UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
ESCUELA DE POSTGRADO Y EDUCACIÓN CONTINUA
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL

PREDICCIÓN DE DEMANDA DE LARGO PLAZO PARA MEJORAR PLANIFICACIÓN LOGÍSTICA EN EMPRESA DEL SECTOR RETAIL

TESIS PARA OPTAR AL GRADO DE
MAGÍSTER EN CIENCIA DE DATOS

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL INDUSTRIAL

PATRICIO VICENTE ORTIZ VARGAS

PROFESOR GUÍA:
SEBASTIÁN RÍOS PÉREZ

MIEMBROS DE LA COMISIÓN:
RICHARD WEBER HAAS
JORGE SILVA SÁNCHEZ

SANTIAGO DE CHILE
2023

Resumen

Este trabajo consiste en la generación de un framework de prediccion de series de tiempo. En particular, se trabajará con una importante empresa del sector retail, que busca mejorar su planificación operacional. En este caso, se quiere reducir el costo en camiones contratados que realiza la firma de manera mensual. Para ello se contó con la información de 5 años de ventas de 128 sublíneas de productos, lo cual es equivalente a más de 200.000 registros.

El primer paso, fue identificar el proceso actual para realizar las predicciones y rediseñarlo introduciendo cambios en la metodología. Luego, se procesan los datos recopilados, de manera que se pudieran aplicar diferentes algoritmos de aprendizaje supervisado sobre ellos. Posteriormente se aplican técnicas de clustering parasegmentar las series de tiempo en 6 grupos diferentes, cada uno con un comportamiento diferente, de manera de entregar profundidad en el análisis.

La estimación de demanda fue realizada utilizando los algoritmos Naïve Forecast, Moving Average, ARIMA, SARIMA, SARIMAX, Holt-Winters, Random Forest, Light Gradient Boosting Machine, Extreme Gradient Boosting Machine, Redes Neuronales Artificiales y Redes Neuronales Recurrentes. Al entrenamiento de estos modelos se le agregó además, diferentes experimentos para corroborar la efectividad de los modelos al ser entrenados con distintos conjuntos de datos. Se concluye que el mejor modelo para realizar las predicciones enfocadas en el caso de uso es el modelo de Extreme Gradient Boosting Machine, que presenta una mejora de 12,4 % de WMAPE respecto al modelo ocupado actualmente. Dicha disminución en el error de las predicciones se traduce en una reducción de costes de entre 23 y 76 millones de pesos mensuales.

Nobody said it was easy.

Agradecimientos

Primero que nada, agradecer a mi familia. A mis padres por el apoyo, la preocupación, y el cariño que me han dado a lo largo de mi vida y que me han hecho llegar hasta este lugar. A mi hermano por la guía que ha simbolizado en mi vida, y el apoyo incondicional que me ha brindado todos estos años, tú iniciaste todo. A mis tías y primos que siempre me han acompañado en cada paso que doy.

A mis amigos, del colegio y de la Universidad, con los que he compartido experiencias, conversaciones y aprendizajes que me han formado como la persona que soy. Mención especial tanto para el Pablo, Panchito y la Cata, los amigos mas antiguos que tengo y con los que mas tiempo he compartido en mi vida, como para la Cami y Felipe, los mejores amigos que me trajo mi estadía en Beauchef, compañeros de mil batallas y mil maratones de estudio hasta las tantas, sin ustedes esto no sería posible.

Agradecimiento especial a los profesores que me han guiado hasta acá. Al profe cubano que transmitía su pasión por la física en las aulas del colegio. Al profe Richard que me ha apoyado en mi camino profesional, acercándome al Machine Learning y dándome un espaldarazo de confianza como su profesor auxiliar. Y especialmente al profe Seba que me apoyó en todo este proceso final de la tesis, siempre dispuesto a dar un consejo y enderezar mi trabajo cuando correspondía.

Un caluroso agradecimiento a mis compañeros de trabajo que me ayudaron a terminar este proceso de tesis, Alfredo, Diego, Aldo y Joaquín, nunca terminaré de agradecerles por su apoyo.

Y finalmente, un agradecimiento desde el fondo de mi corazón a Catalina, por aguantarme todo este año de estrés y llenarme de amor y cariño cuando más lo necesité.

Tabla de Contenido

1. Introducción **1**

1.1. Hipótesis 3

1.2. Objetivos 4

1.3. Resultados esperados 4

1.4. Metodología 4

1.5. Estructura de la tesis 7

2. Marco teórico **8**

2.1. Algoritmos a desarrollar 8

2.1.1. Algoritmos simples 8

2.1.2. Algoritmos estadísticos 9

2.1.3. Algoritmos basados en árboles 11

2.1.4. Métodos de aprendizaje profundo 14

2.2. Métricas de evaluación de desempeño 16

2.2.1. MAE 16

2.2.2. MAPE 16

2.2.3. WMAPE 16

3. Framework y datos **17**

3.1. Framework 17

3.1.1. Requerimientos y situación actual 17

3.1.2. Comprensión del negocio 18

3.1.3. Propuesta de framework 19
3.2. Datos 20
3.2.1. Limpieza de los datos 20
3.2.2. Caracterización general de los datos 25
3.2.3. Datos externos 31
3.3. Segmentación de series temporales 33

4. Resultados **36**
4.1. Procesamiento de Datos 38
4.1.1. Algoritmos simples 38
4.1.2. Algoritmos estadísticos 38
4.1.3. Algoritmos basados en árboles 39
4.1.4. Métodos de aprendizaje profundo 41
4.2. Predicciones 42
4.2.1. Predicción mensual 43
4.2.2. Predicción diario 50
4.3. Análisis de resultados 57
4.3.1. Análisis familia de modelos 57
4.3.2. Análisis experimentos 58
4.3.3. Comparación de experimentos 62
4.4. Evaluación económica 64

5. Conclusiones y trabajo futuro **68**
5.1. Efecto del clustering 68
5.2. Selección de modelo 69
5.3. Trabajo propuesto 71

Bibliografía **73**

Índice de Tablas

3.1. Ejemplo datos utilizados. 20

3.2. Descripción inicial datos. 21

3.3. Top 10 registros más grandes. 21

3.4. Descripción datos - postprocesamiento. 24

3.5. Descripción datos - pre-pandemia. 25

3.6. Descripción datos - pandemia. 27

3.7. Top 15 mix pre pandemia 29

3.8. Top 15 mix pandemia 29

3.9. Caracterización sublíneas. 29

3.10. Muestra de caracterización de sublíneas. 30

3.11. Muestra de campañas. Fuente: Equipo Comercial 32

3.12. Muestra de feriados. Fuente: Equipo Comercial 32

3.13. Muestra de caracterización para clusters. 34

3.14. Caracterización de clusters. 34

4.1. Muestra de datos con eventos, sublínea J3250, agregación mensual 38

4.2. Muestra de transformación con *lags*, sublínea J3250 39

4.3. Muestra de transformación con *lags*, sublínea J3250. 40

4.4. Muestra de caracterización de serie de tiempo, agregación mensual. 40

4.5. Muestra de datos con columna a predecir, sublínea J3250, agregación mensual. 41

4.6. Resultados de robustez agregación general, mensual. 43

4.7. Resultados de robustez cluster rotación normal, mensual. 44
4.8. Resultados de robustez cluster rotación variable, mensual. 45
4.9. Resultados de robustez cluster baja rotación, mensual. 46
4.10. Resultados de robustez cluster alta rotación, mensual. 47
4.11. Resultados de robustez cluster rotación itinerante, mensual. 48
4.12. Resultados de robustez cluster rotación potenciada, mensual. 49
4.13. Resultados de robustez agregación general, diario. 50
4.14. Resultados de robustez cluster rotación normal, diario. 51
4.15. Resultados de robustez cluster rotación variable, diario. 52
4.16. Resultados de robustez cluster baja rotación, diario. 53
4.17. Resultados de robustez cluster alta rotación, diario. 54
4.18. Resultados de robustez cluster rotación itinerante, diario. 55
4.19. Resultados de robustez cluster rotación potenciada, diario. 56
4.20. Resumen aplicación de experimentos. 58
4.21. Resumen efectos de aplicación de experimentos, mensual. 62
4.22. Resumen efectos de aplicación de experimentos, diario. 62
4.23. Máximo semanal para modelo baseline. 64
4.24. Estimación de unidades despachadas desde los CD, modelo baseline. 64
4.25. Estimación de camiones necesarios para despachos, modelo baseline. 65
4.26. Diferencia de camiones solicitados con demanda real, modelo baseline. 65
4.27. Sobrecoste total, modelo baseline. 65
4.28. Sobrecoste total, por escenario. 66

5.1. Resumen mejor modelo por nivel de agregación. 69

Índice de Ilustraciones

1.1. Fases de CRISP-DM. Fuente: Shearer[12] 5
1.2. Fases de CRISP-DM adaptado. Fuente: elaboración propia 6

2.1. Ejemplo de Árbol Binario con seis regiones separadas. 12
2.2. Ilustración del proceso de bagging. Fuente: Orellana A. [9] 12
2.3. Representación de una red neuronal con propagación hacia atrás 14
2.4. Representación de una red neuronal recurrente con propagación hacia atrás . 15

3.1. Ventas históricas, grupo 1. 22
3.2. Ventas históricas corregidas, grupo 1 22
3.3. Ventas históricas, grupo 2. 23
3.4. Distribución de mes de primera venta registrada. 24
3.5. Ventas agregadas 2017/01 - 2022/03 25
3.6. Ventas Pre Pandemia 26
3.7. Desagregación de ventas pre-pandemia. 26
3.8. Ventas pandemia. 27
3.9. Desagregación de ventas pandemia. 28
3.10. Efecto de eventos sobre ventas promedio. 33

4.1. Ejemplo de división de conjuntos de entrenamiento y prueba 36
4.2. Métricas por familia de algoritmos 58
4.3. Efecto de experimento de pandemia 59
4.4. Efecto de experimento de concentración 60

4.5. Efecto de experimento híbrido 61
4.6. Ventas pronosticadas por modelo baseline 64

Capítulo 1

Introducción

En los últimos años, los conceptos de *Data Science* y *Machine Learning* se han vuelto muy importantes para la toma de decisiones dentro de las empresas. Usando estas nuevas herramientas, las organizaciones son capaces tanto de optimizar su asignación de recursos, y con ello, de reducir los costes de su operación como de generar nuevos productos con los cuales aumentar el valor que le crean a sus clientes. Sin embargo, el potencial del *Machine Learning* dentro de las empresas aun tiene mucho espacio para desarrollarse y crear valor para la industria.

En dicho contexto, se desarrolla este trabajo de tesis dentro de una importante empresa del sector retail a nivel nacional, la cual cuenta con diversas líneas de negocios, incluyendo, supermercados, tiendas de ropa y accesorios, tiendas de ferretería, entre otros. En particular, se trabajará con la Gerencia de Supply Chain Corporativa de la firma. Esta gerencia, se encarga de la logística del holding, y por tanto, enfrenta una variedad de problemas que van desde la compra de productos a proveedores, hasta el despacho de productos en última milla, pasando por el almacenamiento de productos en stock y consolidación de pedidos, entre otros.

En ese entramado, la gerencia ha mostrado particular interés por incorporar herramientas de *Inteligencia de Negocios*, con las cuales puedan mejorar los rendimientos de la empresa. Para ese fin, se fundó la subgerencia de *Advanced Analytics*, que vela por incorporar estas nuevas tecnologías y generar productos de datos que puedan ser explotados por los distintos *stakeholders* de la operación, permitiendo obtener eficiencias logísticas que permitan disminuir los costes y/o mejorar la calidad de servicio al cliente.

Un problema que se ha identificado como crucial para la operación, es la planificación de *cotas de picking y shipping* en los Centros de Distribución (CD), es decir, definir una estrategia de contratación de personal y de empresas transportistas, para la consolidación de envíos y despacho de productos desde los CD hasta los *Delivery Points*, entre los cuales se consideran: *Centros de Transferencia Regionales* (CT) o despacho directo al cliente. En promedio, cada día se mueven más de 6500 productos a través de los CD, por lo que esta planificación tiene un impacto muy grande en la operación de la firma.

Para realizar una planificación eficiente en este apartado, así como en otros procesos dentro de la compañía, el *input* clave es tener un **pronóstico de demanda** acertado, es decir,

una proyección de las ventas que tenga un bajo error. En otras palabras, para poder realizar una asignación óptima de recursos, es necesario tener una estimación certera y confiable respecto a cuantos productos se venderán en un horizonte de tiempo destinado para la planificación. Existen dos casos claramente identificables al tener una estrategia de contratación sub-óptima:

- Contratar menos de lo necesario: Lo cual produce ineficiencias operativas, genera un cuello de botella en la cadena logística y afecta la promesa de envío al cliente. Además, se generan conflictos con las empresas contratistas, dificultando las relaciones contractuales con ellas.
- Contratar más de lo necesario: Lo cual produce un sobrecoste en la operación, y tiempos de ocio no aprovechables. Además, al ser un *input* usado en otras áreas, tiene un efecto adverso, no cuantificado, sobre otros procesos de la firma.

El contrato que se tiene actualmente con las empresas contratistas encargadas de los envíos, estipula que se reserva el número de camiones con una periodicidad semanal. Por lo que la empresa en su política de reserva de camiones, contrata el número de camiones que le permita satisfacer la demanda en el momento *peak* de la semana. En otras palabras, la empresa reserva los camiones considerando el máximo de ventas dentro de la semana a planificar. Adicionalmente, se debe tomar en cuenta que la empresa está optando por una estrategia que busca disminuir los tiempos de entrega, por lo que en caso de contratar menos camiones de lo necesario, se hace un pedido para agrandar la flota de camiones, pagando un sobrecoste del 50 % respecto al precio de la primera reserva.

Debido a lo anterior, se propone en este estudio, una reingeniería a la metodología para realizar la **predicción del número de unidades que se venderán en el futuro**. Para ello se presenta un estudio de los datos históricos disponibles, una evaluación de la metodología actual para generar predicciones, un estudio el desempeño de distintos modelos de aprendizaje supervisado para este caso de estudio, y finalmente, una propuesta de framework para realizar el pronóstico de demanda.

