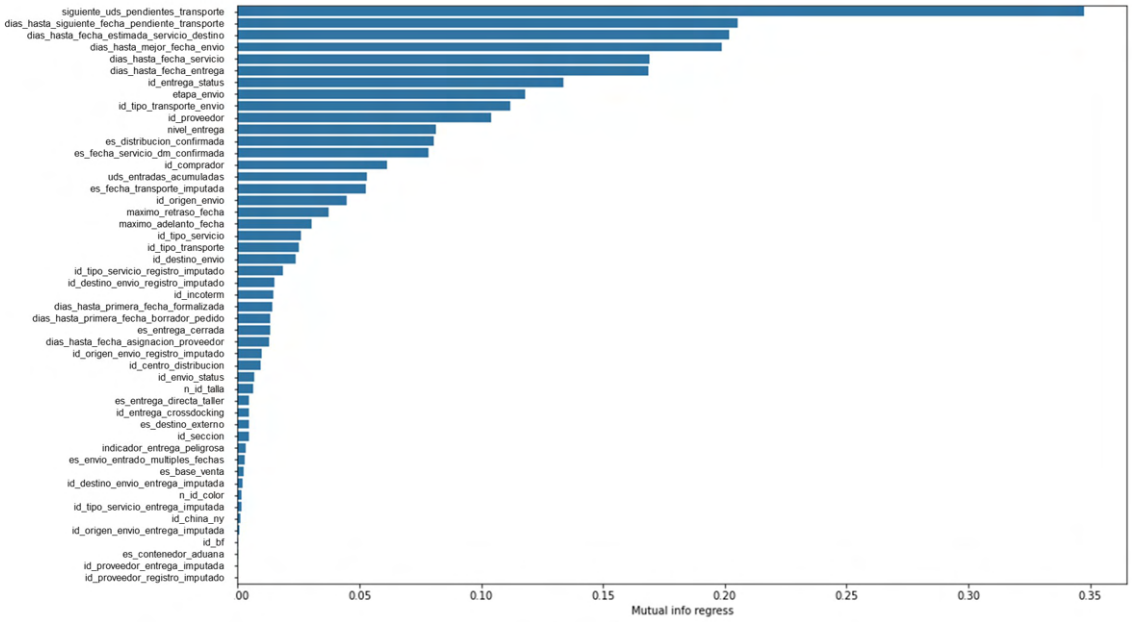

Figura 2.6: Gráficos de barras de la información mutua entre las diferentes variables y la variable `dias_hasta_fecha_total_entrado`.

Contrario a la expectativa inicial, las fluctuaciones relacionadas con estos eventos no parecen ejercer una influencia determinante en la eficiencia del envío. Este hallazgo sugiere que, a pesar de los potenciales desafíos logísticos que representan, los mecanismos de mitigación implementados logran minimizar su impacto sobre los plazos de entrega.

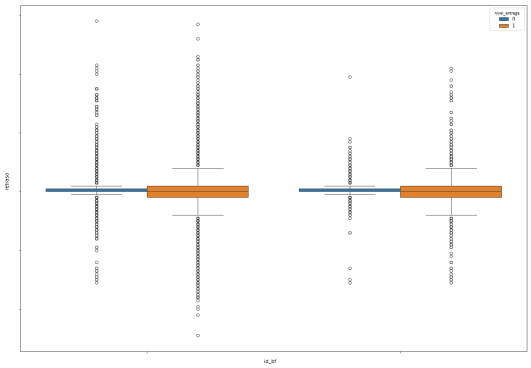

Figura 2.7: Gráfico de cajas del retraso agrupando por Black Friday.

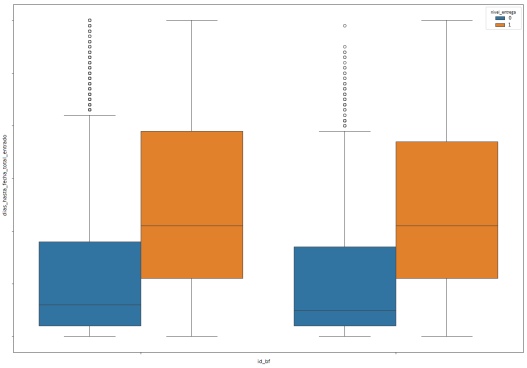

Figura 2.8: Gráfico de cajas de los días hasta la entrega total agrupando por Black Friday.

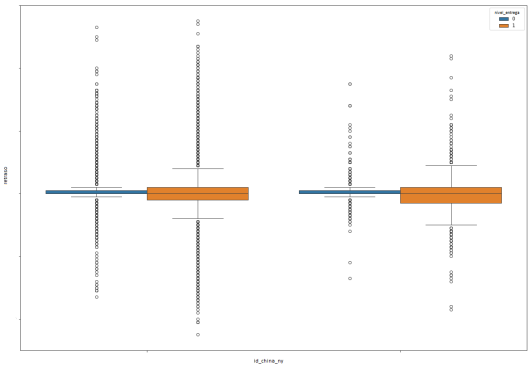

Figura 2.9: Gráfico de cajas del retraso agrupando por Año Nuevo Chino.

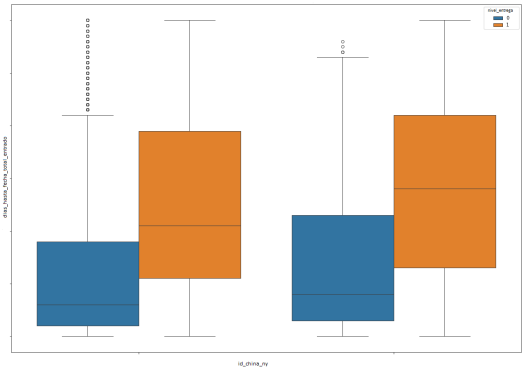

Figura 2.10: Gráfico de cajas de los días hasta la entrega total agrupando por Año Nuevo Chino.

2.3.2. Medidas de error sobre la fecha pendiente actual

En esta sección, nos centramos en el desarrollo y aplicación de medidas de error específicas destinadas a evaluar la precisión de nuestras predicciones y poder así compararlas con la predicción actual (`siguiente_fecha_pendiente_transporte`). Nuestro objetivo es no solo medir la eficacia de las predicciones sino también comprender su impacto en la logística. Para ello, introducimos cuatro medidas clave, cada una diseñada para capturar diferentes aspectos del rendimiento predictivo:

1. | `retraso` | : Esta medida, se calcula como los días de diferencia (en valor absoluto) entre la fecha predicha para la siguiente entrega o *packing* (`siguiente_fecha_pendiente_transporte`) y la fecha en la que se han entregado todas las unidades de dicha entrega o *packing*, según corresponda.

2. `percentage_ontime_arrival`: Esta medida calcula el porcentaje de unidades que llegan antes o en la fecha límite prevista. Esta métrica nos ofrece una visión general de la puntualidad del

proceso de entrega, permitiéndonos identificar qué tan frecuentemente las entregas se realizan en el plazo acordado.

3. `id_total_ontime_arrival`: Actúa como un indicador binario, asignando un valor de 1 cuando el 95 % o más de las unidades llegan a tiempo y 0 en cualquier otro caso. Esta es la medida más restrictiva entre las presentadas, proporcionando un umbral claro para evaluar el éxito de las entregas a tiempo.

4. `exponential_weight_measure`: Esta métrica es la menos restrictiva de las cuatro y está diseñada para evaluar de manera universal el rendimiento de las entregas. La medida se calcula como una suma ponderada del porcentaje entregado cada día respecto al total entregado a lo largo de todos los días, multiplicada por el tanto por uno del total entregado sobre el total pendiente. Los pesos varían según si las entregas se realizan antes o después del tiempo previsto.

   Para los retrasos, los pesos se calculan como:

$$\exp\left(\frac{\texttt{siguiente\_fecha\_pendiente\_transporte} - \texttt{fecha\_servicio\_destino}}{5}\right),$$

   y para los adelantos, como:

$$\exp\left(\frac{\texttt{fecha\_servicio\_destino} - \texttt{siguiente\_fecha\_pendiente\_transporte}}{35}\right).$$

   Esta estrategia de ponderación penaliza más los retrasos que los adelantos, destacando la importancia de la puntualidad en la cadena de suministro. La elección de los coeficientes en las fórmulas se ha realizado mediante un enfoque empírico basado en prueba y error, optimizando los valores para obtener resultados razonables y coherentes. Esta medida no solo considera las variaciones en las fechas de entrega, sino también las discrepancias en el número de unidades entregadas frente al número prometido, proporcionando un análisis exhaustivo del rendimiento de entrega.

Estas medidas, cuidadosamente seleccionadas y diseñadas, nos permiten evaluar de manera integral la precisión y fiabilidad de las predicciones de entrega previas a la creación de los modelos. Al hacerlo, podemos identificar áreas de mejora de cara a crear los modelos predictivos. Seguidamente, se presentan una serie de gráficos de violín que nos permitirán explorar en profundidad las distribuciones de las medidas de error que hemos desarrollado anteriormente. Un gráfico de violín es una herramienta visual que combina las características de un diagrama de caja y bigotes con un gráfico de densidad. Este tipo de gráfico muestra la distribución de los datos a través de su densidad de probabilidad, proporcionando una representación detallada y enriquecida de las distribuciones. Los gráficos de violín son especialmente útiles para comparar múltiples distribuciones de datos en diferentes categorías [9].

A continuación, en las Figuras 2.11 a 2.14, podemos observar los gráficos de violín de las variables `percentage_ontime_arrival`, `id_total_ontime_arrival`, y `exponential_weight_measure`. Estos gráficos se generan para cada valor de la variable `dias_hasta_siguiente_fecha_pendiente_transporte`, mostrando las medias ponderadas de las medidas mencionadas en función del número de unidades del envío, agrupando por `id_proveedor`, y separando los datos en función de su nivel de granularidad, ya sea a nivel de *packing* (en naranja) o de entrega (en azul).

Es importante notar que los violines naranjas, correspondientes a los *packings*, no están presentes para todos los valores de la variable `días_hasta_pendiente`, ya que se generan más cerca de la fecha de entrega. Además, para que un proveedor sea considerado en la evaluación del gráfico de violines correspondiente a un determinado valor de la variable `días_hasta_pendiente`, este debe tener al menos 50 entregas con ese determinado valor de `días_hasta_pendiente`.

Esta metodología nos permite discernir no solo la variabilidad en las diferentes medidas entre proveedores, sino también cómo la anticipación afecta a las medidas de error. Esto es crucial para identificar patrones, tendencias, o anomalías que podrían influir en la estrategia de gestión de la cadena de suministro y en la toma de decisiones operativas y logísticas. A continuación, se comentarán los resultados de los gráficos mencionados, los cuales se pueden ver en las Figuras 2.11, 2.12, 2.13 y 2.14.

Las figuras mencionadas muestran que, como era de esperar, los errores de predicción disminuyen a medida que nos acercamos a la fecha de entrega estimada a nivel de *packing*. Esto se debe a que hay menos incertidumbre en las predicciones cuanto más cerca estamos de esta fecha. Además, en todas estas figuras se observa que la variabilidad de los errores de predicción entre diferentes proveedores también se reduce conforme se aproxima la fecha de entrega.

No obstante, lo inesperado surge al considerar estos mismos gráficos a nivel de entrega: la variabilidad de las predicciones y los errores asociados no disminuyen lo suficiente a medida que se acerca la fecha de entrega. De hecho, los errores de predicción tienden a mejorar hasta que la entrega se encuentra a unos 5-6 días de distancia de la fecha pendiente. Sin embargo, estos errores empeoran al avanzar a partir de ese punto y lo hacen más rápidamente a partir de los 3 días hasta la fecha pendiente. Esto sugiere que si una entrega está a 3 días de la fecha estimada y no tiene información a nivel de *packing*, probablemente enfrente algún problema. Esta situación ha llevado a la creación del indicador `id_entrega_peligrosa` para identificar dichos casos e intentar que los modelos sean capaces de detectarlos con mayor facilidad, con el objetivo de mejorar su capacidad predictiva.

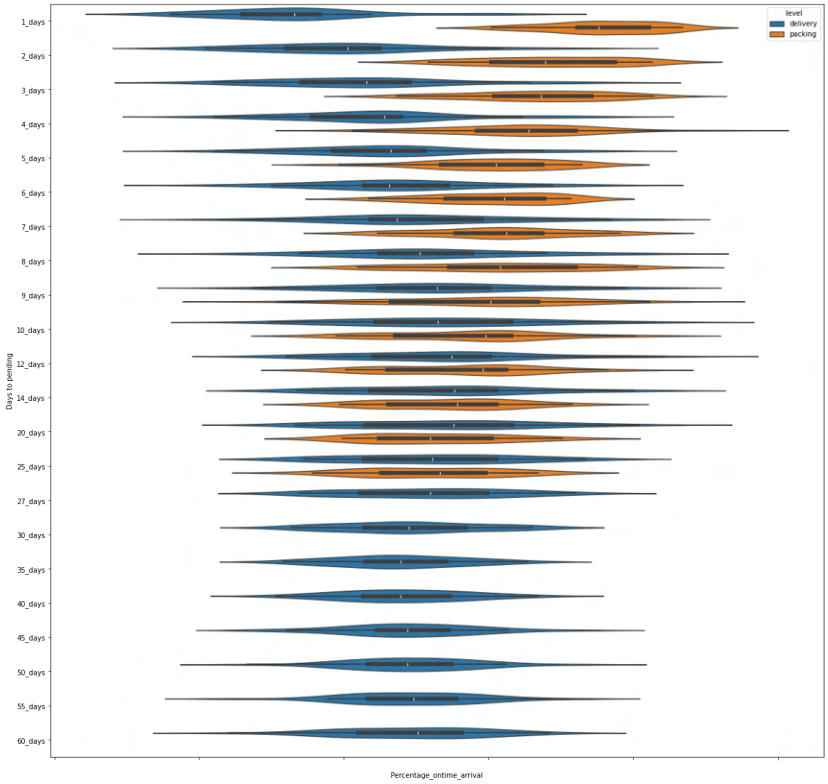

Figura 2.11: Gráficos de violín de `percentage_ontime_arrival` agrupando por `id_proveedor` y filtrando por `dias_hasta_siguiente_fecha_pendiente_transporte`.

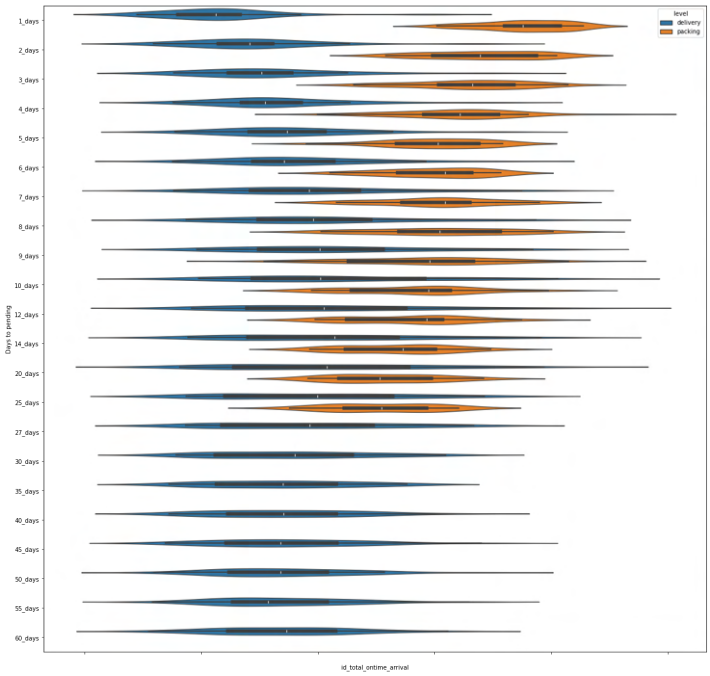

Figura 2.12: Gráficos de violín de `id_total_ontime_arrival` agrupando por `id_proveedor` y filtrando por `dias_hasta_siguiente_fecha_pendiente_transporte`.

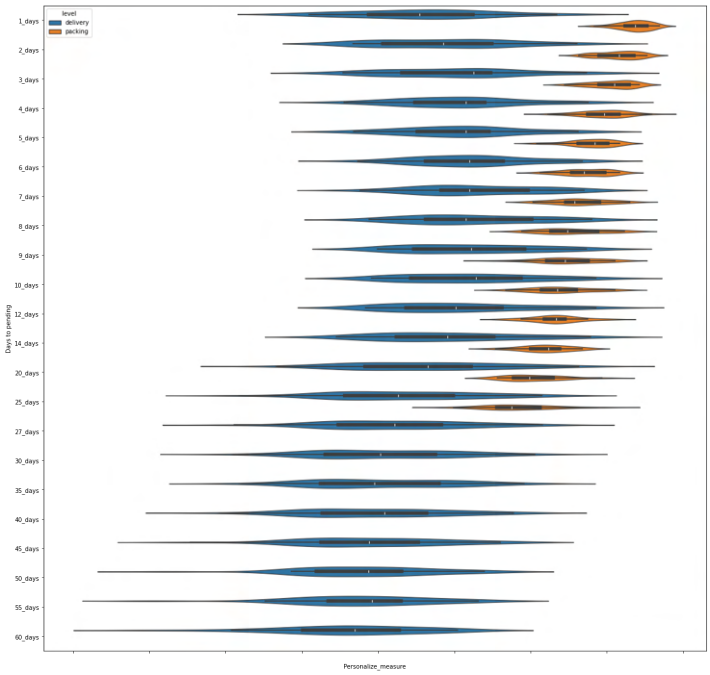

Figura 2.13: Gráficos de violín de `exponetial_weight_measure` agrupando por `id_proveedor` y filtrando por `dias_hasta_siguiente_fecha_pendiente_transporte`.

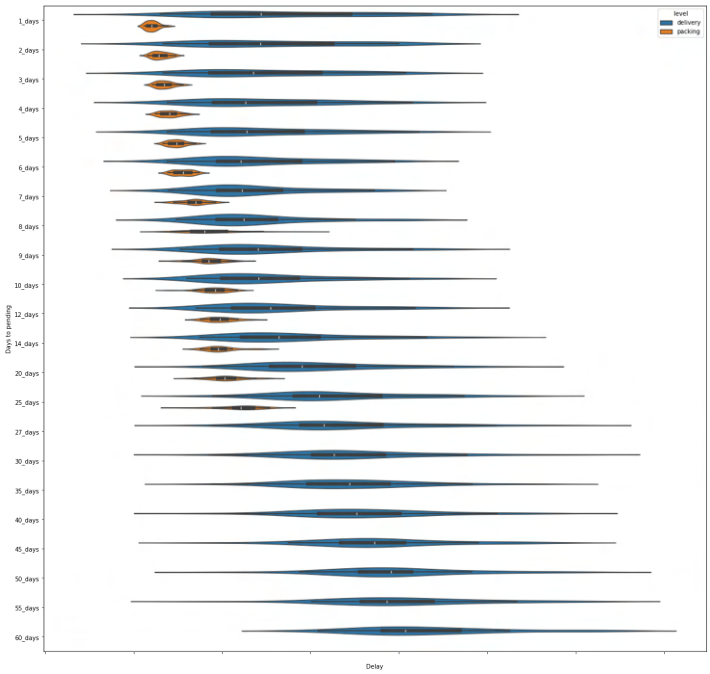

Figura 2.14: Gráficos de violín de `retraso` agrupando por `id_proveedor` y filtrando por `dias_hasta_siguiente_fecha_pendiente_transporte`.