1.1. Hipótesis

La hipótesis principal que se desarrolla en este trabajo es:

"Se pueden introducir mejoras al framework actual de predicción de demanda de la empresa, usando modelos de aprendizaje supervisado, que aporten mejores resultados a la operación"

Lo cual tiene dos implicancias que han de ser abordadas en este trabajo:

1. Existe un planteamiento diferente al actual[1], que permite a la operación tomar decisiones acertadas, cumpliendo los requisitos que deben tener los entregables, sin comprometer rendimiento en las predicciones.
2. Existe un modelamiento del problema que permite mejorar las métricas de desempeño actuales.

Cabe especificar de lo anterior, que el framework de predicciones debe abordar los siguientes puntos:

- Fuentes de información : Espacio de almacenaje donde se recopilan los datos que se aplican al problema.
- Horizonte de predicción : Espacio temporal para el cual se desea generar las predicciones (diario/semanal/mensual)
- Granularidad : Nivel de agregación con el que se realizan las predicciones.
- Algoritmo de predicción : Modelo matemático con el cual generar predicciones.
- Evaluación de resultados : Métrica utilizada para medir la calidad de los resultados entregados.

[1]Se considera un cambio tanto en la metodología para hacer predicciones como en los algoritmos empleados para este fin.

1.2. Objetivos

A continuación se describe el objetivo general de este trabajo de tesis y sus respectivos objetivos específicos.

Objetivo general

Desarrollar un framework para realizar el pronóstico de demanda de la compañia que mejore las métricas de performance actuales y disminuya el costo de la operación en al menos 10 millones de pesos mensuales.

Objetivos específicos

1. Comprender el framework de trabajo actual y levantar los requerimientos del framework futuro.
2. Diseñar, experimentar y evaluar los modelos de pronóstico en la modelación de la demanda de productos.

1.3. Resultados esperados

Los resultados esperados de este trabajo son los siguientes:

1. Diagnosticar el framework de trabajo actual.
2. Estudiar de modelos presentados en la literatura y la comunidad de Machine Learning para la predicción en series de tiempo.
3. Calibrar diferentes modelos matemáticos para realizar las predicciones propuestas.
4. Evaluación económica del framework de trabajo propuesto.

1.4. Metodología

La metodología utilizada en esta investigación es una variante de ***CR**oss **I**ndustry **S**tandard **P**rocess for **D**ata **M**ining* (CRISP-DM por sus siglas). Esta metodología organiza los proyectos de minería de datos en seis fases: entendimiento del negocio, entendimiento de la data, preparación de la data, modelamiento, evaluación e implementación [12].

La figura 1.1 muestra las distintas fases señaladas, detallando las iteraciones específicas que se propone realizar en la medida que cada fase es completada.

En la fase inicial, **entendimiento del negocio**, se determinan los objetivos de la investigación, se realiza un levantamiento de la situación inicial (supuestos, requerimientos y

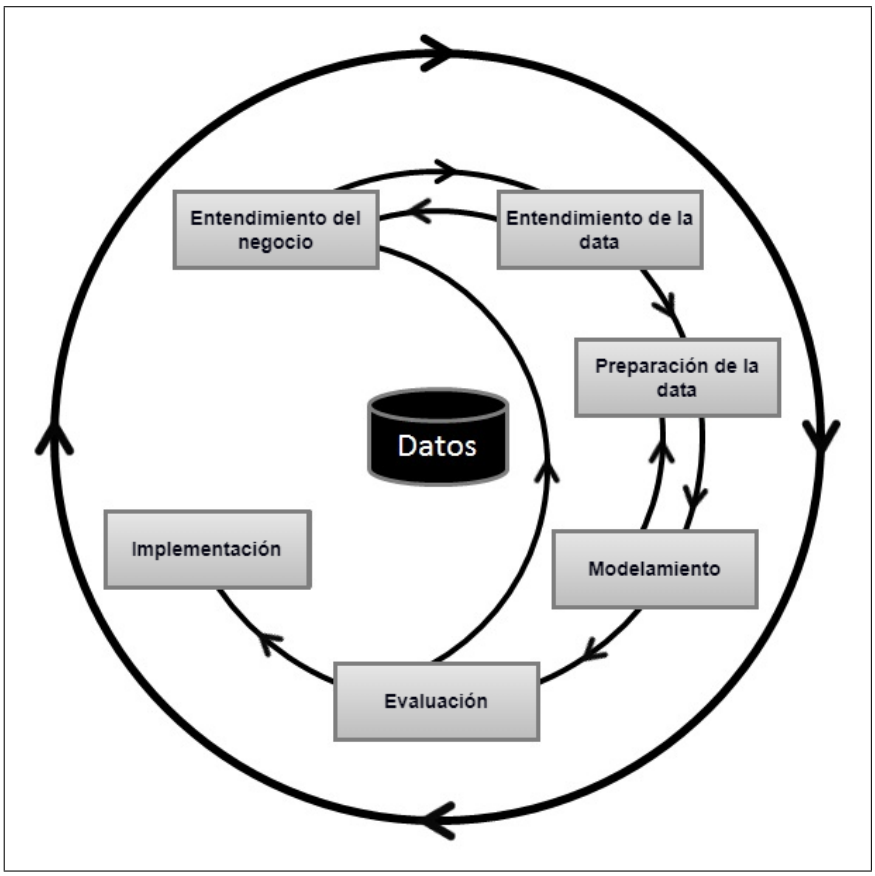

Figura 1.1: Fases de CRISP-DM. Fuente: Shearer[12]

restricciones), se replantean los objetivos con la perspectiva de un estudio de ciencia de datos y se genera un plan de trabajo que permita satisfacer los puntos propuestos inicialmente.

Luego, en la fase de **entendimiento de la data** se busca recopilar las fuentes de datos a utilizar, se realizan análisis descriptivos de los mismos y se verifica la calidad de estos, es decir, se analiza la granularidad de los datos, la presencia de anomalías y limitaciones que presentes los datos recopilados. En el contexto de las iteraciones mostradas en la figura 1.1, desde esta fase se pueden replantear objetivos y restricciones definidos en la fase de **entendimiento del negocio** producto de las condiciones de los datos a trabajar, por lo que un buen estudio de esta etapa es clave para poder re configurar el estudio propuesto.

Posteriormente, en la fase de **preparación de la data** se busca seleccionar los datos con los que se trabajará y realizar las respectivas limpiezas que esta necesita. En este punto, cabe destacar que al utilizar distintos modelos matemáticos se llevaran a cabo distintos procesamientos de los datos, construyendo nuevas variables y cruzando con fuentes de datos externas, según lo permitan los modelos.

La fase siguiente corresponde al **modelamiento**, que es donde se evalúan los distintos modelos matemáticos que sean capaces de resolver el problema con las restricciones planteadas, se configuran parámetros globales y se desarrolla el modelo. Nuevamente en el contexto de las iteraciones planteado, en esta fase se debe reevaluar la fase de **preparación de la data**, pues puede ser necesario la creación de nuevas variables o la parametrización con formato

diferente de variables ya existentes.

En la fase de **evaluación** se realiza una revisión del proceso general para determinar si el modelo propuesto cumple con las condiciones y objetivos definidos al inicio, así como también definir los siguientes pasos a desarrollar. Aquí la iteración se realiza con la primera fase (figura 1.1), precisamente por el contraste de los resultados obtenidos con los esperados que se señaló anteriormente.

Finalmente, en la fase de **implementación** se define un plan de puesta en marcha de la solución propuesta, con el respectivo plan de monitoreo y mantenimiento que este tipo de soluciones requiere.

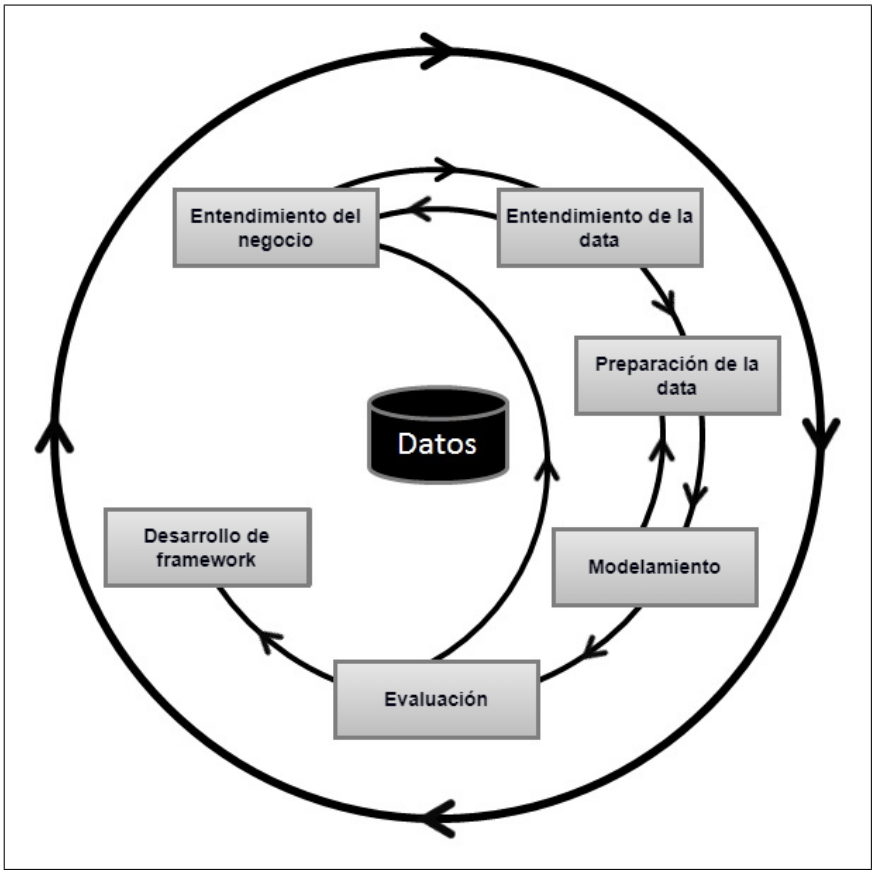

Figura 1.2: Fases de CRISP-DM adaptado. Fuente: elaboración propia

La figura 1.2 detalla la misma metodología pero adaptada a la problemática a abordar, en particular, se reemplaza la etapa de **implementación** por la etapa de **desarrollo de framework** en la cual se consolidan los aprendizajes obtenidos y se genera la propuesta que consolide los pasos previos y proponga una metodología para realizar las predicciones de manera sistemática.

1.5. Estructura de la tesis

La estructura de este trabajo de investigación, se estructura como se detalla a continuación.

El capítulo 2 detalla el marco teórico en que está contenido esta investigación, se muestra la bibliografía asociada a distintas soluciones propuestas a problemáticas similares a la planteada, tanto a nivel mundial como específicas para el caso chileno, además de estipular los sustentos matemáticos del trabajo posterior.

El capítulo 3 presenta los requisitos que debe satisfacer la predicción propuesta, realiza el análisis descriptivo de los datos, se señala las distintas variables disponibles para realizar este estudio y profundiza en cómo se comportan estas variables entre distintos grupos existentes en la muestra.

El capítulo 4 especifica todo lo relacionado al desarrollo del algoritmo que da solución a la problemática planteada. Muestra especial énfasis en el funcionamiento del mismo y en los modelos utilizados para dicho propósito. También se muestra la evaluación ecónomica de los modelos propuestos.

Finalmente, el capítulo 5 muestra los principales resultados y conclusiones obtenidos de los experimentos de datos y da paso a la discusión que genera el estudio desarrollado.

Capítulo 2

Marco teórico

A continuación se describen los aspectos relevantes del estado del arte en los que se enmarca este trabajo. El capítulo se divide en 2 partes, primero se muestran los algoritmos utilizados para realizar la predicción. En segundo lugar, se exploran las métricas de desempeño a aplicar.

2.1. Algoritmos a desarrollar

En esta sección son descritos los métodos que son utilizados para la estimación de la demanda. Estas se pueden englobar en cuatro categorias principales:

- Algoritmos simples.
- Algoritmos estadísticos
- Algoritmos basados en árboles.
- Algoritmos de aprendizaje profundo.

2.1.1. Algoritmos simples

Naïve forecast

Este es el método más básico que se utiliza para predecir. La premisa de este método es que el punto esperado es igual al último punto observado:

$$\hat{y}_{t+1} = y_t \tag{2.1}$$

También se puede asumir que los k puntos esperados son iguales a los k puntos anteriores.

Aunque este método luzca simple o ingenuo es útil para crear un punto de partida en el análisis. Numerosos estudios de predicción lo utilizan cuando los datos no poseen una considerable diferencia entre los días, y algunos demuestran que Naïve Forecasting es mejor que otros métodos como Moving Average o Trend (cuando no se ve mucha variación en los datos [11]).

Moving average

Dada una secuencia $\{a_i\}_{i=1}^{N}$ una n-media móvil o n-moving average se define como una nueva secuencia $\{s_i\}_{i=1}^{N-n+1}$ la cual proviene de la media aritmética de n elementos de la secuencia a_i. Moving average es una técnica para tener una idea de la tendencia del set de datos. Esta metodología es extremadamente útil para predecir tendencias a lo largo del tiempo y además sirve para tener un primer acercamiento con los datos.

2.1.2. Algoritmos estadísticos

Holt-Winters

El método de suavización exponencial de Holt-Winters se utiliza para pronosticar series de tiempo, utilizando para pronosticar los valores de tendencia, temporalidad y estacionalidad. De esta manera, esta metodología utiliza tres ecuaciones para *suavizar*; una para la *atenuación* de la serie de tiempo, otra para la tendencia y una última para la estacionalidad. La ecuación de atenuación de la serie de tiempo o de pronóstico para el período t se calcula de la siguiente forma:

$$A_t = \alpha \left(\frac{y_t}{R_{t-L}} \right) + (1 - \alpha) \left(A_{t-1} + T_{t-1} \right) \tag{2.2}$$

Donde α corresponde a la constante de atenuación la cual toma valores en el intervalo $0 < \alpha < 1$, T_{t1} corresponde a la tendencia del período $t1$ y R_{tL} a la estacionalidad del período tL. L se considera como el largo del ciclo de estacionalidad. La tendencia del período t se modela como sigue:

$$T_t = \beta \left(A_t + A_{t-1} \right) + (1 - \beta) T_{t-1} \tag{2.3}$$

Donde β es el coeficiente de tendencia el cual toma valores entre el intervalo $0 < \beta < 1$. La estacionalidad del período t se formula a continuación:

$$R_t = \gamma \left(\frac{y_t}{A_t} \right) + (1 - \gamma) R_{t-L} \tag{2.4}$$

Donde el parámetro γ se refiere al coeficiente de estacionalidad el cual, al igual que los

coeficientes anteriores, se encuentra en el intervalo $0 < \gamma < 1$. Finalmente, la predicción para k períodos en el futuro dado el período t es:

$$\hat{y}_{t+k} = (A_t + k * T_t)\, R_{t-L+k} \tag{2.5}$$

Se espera que con este modelo se pueda detectar tanto la tendencia como la estacionalidad y la temporalidad de los datos.

ARIMA

El modelo autoregresivo integrado de promedio móvil o ARIMA es un modelo que estudia predicciones de series de tiempo, donde estas pueden considerarse como la realización de un proceso estocástico que se observa secuencialmente a lo largo del tiempo. El modelo ARIMA es un caso particular del modelo ARMA en el cual sí existe una raíz unitaria. El modelo ARMA es a su vez una combinación del proceso autorregresivo AR(p) y el proceso de media móvil MA(q). Ambos procesos son procesos de series de tiempo que intentan explicar la demanda a partir de datos pasados. La diferencia es que el primero tiene memoria a largo plazo por lo que le cuesta reaccionar rápidamente ante "shocks" o perturbaciones y el segundo, tiene corta memoria, reaccionando ágilmente a perturbaciones, pero "olvidando" la información del pasado [3].

Este es un modelo utilizado en estadísticas, econometría e ingeniería por varias razones[6]:

1. Es considerado como uno de los modelos con mejor desempeño en términos de pronóstico.
2. Se utilizan como referencia para modelos más sofisticados.
3. Es de fácil implementación y alta flexibilidad dada su estructura multiplicativa.

Los parámetros de un modelo $ARIMA(p, d, q)$ se definen como sigue:

- p es el número de términos autorregresivos.
- d es el número de diferencias que se aplican a la serie de tiempo para que sea estacionaria.
- q es el número de medias móviles o moving average que realiza el proceso.

De esta forma, se construye un modelo de regresión lineal que incluye el número y el tipo de términos especificados, de tal manera que la serie de tiempo sea estacionaria. Es necesario que la serie de tiempo sea estacionaria para eliminar tendencias y estructuras estacionales que pueden afectar negativamente el modelo de regresión. Finalmente, el modelo de regresión lineal que se busca tiene la siguiente forma:

$$\hat{y}_t = \delta + \phi_1 * y_{t-1} + ... + \phi_p * y_{t-p} - \theta_1 * \varepsilon_{t-1} - ... - \theta_q * \varepsilon_{t-q} \tag{2.6}$$

Con:

- δ una constante.
- y_{t-1}, y_{t-p} las ventas en el período $(t-1)$ y $(t-p)$.
- $\varepsilon_{t-1}, \varepsilon_{t-p}$ los residuos de los períodos $(t-1)$ y $(t-p)$, los cuales constituyen el ruido.
- ϕ, θ los coeficientes de los procesos autorregresivos y de media móvil, respectivamente.

SARIMA

Cuando se tienen efectos de temporalidad, es necesario hacer uso del Seasonal ARIMA (SARIMA) o ARIMA temporal para incluir el efecto de la temporalidad de los datos. En este caso, el modelo ARIMA se denota como $ARIMA(p,d,q)(P,D,Q)s$. Aquí, (p,d,q) son los parámetros no-temporales descritos anteriormente, mientras que (P,D,Q) siguen la misma intuición, pero para la componente temporal de la serie de tiempo. El término s es la periodicidad de la serie.

En la literatura se sugiere que la aplicación de este modelo tiene un desempeño robusto a la hora de enfrentar series de tiempo con un comportamiento estacionario [13].

SARIMAX

La variante *Seasonal AutoRegressive Integrated Moving Average with eXogenous regressors* o SARIMAX considera el planteamiento de SARIMA, pero le agrega al modelamiento la injerencia de variables externas al modelo. Para este caso de uso, se le agregarán los eventos y feriados como regresores externos.

En la literatura se sugiere que la aplicación de este modelo tiene un desempeño robusto para capturar el movimiento de las series de tiempo al incorporar factores externos como las promociones [13].

2.1.3. Algoritmos basados en árboles

Los árboles binarios o árboles de decisión son un tipo de algoritmo de aprendizaje supervisado en la cual existe una variable objetivo predefinida. Las variables de entrada y salida pueden ser categóricas o continuas y este tipo de algoritmos divide el espacio de los predictores (las variables independientes) en regiones distintas y no sobrepuestas [9]. En la figura 2.1 se puede ver un ejemplo de árbol de decisión el cual posee seis regiones separadas.

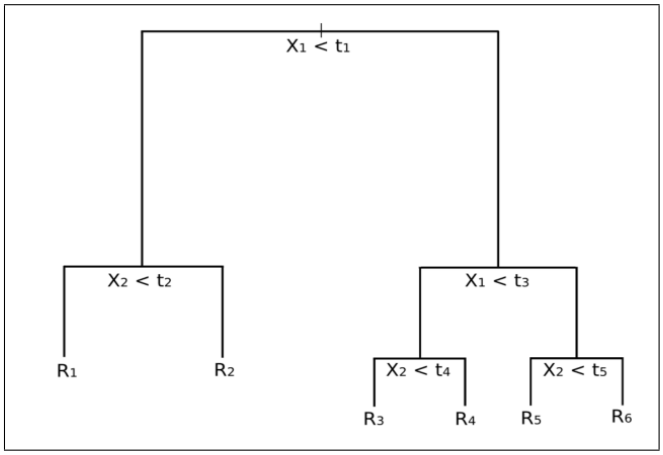

Figura 2.1: Ejemplo de Árbol Binario con seis regiones separadas.

Una de las características de los árboles binarios es que éstos siguen un enfoque de división binaria recursiva, o más conocido como *top-down greedy approach*. En ésta, se analiza la mejor variable para la ramificación en el proceso de división que se esté realizando.

Una de las grandes ventajas de los árboles de decisión es que son fáciles de entender y explicar ya que se sabe cuáles de las variables son las que afectan en el resultado. En este sentido, se puede identificar la importancia de las variables del modelo. Sin embargo, este es un método que lleva generalmente al sobreajuste.

Random forest

Así como todos los modelos descritos anteriormente, un árbol binario también tiene problemas de sesgo y de varianza. Esto se intenta disminuir con la metodología de Random Forest en la cual se utiliza la técnica de *bagging* (figura 2.2) para reducir la varianza de las predicciones. Esta técnica lo que hace es generar subconjuntos de árboles dentro del set de entrenamiento para que la correlación de las variables, si es que existe, no afecte en los resultados, reduciendo la varianza.

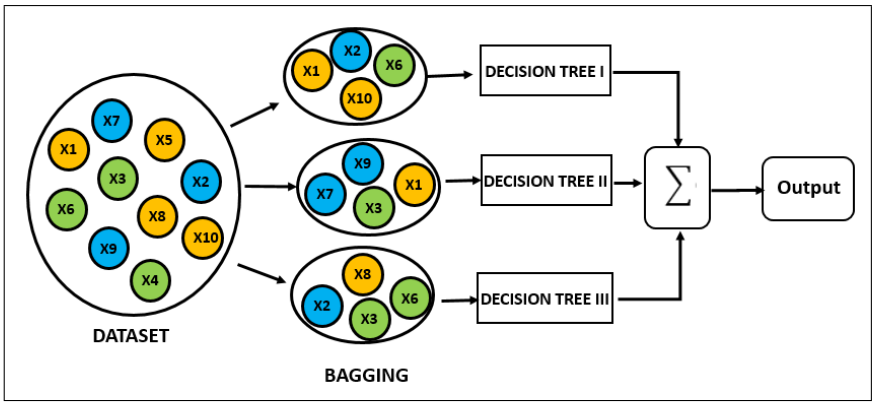

Figura 2.2: Ilustración del proceso de bagging. Fuente: Orellana A. [9]

En otras palabras, Random Forest es una técnica que utiliza múltiples árboles de decisión y

la metodología de bagging, la cual consiste en entrenar cada árbol de decisión en una muestra de datos distinta donde se realiza el muestreo con reemplazo, para reducir la varianza en la predicción. La idea básica que está detrás de este modelo, es de combinar los resultados que arrojan una gran cantidad de árboles de decisiones en vez de dejarle la responsabilidad a uno solo.

La bibliografía indica que estos modelos tienen un rendimiento sobre saliente en estos casos de uso, logrando predecir de manera más precisa la demanda de los productos a predecir [14]

Light gradient boosting machine

El método *Light gradient boosting machine* o *LGBM* es un algoritmo de aprendizaje automático de grado superior basado en árboles que se especializa en manejar grandes conjuntos de datos y características. Utiliza una técnica de construcción de árboles denominada *gradient boosting framework*, en la cual se entrenan árboles sucesivos para corregir los errores cometidos por los árboles anteriores. Para ello Utiliza una técnica de construcción de árboles denominada *gradient-based one-side sampling* que permite construir árboles más profundos y de mayor precisión en comparación con otros algoritmos de gradient boosting.

La ventaja de este método es su rapidez y alta eficiencia en la gestión de memoria, lo que permite manejar grandes conjuntos de datos con facilidad. Además, también tiene una técnica de partición denominada "histograma de gradiente", que permite una mayor eficiencia en la construcción de árboles, una mayor precisión en las predicciones y también es más rápido que otros algoritmos de gradient boosting debido a su implementación de árboles binarios en lugar de árboles de decisión completos.

La bibliografía indica que estos modelos tienen un rendimiento superior a sus pares en este caso de uso [10].

Extreme gradient boosting machine

El algoritmo Extreme gradient boosting machine (XGB) es un algoritmo de gradient boosting similar a LGBM, pero con algunas diferencias en su implementación. El funcionamiento de este algoritmo consiste en agregar secuencialmente predictores a un conjunto. Estos predictores se construyen a partir de los árboles de decisión ya explicados. El trabajo de cada predictor es corregir el anterior. Lo hace ajustando el nuevo predictor a los errores residuales realizados por el predictores anteriores.

Este algoritmo es una implementación de código abierto del *algoritmo de impulso de gradiente* (gradient boosting algorithim). Es muy rápido y escalable. Ofrece características tales como hacerse cargo de la detención de las iteraciones con una implementación de *early stopping*. Tambien es compatible con la paralelización, es decir, el modelo se implementa para entrena con múltiples núcleos de CPU, lo que resulta en una mayor eficiencia y velocidad. Para evitar el sobreajuste, XGB utiliza una técnica de regularización denominada *regularized gradient boosting* . Esto da como resultado la capacidad del modelo para generalizar suficientemente.

La bibliografía indica que estos modelos tienen un excelente rendimiento en las predicciones de series de tiempo, a la vez que ocupan menos recursos computacionales y tiempo de cálculo [1] [2].

2.1.4. Métodos de aprendizaje profundo

Redes neuronales artificiales

Tanto las redes neuronales feedforward como las redes neuronales recurrentes son frecuentemente utilizadas para predecir series de tiempo [5]. Existe una basta cantidad de redes neuronales; en este informe se hace referencia sólo a aquellas que tienen una propagación hacia atrás, las que son conocidas como feed-forward error back-propagation neural nets. En estas redes, los elementos individuales (las neuronas de la red) están organizadas en capas de tal manera que las señales de salida de las neuronas de una capa se transmiten a todas las neuronas de la capa siguiente. En este sentido, el flujo de activación de neuronas va en un solo sentido y pasa capa por capa. El número mínimo de capas que se puede tener son dos capas, la de entrada y la de salida, sin embargo, se pueden agregar capas entremedio llamadas capas ocultas las cuales sirven para aumentar el poder computacional de las redes neuronales. En la Figura 2.3 se muestra una representación de este tipo de redes neuronales. De esta manera, a las Redes Neuronales Artificiales hay que entregarle la cantidad de capas de entradas, de capas ocultas, de neuronas (puntos azules de la Figura 2.4) y de capas de salidas. Además, se le debe entregar el número de *epochs*, que es la cantidad de veces que el aprendizaje ocurre. Así, el proceso de aprendizaje de una red neuronal se repite epoch tras epoch hasta que el rendimiento de la red converge a un valor aceptable.

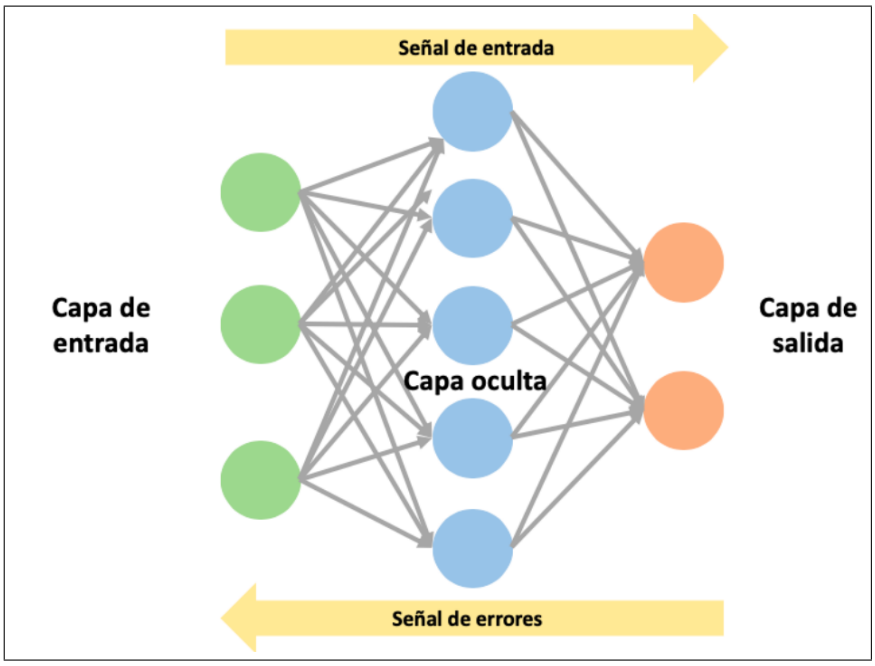

Figura 2.3: Representación de una red neuronal con propagación hacia atrás

Las redes neuronales artificiales están hechas para cumplir con un mapeo requerido uti-

lizando algoritmos de entrenamiento. El algoritmo de entrenamiento común para las redes feed-forward se denomina propagación por error [4]

Este tipo de algoritmos es uno de los más utilizados cuando se tiene una gran cantidad de datos ya que tiene un alto poder predictivo y puede ser fácilmente automatizado; además de tener la habilidad de manejar patrones complejos no lineales [7]. Sin embargo, uno de sus principales puntos en contra es que tiene una muy baja interpretabilidad.