Capítulo 3

Creación de los modelos

Este capítulo está dedicado a la descripción detallada y la evaluación de dos modelos predictivos implementados utilizando *LightGBM*, un algoritmo de aprendizaje automático conocido por su eficiencia, velocidad y capacidad para manejar grandes volúmenes de datos. *LightGBM* ha ganado popularidad en la comunidad de ciencia de datos debido a su capacidad para manejar conjuntos de datos de gran tamaño y su eficiencia en el cálculo de gradientes, lo cual es esencial para el ajuste rápido y preciso de los modelos predictivos.

El proceso de creación de estos modelos se ha estructurado en tres fases clave para optimizar tanto la búsqueda de hiperparámetros como la selección de variables con el fin de obtener las mejores predicciones con la máxima eficiencia computacional.

La primera fase del desarrollo de los modelos involucra una búsqueda inicial de hiperparámetros. En esta etapa, se lleva a cabo una exploración amplia de los hiperparámetros del modelo para identificar una configuración preliminar que ofrezca un rendimiento aceptable. Este paso es crucial ya que establece una base sólida para refinamientos posteriores y asegura que el modelo no esté sesgado por una selección subóptima desde el principio.

Posteriormente, la atención se centra en la selección de variables. Utilizando la configuración de hiperparámetros identificada en la fase anterior como punto de partida, se examina la importancia de las diferentes variables disponibles en el conjunto de datos. Este análisis permite identificar cuáles contribuyen de manera considerable a la capacidad predictiva del modelo. La selección de variables es una etapa crítica para mejorar la eficiencia del modelo, reduciendo la complejidad y focalizando el aprendizaje en la información más relevante.

Finalmente, con un conjunto de variables seleccionadas, se realiza una búsqueda final de hiperparámetros. Esta fase tiene como objetivo afinar la configuración del modelo para maximizar su rendimiento utilizando solo las variables más relevantes y asegurando que se adapte de manera óptima a las peculiaridades del conjunto de datos final.

A lo largo de este capítulo, se explicarán en detalle las metodologías empleadas en cada una de estas fases, incluyendo las técnicas de búsqueda de hiperparámetros y selección de variables, así como las consideraciones teóricas que fundamentan estas elecciones. El objetivo es proporcionar una comprensión clara de cómo se construyeron los modelos *LightGBM*, subrayando la importancia de un enfoque estructurado y metódico en el desarrollo de soluciones de aprendizaje automático eficientes y efectivas.

3.1. Explicación del *LightGBM*

LightGBM, abreviatura de *Light Gradient Boosting Machine* ofrece un enfoque altamente eficiente para el *boosting* de gradientes que permite el manejo de grandes volúmenes de datos con una notable disminución en el tiempo de computación y uso de memoria con respecto otros algoritmos de *boosting* como el *XGBoost*. Este algoritmo se distingue por su capacidad para realizar tareas de clasificación, regresión y clasificación de manera efectiva, empleando técnicas innovadoras que optimizan tanto la velocidad como la precisión del modelo [10]. Cabe destacar que el paquete de python empleado para la implementación se puede ver en [14]

El algoritmo *LightGBM* se basa en la técnica del *gradient boosting*, construyendo modelos secuencialmente, cada uno tratando de corregir los errores del anterior. Antes de comentar que mecanismos innovadores separan al *LightGBM* de otros algoritmos que emplean *gradient boosting*, es necesario explicar en detalle en que consiste el *gradient boosting*.

Para un conjunto de datos $\{(\mathbf{x_i}, y_i)\}_{i=1}^{N}$, donde x_i son las observaciones de las variables explicativas e y_i son las observaciones de la variable respuesta, el objetivo es encontrar una función $F(\mathbf{x})$ que minimice la función de pérdida $L(\mathbf{y}, F(\mathbf{x}))$. El *gradient boosting* establece que esta función se puede obtener como

$$F(\mathbf{x}) = \sum_{m=1}^{M} \rho_m h_m(\mathbf{x}) + F_0, \text{ siendo } F_0 \text{ una constante.}$$

Para ello intenta aproximar esta $F(\mathbf{x})$ con F_0 e irá aproximándose a la función final mediante el siguiente algoritmo *greedy* [6]:

$$F_0(\mathbf{x}) = arg\min_{c \in \mathbb{R}} \sum_{i=1}^{N} L(y_i, c)$$

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + arg\min_{h_m \in \mathcal{H}} \left[\sum_{i=1}^{N} L(y_i, F_{m-1}(\mathbf{x_i}) + \rho_m h_m(\mathbf{x_i})) \right],$$

donde $\mathcal{H}$ es el espacio de funciones de funciones simples parametrizadas de $\mathbf{x}$ (ver [6]) y F_m es la aproximación de F en la m-ésima iteración.

No obstante, debido a que este problema es difícilmente abarcable computacionalmente, en [6] se propone el siguiente algoritmo el cual pretende obtener una aproximación de $F(\mathbf{x})$.

Algorithm 1 *Gradient Boosting*

Paso 1:

$$F_0(x) = \arg\min_{c\in\mathbb{R}} \sum_{i=1}^{N} L(y_i, c).$$

Paso 2: Para $m \in \{1, \ldots, M\}$:

s

2.1 Pseudo - residuos:

$$r_i = -\left[\frac{\partial L\left(y_i, F(\mathbf{x_i})\right)}{\partial F(\mathbf{x_i})}\right]_{F(\mathbf{x})=F_{m-1}(\mathbf{x})}, \text{ para } i = 1, \ldots, N.$$

Si $L\left(y_i, F(\mathbf{x_i})\right) = MSE(y_i, F(\mathbf{x_i}))/2 \Rightarrow r_i = y_i - F(\mathbf{x_i})$, de ahí el nombre de pseudo-residuos. Cabe señalar que el MSE es el error cuadrático medio (MSE por sus siglas en inglés).

s

2.2 Cálculo $\mathbf{a_m}$:

$$(\mathbf{a_m}, \beta_\mathbf{m} = \arg\min_{\mathbf{a},\beta} \sum_{i=1}^{N} \left[r_i - \beta h(\mathbf{x_i}; \mathbf{a})\right]^2,$$

obteniendo así el valor de los parámetros de las funciones h que minimizan el MSE a los pseudo-residuos.

s

2.3 *Line search* del parámetro:

$$\rho_m = \arg\min_{\rho} \sum_{i=1}^{N} L\left(r_i, F_{m-1}(\mathbf{x_i}) + \rho h(\mathbf{x_i}; \mathbf{a_m})\right).$$

Paso 3: Actualización F_m

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \rho_m h(\mathbf{x}; \mathbf{a_m}).$$

Si ahora tenemos en cuenta que *LightGBM* utiliza árboles de regresión, y que la medida de error empleada durante el trabajo es el MSE, el cual es equivalente a minimizar la función $L(\mathbf{y}, F(\mathbf{x})) = MSE(\mathbf{y}, F(\mathbf{x}))/2$ entonces, tal y como se puede ver en [6], el algoritmo de resolución se puede simplificar enormemente, convirtiéndose en un ajuste iterativo de los residuos de los modelos de árbol de decisiones. Veámoslo:

Algorithm 2 Gradient Boosting de árboles de decisión con MSE como función de pérdida.

Paso 1:

$$F_0(\mathbf{x}) = \bar{y}.$$

Paso 2: Para $m \in \{1, \dots, M\}$:

2.1 Pseudo - residuos:

$$r_i = y_i - F_{m-1}(\mathbf{x}) \text{ para } i = 1, \dots, N.$$

2.2 Búsqueda de regiones y coeficientes óptimos del árbol:

$$(\{R_{jm}\}_{j=1}^{J}) = arg \min_{\mathbf{c_m}, \{R_{jm}\}_{j=1}^{J}} \sum_{i=1}^{N} \left[r_i - dt(\mathbf{c_m}, \{R_{jm}\}_{j=1}^{J}) \right]^2,$$

donde dt es un árbol de decisión, J es el número de nodos del árbol, $\mathbf{c_m}$ es el vector de valores que toma el árbol en sus diferentes nodos, y $\{R_{jm}\}_{j=1}^{J}$ es la región del espacio de variables explicativas que acaba en el nodo terminal j-ésimo. Esto se traduce en entrenar un árbol de decisión sobre los residuos de la anterior iteración.

2.3 *Line Search* del parámetro:

$$\rho_m = \arg\min_{\rho} \sum_{i=1}^{N} L\left(r_i, F_{m-1}(\mathbf{x_i}) + \rho \cdot dt(\mathbf{c_m}, \{R_{jm}\}_{j=1}^{J}) \right).$$

Paso 3:

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \sum_{j=1}^{J} \beta_{jm} \mathbb{I}(\mathbf{x} \in R_{jm}),$$

siendo $\beta_{jm} = c_{jm}\rho_m$, y teniendo en cuenta la formulación de los árboles de decisión.

Esta nueva formulación se puede ver como sumar J modelos de árbol distintos en cada una de las regiones R_{jm}, por lo que podemos pensar en estimar los coeficientes en cada nodo terminal de forma separada [6].

$$\{\beta_{jm}\}_{j=1}^{J} = arg \min_{\{\beta_j\}_1^J} \sum_{i=1}^{N} L\left(y_i, F_{m-1}(\mathbf{x_i}) + \sum_{j=1}^{J} \beta_j \mathbb{I}(x \in R_{jm}) \right),$$

y como R_{jm} son disjuntos:

$$\beta_{jm} = arg \min_{\beta} \sum_{x_i \in R_{jm}} L\left(y_i, F_{m-1}(\mathbf{x_i}) + \beta \right),$$

que es la actualización óptima de las constantes de los árboles de decisión para cada nodo terminal teniendo en cuenta la aproximación actual de F, F_{m-1}.

Asimismo, en [6] se menciona que para regular el balance entre sobreajuste y velocidad de entrenamiento, una técnica que da muy buenos resultados es el multiplicar por un valor $\nu \in (0, 1)$ el cada término de la fórmula del paso 3 del algortimo, obteniendo:

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \nu \sum_{j=1}^{J} \beta_{mj} \mathbb{I}(\mathbf{x} \in R_{jm})$$

Esta técnica, habitualmente denominada *shrinkage* se traduce en el *LightGBM* en la introducción de un hiperparámetro al que se denomina *learning rate*. Este es el primero de los hiperparámetros del *LightGBM* mencionados, no obstante, más adelante mencionaremos algunos de los más importantes, pero antes explicaremos como se llevan a cabo los *splits* dentro de los árboles de decisión.

Una vez explicado el *gradient boosting*, vamos a explicar como se lleva a cabo cada *split* en los árboles de decisión de un *LightGBM*, para ello se introduce la ganancia de información, que se basa en la reducción de la varianza después de realizar una división. De acuerdo con [10], y siendo O la muestra de entrenamiento en un nodo fijo de un árbol de decisión, la ganancia de información en dicho nodo V_j condicionada a la muestra de entrenamiento se define como:

$$V_{j|O}(d) = \frac{1}{n_O} \left(\left(\sum_{\mathbf{x_i} \in O : x_{ij} \leq d} g_i \right)^2 \frac{1}{n^j_{l|O}(d)} + \left(\sum_{\mathbf{x_i} \in O : x_{ij} > d} g_i \right)^2 \frac{1}{n^j_{r|O}(d)} \right),$$

donde $n_O = \sum_{i=1}^{N} I[\mathbf{x_i} \in O]$, $n^j_{l|O}(d) = \sum_{i=1}^{N} I[\mathbf{x_i} \in O : x_{ij} \leq d]$ y $n^j_{r|O}(d) = \sum_{i=1}^{N} I[\mathbf{x_i} \in O : x_{ij} > d]$ y g_i es el gradiente negativo de la función de pérdida asociado a $\mathbf{x_i}$.

Consecuentemente, para la variable j, el algoritmo de árbol de decisión selecciona $d^*_j = \arg\max_d V_{j|O}(d)$ y calcula la mayor ganancia $V_{j|O}(d^*_j)$. Luego, los datos se dividen según la característica j^* en el punto d^*_j en los nodos hijo izquierdo y derecho.

Una vez explicado esto, comenzaremos a detallar en que se distingue principalmente el *LightGBM* de otros algoritmos que emplean esta técnica. Tal como se puede ver en [10], es básicamente gracias al uso del *Gradient-based One-Side Sampling* (GOSS), el *Exclusive Feature Bundling* (EFB), y de la creación de histogramas para las variables explicativas.

En primer lugar, el *Gradient-based One-Side Sampling* (GOSS) es una técnica para reducir el tamaño del conjunto de datos manteniendo la información de los gradientes más importantes (mayores). La idea intuitiva es, para cada nodo, quedarse únicamente con las observaciones con mayor error (gradiente mayor) y tan solo con una muestra de las de menor error (pues ya están bien estimadas). Para ello se estiman los gradientes para todas las instancias, se ordenan las instancias por la magnitud del gradiente y se selecciona el top $a \times 100\,\%$ de instancias con los gradientes más grandes (subconjunto de instancias denotado como A) y una muestra aleatoria del subconjunto de instancias restantes A^c al que denotaremos por B. Cabe destacar que este conjunto B tiene $b \times |A^c|$ instancias, donde $a, b \in (0, 1)$ y que tanto A como B son subconjuntos a su vez de O. Una vez establecido esto procedemos a dividir las instancias en función de $\widetilde{V_j(d)}$ sobre $A \bigcup B$, lo que reduce enormemente el costo computacional sin perder apenas precisión en el entrenamiento.

$$\widetilde{V_{j|O}(d)} = \frac{1}{n_O} \left(\left(\sum_{\mathbf{x_i} \in A_l} g_i + \frac{1-a}{b} \sum_{\mathbf{x_i} \in B_l} g_i \right)^2 \frac{1}{n^j_l(d)} + \left(\sum_{\mathbf{x_i} \in A_r} g_i + \frac{1-a}{b} \sum_{x_i \in B_r} g_i \right)^2 \frac{1}{n^j_r(d)} \right),$$

donde $A_l = \{\mathbf{x_i} \in A : x_{ij} \leq d\}$, $A_r = \{\mathbf{x_i} \in A : x_{ij} > d\}$, $B_l = \{\mathbf{x_i} \in B : x_{ij} \leq d\}$, $B_r = \{\mathbf{x_i} \in B : x_{ij} > d\}$, y el coeficiente $\frac{1-a}{b}$ se utiliza para normalizar la suma de los gradientes sobre B al tamaño de A^c.

Gracias a esto podemos llevar a cabo el cálculo de cuáles son la variable j^* y el valor d_j^* que maximizan la ganancia (de forma análoga al cálculo con $V_{j|O}(d)$), pero ahorrando tiempo de computación sin perjudicar demasiado la precisión del algoritmo (véase [10]).

En segundo lugar, el *Exclusive Feature Bundling* (EFB) es una técnica para reducir la dimensión de características combinando características mutuamente excluyentes (que nunca son distintas de 0 a la vez) en una sola característica. Esto permite optimizar el cálculo del histograma reduciendo la carga computacional y la memoria necesaria [10].

En tercer lugar, la construcción de un histograma de características, el cual consiste en agrupar en *bins* los valores de las variables características para así evitar usar todos los posibles valores de una característica, reduciendo la complejidad del algoritmo.

Por último, antes de terminar esta subsección acerca del funcionamiento del *LighGBM* debemos explicar los hiperparámetros fundamentales de este algoritmo. Además del *learning rate* ya mencionado, existen otros hiperparámetros que influyen en en modelo. Estos son los siguientes:

- α se emplea para regularizar la función de pérdida del algoritmo transformando
$$\beta_{jm} = arg\min_{\beta} \sum_{\mathbf{x_i} \in R_{jm}} L\left(y_i, F_{m-1}(\mathbf{x_i}) + \beta\right),$$
en
$$\beta_{jm} = arg\min_{\beta} \sum_{\mathbf{x_i} \in R_{jm}} L\left(y_i, F_{m-1}(\mathbf{x_i}) + \beta\right) + \alpha|\beta|,$$
y evitando así valores muy altos en los coeficientes.
- λ es similar a α solo que actúa sobre la norma L2 de los coeficientes, transformando
$$\beta_{jm} = arg\min_{\beta} \sum_{\mathbf{x_i} \in R_{jm}} L\left(y_i, F_{m-1}(\mathbf{x_i}) + \beta\right),$$
en
$$\beta_{jm} = arg\min_{\beta} \sum_{\mathbf{x_i} \in R_{jm}} L\left(y_i, F_{m-1}(\mathbf{x_i}) + \beta\right) + \lambda\beta^2.$$
- El número máximo de hojas (numLeaves) que un árbol puede tener. Un número mayor aumenta la complejidad del modelo y puede mejorar la precisión, pero también el riesgo de sobreajuste.
- La profundidad máxima de los árboles (maxDepth) es la distancia máxima entre el primer nodo y los nodos terminales. Profundidades más pequeñas hacen el modelo más general, pero pueden reducir su capacidad para capturar relaciones complejas.
- El número de árboles (numIterations) en el modelo. Un número mayor puede mejorar el rendimiento hasta cierto punto, pero aumenta el tiempo de entrenamiento.
- La fracción de características (featureFraction) que permite limitar la proporción de características a usar en cada iteración. Esto reduce la correlación entre árboles y mejora la generalización.
- La fracción de submuestras (baggingFraction) que permite especificar la proporción de datos a usar en cada iteración para entrenar un árbol. Esto ayuda a reducir el sobreajuste.
- El numero de observaciones mínimas en una hoja (minDataInLeaf, que establece el número mínimo de datos que debe tener una nodo, evitando así el sobreajuste.

Una vez explicado en detalle el *LightGBM* procedamos a explicar porqué se escogió este modelo frente a las opciones clásicas de regresión o supervivencia.

3.2. Motivación para la elección del modelo *LightGBM*.

En esta sección, detallaremos las razones que motivaron la elección de los modelos *LightGBM* en comparación con alternativas más simples, como los modelos de supervivencia para estimar tiempos de entrega o modelos logísticos para predecir retrasos en las entregas. Explicaremos las características clave del modelo elegido, así como su aplicabilidad y ventajas dentro del contexto específico de nuestro estudio.

Como mencionamos en la sección anterior, los modelos creados para nuestro proyecto emplean el *LightGBM*, una implementación eficiente de gradient boosting que destaca por su rendimiento en grandes conjuntos de datos y su manejo excepcional de variables categóricas. *LightGBM* utiliza el algoritmo de Fisher para optimizar la codificación de variables categóricas, buscando codificaciones ordinales que resulten en divisiones óptimas en la construcción de árboles de decisión. Esto reduce significativamente la dimensionalidad que resultaría de la aplicación de técnicas de codificación como *One-Hot Encoding* que son las empleadas en la mayoría de los modelos clásicos.

Para entender por qué el *One-Hot Encoding* es un método poco recomendado para nuestro conjunto de datos primero debemos entender qué hace. Pues bien, el *One-Hot Encoding* es una técnica utilizada para convertir variables categóricas en un formato adecuado para modelos estadísticos, convirtiendo cada categoría en una nueva columna binaria (0 o 1) [17]. Si bien no asume ningún orden entre las categorías y permite tratar cada una de manera independiente, su principal desventaja es el aumento de la dimensionalidad, especialmente con variables categóricas de alta cardinalidad. Este incremento en el número de características puede aumentar la complejidad computacional y generar matrices dispersas con muchas entradas de cero, lo que resulta ineficiente en términos de almacenamiento y cálculo, pudiendo incluso llevar a problemas de convergencia en los métodos clásicos [22]. Esta desventaja se hizo evidente en nuestro proyecto cuando intentamos aplicar modelos de supervivencia. El aumento en la dimensionalidad no solo complicó el procesamiento de los datos, sino que también provocó dificultades significativas en la convergencia de los modelos, llevándonos a descartar este tipo de enfoque.