La bibliografía sugiere que los modelos de *Multi Layer Perceptron* son los que mejor desempeño entregan para este caso de uso, a la vez que son de los más visitados por los investigadores [8].

Redes Recurrentes

Las redes neuronales recurrentes se incluyen en este análisis ya que se considera que la demanda de productos puede ser una serie caótica, la cual no posee, en ocasiones, una gran lógica. Las redes neuronales recurrentes lo que hacen es generar un back-propagation o una propagación hacia atrás que permite obtener mejores patrones de aprendizaje a través del tiempo. Esto quiere decir que una red neuronal recurrente puede hacer coincidir cierto patrón (algún patrón de los datos) a través del tiempo el cual se extiende más allá de la ventana de tiempo actual proporcionada [11]. En la Figura 2.4 se muestra una representación de este tipo de redes neuronales.

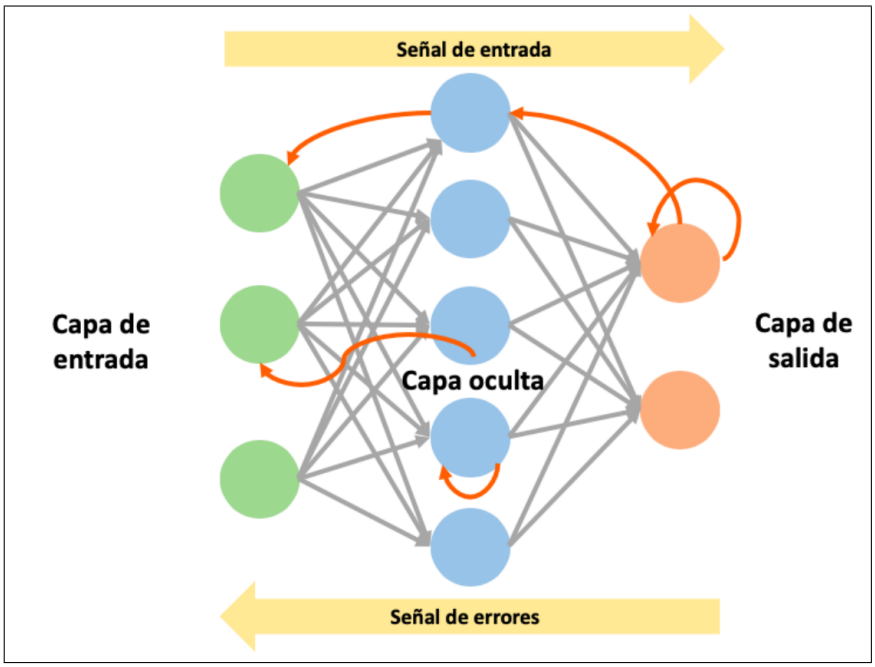

Figura 2.4: Representación de una red neuronal recurrente con propagación hacia atrás

La bibliografía sugiere que el uso de este tipo de arquitecturas tiene gran efectividad para predecir productos de demanda constante [13]. Además, se constata que existe gran interes por parte de los investigadores para estudiar el desempeño de estos modelos [14] [10] por su gran desempeño en otras áreas de estudio.

2.2. Métricas de evaluación de desempeño

Los estadísticos que se utilizan para comparar los modelos anteriormente descritos son MAE, MAPE y WMAPE. Estos se utilizan para medir distintas formas de errores de predicciones.

2.2.1. MAE

El Error Absoluto Medio o MAE, por su acrónimo en inglés, es una métrica que mide la magnitud de los errores en un set de predicciones, sin considerar la dirección del error (error mayor o menor que el valor real). Se calcula como sigue:

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t| \tag{2.7}$$

2.2.2. MAPE

El Error Porcentual Absoluto Medio o MAPE, por su acrónimo en inglés, es un indicador del desempeño del pronóstico de la demanda el cual mide las variaciones porcentuales que existen entre la demanda real y la demanda pronosticada. Se calcula como sigue:

$$MAPE = \frac{\sum_{t=1}^{n}\frac{|y_t - \hat{y}_t|}{|y_t|}}{n} \tag{2.8}$$

Donde y_t es la demanda en t, $\hat{y}_t$ es la demanda pronosticada en t y n es la cantidad de datos pronosticados.

2.2.3. WMAPE

El Error Porcentual Absoluto Medio Ponderado o WMAPE, por su acrónimo en inglés, es un indicador del desempeño del pronóstico de la demanda el cual mide las variaciones porcentuales que existen entre la demanda real y la demanda pronosticada. Se calcula como sigue:

$$WMAPE = \frac{\sum_{t=1}^{n}|y_t - \hat{y}_t|}{\sum_{t=1}^{n} y_t} \tag{2.9}$$

Donde y_t es la demanda en t, $\hat{y}_t$ es la demanda pronosticada en t y n es la cantidad de datos pronosticados. En este trabajo el indicador WMAPE se interpreta como la diferencia porcentual entreel total de las ventas reales y de las pronosticadas para cada sublínea.

Capítulo 3

Framework y datos

Lo presentado en este capítulo responde a los pasos **entendimiento del negocio**, **entendimiento de la data** y **preparación de la data** presentados en la metodología.

3.1. Framework

3.1.1. Requerimientos y situación actual

El primer paso para formular el framework de trabajo, fue identificar los requerimientos que se deben cuplir con las predicciones. Para ello se entrevistó a los subgerentes encargados de la planificación y se enlistan los requisitos a satisfacer:

- Fuente de información : Se debe utilizar la información disponible en el *Data Lake* de la empresa.
- Horizonte de predicción : Largo plazo, equivalente a 150 días (5 meses).
- Granularidad : Cantidad de ventas por *sublínea*[1].
- Algoritmo de predicción : No existe restricción a este respecto.
- Métrica de evaluación : No existe criterio único, debe cumplir con ser simple de entender.

Al estudiar los pronósticos ocupados actualmente, se constata que la metodología realiza las predicciones, ocupando el algoritmo *LGBM* con una granularidad de **ventas diarias por sublínea** y son evaluadas, de manera general[2], usando la métrica de evaluación **MAPE**. Lo anterior se traduce en que para cada sublínea se genera una predicción de 150 días hacia el futuro.

[1]Nivel de agregación intermedio. compuesto por un conjunto de productos diferentes que pertenecen a una misma *familia de productos*.

[2]Se promedia la performance de todas las sublíneas sin realizar ninguna ponderación adicional

Dicha forma de operar, genera múltiples inconvenientes que buscarán ser solventados con la propuesta de este trabajo, estos son:

- Al evaluar de manera agregada se pierde visibilidad respecto a cómo opera el algoritmo en distintos grupos de datos.
- Al generar predicciones para un horizonte de tiempo tan prolongado, el error crece con el tiempo, generando errores mayores en el largo plazo.
- Utilizar solo una métrica para medir el error no permite comprender la calidad de las predicciones.

3.1.2. Comprensión del negocio

Al entrevistar a los analistas que ocupan las predicciones, para poder entender el impacto de negocio de este caso de estudio, se constatan dos factores importante a considerar en el diseño del framework.

1. Las predicciones con horizonte menor a 30 días son utilizadas sin mayores alteraciones por el equipo de operación para la planificación logística, para la contratación de camiones y la toma de decisiones en general.
2. Las predicciones con horizonte mayor a 30 días, son utilizadas de manera agregada. Estas predicciones no son usadas de manera operacional, sino que se usan de manera táctica o estrategica para la busqueda de nuevos proveedores o lanzamiento de campañas.

Para profundizar el conocimiento del negocio, y conocer la magnitud que tiene este caso de estudio y su impacto en los costes de la empresa, se entrevistó a los operarios y encargados de la logistica en los CD[3], con los cuales se obtuvieron los siguientes conocimientos para poder estimar el impacto de la solución en la operación:

- Los camiones se reservan una vez por semana, en base al máximo de ventas estimadas para la semana de evaluación. En caso de que esa estimación sea menor a la demanda real, se contrata una flota adicional que tiene un coste mayor al de la primera reserva.
- Cada camión reservado en la primera etapa tiene un costo de $140.000 mientras que los camiones adicionales tienen un costo de $210.000.
- La carga que cada camión soporta es variable dependiendo del tipo de productos vendidos. Usualmente la carga oscila entre los 60 y los 200 productos por camión.
- De las ventas totales, un 18 % de ellas son despachadas desde un sólo CD.

[3]Centros de distribución

3.1.3. Propuesta de framework

Considerando lo visto previamente, se propone la siguiente extructura para el framework de predicciones:

- Encontrar una segmentación de las series de tiempo que permita desagregar los análisis y observar en detenimiento el desempeño de los algoritmos.
- Dividir la forma en que se hacen las predicciones.

  Primero se aplica un *forecast* general, que prediga a 5 meses con granularidad de ventas mensuales por sublínea, obteniendo la tendencia general de ventas en el largo plazo.

  Luego se realiza un *forecast* desagregado, que prediga los primeros 30 días con granularidad diaria, obteniendo el comportamiento de las ventas dentro del mes. Esto será el input clave para la contratación de camiones.
- Se medirán las predicciones ocupando 3 métricas (MAE, MAPE y WMAPE) que permitan entender el rendimiento de los algoritmos.
- Se realizará una evaluación del impacto económico del modelo utilizando sólo la información disponible de los primeros 30 días, aplicado al caso de la planificación de camiones.

3.2. Datos

Pasando con la etapa de **entendimiento de la data**, se tiene que los datos de ventas se almacenan dentro del *Data Lake* de la compañía, con una granularidad de ventas diarias por sublínea. En particular, los datos consolidados corresponden a la ventas registradas en los últimos 5 años[4] por 128 sublíneas de productos[5].

Lo anterior se traduce en un total de 228.756 observaciones. La tabla 3.1 muestra un extracto de los datos que se tienen para cada sublínea.

Fecha	Sublínea	Ventas
2022-03-26	J0101	14
2022-03-26	J0508	2
2022-03-27	J0101	17
2022-03-28	J0508	2
2022-03-28	J0101	21
2022-03-29	J0508	1
2022-03-29	J0101	12
2022-03-30	J0101	11
2022-03-30	J0508	3
2022-03-31	J0101	15
2022-03-31	J0508	3

Tabla 3.1: Ejemplo datos utilizados.

3.2.1. Limpieza de los datos

Al estudiar los estadísticos principales de los datos (tabla 3.2), se observa que:

- La desviación estándar de las ventas es varias veces mayor al promedio, indicando alta variablidad en las unidades vendidas por día.
- El valor máximo registrado está a más de 100 desviaciones del promedio, indicado que la base datos puede contener valores *outliers*.
- El valor mínimo registrado, es de 1 unidad, indicando que los días en los que no se generan ventas no se registran dentro de las bases de datos.
- No existen valores negativos en las ventas.

[4]Las ventas registradas se ubican entre el 01-01-2017 y el 31-03-2022

[5]Por motivos de confidencialidad, se ha anonimizado el nombre de las sublíneas con un identificador único

Promedio	Desviación estándar	Mínimo	Q1	Q2	Q3	Máximo
303	1.194	1	25	90	276	393.419

Tabla 3.2: Descripción inicial datos.

Con dicha información se decide separar la limpieza de los datos en dos etapas. La primera será realizar una revisión de los valores extremos, en la que se comprobará si las observaciones presentan errores en la imputación que deban ser solucionados. En la segunda etapa se revisarán los valores faltantes, buscando la estrategia correcta para imputar los registros que indiquen que no hubo una venta.

Valores extremos

Para estudiar la presencia de valores *outliers* se identificaron los 10 registros de mayores ventas (tabla 3.3). Al observar dichos registros, se observa que hay dos grupos diferenciables.

El primero, compuesto por los tres mayores valores, al cual se le llamará grupo 1. Y el segundo, que corresponde a aquellos registros menores a 40.000 ventas, que se le llamará grupo 2.

Fecha	Sublínea	Venta
2022-03-27	J0305	393.419
2022-03-27	J0601	132.571
2022-03-27	J1301	65.946
2021-05-31	J1109	37.039
2021-10-04	J1109	34.557
2021-10-04	J0802	34.307
2021-05-31	J1006	34.146
2020-08-31	J1109	32.318
2021-10-04	J1002	31.351
2021-05-31	J0802	29.345

Tabla 3.3: Top 10 registros más grandes.

Grupo 1

Al estudiar el comportamiento de las sublíneas asociadas a este grupo, se observa claramente que estos son valores atípicos (figura 3.1). Por ello se decide reemplazar estos valores por la mediana de las ventas de la sublínea, lo cual corrige esa anomalía (figura 3.2).

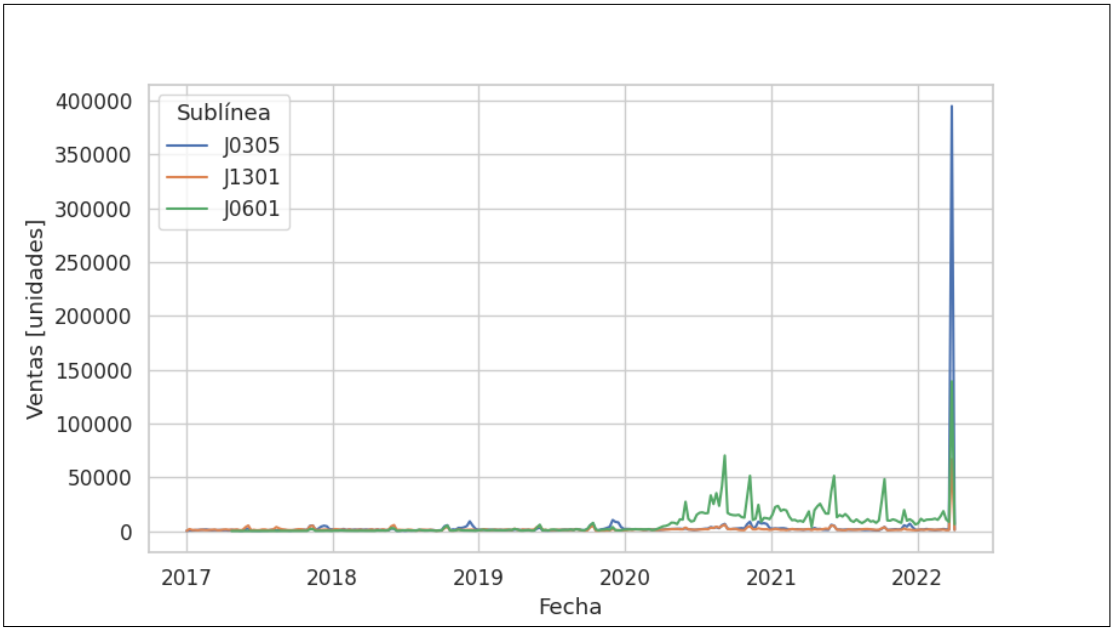

Figura 3.1: Ventas históricas, grupo 1.

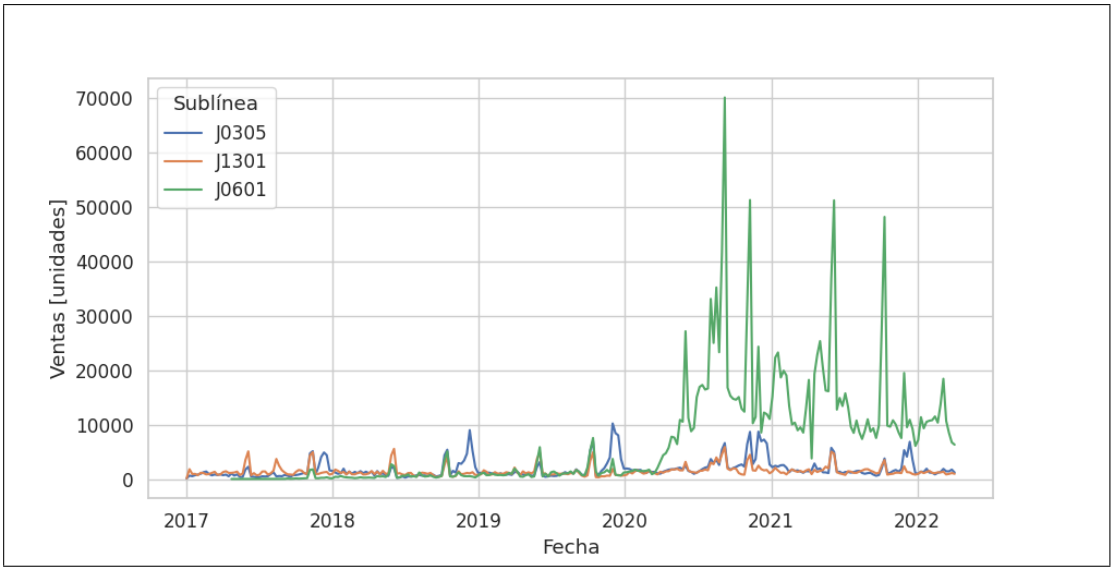

Figura 3.2: Ventas históricas corregidas, grupo 1

Grupo 2

Al estudiar el comportamiento de las sublíneas asociadas a este grupo, se observa que si bien presentan *peaks* de valores altos, estos son consistentes tanto con su historia previa como con la fecha en que las otras sublíneas tienen *peaks* de ventas (figura 3.3). Por ello se decide no aplicar ningún procesamiento sobre estos datos.

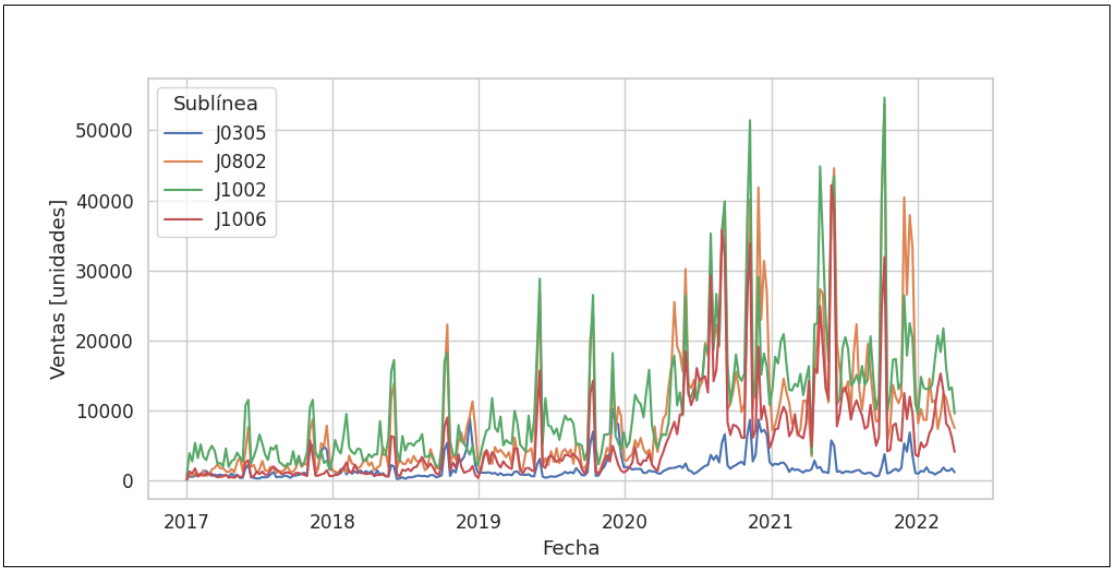

Figura 3.3: Ventas históricas, grupo 2.

Datos faltantes

Existen dos casuísticas que explican la falta de registros en una fecha:

1. Itinerancia en las ventas : Comportamiento general que se atribuye a que la sublínea no presenta ventas todos los días sino que tiene un comportamiento *itinerante*. Como ejemplo, vemos que la sublínea **J0508** no tiene registro para el 27-03-2022 (tabla 3.1).
2. Lanzamiento de nuevas sublíneas : Comportamiento específico que contempla a las sublíneas que no registran ventas desde el comienzo, ya que su primera venta se realizó en algún instante posterior. En particular, se tiene que 111 de las 128 sublíneas (86,7 %) presentan su primera venta en Enero del 2017, mientras que las 17 restantes presentan su primera venta entre Febrero del 2017 y Enero del 2019 (figura 3.4).

Tomando en cuenta aquello, se imputan registros con valor 0 para cada sublínea a partir del primer día en que presentan ventas. Con ello, el dataframe pasa a tener 240.612 registros, simbolizando un aumento de 11.856 registros.

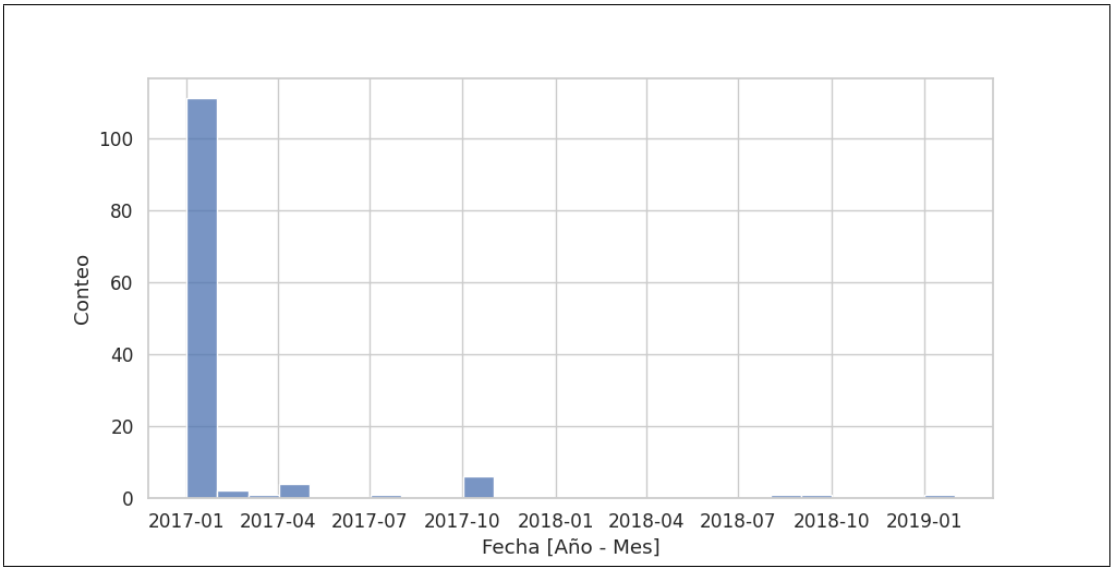

Figura 3.4: Distribución de mes de primera venta registrada.

Resumen limpieza

Al revisar nuevamente los estadísticos principales de los datos, una vez aplicada esta limpieza (tabla 3.4), se observa que:

- El promedio de los registros disminuye, lo cual es consistente con la limpieza realizada.
- La desviación estándar disminuye, lo cual nos indica que los datos están más concentrados que antes, no obstante sigue siendo bastante mayor al promedio.

Promedio	**Desviación estándar**	**Mínimo**	**Q1**	**Q2**	**Q3**	**Máximo**
285	791	0	19	80	259	37.039

Tabla 3.4: Descripción datos - postprocesamiento.

3.2.2. Caracterización general de los datos

En esta sección, se realiza una caracterización de los datos, tanto de manera agregada como desagregada, con el fin de entender las macro-tendencias presentes y las dinámicas internas entre las sublíneas.

Comportamiento agregado

Al estudiar el comportamiento agregado de las ventas (figura 3.5) se observa que estas tienen dos etapas de comportamiento, las cuales se separan por el inicio de la pandemia del Covid 19 en Marzo del 2020. Cada una presenta comportamientos diferenciables, por lo cual se propone un estudio por separado de ambos.

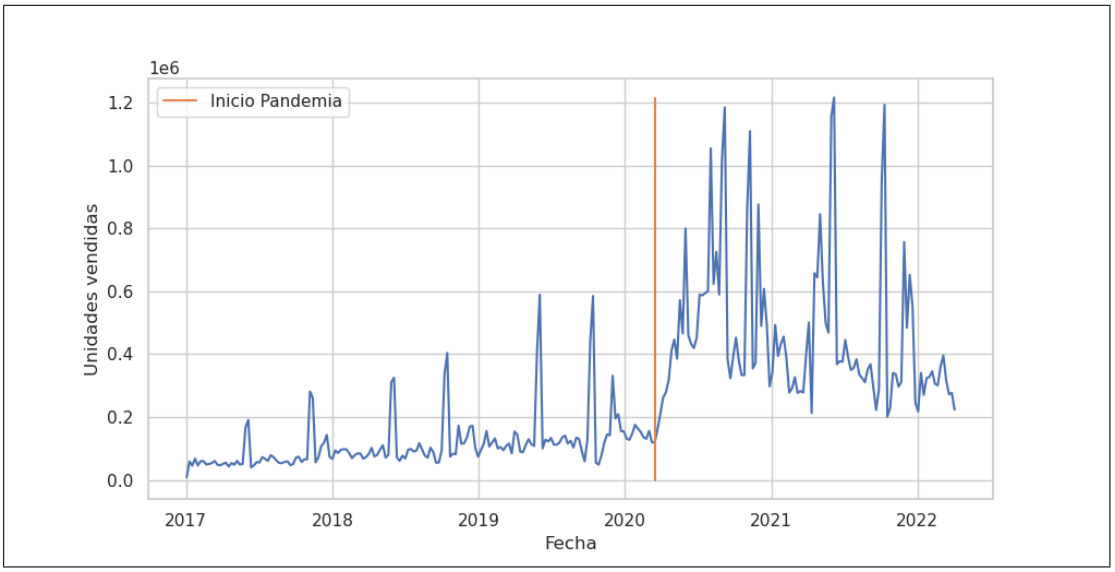

Figura 3.5: Ventas agregadas 2017/01 - 2022/03

Comportamiento pre-pandemia

Se observa que hay gran variabilidad dentro de las ventas en este período, lo que se representa en una desviación estándar 3 veces mayor al promedio (tabla 3.5). Además, se observa que existe cierta estacionalidad, pues se observa que los *peaks* de demanda están distribuidos en el tiempo de manera relativamente uniforme (figura 3.6).

Promedio	Desviación estándar	Mínimo	Q1	Q2	Q3	Máximo
131	404	0	11	42	134	22.537

Tabla 3.5: Descripción datos - pre-pandemia.