En contraste, el algoritmo de Fisher, que se centra en la agrupación de observaciones para maximizar la homogeneidad dentro de los grupos [4], se emplea en *LightGBM* para mejorar la pureza de los nodos en los árboles de decisión. La pureza de un nodo se refiere a la homogeneidad de las observaciones dentro de ese nodo respecto a la variable objetivo. Este enfoque es crucial cuando se manejan variables categóricas con alta cardinalidad, como `id_proveedor`, `id_origen_envio`, y `id_destino_envio`, que tienen más de 900, 200 y 100 categorías distintas respectivamente. Agrupar categorías similares de manera efectiva permite a *LightGBM* realizar divisiones más precisas y manejar variables complejas sin expandir masivamente el espacio de características.

Además de su capacidad para manejar variables categóricas de alta cardinalidad sin recurrir a *One-Hot Encoding*, *LightGBM* tiene la ventaja adicional de gestionar datos faltantes de manera eficiente sin necesidad de imputarlos. Esta capacidad es particularmente valiosa en nuestro contexto, ya que los datos faltantes pueden tener un significado específico, como indicar que un determinado evento aún no ha ocurrido. En modelos tradicionales, como la regresión lineal o GAMs, los datos faltantes suelen requerir imputación, lo que puede llevar a la pérdida de información crítica y potencialmente a sesgos en los resultados. *LightGBM* trata los datos faltantes de manera natural, permitiendo que el modelo aprenda de la ausencia de datos como una característica más, lo que enriquece la capacidad predictiva del mismo.

El uso de *LightGBM* nos permite manejar de manera más eficiente y efectiva el gran volumen de datos y la alta cardinalidad de las variables categóricas presentes en nuestro estudio, reduciendo la complejidad del mismo sin sacrificar precisión. Esta combinación de manejo avanzado de variables categóricas y capacidad para gestionar datos faltantes hace que *LightGBM* sea una elección superior

frente a los métodos convencionales, y ese es el motivo de haberlo elegido.

3.3. Estructura de los modelo creados

En esta sección, describiremos la metodología adoptada para el desarrollo de los modelos empleados (tanto para la predicción de `retraso` como de `dias_hasta_fecha_total_entrado`), comenzando con la búsqueda inicial de hiperparámetros, cuyo resultado se empleó en la selección de variables a través de un algoritmo genético. Finalmente, una vez escogidas dichas variables se concluyó con la búsqueda final de hiperparámetros para optimizar el rendimiento del modelo.

Este primer paso de la búsqueda de hiperparámetros se aborda a través de la optimización bayesiana, una técnica que se distingue por su enfoque probabilístico. A diferencia de métodos más tradicionales como la búsqueda en rejilla o aleatoria, la optimización bayesiana construye un modelo probabilístico de la función objetivo en relación con los hiperparámetros. Este modelo se utiliza para predecir los hiperparámetros más prometedores, enfocando la búsqueda en áreas del espacio de hiperparámetros que probablemente ofrezcan mejores valores de la función objetivo. Este enfoque no solo ahorra recursos, sino que también acelera significativamente el proceso de optimización. En [21] se ofrece una visión profunda de cómo la optimización bayesiana puede aplicarse de manera efectiva en este contexto, ilustrando su superioridad sobre métodos de búsqueda menos eficientes. Igualmente, en la Sección 3.3.1 entraremos en detalle acerca del funcionamiento de este algoritmo.

Seguidamente y tras identificar un conjunto prometedor de hiperparámetros iniciales, el proceso de optimización se adentra en la selección de variables, utilizando algoritmos genéticos. Inspirados en los principios de la evolución biológica, estos algoritmos simulan la selección natural, donde solo las soluciones más aptas sobreviven y se reproducen. Aplicando operadores genéticos como la selección, el cruce y la mutación, los algoritmos genéticos exploran el espacio de variables de forma iterativa para encontrar las combinaciones que mejor se adaptan al problema en cuestión. Este método es particularmente eficaz para manejar espacios de búsqueda grandes y complejos, donde la intuición humana o los métodos más simples podrían no ser suficientes. El trabajo reflejado en [15] proporciona una base teórica sólida y ejemplos prácticos de cómo estos algoritmos pueden ser implementados para la selección de variables, destacando su capacidad para descubrir interacciones no evidentes entre variables.

Con las variables óptimamente seleccionadas, se procede a una búsqueda final de hiperparámetros. Este último paso refina los hiperparámetros iniciales en el contexto del conjunto reducido y optimizado de variables, buscando el conjunto de hiperparámetros que maximizará el rendimiento del modelo. Este proceso iterativo garantiza que tanto la estructura del modelo (a través de sus hiperparámetros) como su contenido (las variables seleccionadas) estén óptimamente alineados para abordar el problema con la máxima eficiencia y eficacia.

3.3.1. Búsqueda de hiperparámetros

La optimización bayesiana es un método avanzado para encontrar el mínimo de una función objetivo $f(x)$ que es desconocida y costosa de evaluar. Por consiguiente, esta técnica es ideal para la búsqueda de hiperparámetros en modelos complejos como *LightGBM*. En este trabajo empleamos el paquete *scikit-optimize* para implementarlo (véase [8]). Seguidamente, explicaremos en detalle en que consiste la optimización bayesiana.

Como acabamos de mencionar la optimización bayesiana es un método ideal para la búsqueda de hiperparámetros de algoritmos de aprendizaje automático que se basa en la teoría bayesiana y utiliza procesos gaussianos para modelar una función objetivo desconocida. Dentro de este contexto, el

objetivo es encontrar el conjunto de hiperparámetros que maximice el rendimiento del modelo con el menor número de evaluaciones posible. Para ello, la optimización bayesiana sigue un enfoque iterativo donde cada evaluación proporciona información que se usa para actualizar la creencia sobre la función objetivo y seleccionar el siguiente punto de evaluación [5]. Seguidamente, y antes de entrar en detalles acerca de esta técnica explicaremos brevemente en que consiste un procesos gausiano.

Un proceso gausiano (GP) es una generalización del concepto de una distribución normal multivariante a funciones continuas. Mientras que una distribución normal multivariante describe la distribución de un vector de variables aleatorias, un proceso gausiano describe la distribución de una función aleatoria $f(x)$ en todo su dominio. Es decir, en lugar de definir una media y una covarianza para un conjunto finito de puntos, un GP define una media y una función de covarianza para todos los puntos posibles en el espacio de entrada.

En el contexto de la optimización bayesiana, el GP se utiliza para modelar la función objetivo f. Esta modelización permite predecir el valor de f en nuevos puntos y cuantificar la incertidumbre de estas predicciones. El GP se define por dos componentes: una función de media que representa la media esperada de la función en cada punto x, y una función de covarianza que describe cómo varían conjuntamente los valores de f en dos puntos distintos del espacio [5].

Antes de observar cualquier dato, el GP se define por su distribución *a priori*. Esta distribución refleja nuestras creencias iniciales sobre la función objetivo antes de observar los datos y se caracteriza por la media $\mu_0(x)$ y la covarianza $\Sigma_0(x, x')$. La notación formal de esta distribución *a priori* para un conjunto de puntos $x_1, \ldots, x_k$ es:

$$f(x_{1:k}) \sim \text{Normal}(\mu_0(x_{1:k}), \Sigma_0(x_{1:k}, x_{1:k}))$$

donde:

$$x_{1:k} = [x_1, \ldots, x_k]$$
$$f(x_{1:k}) = [f(x_1), \ldots, f(x_k)]$$
$$\mu_0(x_{1:k}) = [\mu_0(x_1), \ldots, \mu_0(x_k)]$$
$$\Sigma_0(x_{1:k}, x_{1:k}) = \begin{pmatrix} \Sigma_0(x_1, x_1) & \cdots & \Sigma_0(x_1, x_k) \\ \vdots & \ddots & \vdots \\ \Sigma_0(x_k, x_1) & \cdots & \Sigma_0(x_k, x_k) \end{pmatrix}.$$

Una vez que se observan los datos, se actualiza esta creencia a una distribución *a posteriori* utilizando el teorema de Bayes. La distribución *a posteriori* combina la información de la distribución *a priori* con los datos observados para proporcionar una predicción actualizada de la función objetivo. En términos del GP, la media predicha $\mu_n(x)$ y la varianza predicha $\sigma_n^2(x)$ se actualizan como sigue:

$$f(x)|f(x_{1:n}) \sim \text{Normal}(\mu_n(x), \sigma_n^2(x)),$$

donde:

$$\mu_n(x) = \Sigma_0(x, x_{1:n})\Sigma_0(x_{1:n}, x_{1:n})^{-1}(f(x_{1:n}) - \mu_0(x_{1:n})) + \mu_0(x),$$

$$\sigma_n^2(x) = \Sigma_0(x, x) - \Sigma_0(x, x_{1:n})\Sigma_0(x_{1:n}, x_{1:n})^{-1}\Sigma_0(x_{1:n}, x),$$

donde Σ_0 es la función de covarianza del GP, $x_{1:n}$ son los puntos ya evaluados, $f(x_{1:n})$ son los valores observados en esos puntos, y μ_0 es la media inicial del GP (véase [5]).

Una vez explicado el procedimiento por el cual se estima la función objetivo, solo queda entender como se escoge el siguiente punto para la evaluación. Para esta tarea la optimización bayesiana utiliza una función de adquisición basada en el *GP* actual. Una de las funciones de adquisición más comunes, que es además la que hemos empleado en este trabajo, es el *Expected Improvement (EI)*, que se define como:

$$EI_n(x) = \mathbb{E}_n \left[(f(x) - f_n^*)^+ \right]$$

donde f_n^* es el mejor valor observado hasta el momento, y $\mathbb{E}_n$ denota la expectativa bajo la distribución *a posteriori*. El EI puede calcularse en forma cerrada como:

$$EI_n(x) = [\Delta_n(x)]^+ + \sigma_n(x)\phi\left(\frac{\Delta_n(x)}{\sigma_n(x)}\right) - |\Delta_n(x)|\Phi\left(\frac{\Delta_n(x)}{\sigma_n(x)}\right)$$

donde $\Delta_n(x) = \mu_n(x) - f_n^*$, ϕ es la función de densidad de la normal estándar y Φ es la función de distribución acumulativa de la normal estándar.

La optimización bayesiana actualiza iterativamente el modelo *GP* con las nuevas evaluaciones, recalcula la función de adquisición, y selecciona el próximo punto a evaluar maximizando la función de adquisición:

$$x_{n+1} = arg\,\text{máx}\, EI_n(x).$$

Este enfoque balancea la exploración de áreas con alta incertidumbre (alta $\sigma_n(x)$) y la explotación de áreas con alta media esperada (alta $\mu_n(x)$).

En resumen, la optimización bayesiana es un enfoque eficiente para la optimización de hiperparámetros, especialmente útil cuando las evaluaciones de la función objetivo son costosas. Utiliza un modelo probabilístico para hacer un balance entre la exploración y la explotación, mejorando iterativamente la precisión de las predicciones sobre la función objetivo.

3.3.2. Selección de variables: Algoritmo genético

Un algoritmo genético (GA, por sus siglas en inglés) es una técnica de optimización y búsqueda inspirada en los principios de la selección natural y la genética. Los algoritmos genéticos pertenecen a la categoría de algoritmos evolutivos y son utilizados para resolver problemas de optimización en los que otras técnicas pueden ser ineficaces. En la Figura 3.1 se describen los componentes y el funcionamiento de un algoritmo genético aplicado a la selección de variables para un modelo *LightGBM*:

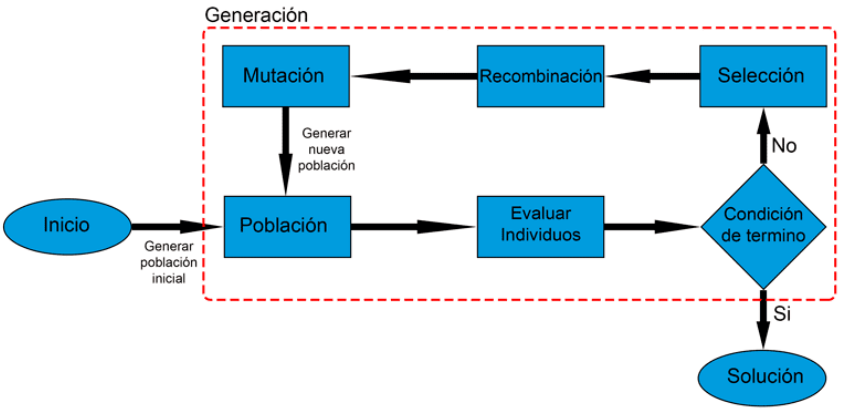

Figura 3.1: Diagrama de flujo de un algoritmo genético.

El algoritmo genético comienza con una población inicial de soluciones potenciales, también conocidas como individuos o cromosomas. Estos individuos son usualmente representados mediante cadenas de caracteres, números binarios, o cualquier otra estructura adecuada para el problema en cuestión [7]. La representación binaria es común, donde cada cromosoma puede verse como una cadena de bits. En este contexto, cada bit de la cadena representa la presencia (1) o ausencia (0) de una variable en el modelo *LightGBM*.

Cada individuo en la población es evaluado mediante una función de aptitud o *fitness* que cuantifica qué tan buena es la solución representada por el individuo para el problema específico de selección de variables. Esta función de evaluación es crucial ya que guía el proceso evolutivo [7]. La función de aptitud $f(x)$ mide la calidad de la solución x en el contexto del rendimiento predictivo del modelo *LightGBM*, y nosotros decicimos emplear el error cuadrático medio (*MSE* por sus siglas en inglés).

$$f(x) = MSE(\text{Modelo con las variables codificadas en el individuo } x).$$

Basándose en los valores de aptitud, se seleccionan individuos para reproducirse y generar la siguiente generación. Métodos comunes de selección incluyen la selección por torneo, la ruleta y la selección por rango [16].

En la selección por ruleta, los individuos son seleccionados de manera proporcional a su aptitud, similar a una ruleta en la que los sectores tienen tamaños proporcionales a las aptitudes. La selección por rango ordena a los individuos por aptitud y les asigna probabilidades de selección basadas en su orden, lo que puede reducir el impacto de diferencias extremas en aptitudes. La selección por torneo, que hemos empleado, implica elegir al azar grupos pequeños de individuos y seleccionar al mejor de cada grupo para reproducirse. Este método es eficaz para mantener la diversidad genética y favorecer a los individuos de alto rendimiento, ya que permite que incluso los individuos con menor aptitud tengan una oportunidad de ser seleccionados, especialmente en torneos pequeños. La idea es que los individuos con mejores aptitudes tienen una mayor probabilidad de ser seleccionados para reproducirse, pero sin eliminar completamente la posibilidad de selección de los individuos menos aptos, lo que ayuda a explorar más ampliamente el espacio de soluciones.

A continuación, se lleva a cabo la recombinación, una etapa donde los operadores genéticos son fundamentales. El operador de cruce combina partes de dos o más individuos (padres) para producir nuevos individuos (hijos).

El cruce puede ser de un punto, en el cual la cadena de genes de los padres se divide en un punto específico y se intercambian las partes para formar los hijos; de dos puntos, donde se eligen dos puntos de cruce y se intercambian los segmentos entre estos puntos; o uniforme, el cual hemos empleado, donde cada gen del descendiente se elige aleatoriamente de uno de los dos padres con igual probabilidad. Este método uniforme facilita la exploración equilibrada del espacio de búsqueda, ya que permite una mezcla más variada y completa de las características de ambos padres en los hijos [7].

Padre 1	Padre 2
010110	110101
↓	↓
Hijo 1	Hijo 2
010101	110110

Seguidamente, el algoritmo lleva a cabo la mutación, donde se introducen variaciones aleatorias en los individuos, modificando uno o más genes en un cromosoma. En este caso, se empleó la mutación por *flipbit*, que consiste en invertir el valor de un bit en la cadena binaria del cromosoma, cambiando de 0 a 1 o de 1 a 0. La mutación por *flipbit* es simple pero efectiva, ya que permite explorar nuevas combinaciones de variables que pueden no ser generadas a través del cruce. Este tipo de mutación es crucial para mantener la diversidad genética dentro de la población y para evitar la convergencia prematura a soluciones subóptimas [7]. Al introducir variaciones de esta manera, el algoritmo puede escapar de óptimos locales y continuar buscando mejores soluciones en el espacio de búsqueda.

$$\text{Mutación:} \quad 010101 \rightarrow 010111.$$

En nuestro caso, los nuevos individuos generados a través del cruce y la mutación reemplazan completamente a todos los individuos de la población actual, formando una nueva generación. Este enfoque de reemplazo total garantiza que cada nueva generación esté completamente formada por descendientes, lo que puede ayudar a explorar más ampliamente el espacio de soluciones y evitar la convergencia prematura a soluciones subóptimas [16].

El proceso de selección, cruce, mutación y reemplazo se repite durante muchas generaciones. Con cada iteración, la población evoluciona y las soluciones tienden a mejorar. El criterio de terminación puede ser un número fijo de generaciones, la convergencia de la población, o el tiempo de ejecución [7].

Idealmente, después de varias iteraciones, la población converge hacia una solución óptima o casi óptima del problema. Sin embargo, en la práctica, puede haber desafíos como la convergencia prematura o la necesidad de un balance entre exploración y explotación [7].

En resumen, los algoritmos genéticos son una poderosa herramienta para la búsqueda y optimización, inspirados en los mecanismos de la evolución natural. Gracias a sus operadores genéticos y su capacidad de explorar grandes espacios de búsqueda, son aplicables en una amplia gama de problemas complejos en ingeniería, ciencia y negocios.

Una de las ventajas clave de los algoritmos genéticos es su flexibilidad, ya que pueden aplicarse a una diversidad de problemas sin requerir suposiciones específicas sobre el problema. Además, su capacidad de búsqueda global les permite encontrar soluciones en problemas con múltiples óptimos locales gracias a su capacidad de explorar extensamente el espacio de búsqueda [11]. La evaluación de la aptitud de cada individuo puede realizarse en paralelo, lo que reduce el tiempo de computación [2].

Además, su enfoque poblacional y estocástico los hace robustos en la selección de variables frente a otros algoritmos *greedy* [12].

Sin embargo, los algoritmos genéticos presentan algunas desventajas. Por ejemplo, en comparación con otros métodos de selección de variables, pueden requerir más tiempo para converger a una solución óptima, especialmente en problemas de alta dimensión [12]. La eficiencia de un algoritmo genético depende significativamente de la elección adecuada de sus parámetros [15]. Existe también el riesgo de convergencia prematura a soluciones subóptimas si la diversidad genética se pierde demasiado rápido [2].

Una vez explicados en detalle estos algoritmos pasaremos a explicar la fase final de la estructura del modelo: la búsqueda final de hiperparámetros.

3.3.3. Búsqueda final de hiperparámetros

Una vez determinado el conjunto óptimo de variables mediante el uso de un algoritmo genético, se procede a la etapa de búsqueda final de hiperparámetros. Este proceso es esencial, ya que permite realizar ajustes finos al modelo, adaptándolo de manera más precisa al conjunto de variables seleccionadas previamente. Esta búsqueda final de hiperparámetros se centra en optimizar aquellos que pueden influir sustancialmente en el rendimiento del modelo, asegurando que este opere con la máxima eficacia posible dentro del espacio de soluciones viables.

Este es el momento en el que el modelo se calibra finamente para alinearse con las características específicas del conjunto de datos y las variables seleccionadas. Este ajuste minucioso contribuye significativamente a la robustez y precisión del modelo, lo cual es vital para asegurar que los resultados finales sean confiables y aplicables al problema en cuestión. Además, este enfoque sistemático y metódico para la optimización de hiperparámetros garantiza que el modelo final esté completamente optimizado, no solo en términos de las variables que utiliza, sino también en cómo estas interactúan y contribuyen al rendimiento general del modelo.