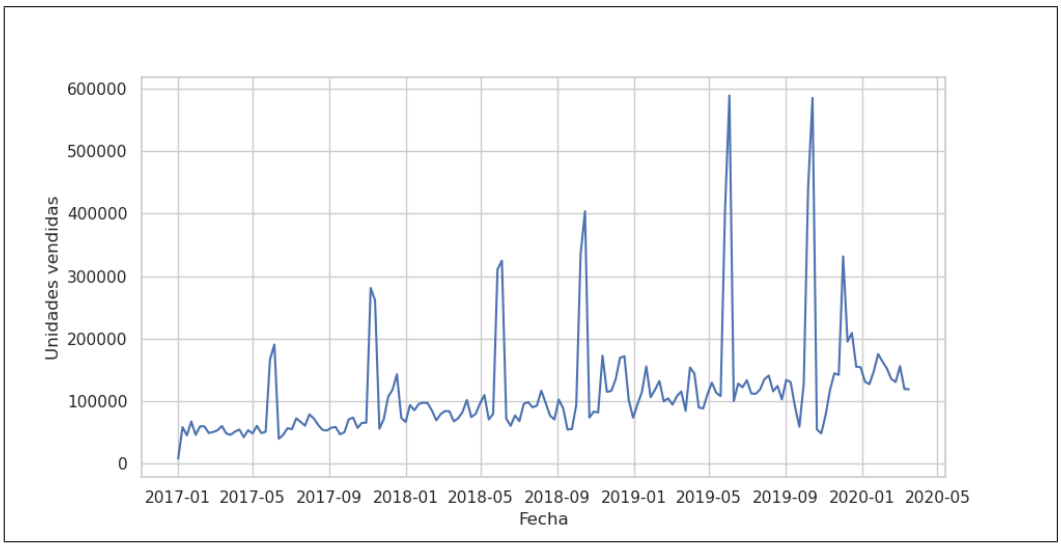

Figura 3.6: Ventas Pre Pandemia

Al estudiar el comportamiento de esta serie de tiempo a través de los meses (figura 3.7), se pueden destacar los siguientes puntos:

- Los meses de Enero, Febrero, Marzo, Abril y Junio tienen las menores ventas del año.
- En el mes de Mayo se vende la mayor cantidad de productos.
- A partir del año 2018, existe un *peak* de demanda en el mes de Octubre.
- Existe una tendencia al alza en las unidades vendidas.

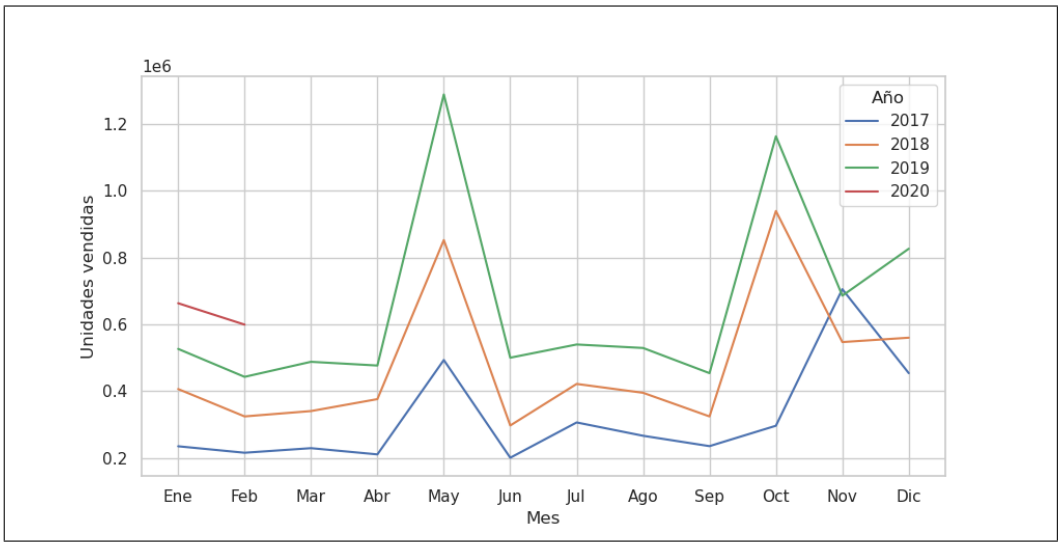

Figura 3.7: Desagregación de ventas pre-pandemia.

Comportamiento pandemia

Se observa que el promedio de las ventas es mayor en este período que en el anterior (tabla 3.6). A diferencia del periodo anterior, no se observa estacionalidad en las ventas (figura 3.8).

Promedio	Desviación estándar	Mínimo	Q1	Q2	Q3	Máximo
512	1.105	0	62	198	584	37.039

Tabla 3.6: Descripción datos - pandemia.

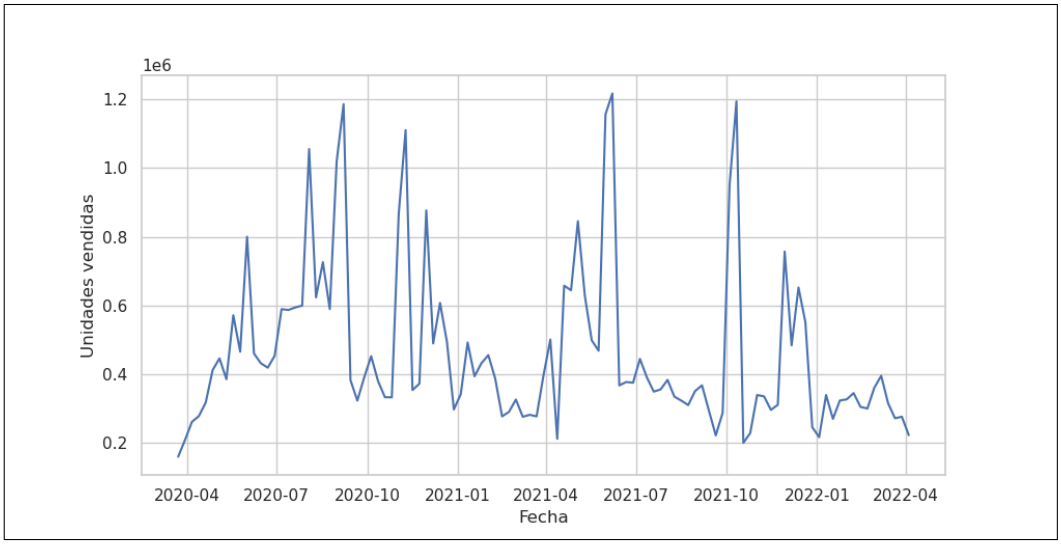

Figura 3.8: Ventas pandemia.

Al estudiar el comportamiento de esta serie de tiempo a través de los meses (figura 3.9), se pueden destacar los siguientes puntos:

- Los meses de Enero, Febrero, Marzo, Abril y Junio se mantienen como los meses con menores ventas del año.
- El mes de Mayo tiene un *peak* de ventas, siendo el mes en que más productos se venden en el primer semestre, no obstante ya no es el mes con mayor nivel de ventas en el año.
- Desaparece el *peak* de demanda en el mes de Octubre.
- No existe una tendencia clara respecto a las unidades vendidas en este periodo, si bien existe un alza en las ventas del año 2021 respecto al año 2020, esta no es consistente a lo largo de los meses.
- El comportamiento del segundo semestre no sigue ningún patrón observable.

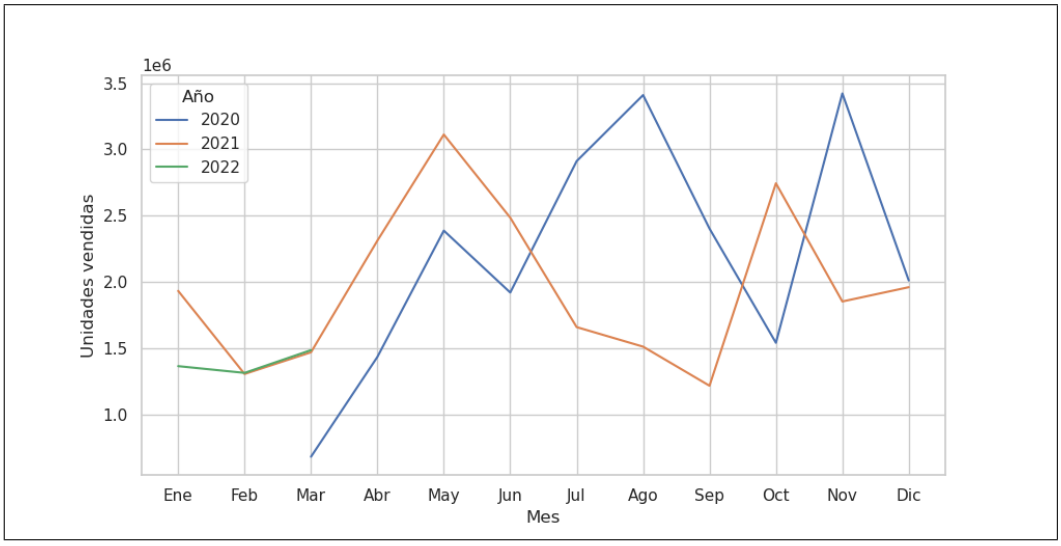

Figura 3.9: Desagregación de ventas pandemia.

Comportamiento desagregado

Para estudiar el comportamiento de las dinámicas internas de cada sublínea, hay dos temáticas que deben ser resueltas.

1. Efecto pandemia : ¿Existe diferencia en el comportamiento de ventas de las sublíneas entre los periodos pre-pandemia y pandemia?
2. Distribución de valores : La presencia de valores extremos (valores muy altos o muy bajos) ¿es igual entre las sublíneas o hay un comportamiento diferenciable?

Efecto pandemia

Para analizar este efecto, se estudia la composición que cada sublínea representa respecto al total de las ventas, tanto en el periodo pre-pandemia (tabla 3.7) como en el periodo de pandemia (tabla 3.8).

Al respecto, se observa que las sublíneas que mas peso tienen dentro del mix (las sublínea **J1109** y **J1002**) pierden importancia relativa en el periodo de pandemia, pues pasan de representar el 5,9 % y 5,3 % de las ventas respectivamente, a representar un 4,2 % y un 3,9 %.

Al mismo tiempo, se observa que las sublíneas **J0908**, **J1003**, **J1101**, **J0913** y **J1104** desaparecen del top 15; destaca especialmente la sublínea **J0908** la cual pasa de representar un 2,5 % del total, a un 1,1 %.

En sentido contrario ocurre un fenomeno similar, pues en el periodo de pandemia las sublíneas **J0601**, **J0801**, **J0406**, **J0501** y **J0110** se integran al top 15; destacando especialmente la sublínea **J0601** la cual pasó de un 0,7 % a un 3,3 % del mix.

Sublínea	Porcentaje del mix total
J1109	5,93 %
J1002	5,32 %
J0802	3,46 %
J1105	3,32 %
J1114	3,07 %
J0909	2,71 %
J0908	2,49 %
J1003	2,40 %
J1501	2,29 %
J1201	2,29 %
J1101	2,12 %
J1102	1,96 %
J1006	1,91 %
J0913	1,81 %
J1104	1,75 %

Tabla 3.7: Top 15 mix pre pandemia

Sublínea	Porcentaje del mix total
J1109	4,24 %
J1002	3,87 %
J0802	3,59 %
J1114	3,54 %
J0601	**3,28 %**
J0909	2,82 %
J0801	**2,62 %**
J1105	2,45 %
J1006	2,44 %
J0406	**2,23 %**
J1201	2,16 %
J1102	2,09 %
J0501	**2,07 %**
J0110	**1,96 %**
J1501	1,95 %

Tabla 3.8: Top 15 mix pandemia

Distribución de valores

Para analizar si la distribución de los valores extremos es igual en las diferentes sublíneas, se genera una caracterización individual de cada sublínea (tabla 3.9) que considera 3 variables:

- Ventas altas: Porcentaje del tiempo que vende más de 1000 unidades en un día.
- Ventas bajas: Porcentaje del tiempo que vende menos de 10 unidades en un día.
- Ventas nulas: Porcentaje del tiempo que no se venden unidades en un día.

Sublínea	Ventas altas	Ventas bajas	Ventas nulas
J0101	0,05 %	58,91 %	5,65 %
J0102	0,00 %	54,30 %	10,95 %
J0103	0,00 %	90,96 %	48,13 %
		⋮	
J3225	0,00 %	91,92 %	42,29 %
J3250	0,00 %	89,25 %	13,90 %

Tabla 3.9: Caracterización sublíneas.

Al tomar una muestra aleatoria de sublíneas (tabla 3.10) se observan comportamientos bastante disimiles. como pueden ser:

- Sublínea J1109 : Presenta un 49 % de registros catalogados como ventas altas, y 0 % tanto en ventas bajas como nulas.
- Sublínea J0805 : Presenta registros nulos un 15,7 % del tiempo, y ventas bajas un 27,9 % del tiempo.
- Sublínea J0408 : Presenta un porcentaje pequeño tanto en registros de venta alta como de venta baja.

Sublínea	**Ventas altas**	**Ventas bajas**	**Ventas nulas**
J0306	0,47 %	4,54 %	0,00 %
J1109	49,01 %	0,00 %	0,00 %
J0201	2,97 %	4,28 %	0,05 %
J0302	7,72 %	0,73 %	0,00 %
J0805	4,54 %	27,94 %	15,72 %
J0601	37,31 %	9,62 %	4,81 %
J0801	29,23 %	0,00 %	0,00 %
J1110	6,94 %	0,42 %	0,00 %
J0408	1,57 %	1,36 %	0,00 %
J0602	10,02 %	18,46 %	9,86 %

Tabla 3.10: Muestra de caracterización de sublíneas.

Con este estudio se comprueba que existen patrones de comportamiento diferenciables entre las diferentes sublíneas. No obstante también es posible reconocer sublíneas que presentan comportamientos similares, como por ejemplo:

- Sublíneas J1109 y J0801 : Las cuales no presentan procentajes de ventas bajos, y tienen un significativo valor en *Ventas altas*. Lo cual nos indica que ambas sublíneas comparten la caracteristica de tener importantes peaks de ventas.
- Sublíneas 1110 y J0306 : Las cuales si bien no tienen alto pocentaje de ventas altas, no tienen ventas nulas, lo cual sería indicador de que son sublíneas que suelen venderse todos los días, de manera regular, pero sin generar *peaks* de demanda

Síntesis de la caracterización

Luego de realizados los análisis propuestos en esta sección, se extraen las siguientes conclusiones:

1. La pandemia afectó el comportamiento de las series de tiempo fuertemente. Lo cual se tradujo en un aumento del 400 % en el promedio de unidades vendidas.
2. La pandemia afectó el comportamiento de los consumidores, los cuales empezaron a cambiar el tipo de productos que compraban. Lo anterior, se infiera a partir del cambio en el mix de sublíneas de mayor importancia respecto a las ventas totales.
3. Las sublíneas tienen comportamientos muy disimiles entre ellas, aún así es posible reconocer grupos que tienen comportamientos similares. (tablas 3.7 y 3.8)
4. Existen días en los cuales la demanda alcanza valores *peak*. Es importante corroborar si existe alguna variable externa que explique este comportamiento.
5. Previo a la pandemia existía un patrón estacionario de la serie agregada a lo largo del tiempo, no obstante este se vió modificado después de Marzo del 2020.

3.2.3. Datos externos

Para enriquecer el análisis de los datos y fortalecer los modelos a desarrollar, se incluye la existencia de campañas o promociones que la empresa haya impulsado, y de fechas externas importantes que puedan explicar los cambios que sufrió la demanda.

1. Eventos : Contiene las campañas o eventos implementados por la empresa para potenciar las ventas.
2. Feriados : Contiene información sobre los feriados nacionales.

A continuación se enseña una muestra de estos datos y se profundiza el origen y tratamiento que tienen.

Eventos

Los datos recopilados respecto a las campañas implementadas por la compañia, considera los 8 macro eventos más importantes para la compañia (tabla 3.11). Dichos macro eventos, fueron definidos por el equipo comercial de la empresa, quienes se encargan de la implementación de los mismos. Se observa que cada campaña es aplicada por varios días y **se considera que tienen alcance masivo, y por tanto afectan a todas las sublíneas.**

Fecha	Black	CMR	Cyber	Padre	Escolar	Hot Sale	Liquidación	Valentín
2017-01-02	0	0	0	0	0	0	1	0
2017-01-03	0	0	0	0	0	0	1	0
2017-01-04	0	0	0	0	0	0	1	0
	⋮			⋮			⋮	
2022-03-06	0	0	0	0	0	1	0	0
2022-03-07	0	0	0	0	0	1	0	0
2022-03-08	0	0	0	0	0	1	0	0

Tabla 3.11: Muestra de campañas. Fuente: Equipo Comercial

Feriados

Los datos recopilados respecto a los feriados nacionales, también fueron consolidados por el equipo comercial (tabla 3.12). Esta tabla considera únicamente si hubo o no un feriado en la fecha correspondiente, sin discriminar por el tipo de feriado que este hace alusión.

Fecha	Feriado
2017-01-01	1
2017-04-19	1
2017-05-01	1
⋮	
2021-12-25	1
2022-01-01	1

Tabla 3.12: Muestra de feriados. Fuente: Equipo Comercial

Impacto de los datos externos

Para estudiar tanto el impacto que tanto los eventos como los feriados tienen sobre las ventas, se calcula un indicador que captura el efecto que tienen sobre las ventas promedio de los periodos en los que estos ocurren (figura 3.10). Se observa que alguna de estas instancias potencian las ventas, mientras que otras generan el efecto inverso, es decir, disminuyen el promedio de las ventas. Lo anterior indica que ninguna de las circunstancias estudiadas tiene un efecto neutro sobre las sublíneas, lo cual permitirá mejorar la capacidad predictiva de los modelos.

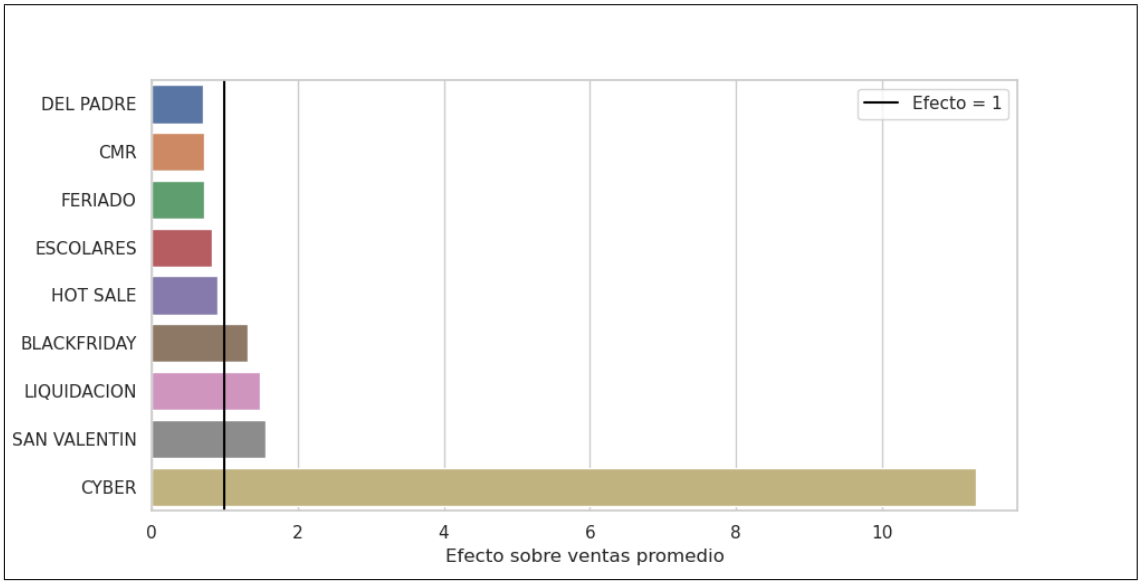

Figura 3.10: Efecto de eventos sobre ventas promedio.

3.3. Segmentación de series temporales

Tras los análisis realizados previamente se evidenció que existe mucha disimilitud en el comportamiento de las ventas de cada sublínea, por ello se desea agrupar aquellas sublíneas que tengan comportamientos similares, de manera de poder diseccionar de mejor manera los resultados y enriquecer el análisis respectivo.

Para poder realizar lo anterior, y en base a lo visto en las secciones previas, se caracterizó cada serie temporal en 6 variables (tabla 3.13), las cuales se explican a continuación:

- Ventas diarias (V1) : Corresponde al número de unidades que se venden, en promedio, de cada sublínea por día.
- Variación de ventas (V2) : Corresponde a la desviación estándar de las unidades vendidas de cada sublínea por día.
- Itinerancia en ventas (V3) : Corresponde al porcentaje de días en las cuales la sublínea no presenta ventas.
- Ventas bajas (V4) : Corresponde al al porcentaje de días en las cuales la sublínea vende 10 o menos unidades.
- Ventas elevadas (V5) : Corresponde al al porcentaje de días en las cuales la sublínea vende 1000 o mas unidades.
- Multiplicador Pandemia (V6) : Corresponde al aumento que sufrió el promedio de las ventas diarias de la sublínea producto del *efecto pandemia*.

Sublínea	V1	V2	V3	V4	V5	V6
J0101	17	51	5,7 %	58,9 %	0,1 %	0,9
J0102	16	27	11.0 %	54,3 %	0,0 %	2,6
J0103	4	10	48,1 %	91,0 %	0,0 %	3,3
	⋮			⋮		⋮
J3205	81	108	0,4 %	9,1 %	0,3 %	2,5
J3225	3	6	42,3 %	91,9 %	0,0 %	6,7
J3250	5	6	13,9 %	89,3 %	0,0 %	1,2

Tabla 3.13: Muestra de caracterización para clusters.

Con dichos datos, se aplicó el algoritmo de clustering **K-Means** para poder identificar aquellas series de tiempo que comportan caracteristicas principales. Con ello, se identificaron 6 clusters[6][7] en los cuales se puede agrupar a las 128 sublíneas de la base de datos (tabla 3.14).

Cluster	V1	V2	V3	V4	V5	V6
Cluster 1 Rotación normal	115	191	0,5 %	9,9 %	0,9 %	3,6
Cluster 2 Rotación variable	397	738	0,6 %	3,7 %	7,3 %	4,4
Cluster 3 Baja rotación	43	83	8,5 %	44,2 %	0,2 %	4,6
Cluster 4 Alta rotación	964	1.568	0,3 %	0,9 %	30,7 %	5,9
Cluster 5 Rotación itinerante	5	13	47,2 %	88,3 %	0,0 %	5,4
Cluster 6 Rotación potenciada	228	473	8,6 %	34,4 %	5,0 %	16,5

Tabla 3.14: Caracterización de clusters.

Una vez consolidados los clusters, se procede con la descripción de ellos.

- **Cluster 1 - Rotación normal** : Segmento con la mayor cantidad de sublíneas, agrupa 43 de las 128 sublíneas (33,5 %). Se caracteriza por tener un comportamiento *estable* lo cual se evidencia en una variación de ventas (V2) muy similar a las ventas diarias (V1).

[6]Para la elección del número de clusters se utilizó el método del codo.

[7]Posterior a la identificación de los clusters se hizo una validación con el negocio para corroborar la consistencia de los hallazgos con la información comercial que se maneja en la firma.

- **Cluster 2 - Rotación variable** : Segundo segmento más grande, agrupa 36 de las 128 sublíneas (28,1 %). Se caracteriza por tener un comportamiento altamente volatil, lo cual se evidencia en una variación de ventas (V2) mucho mayor a las ventas diarias (V1). Como es de esperarse, al tener alta variación estas sublíneas tienen un porcentaje de tiempo no nulo tanto en ventas elevadas (V5) como en ventas bajas (V4).

- **Cluster 3 - Baja rotación** : Se agrupa a 20 de las 128 sublíneas (15,6 %). Se caracteriza por tener un promedio de ventas diarias pequeño (43). Consistente con lo anterior, este segmento tiene un alto porcentaje en la variable de ventas bajas (44,2 %), lo que simboliza que las sublíneas de este segmento pasan un alto porcentaje del tiempo teniendo ventas de menos de 10 unidades diarias.

- **Cluster 4 - Alta rotación** : Se agrupa a 15 de las 128 sublíneas (11,7 %). Se caracteriza por tener el promedio de ventas más grande (964). Consistente con lo anterior, este segmento tiene el porcentaje más alto en la variable de ventas altas (30,7 %), lo que simboliza que las sublíneas de este segmento pasan un alto porcentaje del tiempo teniendo ventas de más de 1000 unidades diarias.

- **Cluster 5 - Rotación itinerante** : Segundo segmento más pequeño, agrupa 8 de las 128 sublíneas (6,3 %). Su principal caracteristica es tener el porcentaje más alto en la variable de itinerancia en ventas (88,3 %) lo que simboliza que las sublíneas de este segmento pasan un alto porcentaje del tiempo sin tener ventas.

- **Cluster 6 Rotación potenciada** : Ssegmento más pequeño, agrupa sólo a 6 de las 128 sublíneas (4,7 %). Se caracteriza por tener el valor más alto asociado a la variable multiplicador en pandemia (V6), lo que significa que las sublíneas de este grupo tuvieron un aumento muy grande en sus ventas desde que comenzó la pandemia.

Capítulo 4

Resultados

En esta sección se implementan los modelos descritos en el capítulo 2 y se reportan los resultados que estos tienen al ser evaluados. Es importante notar que para que los diferentes algoritmos presentados puedan ser aplicados, se debe realizar un procesamiento adicional sobre los datos previamente mostrados. Los procesamientos respectivos para implementar los diferentes métodos se mostrarán en la sección *procesamiento de datos* de este capítulo.

Tal como se ha mencionado en este documento las predicciones deben tener un horizonte de 150 días (5 meses) por lo que se dividieron los datos, considerando a los registros generados hasta el día 31 de Octubre del 2021 (58 meses) como parte del conjunto de entrenamiento, mientras que los registros generados entre 1 de Noviembre del 2021 y el 30 de Marzo del 2022 (5 meses) serían el conjunto de prueba (figura 4.1).

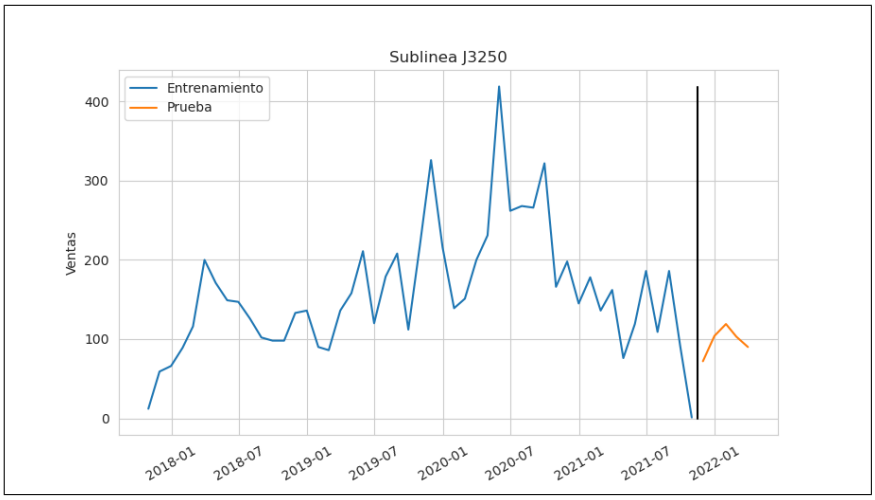

Figura 4.1: Ejemplo de división de conjuntos de entrenamiento y prueba

Para la aplicación de los modelos, se decidió incorporar dos tipos de experimentos, según lo permitieran los métodos empleados, estos fueron:

- Experimento de temporalidad : Se entrena el modelo correspondiente utilizando únicamente la información generada desde el comienzo de la pandemia, es decir, los registros generados a partir de Marzo del 2020. A estos modelos se les distinguirá pues se les agregará el subíndice **PANDEMIA** para reconocer la diferencia en los resultados con el caso base.
- Experimento de concentración : Se aplica el modelo correspondiente utilizando unicamente la información registrada por las sublíneas de un cluster. Este experimento cambia un poco la lógica de predicción, pues en vez de predecir todos los datos con un modelo, ajusta un modelo para cada cluster. A estos modelos se les distinguirá pues se les agregará el subíndice **FOCUS** para reconocer la diferencia en los resultados con el caso base.

Cabe destacar que estos experimentos podrían ser aplicados de manera simultánea, en cuyo caso se agregará el subíndice **HIBRIDO**.