Por lo tanto, la integración de la optimización bayesiana en la fase de afinamiento de hiperparámetros refuerza la metodología general de desarrollo del modelo, proporcionando un marco sólido para alcanzar un rendimiento óptimo y ofreciendo una mayor garantía de que el modelo será efectivo y eficiente en la práctica real.

3.4. Descripción detallada del proceso en los modelos finales

En esta sección, detallamos el proceso de desarrollo y afinamiento de los modelos finales siguiendo la metodología descrita en la Sección 3.3. Este proceso se divide en tres etapas principales: selección inicial de hiperparámetros, selección de variables y búsqueda final de hiperparámetros.

3.4.1. Modelo para la variable `dias_hasta_fecha_total_entrado`

Este modelo tiene como objetivo estimar la variable dias_hasta_fecha_total_entrado utilizando un conjunto inicial de variables predictoras, detalladas en la Sección A.1 del apéndice, a través de un proceso de selección de variables y búsqueda de hiperparámetros.

Para el ajuste inicial del modelo *LightGBM* antes de la selección de variables, hemos afinado varios hiperparámetros clave para optimizar su rendimiento. A continuación, se describen cada uno de estos hiperparámetros, explicando su función y los rangos de valores considerados durante la optimización.

Búsqueda Inicial de Hiperparámetros

El hiperparámetro `alpha` controla la regularización L1, que ayuda a prevenir el sobreajuste penalizando el valor absoluto de los coeficientes, y se seleccionó de manera uniforme en el rango $[0, 1]$.

El `learningRate` (tasa de aprendizaje) determina el tamaño de los pasos que el modelo da al ajustar los pesos con respecto a la función de pérdida. Un valor más bajo hace que el modelo aprenda más lentamente, pero puede resultar en una mejor generalización. Para este hiperparámetro, se utilizó una distribución log-uniforme en el rango [exp(-5), exp(0)], lo que permite explorar una amplia gama de valores de tasa de aprendizaje en una escala logarítmica.

El hiperparámetro `numLeaves` especifica el número máximo de hojas en cada árbol de decisión. Más hojas pueden capturar más detalles de los datos, pero también pueden llevar a un sobreajuste. Los valores para `numLeaves` se seleccionaron usando una distribución uniforme cuantizada entre 20 y 200.

El `numIterations` indica el número de iteraciones (o árboles) que el modelo utilizará. Más iteraciones pueden mejorar el ajuste del modelo, pero también aumentan el riesgo de sobreajuste y el tiempo de entrenamiento. Los valores de `numIterations` se seleccionaron usando una distribución uniforme cuantizada en el rango [50, 1000].

El `maxDepth` define la profundidad máxima de cada árbol. Limitar la profundidad ayuda a prevenir el sobreajuste. Los valores de `maxDepth` se eligieron usando una distribución uniforme cuantizada entre 1 y 50.

El `featureFraction` indica la fracción de características (features) que se seleccionarán aleatoriamente en cada iteración para construir los árboles. Un valor más bajo puede ayudar a reducir el sobreajuste. Los valores se seleccionaron de manera uniforme en el rango $[0, 1]$. Similar a `featureFraction`, pero aplicable a las muestras de datos, el `baggingFraction` controla la proporción de datos que se utilizarán en cada iteración. Los valores de `baggingFraction` se seleccionaron de manera uniforme en el rango [0, 1.0].

El `minDataInLeaf` especifica el número mínimo de datos que debe tener una hoja. Este hiperparámetro ayuda a evitar que el modelo se ajuste demasiado a ruidos específicos del conjunto de entrenamiento. Los valores para `minDataInLeaf` se seleccionaron usando una distribución uniforme cuantizada en el rango $[2, 70]$.

Para optimizar estos hiperparámetros, se utilizó un proceso de optimización bayesiana, una técnica eficiente para encontrar los valores óptimos en espacios de hiperparámetros complejos (ver Sección 3.3). Se realizaron un total de 150 evaluaciones, número elegido basado en experimentación previa, asegurando un balance entre el tiempo de computación y la probabilidad de encontrar un conjunto óptimo de hiperparámetros. Se implementó un criterio de parada temprana que detiene el proceso si no hay progreso en la reducción de la función de pérdida (*MSE*) durante 25 evaluaciones consecutivas, ayudando a evitar un uso innecesario de recursos computacionales una vez que se ha alcanzado un rendimiento estable.

Después de completar el proceso de optimización, se identificaron los siguientes hiperparámetros como óptimos para nuestro modelo *LightGBM*: `alpha` de 0.4469, `learningRate` de 0.2507, `numLeaves` de 164, `numIterations` de 896, `maxDepth` de 39, `featureFraction` de 0.2622, `baggingFraction` de 0.6218, y `minDataInLeaf` de 3. El modelo entrenado con estos hiperparámetros óptimos alcanzó un error cuadrático medio (RMSE) de 2.7831, indicando un ajuste adecuado a los datos de entrenamiento y validación.

Selección de variables

En la etapa de selección de variables del modelo *LightGBM*, se utilizó un algoritmo genético para identificar el subconjunto óptimo de variables que maximiza el rendimiento del modelo. Este enfoque nos permitió reducir la complejidad del modelo sin sacrificar su precisión. El fundamento teórico de los algoritmos genéticos ha sido explicado previamente en la Sección 3.3. A continuación, se detallan los parámetros utilizados en el algoritmo y los resultados obtenidos.

El tamaño de la población se estableció en 200 individuos, lo que proporciona una amplia diversidad genética para la exploración de soluciones. La probabilidad de *crossover* (cruzamiento) se fijó en 0.5, permitiendo que la mitad de los individuos en cada generación se combinen para crear nuevos individuos. La probabilidad de que un individuo sufra una mutación fue del 0.1, asegurando que se mantenga cierta variabilidad en la población. Adicionalmente, la probabilidad de mutación en cada bit del individuo se fijó también en 0.1, lo que permite cambios menores en las soluciones propuestas.

El algoritmo utilizó torneos de selección con un tamaño de 40 individuos, donde se seleccionaron 4 ganadores para contribuir a la próxima generación. Este método de selección ayuda a preservar los mejores individuos y a mantener una presión selectiva adecuada.

Como resultado de este proceso, se logró un modelo con un *RMSE* de 2.70, ligeramente mejor que el obtenido en la optimización de hiperparámetros inicial pero con menos variables. De las variables iniciales consideradas, se eliminaron las variables `correccion_fecha`, `es_distribucion_confirmada`, `es_contenedor_aduana`, `es_destino_externo`, `dias_hasta_fecha_handover_forwarder`, `dias_hasta_fecha_asignacion_proveedor`, `dias_hasta_primera_fecha_formalizada`, `dias_hasta_fecha_servicio`, `id_origen_envio_entrega_imputada`, `porcentaje_mediano_entrado_cd_seccion`, simplificando así el modelo sin pérdida significativa de precisión. Las variables con las que se llevó a cabo el modelo se pueden consultar en la Sección A.1 del apéndice.

Búsqueda final de hiperparámetros

Después de realizar la selección de variables, procedimos a una nueva búsqueda de hiperparámetros para el modelo *LightGBM*. Esta búsqueda se llevó a cabo de manera similar a la búsqueda inicial de hiperparámetros, utilizando el mismo espacio de búsqueda que en la etapa inicial.

Para esta segunda fase de optimización, se realizaron un total de 180 evaluaciones. Este número se eligió basado en recomendaciones previas y la necesidad de explorar el espacio de hiperparámetros de manera exhaustiva después de la selección de variables. Además, se implementó un criterio de parada temprana que detiene el proceso si no hay progreso en la reducción de la función de pérdida durante 25 evaluaciones consecutivas, asegurando que el proceso de optimización no continúe innecesariamente una vez que se haya alcanzado un rendimiento estable.

Después de completar el proceso de optimización, se identificaron los siguientes hiperparámetros como óptimos para nuestro modelo *LightGBM*:
`alpha` = 0,4787, `learningRate` = 0,2539, `numLeaves` = 175, `numIterations` = 1000, `maxDepth`= 34, `featureFraction`= 0,8836, `baggingFraction` = 0,3178, y `minDataInLeaf` = 5.

El modelo entrenado con estos hiperparámetros óptimos, después de la selección de variables, alcanzó un error cuadrático medio (*RMSE*) de 2.6297 en el conjunto de datos de prueba. Esto representa una mejora en comparación con el *RMSE* obtenido durante la búsqueda inicial de hiperparámetros, destacando la eficacia del proceso de selección de variables y la posterior optimización de hiperparámetros.

3.4.2. Modelo para la variable `retraso`

Este modelo tiene como objetivo estimar la variable retraso utilizando un conjunto inicial de variables predictoras, detalladas en en la Sección A.1 del apéndice, a través de un proceso de selección de variables y búsqueda de hiperparámetros.

Para el ajuste inicial del segundo modelo *LightGBM*, seguimos un proceso similar al del primer modelo, ajustando varios hiperparámetros y llevando a cabe un proceso de selección de variables. A continuación, se describen estos hiperparámetros y los valores seleccionados durante la optimización inicial.

Búsqueda Inicial de Hiperparámetros

La búsqueda inicial de hiperparámetros se llevó a cabo de forma análoga a la del modelo anterior, empleando la optimización bayesiana sobre el mismo espacio de búsqueda y con el mismo número de iteraciones e idéntica condición de parada. En estas condiciones se obtuvieron los siguientes hiperparámetros: `alpha`: 0.49299, `learningRate`: 0.30018, `numLeaves`: 168, `numIterations`: 812, `maxDepth`: 43, `featureFraction`: 0.99553, `baggingFraction`: 0.35245, y `minDataInLeaf`: 35. Este modelo alcanzó un *RMSE* de 2.7339, indicando un buen ajuste inicial.

Selección de variables

En la etapa de selección de variables, se utilizó nuevamente un algoritmo genético para identificar el subconjunto óptimo de variables, siguiendo la metodología descrita en la sección correspondiente al primer modelo. Este proceso nos permitió simplificar el modelo al eliminar las variables: `uds_entradas_acumuladas`, `id_proveedor_registro_imputado`, `correccion_fecha`, `dias_hasta_fecha_servicio_destino_programada`, `etapa_envio`, `id_proveedor_entrega_imputada` y `dias_hasta_fecha_envio`. Las variables con las que se llevó a cabo el modelo se pueden consultar en la Sección A.1 del apéndice.

Después de esta selección, el modelo mostró una mejora con un *RMSE* de 2.7128, manteniendo la precisión mientras se reducía la complejidad del modelo.

Búsqueda final de hiperparámetros

Posteriormente, realizamos una búsqueda final de hiperparámetros utilizando el mismo espacio de búsqueda y criterios de optimización que en la etapa final del primer modelo. Para esta segunda fase, y al igual que para el modelo sobre los días hasta la fecha de entrega, se realizaron 180 evaluaciones, incluyendo un criterio de parada temprana si no se observaba progreso en la reducción del MSE durante 25 evaluaciones consecutivas.

Los hiperparámetros óptimos identificados en esta etapa fueron:
`alpha`: 0.19076, `learningRate`: 0.39238, `numLeaves`: 197, `numIterations`: 920, `maxDepth`: 18, `featureFraction`: 0.61836, `baggingFraction`: 0.48694, y `minDataInLeaf`: 22.

El modelo final, entrenado con estos hiperparámetros después de la selección de variables, alcanzó un *RMSE* de 2.6893, mostrando una mejora respecto a la búsqueda inicial y destacando la eficacia del proceso de selección de variables y optimización de hiperparámetros similar al del primer modelo.

Capítulo 4

Medidas de error e interpretación de modelos

En este capítulo, se presentan y analizan los resultados de los modelos predictivos desarrollados en este trabajo de fin de máster. Se examinan las medidas de error obtenidas, utilizando como principal métrica el Error Absoluto Medio (MAE). La elección del MAE se debe a que es una medida muy intuitiva que mantiene las unidades originales del problema, lo cual permite, de un vistazo a la gráfica, saber cuál es el error en días. Las métricas de error se compararán con las del algoritmo actualmente implementado en la compañía. Este algoritmo utiliza la mejor fecha disponible en cada momento, comenzando con un conjunto de lógicas basadas en el tipo de transporte, servicio, destino y origen para proporcionar una estimación inicial del tiempo de llegada, lo cual ayuda al comprador a planificar la llegada de la mercancía. A medida que avanza el proceso, el modelo se actualiza con las estimaciones proporcionadas primero por el proveedor, luego tras el handover, con la información del encargado del transporte, y finalmente se ajusta según la cita de llegada de la mercancía en el centro de distribución.

Además, se profundiza en la interpretación de los modelos a través de técnicas avanzadas como los valores de Shapley. Este análisis se lleva a cabo tanto a nivel general como desglosado por centro de distribución, tipo de transporte y secciones específicas, proporcionando una visión detallada del rendimiento y comportamiento de los modelos, y permitiendo una evaluación comparativa con el modelo existente en la compañía.

4.1. Modelo para los Días Hasta la Fecha de Entrada de los Envíos

En esta sección, se presentan los resultados del modelo diseñado para predecir los días hasta la fecha de entrada de los envíos. Se realizarán análisis detallados de los errores cometidos por el modelo, seguidos de un estudio sobre la interpretabilidad del mismo mediante el uso de valores de Shapley.

4.1.1. Resultados

A continuación, se presenta una descripción general de las gráficas utilizadas para analizar el Error Absoluto Medio (MAE) en función de los días hasta la fecha de entrega total. Es importante destacar que, a nivel global, se logró una mejora del 41 % en términos de MAE y un 45 % en términos de $RMSE$.

En las Figuras de la 4.1 a la 4.6, los datos se representan como sigue.

En azul, se observan las métricas a nivel de *packing*. La línea continua azul muestra el error obtenido con la fecha pendiente actual (sin modelo) cuando disponemos de información del *packing*, mientras que la línea discontinua azul indica el error predicho por el modelo cuando se tiene esta información.

En naranja, se presentan las métricas a nivel de entrega. La línea continua naranja representa el error obtenido con la fecha pendiente actual (sin modelo) cuando solo se dispone de información a nivel de entrega, y la línea discontinua naranja muestra el error predicho por el modelo en estas mismas condiciones.

En las siguientes figuras se muestran las comparaciones de los errores obtenidos mediante las fechas actuales y las predicciones del modelo.

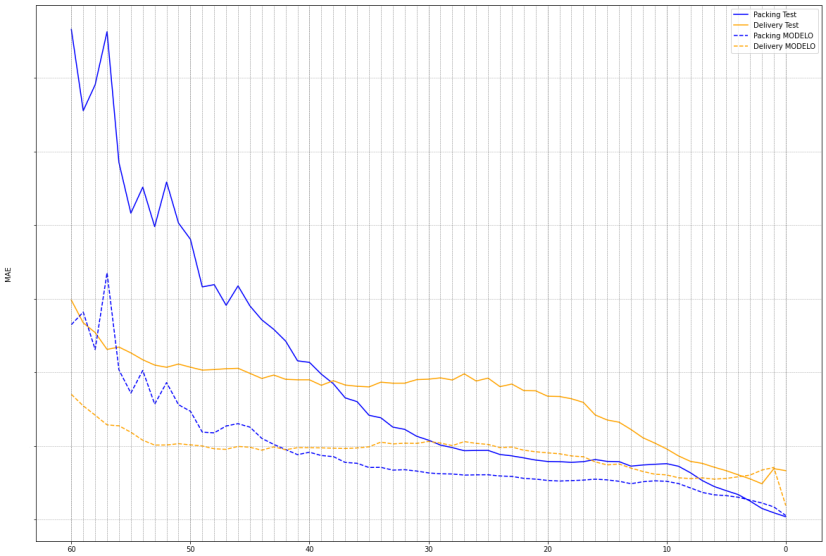

Figura 4.1: Gráficos del *MAE* obtenido con la `siguiente_fecha_pendiente_transporte` del algoritmo actual (linea continua) y con el modelo (linea discontinua), comparando este error a nivel de *packing* (línea azul) y de entrega (línea naranja). El eje de la X representa los días hasta la fecha prevista.

En la Figura 4.1 se muestra en el eje Y el *MAE*, mientras que el eje X muestra los días restantes hasta la fecha de entrega. El análisis de esta gráfica muestra que, hasta aproximadamente 3 días antes de la fecha de entrega, las predicciones del modelo (líneas discontinuas) son generalmente mejores, es decir, con menor *MAE* que las obtenidas con la `siguiente_fecha_pendiente_transporte` (líneas continuas). Esto se observa tanto en el packing (líneas azules) como en la entrega (líneas naranjas). Sin embargo, a medida que nos acercamos a 3 días antes de la fecha de entrada, la `siguiente_fecha_pendiente_transporte` mejora las predicciones del modelo, resultando en un menor *MAE*.

Este comportamiento indica que el modelo es más efectivo en sus predicciones cuando la fecha de entrega está más alejada, mientras que las fechas dadas por el sistema actual son más precisas cuando estamos más cerca de la entrega. Esta información es crucial para optimizar los procesos logísticos, ya que sugiere que el modelo puede ser particularmente útil para predicciones a largo plazo, pero que a corto plazo, confiar en las fechas actuales puede ser más ventajoso.

Como veremos en las siguientes figuras, este comportamiento se mantendrá con mayor o menor in-

tensidad a lo largo de diferentes centros de distribución, secciones y tipos de transporte, lo que sugiere que es un comportamiento generalizado del modelo. Esto puede tener múltiples explicaciones, pero una de ellas es que la fecha de entrega se ve influida por muchas variables al comienzo del proceso y a medida que pasa el tiempo, estas variables dejan de tener un efecto significativo sobre la fecha de entrega y el modelo no es capaz de responder adecuadamente a este tipo de situaciones.

A continuación, se presentarán las diferentes gráficas de error para algunos de los centros de distribución, teniendo en cuenta las secciones asociadas. Por motivos de confidencialidad, enumeraremos tanto los centros de distribución como las secciones en el orden en que aparecen en el texto y nos referiremos a ellos con dichos números a lo largo del mismo.

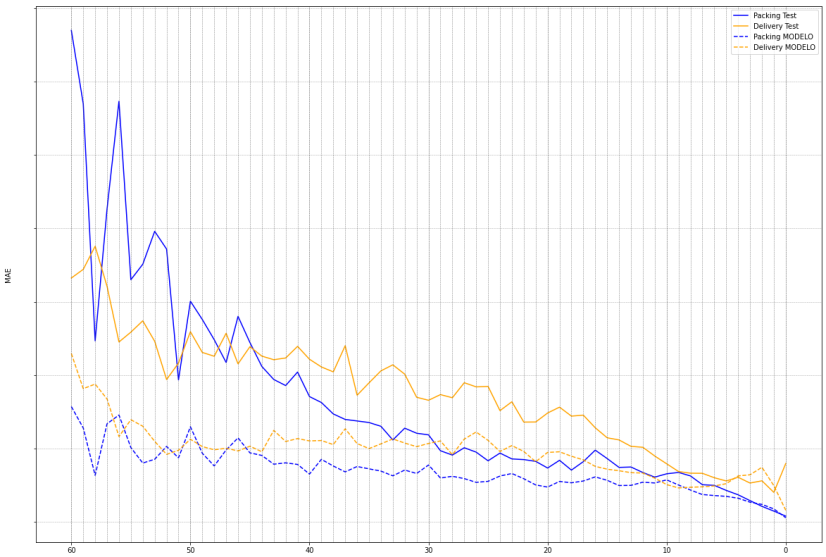

Figura 4.2: Gráficos del *MAE* para el centro de distribución 1 y la sección 1 obtenido con la `siguiente_fecha_pendiente_transporte` del algoritmo actual (linea continua) y con el modelo (linea discontinua), comparando este error a nivel de *packing* (línea azul) y de entrega (línea naranja). El eje de la X representa los días hasta la fecha prevista.

En la Figura 4.2 se puede observar que, aunque a nivel de *packing* el modelo sigue teniendo un error mayor que para la `siguiente_fecha_pendiente_transporte` cuando quedan menos de 3 días hasta la fecha total de entrega, en este caso la diferencia es mínima. Asimismo, este centro de distribución y sección tienen unos de los errores más bajos de todos. Esto puede deberse a que gran parte de los envíos son por camión, lo que los hace mucho más fáciles de predecir. Esto también explica por qué es la única figura de las presentadas con una disminución constante del error tanto a nivel de *packing* como a nivel de entrega.

En la Figura 4.3 podemos observar que el error del modelo a nivel de *packing* tiene un comportamiento bastante peor que en el resto de centros de distribución cuando la antelación supera los 40 días. Esto se debe a la escasez de envíos que generen *packing* con tanta antelación en este centro de distribución, lo que provoca que el error tenga tantas diferencias entre días consecutivos (debido a la falta de observaciones).

En esta misma figura también destaca la menor diferencia entre las predicciones a nivel de *packing* y a nivel de entrega tanto para el modelo como para la fecha que arrojan los sistemas actuales. Con-

secuentemente, este es el centro de distribución para el cual la mejora parece menos notable.

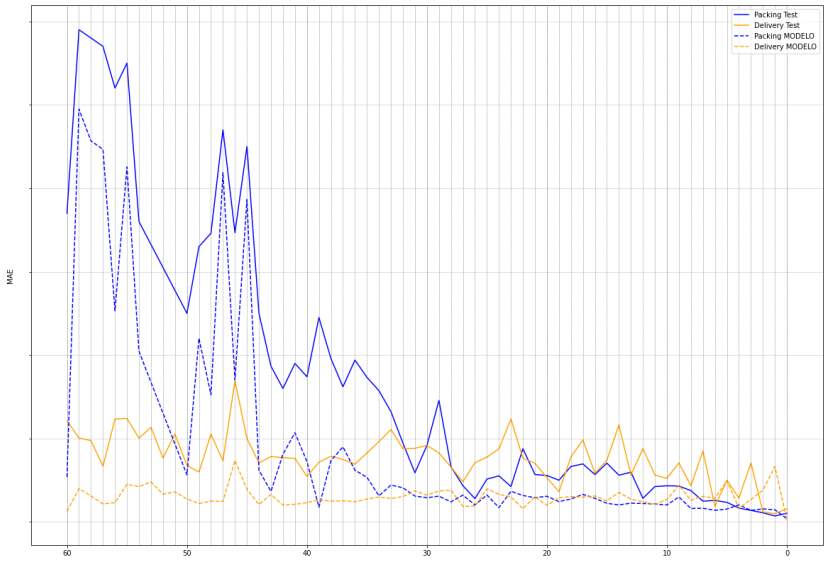

Figura 4.3: Gráficos del *MAE* para el centro de distribución 2 y la sección 2 obtenido con la `siguiente_fecha_pendiente_transporte` del algoritmo actual (linea continua) y con el modelo (linea discontinua), comparando este error a nivel de *packing* (línea azul) y de entrega (línea naranja). El eje de la X representa los días hasta la fecha prevista.

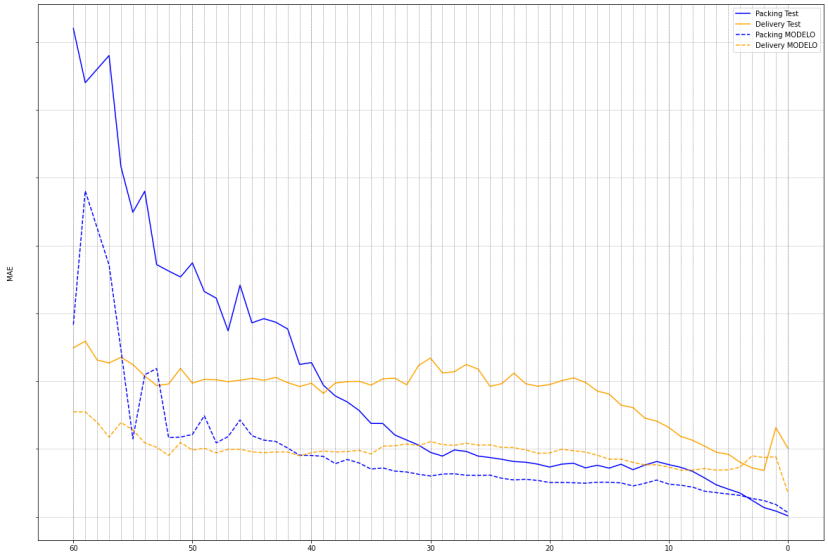

Figura 4.4: Gráficos del *MAE* para el centro de distribución 3 y la sección 2 obtenido con la `siguiente_fecha_pendiente_transporte` del algoritmo actual (linea continua) y con el modelo (linea discontinua), comparando este error a nivel de *packing* (línea azul) y de entrega (línea naranja). El eje de la X representa los días hasta la fecha prevista.

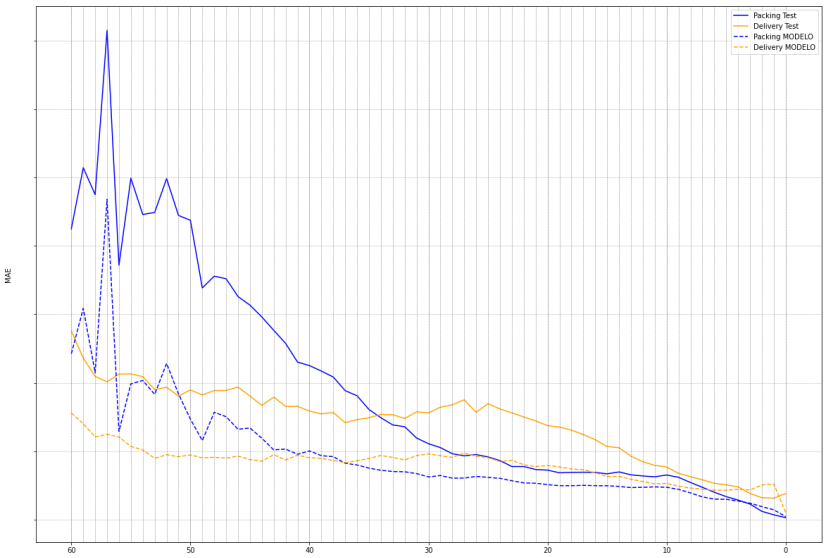

Figura 4.5: Gráficos del *MAE* para el centro de distribución 4 y la sección 1 obtenido con la `siguiente_fecha_pendiente_transporte` del algoritmo actual (linea continua) y con el modelo (linea discontinua), comparando este error a nivel de *packing* (línea azul) y de entrega (línea naranja). El eje de la X representa los días hasta la fecha prevista.

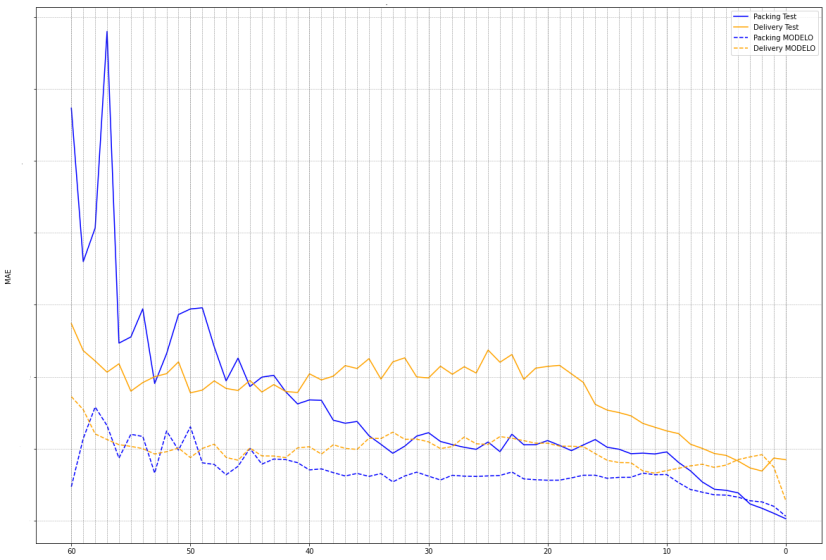

Figura 4.6: Gráficos del *MAE* para el centro de distribución 5 y la sección 3 obtenido con la `siguiente_fecha_pendiente_transporte` del algoritmo actual (linea continua) y con el modelo (linea discontinua), comparando este error a nivel de *packing* (línea azul) y de entrega (línea naranja). El eje de la X representa los días hasta la fecha prevista.

En las Figuras 4.4, 4.5 y 4.6 se pueden observar claras similitudes. En todas ellas se aprecia una disminución del error a nivel de *packing* cuando quedan menos de 10 días para la fecha de entrega, fenómeno observable tanto en las predicciones del modelo como en las

`siguiente_fecha_pendiente_transporte`. Además, cabe destacar que el error del modelo a nivel de *packing* permanece casi constante desde aproximadamente 30 días antes de la entrega de la mercancía.

A nivel de entrega, ocurre una situación similar. En las Figuras 4.4 y 4.6, el error de la `siguiente_fecha_pendiente_transporte` disminuye a partir de 15 días antes de la fecha de entrega, mientras que en la Figura 4.5, esta disminución empieza un poco antes, alrededor de 20 días antes de la entrega. Una vez llegado a ese punto, al igual que en los casos anteriores, el error experimenta una disminución lineal casi hasta el día de la entrega. No obstante, en el caso del error del modelo a nivel de entrega, este experimenta una evolución mucho más suave a medida que van pasando los días, aunque manteniéndose por debajo del error de la `siguiente_fecha_pendiente_transporte` hasta estar a menos de 3 días de la fecha de entrega.

En resumen, los resultados obtenidos demuestran que el modelo propuesto mejora significativamente la precisión de las predicciones a largo plazo, aunque para predicciones a corto plazo, las fechas actuales proporcionadas por el sistema son más fiables. Este análisis exhaustivo de las distintas gráficas por centro de distribución y sección refuerza la idea de que la implementación de este modelo puede optimizar la logística y planificación de envíos en distintas etapas del proceso.

En la siguiente sección, se profundizará en la interpretación de estos modelos utilizando los valores de Shapley, que proporcionarán una comprensión más detallada de las contribuciones individuales de las variables en las predicciones del modelo, ayudando así a identificar las áreas específicas de mejora y ajustar las estrategias de predicción de manera más efectiva.

4.1.2. Interpretabilidad

Para comprender mejor cómo el modelo realiza sus predicciones, se utilizan los valores de Shapley. Los valores de Shapley provienen de la teoría de juegos y permiten descomponer la predicción de cada instancia en contribuciones individuales de cada variable, proporcionando una visión clara de la importancia y el impacto de cada variable en la predicción final [19]. Esta técnica facilita la explicación de las decisiones del modelo al distribuir de manera justa el valor de la predicción entre las variables involucradas, basándose en su contribución marginal [13]. La interpretabilidad de los modelos de machine learning es crucial, especialmente en contextos donde las decisiones tienen un impacto significativo en la logística de una empresa. A través de los valores de Shapley, no solo se entiende mejor el comportamiento del modelo, sino que también se identifican áreas potenciales de mejora.

La gráfica mostrada en la Figura 4.7 es un gráfico de valores de SHAP (SHapley Additive exPlanations), que muestra la importancia de las características en el modelo sobre la variable respuesta `dias_hasta_fecha_total_entrado`. En esta figura, cada punto representa un valor SHAP para una instancia específica del conjunto de datos. En el eje Y se enumeran las características del modelo, algunas de las principales incluyen `dias_hasta_mejor_fecha_envio`, `id_tipo_transporte` e `dias_hasta_siguiente_fecha_pendiente_transporte`, entre otras. El eje X representa el impacto de las características en la salida del modelo, con valores SHAP positivos y negativos indicando el aumento o disminución de la predicción del modelo, respectivamente. La distribución horizontal de los puntos muestra cómo los valores de las características afectan a las predicciones del modelo: los puntos azules indican valores bajos de las características, mientras que los puntos rojos indican valores altos. A continuación, comentaremos las variables con más efecto sobres las predicciones.

La variable `dias_hasta_mejor_fecha_envio` tiene un impacto significativo en el modelo, con una amplia distribución de valores SHAP que van desde aproximadamente -10 hasta +15. De manera similar, la variable `dias_hasta_fecha_estimada_servicio_destino` también influye notablemente en las predicciones, aunque ligeramente menos. Asimismo, `id_tipo_transporte` y

`dias_hasta_fecha_handover` también muestran una influencia considerable.

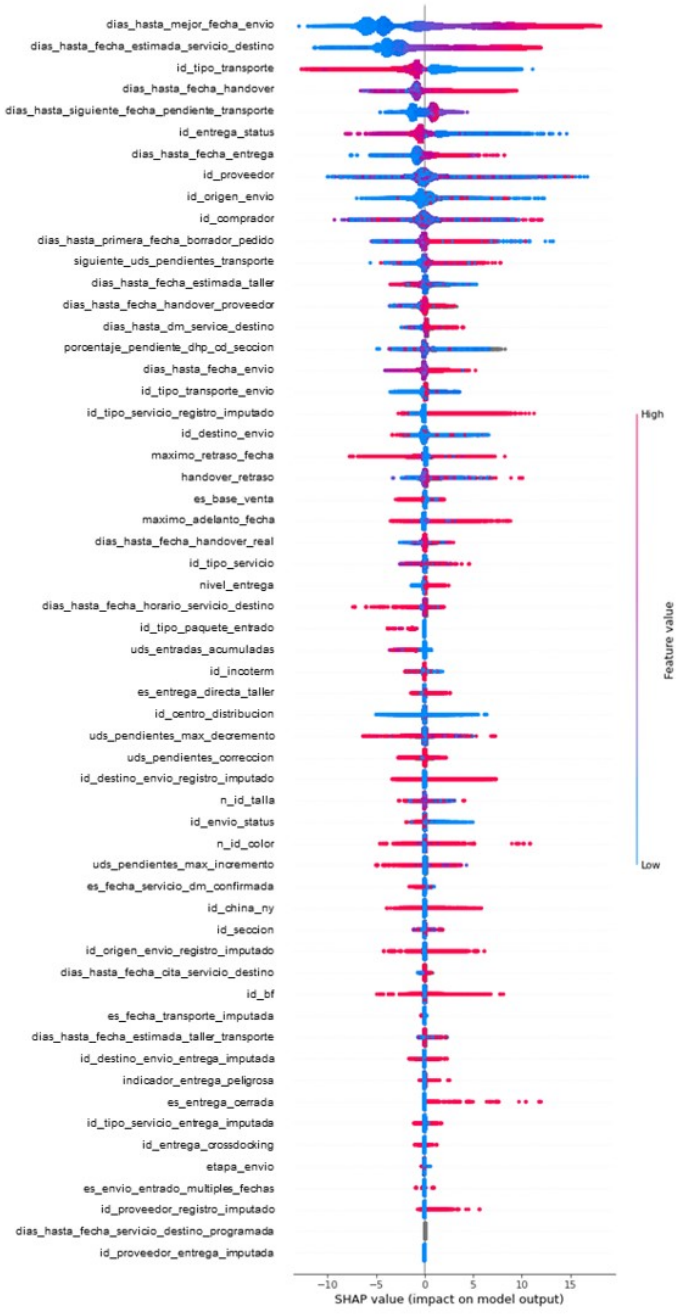

Figura 4.7: Gráfico de valores SHAP que muestra la importancia de las características en el modelo.

Para variables explicativas como `dias_hasta_mejor_fecha_envio`, los valores más altos tienden a aumentar la predicción del modelo, mientras que los valores bajos tienden a tener un impacto más negativo. Algunas variables, como `id_tipo_transporte`, muestran una clara distinción entre los valores

altos y bajos, resultando los valores altos en una disminución de la predicción, pues se corresponden con tipos de transporte con una duración menor del trayecto. Otras, como `id_entrega_status` e `dias_hasta_siguiente_fecha_pendiente_transporte`, muestran una amplia dispersión de valores SHAP para valores similares de esas variables, indicando una influencia variada en diferentes instancias.

En resumen, la Figura 4.7 proporciona una visión detallada de cómo cada característica afecta las predicciones del modelo. Las características con mayor dispersión en el eje X son las más importantes, ya que tienen el mayor impacto en las predicciones. Además, los colores de los puntos ayudan a entender cómo los valores específicos de las características influyen en el modelo. Seguidamente, comentaremos algunas gráficas que explican con más detalle el efecto de las variables más importantes sobre las predicciones.

La gráfica en la Figura 4.8 muestra el impacto de la característica `dias_hasta_siguiente_fecha_pendiente_transporte` en las predicciones del modelo. En el eje X se encuentra `dias_hasta_siguiente_fecha_pendiente_transporte`, mientras que el eje Y muestra los valores SHAP correspondientes. El color de los puntos indica el efecto de `id_tipo_transporte`, donde los puntos azules representan valores bajos y los puntos rojos valores altos.

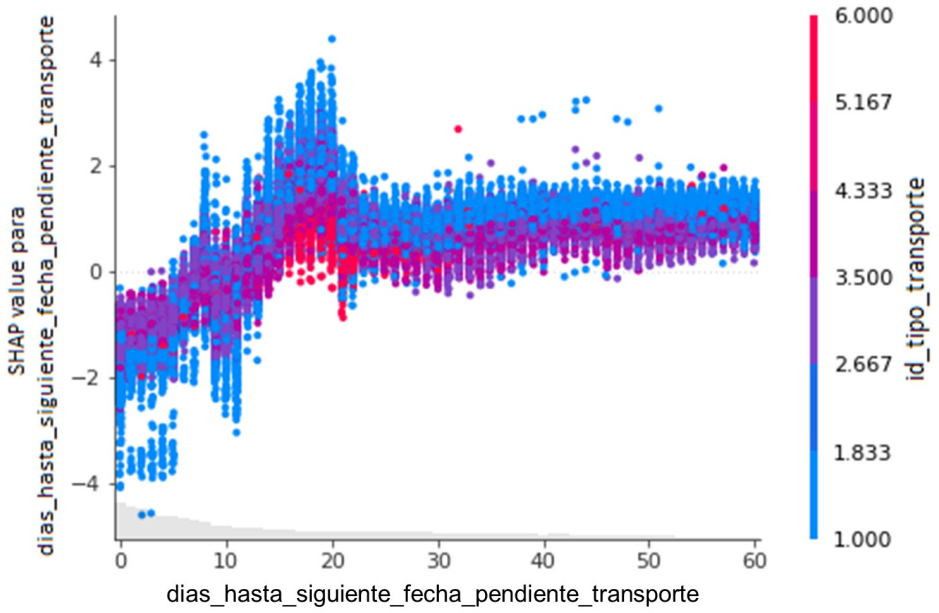

Figura 4.8: Gráfico de valores SHAP que muestra el impacto de la característica `dias_hasta_siguiente_fecha_pendiente_transporte` en las predicciones del modelo.

Observamos que los valores SHAP tienden a ser negativos para valores bajos de `dias_hasta_siguiente_fecha_pendiente_transporte` y se vuelven positivos a medida que `dias_hasta_siguiente_fecha_pendiente_transporte` aumenta. Esto sugiere que a medida que el número de días pendientes crece, su impacto en la predicción del modelo cambia de negativo a positivo. Esta relación, aunque es no lineal, también podría verse como lineal a trozos con dos partes muy diferenciadas en cuanto la variable `dias_hasta_siguiente_fecha_pendiente_transporte` toma valores entre 18 y 20.

El color de los puntos añade una dimensión adicional de interpretación. Los puntos rojos, que

representan valores altos de `id_tipo_transporte`, aparecen cuando queda menos hasta la entrega, en comparación con los puntos azules, y son los que menos se alejan del 0, lo cual parece deberse a que se trata de un tipo de transporte que no requiere de mucha antelación y al que, salvo imprevistos, tiene una predicción bastante fiable. Además, en esta figura se puede apreciar que el tipo de transporte tiene un impacto moderado pero consistente a lo largo de todos los valores de `dias_hasta_siguiente_fecha_pendiente_transporte`. Esto se puede ver en que los valores bajos de `id_tipo_transporte` devuelven una predicción mayor que los valores altos cuando la variable `dias_hasta_siguiente_fecha_pendiente_transporte` supera los 15 días y sucede al revés cuando `dias_hasta_siguiente_fecha_pendiente_transporte` es menor de 15 días.

En comparación con las siguientes figuras, esta muestra una relación más lineal y directa entre `dias_hasta_siguiente_fecha_pendiente_transporte` y los valores SHAP, contrastando con las relaciones no lineales observadas con `dias_hasta_fecha_envio` y `dias_hasta_fecha_handover`.

A continuación, se comentará el efecto entre la variable `dias_hasta_mejor_fecha_envio` y las predicciones del modelo.

La gráfica de la Figura 4.9 muestra el impacto de `dias_hasta_fecha_handover` en las predicciones del modelo. En el eje X se encuentra `dias_hasta_fecha_handover`, mientras que el eje Y muestra los valores SHAP correspondientes. El color de los puntos indica el efecto de `dias_hasta_mejor_fecha_envio`, donde los puntos azules representan valores bajos y los puntos rojos valores altos.

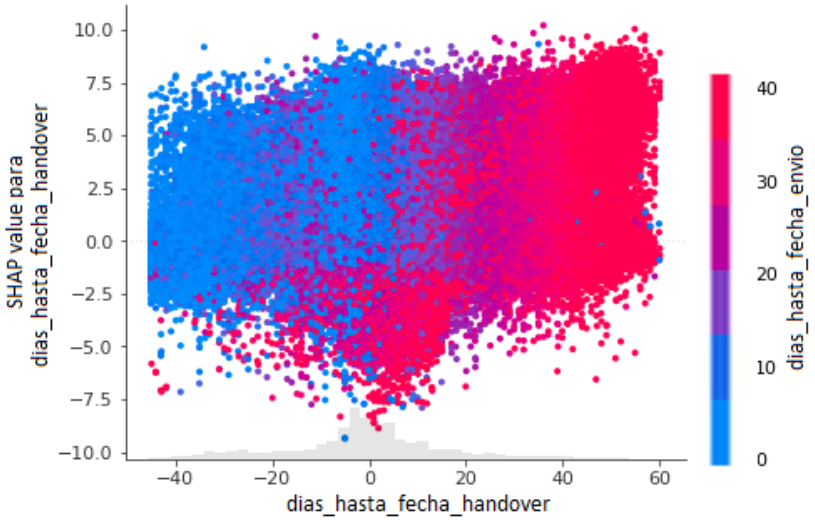

Figura 4.9: Gráfico de valores SHAP que muestra el impacto de la característica `dias_hasta_fecha_handover` en las predicciones del modelo.

En esta figura observamos una relación no lineal entre `dias_hasta_fecha_handover` y su impacto en el modelo. Los valores SHAP tienden a ser negativos para valores bajos y medios de `dias_hasta_fecha_handover`, mientras que para valores altos, los valores SHAP tienden a ser positivos. Esto sugiere que los días hasta el *handover* tienen un impacto significativo y complejo en la predicción del modelo, pues es razonable que los días hasta el *handover* influyan en los días hasta la entrega final. No obstante, se puede observar que la relación no es lineal, sino que dependiendo del valor de otras características como los días hasta la mejor fecha de envío, esta puede variar bastante.

En esta figura cabe destacar el efecto de la variable adicional. Aquí, el color de los puntos representa los valores de `dias_hasta_mejor_fecha_envio`. Los puntos rojos, que representan valores altos de `dias_hasta_mejor_fecha_envio`, tienden a concentrarse en la parte superior derecha de la gráfica, sugiriendo que cuando el número de días hasta la mejor fecha de envío es alto, y los días hasta la fecha de entrega también son altos, el impacto en la predicción es más significativo y al alza. Por otro lado, los puntos azules, que representan valores bajos de `dias_hasta_mejor_fecha_envio`, están más dispersos en la parte central e izquierda, indicando que con valores bajos de días hasta la mejor fecha de envío, el impacto de `dias_hasta_fecha_handover` es más moderado. Asimismo, cabe destacar que el momento en el cual se ve un mayor efecto de los `dias_hasta_mejor_fecha_envio` es cuando los `dias_hasta_fecha_handover` se acercan a 0. En dicho instante valores altos de `dias_hasta_mejor_fecha_envio` generan predicciones a la baja mientras que valores bajos apenas las modifican.

En resumen, esta gráfica de valores SHAP revela una relación compleja entre `dias_hasta_fecha_handover` y las predicciones del modelo, modulada por `dias_hasta_mejor_fecha_envio`. La comparación con la gráfica anterior muestra patrones similares de interacción compleja, aunque con diferentes características moduladoras, proporcionando una visión más profunda de cómo estas variables influyen en el resultado del modelo. Seguidamente se comentará la influencia de la variable `dias_hasta_fecha_envio` sobre la predicción del modelo.

La gráfica en la Figura 4.10 muestra el impacto de `dias_hasta_fecha_envio` en las predicciones del modelo. En el eje X se encuentra `dias_hasta_fecha_envio`, mientras que el eje Y muestra los valores SHAP correspondientes. El color de los puntos indica el efecto de `dias_hasta_siguiente_fecha_pendiente_transporte`, donde los puntos azules representan valores bajos y los puntos rojos valores altos.

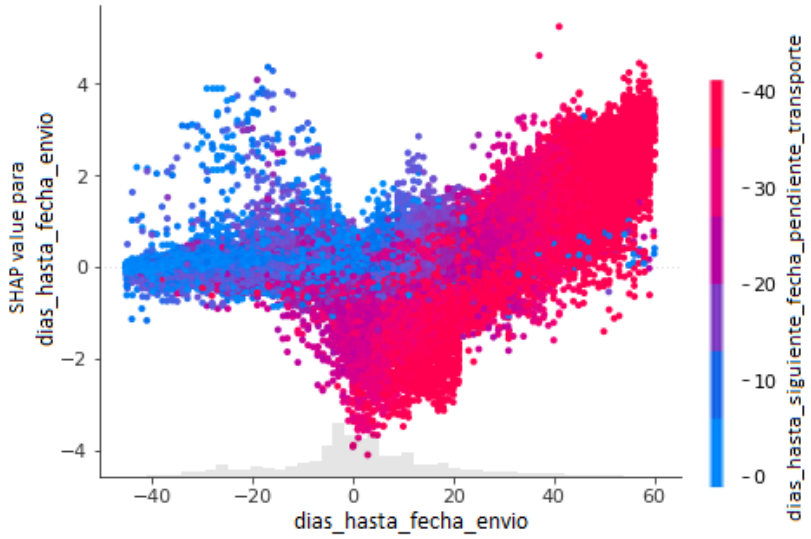

Figura 4.10: Gráfico de valores SHAP que muestra el impacto de la característica `dias_hasta_fecha_envio` en las predicciones del modelo.

La distribución de puntos sugiere una relación no lineal entre `dias_hasta_fecha_envio` y su impacto en el modelo. Los valores SHAP tienden a ser negativos para valores medios de `dias_hasta_fecha_envio`,

mientras que para valores altos y bajos, los valores SHAP tienden a ser positivos (aunque sobre todo en los altos). Esto indica que los días hasta la fecha de envío tienen un impacto significativo y complejo en la predicción del modelo, afectando tanto de manera positiva como negativa dependiendo del rango de la característica.

El color de los puntos añade otra dimensión de interpretación. Los puntos rojos, que representan valores altos de `dias_hasta_siguiente_fecha_pendiente_transporte`, tienden a concentrarse en la parte superior derecha de la gráfica, sugiriendo que cuando el número de días pendientes es alto, y los días hasta la fecha de envío también son altos, el impacto en la predicción es más significativo. Por otro lado, los puntos azules, que representan valores bajos de `dias_hasta_siguiente_fecha_pendiente_transporte`, están más dispersos en la parte central y izquierda, indicando que con valores bajos de días pendientes, el impacto de `dias_hasta_fecha_envio` es más moderado. Asimismo, al igual que sucedía en la Figura 4.9, cabe destacar que el momento en el cual se ve un mayor efecto de los `dias_hasta_best_pendiente` es cuando los `dias_hasta_fecha_envio` se acercan a 0. En dicho instante valores altos de `dias_hasta_siguiente_fecha_pendiente_transporte` generan predicciones a la baja mientras que valores bajos apenas las modifican.

En resumen, esta gráfica de valores SHAP revela una relación compleja entre `dias_hasta_fecha_envio` y las predicciones del modelo, modulada por `dias_hasta_siguiente_fecha_pendiente_transporte`. La comprensión de esta relación ayuda a interpretar cómo estas dos características interactúan para influir en el resultado del modelo, proporcionando información valiosa para mejorar la precisión y la interpretabilidad del mismo.

En la próxima sección, se explorará en detalle tanto las medidas de error como la utilización de los valores de Shapley para interpretar las predicciones del otro modelo desarrollado en este trabajo. Este análisis contribuirá a una comprensión integral de los factores que influyen en las predicciones, reforzando la robustez y la fiabilidad de las decisiones basadas en estos modelos.

4.2. Modelo para el Retraso de los Envíos

De manera similar a la Sección 4.1, en esta sección se presentan y analizan los resultados del modelo encargado de predecir el retraso en los envíos. Esta tiene como objetivo evaluar la efectividad del modelo en la predicción de retrasos y comparar su rendimiento con el método actual, proporcionando así una visión clara de sus ventajas y áreas de mejora. Además, analizaremos los valores de Shapley para dar una interpretación del modelo.

4.2.1. Resultados

Debido a que los patrones de los errores apenas tienen diferencias con los del modelo de la Sección 4.1, los detalles completos de las gráficas y el análisis del Error Absoluto Medio (MAE) en función del retraso en la entrega se encuentran en la Sección B.1 del apéndice. Es importante destacar que, a nivel global, se logró una mejora del 42 % en términos de MAE y un 46 % en términos de $RMSE$ con respecto al algoritmo que actualmente emplea la compañía. En las gráficas del apéndice, los datos se representan con los mismos códigos de colores que en la Sección 4.1 y lo mismo sucede para el tipo de trazo de las curvas (continuo o discontinuo). Para más detalles, consulte el apéndice.

En base a lo expuesto en la Sección B.1 del apéndice, el modelo para predecir el retraso de los envíos muestra una mejora significativa en comparación con el algoritmo actual, especialmente en predicciones a largo plazo. Las gráficas presentadas en el apéndice indican que el modelo es capaz de mantener una

precisión estable en sus predicciones a medida que se acerca la fecha de entrega, aunque con una ligera disminución en la precisión en los últimos días antes de la entrega.

El análisis detallado de las diferentes gráficas de error para los centros de distribución y secciones ha permitido identificar patrones consistentes y áreas de mejora. La estabilidad del modelo a largo plazo sugiere que es una herramienta valiosa para la planificación y gestión logística, permitiendo anticipar y mitigar posibles retrasos en los envíos.

La comparación con el modelo presentado en la Sección 4.1.1 resalta la importancia de ajustar y optimizar las predicciones a corto plazo, donde el algoritmo actual aún ofrece una precisión ligeramente superior. Sin embargo, el rendimiento global del modelo propuesto demuestra su potencial para mejorar significativamente la eficiencia operativa.

En la próxima subsección, se explorarán más a fondo los valores de Shapley para interpretar las predicciones de este modelo, proporcionando una comprensión más profunda de los factores que influyen en los retrasos y cómo pueden ser gestionados de manera más efectiva.

4.2.2. Interpretabilidad

Utilizando nuevamente los valores de Shapley, se realiza un análisis detallado de cómo el modelo hace sus predicciones de retraso en los envíos. La interpretabilidad es esencial para entender las decisiones del modelo y asegurar que las predicciones sean precisas y justas.

La Figura 4.11, al igual que la Figura 4.7, también muestra la importancia de las variables del modelo utilizando valores de Shapley. En esta gráfica, las variables más influyentes incluyen `dias_hasta_siguiente_fecha_pendiente_transporte`, `id_tipo_transporte`, y `dias_hasta_fecha_handover`.

Comparando ambas gráficas, se puede observar que la importancia relativa de las variables es diferente entre los dos modelos. En la Figura 4.7, `dias_hasta_siguiente_fecha_pendiente_transporte` es la variable más influyente, mientras que en la Figura 4.11, `dias_hasta_mejor_fecha_envio` tiene un mayor impacto. Esto sugiere que los dos modelos están influenciados por factores diferentes, pero que aportan una información similar.

La distribución de los valores de Shapley para las variables principales también muestra diferencias. En la Figura 4.7, la variabilidad de los valores de Shapley es más pronunciada para `dias_hasta_siguiente_fecha_pendiente_transporte`, mientras que en la Figura 4.11, esta variabilidad es más notable para `dias_hasta_mejor_fecha_envio`. Esto indica que las variables clave no solo son diferentes, sino que su impacto en los modelos también varía significativamente.

En resumen, las gráficas presentan diferencias importantes tanto en las variables más influyentes como en la distribución de sus valores de Shapley. Estas diferencias sugieren que los modelos analizados están influenciados por diferentes factores y capturan patrones distintos en los datos, ya que las variables respuesta son diferentes. Seguidamente, comentaremos algunas gráficas que explican con más detalle el efecto de las variables más importantes sobre las predicciones.

La Figura 4.12, presenta la relación entre `dias_hasta_siguiente_fecha_pendiente_transporte` y sus valores de Shapley, esta vez coloreada por la variable `dias_hasta_fecha_handover`. La elección de una variable secundaria diferente a la de la Figura 4.8 se debe a que esta muestra una mayor interacción con la variable principal. En esta gráfica, se puede observar que lo que realmente influye en el valor de Shapley no son los días hasta pendiente, sino la combinación de esta variable con los días hasta el handover. Esto puede deberse a que los días hasta el pendiente para envíos que vienen de diferentes orígenes pueden estar en situaciones muy diferentes. Es por ello que los días hasta el handover son tan

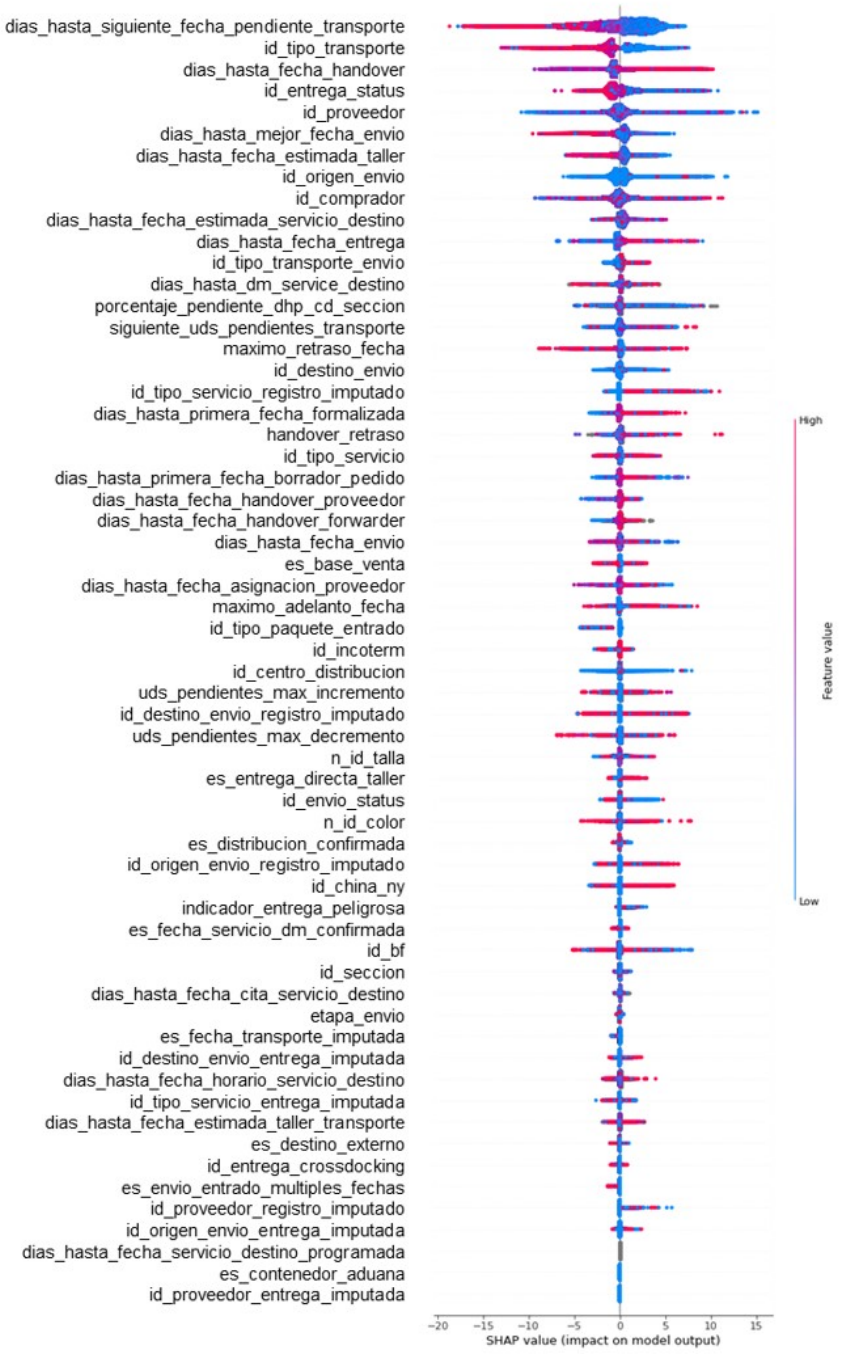

Figura 4.11: Gráfico de valores SHAP que muestra la importancia de las características en el modelo.

influyentes, pues ayudan a dar contexto al envío. Concretamente, cuando los valores del handover son elevados, el modelo da una predicción al alza, mientras que si el handover se llevó a cabo días atrás, la predicción tiende a ser a la baja.

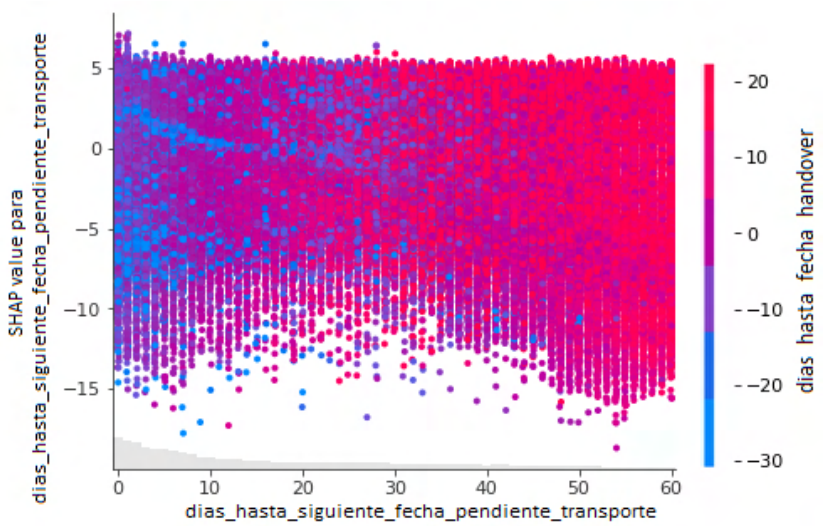

Figura 4.12: Gráfico de valores SHAP que muestra el impacto de la característica `dias_hasta_siguiente_fecha_pendiente_transporte` en las predicciones del modelo.

La Figura 4.13, muestra la relación entre la variable `dias_hasta_fecha_handover` y su valor de Shapley, influenciada por la variable `dias_hasta_mejor_fecha_envio`. Se observa que, al contrario que en la Figura 4.9, los valores de Shapley de la variable `dias_hasta_fecha_handover` no tienen una tendencia alcista tan clara, sino que más bien parecen mantener una distribución uniforme con una leve inclinación hacia valores positivos a medida que `dias_hasta_fecha_handover` se incrementa. No obstante, a medida que los días hasta el handover se aproximan a 0, sí parece haber un efecto relevante de la variable `dias_hasta_mejor_fecha_envio`, la cual genera valores de Shapley negativos cuando los días hasta la fecha de entrega son mayores y viceversa. Esto deja entrever que cuando el handover está cerca de producirse, el modelo prevé un retraso menor si los días hasta la entrega son mayores y prevé un retraso mayor en caso contrario.

En cuanto a la Figura 4.14, muestra la relación entre la variable `dias_hasta_fecha_envio` y su valor de Shapley, influenciado por la variable `dias_hasta_siguiente_fecha_pendiente_transporte`. Se observa una leve tendencia parabólica de `dias_hasta_fecha_envio` sobre los valores de Shapley, aunque no muy lejana de una distribución uniforme. Al comparar la Figura 4.14 con la Figura 4.10, se puede observar que la primera muestra una distribución más uniforme de los valores de Shapley a lo largo de `dias_hasta_fecha_envio`, mientras que la segunda muestra una tendencia clara al alza. Asimismo, la variable secundaria parece tener menos efecto en el caso de este último modelo.

En resumen, las gráficas muestran diferencias significativas en las tendencias observadas en los valores de Shapley para `dias_hasta_fecha_envio`, sugiriendo que las dos variables auxiliares impactan de manera distinta en el modelo.

En este capítulo, se han analizado en detalle las gráficas de error y los valores de Shapley de dos modelos estadísticos, identificando las variables más influyentes en cada uno y cómo estas afectan sus

Figura 4.13: Gráfico de valores SHAP que muestra el impacto de la característica `dias_hasta_fecha_handover` en las predicciones del modelo.

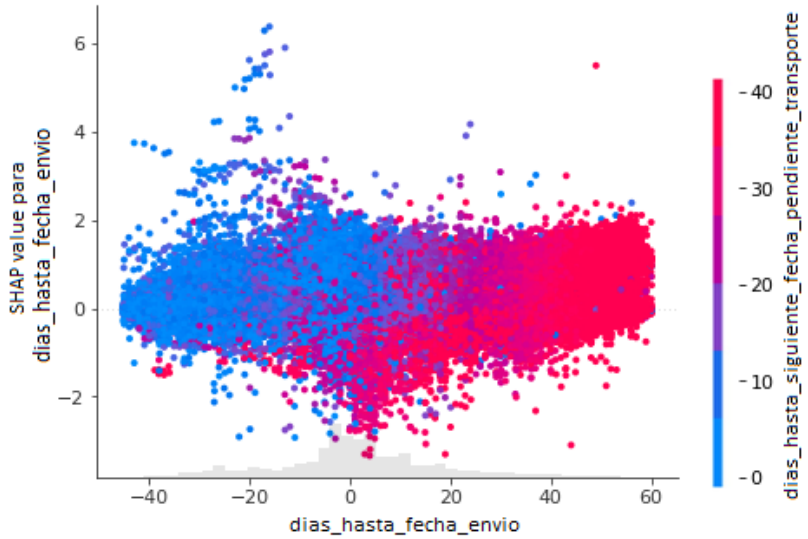

Figura 4.14: Gráfico de valores SHAP que muestra el impacto de la característica `dias_hasta_fecha_envio` en las predicciones del modelo.

predicciones. Las gráficas de valores de Shapley han permitido visualizar la importancia y el impacto de cada variable, proporcionando una comprensión más profunda de los mecanismos subyacentes de los modelos.

Las comparaciones entre las diferentes gráficas han revelado variaciones significativas en la respuesta de los modelos a ciertas variables. Estas diferencias subrayan la importancia de considerar múltiples perspectivas al interpretar los resultados. Asimismo, las gráficas de error han facilitado la evaluación del rendimiento, permitiendo identificar áreas de mejora y ajustar los modelos en consecuencia.

El análisis realizado ha proporcionado valiosas ideas sobre el comportamiento y el rendimiento de los modelos, validando sus resultados y ayudando a comprender mejor sus limitaciones. Este análisis es crucial para garantizar que los modelos sean robustos y fiables en aplicaciones prácticas.

Los valores SHAP son una herramienta útil para la interpretabilidad de los modelos de machine learning, permitiendo descomponer las predicciones en contribuciones individuales de las características. Esto proporciona una visión clara de la importancia de cada característica y facilita la identificación de patrones y relaciones complejas que pueden no ser evidentes en un análisis superficial. La comprensión profunda de estas interacciones es esencial para optimizar y ajustar los modelos predictivos, asegurando decisiones precisas y justificables.

La aplicabilidad de los valores de Shapley en este estudio ha demostrado ser valiosa, permitiendo un análisis detallado de cómo diferentes variables afectan las predicciones del modelo de días hasta la fecha de entrada de los envíos. Esta información facilita ajustes y mejoras en el modelo, enfocándose en las variables de mayor impacto y optimizando aquellas con relaciones complejas o no lineales.

En el siguiente capítulo, se presentarán las conclusiones del proyecto, resumiendo los hallazgos clave, discutiendo las implicaciones de los resultados y proponiendo posibles direcciones para investigaciones futuras.

Capítulo 5

Conclusiones

En este capítulo se resumen los hallazgos principales del Trabajo de Fin de Máster, centrado en el desarrollo de un sistema de alertas predictivo para la cadena de suministro. Este estudio ha subrayado la importancia de la anticipación y gestión proactiva de posibles anomalías en las entregas de pedidos, y cómo estas medidas pueden mejorar significativamente la eficiencia operativa en un entorno logístico complejo.

5.1. Metodología y Principales Resultados

Para lograr una mejora en el sistema de alertas, se llevó a cabo un análisis exhaustivo de los datos disponibles, utilizando técnicas avanzadas de preprocesamiento, análisis exploratorio y modelos predictivos. La metodología empleada incluyó:

- **Preprocesamiento de Datos**: Filtrado y limpieza de datos para eliminar inconsistencias y valores atípicos, creación de nuevas variables relevantes, y tratamiento de datos faltantes mediante técnicas de imputación.
- **Análisis Exploratorio**: Identificación de patrones y relaciones entre variables clave mediante visualizaciones y análisis estadísticos.
- **Modelado Predictivo**: Uso de modelos de machine learning, específicamente *LightGBM*, para predecir fechas de entrega y retrasos, optimizando los hiperparámetros y seleccionando las variables más importantes mediante algoritmos genéticos.

Los resultados obtenidos destacan varios puntos clave que demuestran la eficacia del sistema desarrollado:

1. **Precisión en la Predicción de Fechas de Entrega**: Los modelos lograron predecir con alta precisión las fechas de entrega de los pedidos, reduciendo considerablemente el error medio absoluto (*MAE*) y ayudando así en la toma de decisiones logísticas. La importancia de las variables en los modelos se evaluó utilizando valores *SHAP (Shapley Additive Explanations)*, identificando las características más influyentes en las predicciones.
2. **Identificación de Retrasos y Factores Críticos**: Los modelos utilizados permitieron identificar los principales factores que influyen en los retrasos de las entregas. Variables como el origen del envío, el destino, el tipo de transporte y el proveedor tienen una alta influencia en los tiempos de entrega.
3. **Evaluación de Modelos y Métricas de Error**: Se utilizó un enfoque riguroso para evaluar la precisión de los modelos, incluyendo validación cruzada y comparación con modelos base.

Los modelos desarrollados presentaron una mejora significativa en comparación con los métodos tradicionales, con una reducción notable en el error de predicción.

4. **Importancia de las Variables**: El análisis de los valores SHAP reveló que variables relacionadas con la logística y el transporte, como el tipo de transporte y los *INCOTERMs*, fueron las más influyentes en las predicciones. Esta información es vital para entender qué factores deben ser monitoreados más de cerca para prevenir retrasos.

5.2. Discusión de Resultados

Los hallazgos de este estudio proporcionan una comprensión profunda de las dinámicas de la cadena de suministro y demuestran el valor de los modelos predictivos en la gestión proactiva de entregas. La capacidad de anticipar retrasos y gestionar las entregas permite optimizar los recursos logísticos, mejorar la planificación de inventarios y en última instancia aumentar la satisfacción del cliente.

La importancia de las variables identificadas mediante los valores SHAP destaca la necesidad de un monitoreo constante de ciertos aspectos logísticos. Por ejemplo, la influencia significativa del tipo de transporte sugiere que ajustes en las estrategias de transporte pueden tener un impacto considerable en la reducción de retrasos.

5.3. Implicaciones Prácticas

La implementación del sistema de alertas desarrollado tiene varias implicaciones prácticas importantes:

- **Mejora en la Gestión de Inventarios**: Anticipar las fechas de entrega permite una mejor planificación y gestión de inventarios, reduciendo el riesgo de desabastecimiento o sobrestock.
- **Optimización de Recursos Logísticos**: Conocer con antelación los posibles retrasos permite optimizar el uso de recursos logísticos, mejorando la eficiencia y reduciendo costos.
- **Aumento de la Satisfacción del Cliente Interno**: La capacidad de cumplir con los plazos de entrega mejora la satisfacción del cliente interno, que puede planificar el resto de la operativa con más margen y seguridad, permitiendo además que centre sus esfuerzos en otras tareas en las que aumente su aporte de valor.

5.4. Limitaciones del Estudio

Una limitación del estudio es la complejidad y variabilidad inherente a la cadena de suministro global. A pesar de la precisión de los modelos, existen factores externos impredecibles, como condiciones meteorológicas extremas o cambios políticos, que pueden afectar las entregas y no siempre son capturados por los modelos predictivos.

Además, el modelo actual no considera otras posibles fuentes de variabilidad como cambios en las regulaciones aduaneras, interrupciones laborales, o desastres naturales que pueden impactar significativamente los tiempos de entrega. La variabilidad en la demanda del mercado, que puede causar cambios abruptos en los volúmenes de pedidos y afectar la capacidad de respuesta de la cadena de suministro, también ha sido omitida en el análisis actual.

Asimismo, sería beneficioso incorporar datos en tiempo real de seguimiento de envíos y sensores que proporcionen información actualizada sobre las condiciones de transporte y almacenamiento. Además,

la integración de datos meteorológicos, y de canales noticias podría ayudar a predecir tanto impactos de eventos climáticos como socioeconómicos y políticos.

5.5. Recomendaciones para Futuros Trabajos y Próximos Pasos

5.5.1. Recomendaciones para Futuros Trabajos

Para futuras investigaciones, se sugieren las siguientes líneas de trabajo:

- **Ampliación de la Base de Datos**: Incorporar más datos históricos y variables adicionales podría mejorar la precisión de los modelos predictivos.
- **Integración de Nuevas Técnicas de Machine Learning**: Explorar técnicas más avanzadas de machine learning y deep learning podría proporcionar mejoras adicionales en la capacidad predictiva del sistema.
- **Estudios Comparativos**: Realizar estudios comparativos con otras empresas del sector para validar y generalizar los hallazgos de este trabajo.

5.5.2. Próximos Pasos

El desarrollo de un sistema de alertas predictivo para la cadena de suministro ha demostrado ser una herramienta valiosa para mejorar la eficiencia operativa y la toma de decisiones estratégicas. Los resultados obtenidos no solo cumplen con los objetivos planteados, sino que también abren nuevas oportunidades para optimizar la gestión logística de la empresa.

Actualmente, estamos poniendo el modelo en producción, con el objetivo de proporcionar en tiempo real una predicción actualizada de la fecha de llegada de cada pedido a los diferentes centros de distribución. Esta implementación en tiempo real es crucial, ya que permite que otros algoritmos y procesos relacionados con la distribución a tienda se nutran de información más precisa y actualizada, mejorando así la eficiencia global de la cadena de suministro.

Uno de los clientes internos clave de este modelo es el departamento de logística y transporte. Este departamento está interesado en obtener, a partir del modelo creado o con algunas modificaciones, una predicción diaria de las cantidades que van a ingresar en cada uno de los centros de distribución. Esta capacidad predictiva permitirá llevar a cabo un dimensionado anticipado de los recursos, facilitando la contratación de personal y la asignación de recursos de manera más eficiente y ajustada a las necesidades reales.

Finalmente, se pretende extender el análisis realizado hacia un análisis de los proveedores. Este nuevo enfoque buscará identificar aquellos proveedores con peor desempeño, tanto en términos de retraso final como de retraso en el handover. Este análisis detallado permitirá implementar estrategias de mejora y renegociación de contratos con los proveedores que presenten un desempeño subóptimo, contribuyendo así a una cadena de suministro más robusta y eficiente.

Estos próximos pasos no solo buscan consolidar los avances obtenidos hasta ahora, sino también abrir nuevas vías de optimización y mejora continua en la gestión logística de la empresa, asegurando una respuesta más ágil y precisa a las demandas del mercado y a las necesidades internas de la organización.

Apéndice A

A.1. Variables empleadas en los modelos creados

Las variables empleadas inicialmente en ambos modelos son:

- dias_hasta_mejor_fecha_envio
- dias_hasta_fecha_estimada_servicio_destino
- id_tipo_transporte
- dias_hasta_fecha_handover
- dias_hasta_siguiente_fecha_pendiente_transporte
- id_entrega_status
- dias_hasta_fecha_entrega
- id_proveedor
- id_origen_envio
- id_comprador
- dias_hasta_primera_fecha_borrador_pedido
- siguiente_uds_pendientes_transporte
- dias_hasta_fecha_estimada_taller
- dias_hasta_fecha_handover_proveedor
- dias_hasta_fecha_servicio_destino_dm
- porcentaje_pendiente_dhp_cd_seccion
- dias_hasta_fecha_envio
- id_tipo_transporte_envio
- id_tipo_servicio_registro_imputado
- id_destino_envio
- maximo_retraso_fecha
- handover_retraso

- es_base_venta
- maximo_adelanto_fecha
- dias_hasta_fecha_handover_real
- id_tipo_servicio
- nivel_entrega
- dias_hasta_fecha_horario_servicio_destino
- id_tipo_paquete_entrado
- uds_entradas_acumuladas
- id_incoterm
- es_entrega_directa_taller
- id_centro_distribucion
- uds_pendientes_max_decremento
- uds_pendientes_correccion
- id_destino_envio_registro_imputado
- n_id_talla
- id_envio_status
- n_id_color
- uds_pendientes_max_incremento
- es_fecha_servicio_dm_confirmada
- id_china_ny
- id_seccion
- id_origen_envio_registro_imputado
- dias_hasta_fecha_cita_servicio_destino
- id_bf
- es_fecha_transporte_imputada
- dias_hasta_fecha_estimada_taller_transporte
- id_destino_envio_entrega_imputada
- indicador_entrega_peligrosa
- es_entrega_cerrada
- id_tipo_servicio_entrega_imputada
- id_entrega_crossdocking
- etapa_envio

- es_envio_entrado_multiples_fechas
- id_proveedor_registro_imputado
- dias_hasta_fecha_servicio_destino_programada
- id_proveedor_entrega_imputada
- correccion_fecha
- es_distribucion_confirmada
- es_contenedor_aduana
- es_destino_externo
- dias_hasta_fecha_handover_forwarder
- dias_hasta_fecha_asignacion_proveedor
- dias_hasta_primera_fecha_formalizada
- dias_hasta_fecha_servicio
- id_origen_envio_entrega_imputada
- porcentaje_mediano_entrado_cd_seccion

Finalmente, tras la selección de variables obtenemos que el modelo final que predice la variable dias_hasta_fecha_total_entrado emplea las variables:

- dias_hasta_mejor_fecha_envio
- dias_hasta_fecha_estimada_servicio_destino
- id_tipo_transporte
- dias_hasta_fecha_handover
- dias_hasta_siguiente_fecha_pendiente_transporte
- id_entrega_status
- dias_hasta_fecha_entrega
- id_proveedor
- id_origen_envio
- id_comprador
- dias_hasta_primera_fecha_borrador_pedido
- siguiente_uds_pendientes_transporte
- dias_hasta_fecha_estimada_taller
- dias_hasta_fecha_handover_proveedor
- dias_hasta_fecha_servicio_destino_dm
- porcentaje_pendiente_dhp_cd_seccion

- dias_hasta_fecha_envio
- id_tipo_transporte_envio
- id_tipo_servicio_registro_imputado
- id_destino_envio
- maximo_retraso_fecha
- handover_retraso
- es_base_venta
- maximo_adelanto_fecha
- dias_hasta_fecha_handover_real
- id_tipo_servicio
- nivel_entrega
- dias_hasta_fecha_horario_servicio_destino
- id_tipo_paquete_entrado
- uds_entradas_acumuladas
- id_incoterm
- es_entrega_directa_taller
- id_centro_distribucion
- uds_pendientes_max_decremento
- uds_pendientes_correccion
- id_destino_envio_registro_imputado
- n_id_talla
- id_envio_status
- n_id_color
- uds_pendientes_max_incremento
- es_fecha_servicio_dm_confirmada
- id_china_ny
- id_seccion
- id_origen_envio_registro_imputado
- dias_hasta_fecha_cita_servicio_destino
- id_bf
- es_fecha_transporte_imputada
- dias_hasta_fecha_estimada_taller_transporte

- id_destino_envio_entrega_imputada
- indicador_entrega_peligrosa
- es_entrega_cerrada
- id_tipo_servicio_entrega_imputada
- id_entrega_crossdocking
- etapa_envio
- es_envio_entrado_multiples_fechas
- id_proveedor_registro_imputado
- dias_hasta_fecha_servicio_destino_programada
- id_proveedor_entrega_imputada.

Mientras que el modelo que predice el `retraso` emplea las variables:

- dias_hasta_mejor_fecha_envio
- dias_hasta_fecha_estimada_servicio_destino
- id_tipo_transporte
- dias_hasta_fecha_handover
- dias_hasta_siguiente_fecha_pendiente_transporte
- id_entrega_status
- dias_hasta_fecha_entrega
- id_proveedor
- id_origen_envio
- id_comprador
- dias_hasta_primera_fecha_borrador_pedido
- siguiente_uds_pendientes_transporte
- dias_hasta_fecha_estimada_taller
- dias_hasta_fecha_handover_proveedor
- dias_hasta_fecha_servicio_destino_dm
- porcentaje_pendiente_dhp_cd_seccion
- id_tipo_transporte_envio
- id_tipo_servicio_registro_imputado
- id_destino_envio
- maximo_retraso_fecha
- handover_retraso

- es_base_venta
- maximo_adelanto_fecha
- dias_hasta_fecha_handover_real
- id_tipo_servicio
- nivel_entrega
- dias_hasta_fecha_horario_servicio_destino
- id_tipo_paquete_entrado
- id_incoterm
- es_entrega_directa_taller
- id_centro_distribucion
- uds_pendientes_max_decremento
- uds_pendientes_correccion
- id_destino_envio_registro_imputado
- n_id_talla
- id_envio_status
- n_id_color
- uds_pendientes_max_incremento
- es_fecha_servicio_dm_confirmada
- id_china_ny
- id_seccion
- id_origen_envio_registro_imputado
- dias_hasta_fecha_cita_servicio_destino
- id_bf
- es_fecha_transporte_imputada
- dias_hasta_fecha_estimada_taller_transporte
- id_destino_envio_entrega_imputada
- indicador_entrega_peligrosa
- es_entrega_cerrada
- id_tipo_servicio_entrega_imputada
- id_entrega_crossdocking
- es_envio_entrado_multiples_fechas
- es_distribucion_confirmada

- es_contenedor_aduana
- es_destino_externo
- dias_hasta_fecha_handover_forwarder
- dias_hasta_fecha_asignacion_proveedor
- dias_hasta_primera_fecha_formalizada
- dias_hasta_fecha_servicio
- id_origen_envio_entrega_imputada
- porcentaje_mediano_entrado_cd_seccion.

Apéndice B

B.1. Modelo para el Retraso de los Envíos

De manera similar a la Sección 4.1, aquí se presentan y analizan los resultados del modelo encargado de predecir el retraso en los envíos. Esta sección tiene como objetivo evaluar la efectividad del modelo en la predicción de retrasos y comparar su rendimiento con el método actual, proporcionando así una visión clara de sus ventajas y áreas de mejora.

B.1.1. Resultados

A continuación, se presenta una descripción general de las gráficas utilizadas para analizar el Error Absoluto Medio (*MAE*) en función del retraso en la entrega. Es importante destacar que, a nivel global, se logró una mejora del 42 % en términos de *MAE* y un 46 % en términos de *RMSE* con respecto al algoritmo que emplea actualmente la compañía. En las gráficas, los datos se representan con los mismos códigos de colores que en la Sección 4.1 y lo mismo sucede para el tipo de trazo de las curvas (continuo o discontinuo). Cabe destacar que, debido a que ya hemos comentado en detalle las gráficas de la Sección 4.1, en esta no entraremos en tanto detalle, sino que nos centraremos más en las diferencias entre las gráficas de error de los diferentes modelos.

En la Figura B.1, el eje Y representa el *MAE*, mientras que el eje X muestra los días hasta la fecha de entrega. Las conclusiones de esta gráfica son análogas a las de la Sección 4.1.1, donde el error del modelo mejoraba a la `siguiente_fecha_pendiente_transporte` hasta 3 días antes de la entrega, dando una estimación ligeramente peor a partir de dicho instante. Al igual que en la Sección 4.1.1, en las siguientes figuras, este comportamiento se mantendrá con mayor o menor intensidad a lo largo de diferentes centros de distribución, secciones y tipos de transporte, lo que sugiere que, de nuevo, es un comportamiento generalizado del modelo.

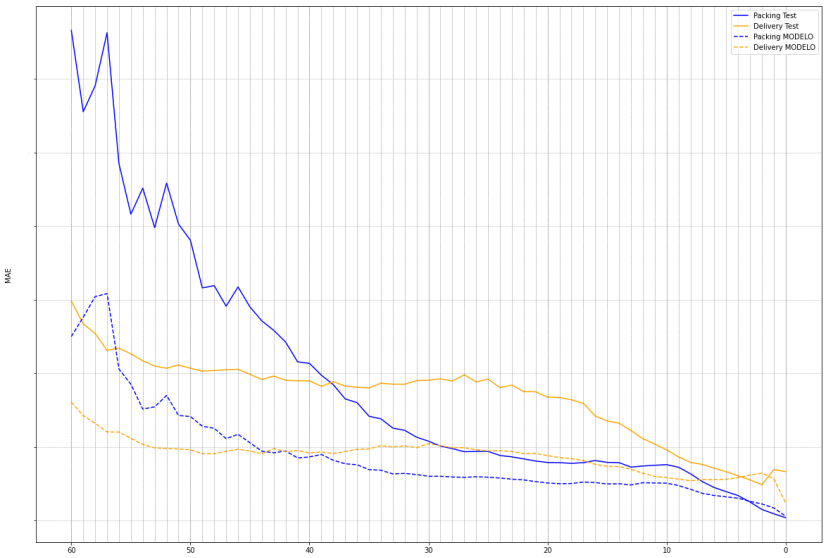

Figura B.1: Gráfico del *MAE* del modelo comparando este a nivel de *packing* y entrega con el obtenido por la `siguiente_fecha_pendiente_transporte`.

A continuación, se presentarán las diferentes gráficas de error para algunos de los centros de distribución, teniendo en cuenta las secciones asociadas. Por motivos de confidencialidad, enumeraremos tanto los centros de distribución como las secciones en el orden en que aparecen en el texto y nos referiremos a ellos con dichos números a lo largo del mismo.

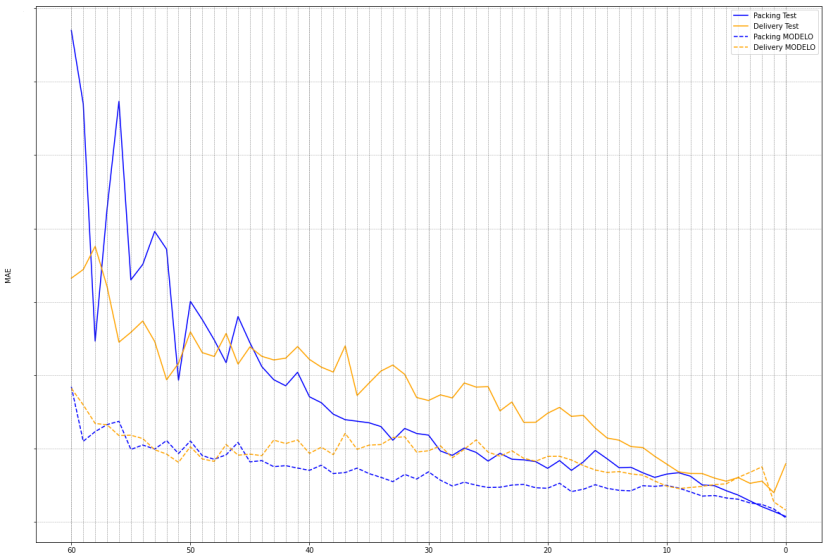

Figura B.2: Gráfico del *MAE* para el centro de distribución 1 y la sección 1. En él comparamos el error del modelo a nivel de *packing* y entrega con el obtenido por la `siguiente_fecha_pendiente_transporte`.

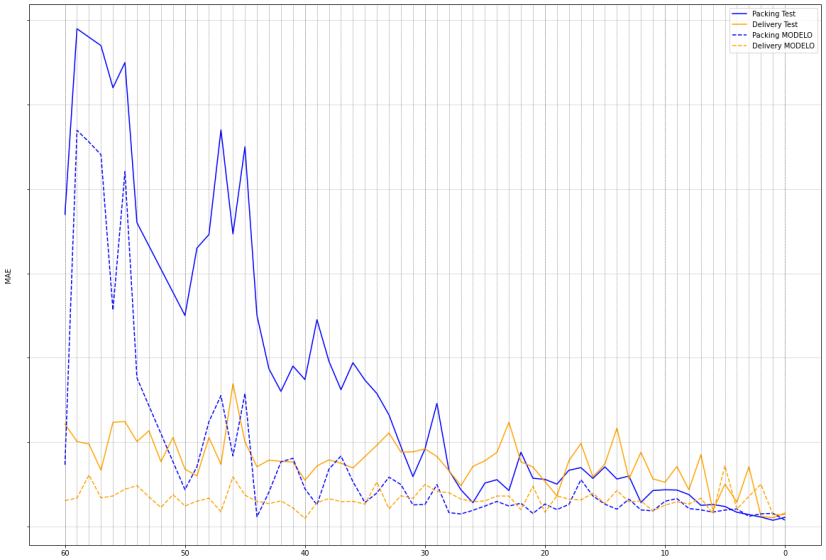

Figura B.3: Gráfico del *MAE* para el centro de distribución 2 y la sección 2. En él comparamos el error del modelo a nivel de *packing* y entrega con el obtenido por la `siguiente_fecha_pendiente_transporte`.

En la Figura B.2 se puede observar que, al igual que en la Figura 4.2, aunque a nivel de *packing* el modelo sigue teniendo un error mayor que la `siguiente_fecha_pendiente_transporte` cuando quedan menos de 3 días hasta la fecha total de entrega, en este caso la diferencia es mínima. No obstante, cabe destacar que tanto en los registros a nivel de *packing* como a nivel de entrega, este modelo genera unos errores ligeramente más estables que el modelo de la Sección 4.1 cuando los días hasta la entrega son cercanos a 60.

En cuanto a la Figura B.3, al igual que en la Figura 4.3, podemos observar que el error del modelo a nivel de *packing* tiene un comportamiento bastante peor que en el resto de centros de distribución cuando la antelación es elevada. No obstante, en el caso de la gráfica del error del modelo sobre el `retraso`, el empeoramiento es a partir de los 50 días, en lugar de los 40 días del modelo tratado en la Sección 4.1.

Finalmente, en las Figuras B.4, B.5 y B.6 se pueden observar claras similitudes entre sí. En todas se aprecia una disminución del error a nivel de *packing* cuando quedan menos de 10 días para la fecha de entrega total, fenómeno observable tanto en las predicciones del modelo como en la `siguiente_fecha_pendiente_transporte`. Además, cabe destacar que el error del modelo a nivel de *packing* permanece casi constante desde aproximadamente 30 días antes de la entrega de la mercancía.

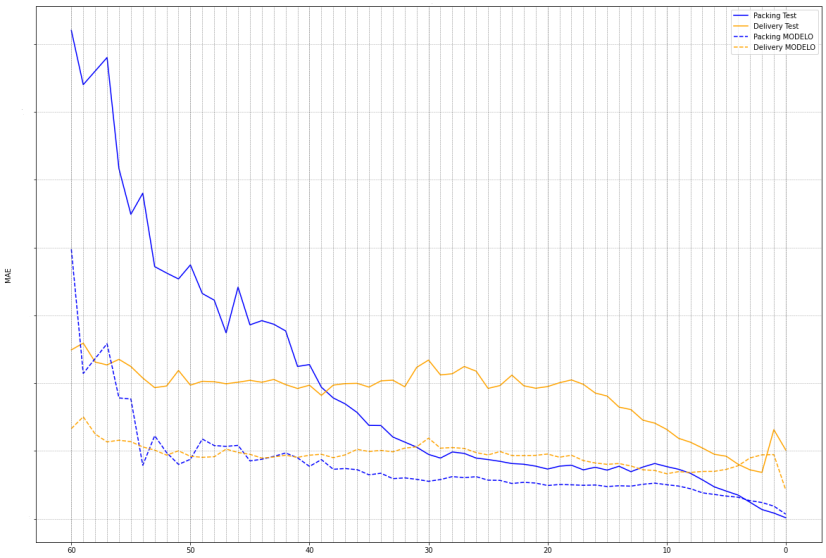

Figura B.4: Gráfico del *MAE* para el centro de distribución 3 y la sección 2. En él comparamos el error del modelo a nivel de *packing* y entrega con el obtenido por la `siguiente_fecha_pendiente_transporte`.

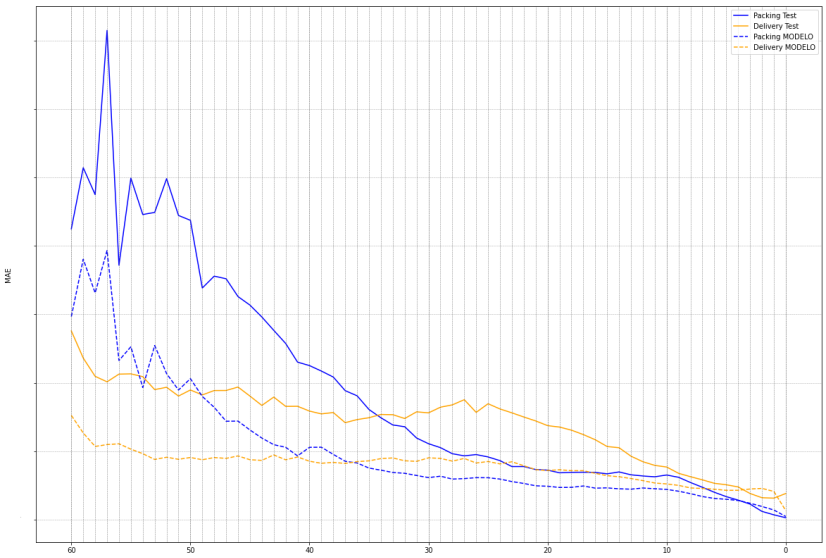

Figura B.5: Gráfico del *MAE* para el centro de distribución 4 y la sección 1. En él comparamos el error del modelo a nivel de *packing* y entrega con el obtenido por la `siguiente_fecha_pendiente_transporte`.

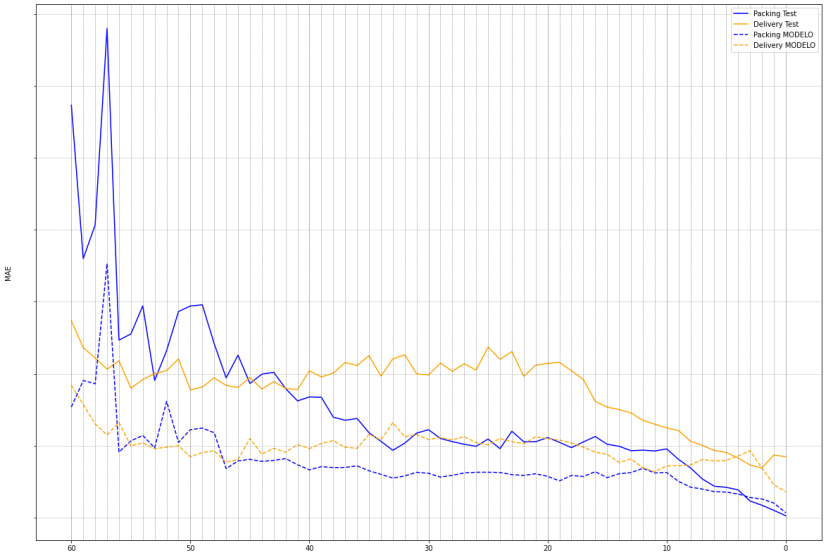

Figura B.6: Gráfico del *MAE* para el centro de distribución 5 y la sección 3. En él comparamos el error del modelo a nivel de *packing* y entrega con el obtenido por la `siguiente_fecha_pendiente_transporte`.

A nivel de entrega, ocurre una situación similar. En las Figuras B.4 y B.6, el error de la `siguiente_fecha_pendiente_transporte` disminuye a partir de 15 días antes de la fecha de entrega, mientras que en la Figura B.5, esta disminución empieza un poco antes, alrededor de 20 días antes de la entrega. Luego, al igual que en los casos anteriores, el error experimenta una disminución lineal casi hasta el día de la entrega. No obstante, en el caso del error del modelo a nivel de entrega, este experimenta una mejora mucho más suave a medida que van pasando los días.

En las Figuras B.4, B.5 y B.6, se observa una disminución del error tanto en las predicciones del modelo como en la `siguiente_fecha_pendiente_transporte`. Esta reducción se hace evidente cuando quedan menos de 10 días para la fecha de entrega, mostrando un comportamiento consistente a través de diferentes centros de distribución y secciones. El error del modelo a nivel de *packing* se mantiene casi constante desde aproximadamente 30 días antes de la entrega, lo que indica una estabilidad en las predicciones a largo plazo.

A nivel de entrega, se observa una tendencia similar. El error disminuye notablemente a partir de 15 días antes de la fecha de entrega en las Figuras B.4 y B.6, mientras que en la Figura B.5, esta disminución comienza alrededor de 20 días antes de la entrega. Esta reducción lineal del error hasta el día de la entrega sugiere que el modelo se adapta bien a las condiciones cambiantes a medida que se acerca la fecha de entrega. No obstante, el error del modelo a nivel de entrega mejora de manera más suave en comparación con las predicciones basadas en la `siguiente_fecha_pendiente_transporte`, aunque sigue siendo más preciso en general.

Bibliografía

[1] Adler, A. I., y Painsky, A. (2022). Feature Importance in Gradient Boosting Trees with Cross-Validation Feature Selection. *Entropy, 24*(5), 687.

[2] Bodenhofer, U. (2003). Genetic algorithms: theory and applications. *Johannes Kepler University en Linz.*

[3] Cover, T. M., y Thomas, J. A. (2006). *Elements of Information Theory (2nd ed.).* Wiley-Interscience.

[4] Fisher, W. D. (1958). On Grouping for Maximum Homogeneity. *Journal of the American Statistical Association, 53*(284), 789-798.

[5] Frazier, P. I. (2018). A tutorial on Bayesian optimization. arXiv.

[6] Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 1189-1232.

[7] Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley.

[8] Head, T., MechCoder, L., Louppe, G., Shcherbatyi, I., fcharras, Dror, et al. (2018). Scikit-Optimize [Software]. Disponible en https://github.com/scikit-optimize/scikit-optimize. Fecha de consulta: 15-02-2024.

[9] Hintze, J. L., Y Nelson, R. D. (1998). Violin plots: a box plot-density trace synergism. *The American Statistician, 52*(2), 181-184.

[10] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., y Liu, T. Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems, 30*, 3149-3157.

[11] Lambora, A., Gupta, K., y Chopra, K. (2019). Genetic algorithm-A literature review. *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon),* 380-384.

[12] Lanzi, P. L. (1997, April). Fast feature selection with genetic algorithms: a filter approach. En *Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC'97)*, 537-540. IEEE.

[13] Lundberg, S. M., y Lee, S. I. (2017). A Unified Approach to Interpreting Model Predictions. En *Advances in Neural Information Processing Systems*, 4765-4774.

[14] Microsoft Corporation. (2021). LightGBM (Version 3.2.1) [Software]. Disponible en https://github.com/microsoft/LightGBM. Fecha de consulta: 24-01-2024.

[15] Miller, B. L., y Goldberg, D. E. (1995). Genetic algorithms, tournament selection, and the effects of noise. *Complex systems, 9*(3), 193-212.

[16] Mitchell, M. (1998). *An Introduction to Genetic Algorithms*. MIT Press.

[17] Müller, A. C., y Guido, S. (2016). *Introduction to machine learning with Python: A guide for data scientists*. O'Reilly Media, Inc.

[18] Shaked, S., y Raviv, A. (2021). Dython: A collection of data science tools in Python [Software]. Disponible en https://github.com/shakedzy/dython. Fecha de consulta: 13-11-2023.

[19] Shapley, L. S. (1953). A Value for n-Person Games. En *Contributions to the Theory of Games*, 307-317. Princeton University Press.

[20] Snedecor, G. W., y Cochran, W. G. (1991). *Statistical Methods* (8th ed.) Wiley.

[21] Snoek, J., Larochelle, H., y Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. En *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012, NIPS 2012*, 2951-2959.

[22] Trendafilov, N., Kleinsteuber, M., y Zou, H. (2014). Sparse matrices in data analysis. *Computational Statistics, 29*(3-4), 403-405.

[23] van Buuren, S., y Groothuis-Oudshoorn, K. (2011). mice: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software, 45* (3), 1-67.

[24] Wilson, S. (2020). miceforest: Fast, Memory Efficient Imputation with LightGBM [Software]. Disponible en https://github.com/AnotherSamWilson/miceforest. Fecha de consulta: 01-12-2023