En la sección *Predicciones* se presentan todos los resultados obtenidos por los 13 modelos, con sus respectivas experimentaciones. En particular, se mostrarán, de manera separada las predicciones mensuales y las predicciones diarias. Además, cada tipo de predicciones será desagregada por el rendimiento que tienen los modelos en cada cluster. De manera de poder observa cual algoritmo tiene mejor rendimiento en cada segmento de sublíneas.

Luego se presenta la sección *Análisis de resultados* en la cual se realiza una comparación entre los resultados obtenidos, de manera de tener una síntesis de estos. Con lo anterior se espera poder concluir respecto a la conveniencia de utilizar los modelos propuestos.

Finalmente, en la sección *Evaluación económica* se entrega un desglose de los costes asociados a las predicciones de cada modelo, con el afán de poder comparar con los costes incurridos con el framework actual y calcular la diferencia de rendimiento de cada modelo.

4.1. Procesamiento de Datos

En esta sección se explora los diferentes procesamientos que debieron realizarse sobre los datos para hacer posible las implementaciones de los diferentes algoritmos.

4.1.1. Algoritmos simples

En este inciso se contemplan los modelos **NAIVE FORECAST (NAIVE)** y **MOVING AVERAGE (MA)**, los cuales consideran únicamente la variable de unidades vendidas para realizar el prónostico. Cabe destacar que con estos algoritmos no es posible aplicar ninguno de los experimentos propuestos.

4.1.2. Algoritmos estadísticos

En este inciso se contemplan los modelos **HOLT-WINTERS (HW)**, **ARIMA**, **SARIMA** y **SARIMAX**. Ninguno de los tres primeros requiere de algún procesamiento o consideración adicional, pues únicamente consideran la variable de unidades vendidas para realizar el prónostico.

No obstante, el modelo **SARIMAX** soporta la inclusión de variables externas, por lo que se agrega la información asociada a la aplicación de campañas y feriados. Lo anterior genera un dataframe diferente que contiene tanto las ventas diarias como la aplicación de campañas (tabla 4.1).

Fecha	Ventas	...	ESCOLARES	BLACKFRIDAY
2017-10-31	11	...	0	0
2017-11-30	58	...	0	0
2017-12-31	65	...	0	0
2018-01-31	88	...	1	0
2018-02-28	115	...	1	0
2018-03-31	199	...	0	0

Tabla 4.1: Muestra de datos con eventos, sublínea J3250, agregación mensual

Para la aplicación de los algoritmos **ARIMA**, **SARIMA** y **SARIMAX** se ocupa la librería **AutoArima** para encontrar la mejor combinación de los parámetros p, d, q, P, D, Q y s para cada una de las sublíneas. Por contra parte, para el algoritmo **Holt-Winters** se eligió un combinación global de los párametros α, β γ que maximizara los resultados del algoritmo sobre un conjunto de validación.

Finalmente, cabe destacar que se aplica el experimento de temporalidad sobre los algoritmos **ARIMA**, **SARIMA** y **SARIMAX**.

4.1.3. Algoritmos basados en árboles

En este inciso se contemplan los modelos **Random forest (RF)**, **Light gradient Bboosting machine (LGBM)** y **Extreme gradient boosting (XGB)**. Para su implementación fue necesario realizar un amplio preocesamiento a los datos originales, de manera de adaptar el formato de series de tiempo a un formato tabular que pudiera ser procesado por los algoritmos basados en árboles.

En primer lugar se crearon los *lags*, esto es, elegir el número de días de historia que se utilizaran hacia el pasado. La tabla 4.2 ejemplifica este procesamiento, para la sublínea **J3250**, mostrando como quedan los datos una vez procesados utilizando número de *lags* igual a cinco.

Fecha	$Ventas_t$	$Ventas_{t-1}$	$Ventas_{t-2}$	$Ventas_{t-3}$	$Ventas_{t-4}$	$Ventas_{t-5}$
2017-10-31	11					
2017-11-30	58	11				
2017-12-31	65	58	11			
2018-01-31	88	65	58	11		
2018-02-28	115	88	65	58	11	
2018-03-31	199	115	88	65	58	11

Tabla 4.2: Muestra de transformación con *lags*, sublínea J3250

En segundo lugar se crearon los *features sinusoidales* o *cyclical features encoding* sobre las fechas, esto es, aplicar transformaciones con seno y coseno sobre las fechas, de tal manera de captar el movimiento cíclico que tienen las fechas. En particular para el caso mensual se aplica esta técnica sobre los meses donde se registra la venta, lo cual sigue la siguiente formula:

$$coseno_{mes_i} = cos\left(\frac{2 * \pi * mes_i}{12}\right) \tag{4.1}$$

$$seno_{mes_i} = sin\left(\frac{2 * \pi * mes_i}{12}\right) \tag{4.2}$$

Por ejemplo, para el mes de Diciembre (12), se tendría el siguiente calculo:

$$coseno_{12} = cos\left(\frac{2 * \pi * 12}{12}\right) = 1 \tag{4.3}$$

$$seno_{12} = sin\left(\frac{2 * \pi * 12}{12}\right) = 0 \tag{4.4}$$

La tabla 4.3 muestra de manera tabular como quedan estructurados los datos después de este procesamiento.

Fecha	$Coseno_{mes_i}$	$Seno_{mes_i}$
2017-10-31	0,5	-0,9
2017-11-30	0,9	-0.5
2017-12-31	1,0	0,0
2018-01-31	0,9	0,5
2018-02-28	0,5	0,9
2018-03-31	0,0	1,0

Tabla 4.3: Muestra de transformación con *lags*, sublínea J3250.

Análogamente, para el caso de predicciones diarias, se aplica un procesamiento similar sobre el atributo día y semana, siguiendo una formula similar a lo visto previamente:

$$coseno_{dia_i} = cos\left(\frac{2 * \pi * dia_i}{30}\right) \tag{4.5}$$

$$seno_{dia_i} = sin\left(\frac{2 * \pi * dia_i}{30}\right) \tag{4.6}$$

$$coseno_{semana_i} = cos\left(\frac{2 * \pi * semana_i}{52}\right) \tag{4.7}$$

$$seno_{semana_i} = sin\left(\frac{2 * \pi * semana_i}{52}\right) \tag{4.8}$$

En tercer lugar, se genera una caracterización general de la serie de tiempo, esto es, se calculan los estadisticos que representan a la serie de tiempo hasta el instante en que son procesados. Se muestra en la tabla 4.4 un ejemplo de esto, para la sublínea J3250.

Fecha	$Promedio_{ventas}$	$Suma_{ventas}$	$Desviación_{ventas}$	$Maximo_{ventas}$	$Mínimo_{ventas}$
2017-12-31	34,5	69	33,2	58	11
2018-01-31	44,7	134	29,4	65	11
2018-02-28	55,5	222	32,3	88	11
2018-03-31	67,4	337	38,6	115	11
2018-04-30	89,3	536	63,9	199	11
2018-05-31	100,9	706	65,8	199	11

Tabla 4.4: Muestra de caracterización de serie de tiempo, agregación mensual.

Además de los procesamientos mostrados, también se incluyen los eventos de igual manera que en el caso del modelo SARIMAX (tabla 4.1).

Finalmente, se agrega al dataframe generado los valores a predecir. Como se mencionó previamente se aplicará la estrategia directa para predecir los valores futuros, por lo que se deben agregar estos valores como columnas adicionales, tal como se observa en la tabla 4.5

Fecha	$Ventas_t$	$Ventas_{t-1}$	$Ventas_{t-2}$	$Ventas_{t-3}$	$Ventas_{t-4}$	$Ventas_{t-5}$
2017-10-31	11	58	65	88	115	199
2017-11-30	58	65	88	115	199	170
2017-12-31	65	88	115	199	170	148
2018-01-31	88	115	199	170	148	146
2018-02-28	115	199	170	148	146	125
2018-03-31	199	170	148	146	125	101

Tabla 4.5: Muestra de datos con columna a predecir, sublínea J3250, agregación mensual.

Para la búsqueda de hiperparámetros se realizó una búsqueda de ellos ocupando el algoritmo **HalvingGridSearch** implementado por la líbreria **Scikit-Learn**.

Cabe destacar que sobre estos algoritmos se aplicó tanto el experimento de temporalidad como el de concentración.

4.1.4. Métodos de aprendizaje profundo

En esta sección se aborda el procesamiento realizado para implementar los algoritmos **Multi-Layer Perceptron (MLP)** y **Redes Recurrentes (RR)**

Para la implmentación de los algoritmos **MLP** fue aplicado el mismo procesamiento sobre los datos originales que se mostró para los algoritmos basados en árboles. Con respecto a las arquitecturas empleadas, se decidió aplicar dos tipos distintos:

- *Multi Layer Perceptron with 1 Dense Hidden Layer (MLP 1L)* : Esta arquitectura contempla la implementación de solamente **una capa oculta** que cuente con un **gran número de neuronas** en la capa escondida.
- *Multi Layer Perceptron with Multiple Hidden Layers (MLP ML)*: Esta arquitectura contempla la implementación de **multiples capas ocultas** que cuenten con un **número menor de neuronas por capa**.

Por otra parte, para los algoritmos **RR** se realiza un procesamiento especial, pues se debe modificar la estructura de datos con la cual se trabaja, pasando de un *dataframe* a un *tensor*. Este cambio se realiza pues la estructura de tensor tiene propiedades que permiten mejorar el costo computacional de la implementación. Es importante señalar, que este cambio no agrega nuevas variables ni datos adicionales.

Con respecto a las arquitecturas de redes recurrentes, se ocupó:

- *Vanilla Recurrent Neural Network* (RNN) : Arquitectura de redes recurrentes sencilla, tal como fue explicada en Capítulo 2.
- *Long Short-Term Memory* (LSTM) : Arquitectura de redes recurrentes que incorpora tres tipos de *compuerta*: entrada, salida y memoria.
- *Gated Recurrent Unit* (GRU) : Arquitectura de redes recurrentes que incorpora dos tipos de *compuerta*: reinicio y actualización.

Para la búsuqeda de configuración de párametros[1] óptima para las distintas arquitecturas se utilizó la líbreria **Ax-Platform**.

Finalmente, cabe mencionar que a las arquitecturas **MLP** se le aplicaron tanto el experimento de temporalidad como el de concentración, mientras que a las arquitecturas **RR** solo se le aplicó el de temporalidad.

4.2. Predicciones

En esta sección se explora los resultados al evaluar las predicciones hechas para cada modelo. Para facilitar los análisis posteriores se han separado en dos incisos diferentes tanto las predicciones mensuales como las diarias. Además, cada inciso tiene además, en su interior una desagregación de los resultados, mostrando el rendimiento de los modelos tanto de manera general sobre las 128 sublíneas, como de manera desagregada sobre cada uno de los segmentos de series de tiempo.

Los modelos se presentan ordenados de mejor a peor, considerando principalmente las métricas **WMAPE** y **MAPE**, pues son las que indican de mejor manera la robustez de las predicciones realizadas.

Además, en cada tabla se presenta el modelo **BASELINE** el cual corresponde a las predicciones realizadas por la empresa para el conjunto de testeo. Como ya se comentó estas predicciones fueron realizadas utilizando el modelo **LGBM**, pero no se conoce cuál es el *feature engineering* que aplican actualmente sobre los datos, por lo cual este modelo sólo quedará como una comparación con la cual medir el impacto de los modelos propuestos.

[1]Esto incluye: número de capas, número de neuronas por capa, tasa de aprendizaje y tasa de *dropout*

4.2.1. Predicción mensual

General

Al observar los resultados de las predicciones mensuales de manera agregada (tabla 4.6) se observa que el algoritmo que mejor desempeño tuvo fue **Light Gradient Boosting Machine** aplicando el experimento de *temporalidad*[2]. Este modelo mejora el rendimiento del baseline en 1,0 % con respecto a la métrica WMAPE.

Se observa, a su vez, que el modelo **Extreme Gradient Boosting** también se evalua mejor que al baseline, pues pese a tener un WMAPE 0,2 % peor, la desviación estándar del modelo es 2,5 % mejor que la del modelo de comparación, lo que nos habla de mejores predicciones en general.

Como punto negativo, se debe mencionar al modelo **SARIMAX** que es el que presenta peor desempeño de todos los modelos, teniendo un WMAPE de 180,2 % con desviación estándar 205,5 %. Lo que indica que el algoritmo no es capaz de realizar predicciones adecuadas con este nivel de granularidad.

Modelo	**WMAPE**	**MAPE**	**MAE**
$\text{LGBM}_{PANDEMIA}$	45,1 % ± 40,8 %	54,3 % ± 80,8 %	4.299 ± 5.298
$\text{XGB}_{HIBRIDO}$	46,3 % ± 32,1 %	54,9 % ± 63,5 %	4.452 ± 5.138
BASELINE	46,1 % ± 34,6 %	53,4 % ± 86,7 %	4.576 ± 6.070
MLP ML_{FOCUS}	47,6 % ± 27,8 %	53,2 % ± 50,2 %	4.711 ± 4.955
$\text{RF}_{PANDEMIA}$	48,1 % ± 35,2 %	55,4 % ± 55,1 %	4.429 ± 4.981
MLP 1L_{FOCUS}	49,6 % ± 44,9 %	63,8 % ± 100,3 %	4.316 ± 5.211
$\text{RNN}_{PANDEMIA}$	51,8 % ± 128,5 %	59,4 % ± 222,1 %	4.622 ± 6.436
HW	58,5 % ± 96,5 %	77,1 % ± 170,2 %	3.918 ± 4.407
$\text{LSTM}_{PANDEMIA}$	63,6 % ± 145,7 %	80,8 % ± 241,7 %	4.173 ± 5.258
$\text{SARIMA}_{PANDEMIA}$	72,3 % ± 80,8 %	90,4 % ± 126,6 %	5.965 ± 68.922
MA	77,7 % ± 84,5 %	101,9 % ± 166,0 %	5.953 ± 6.544
$\text{ARIMA}_{PANDEMIA}$	90,2 % ± 102,9 %	117,0 % ± 154,8 %	8.392 ± 12.977
$\text{GRU}_{PANDEMIA}$	92,9 % ± 555,5 %	134,6 % ± 949,3 %	3.762 ± 4.920
NAIVE	95,9 % ± 84,5 %	133,3 % ± 220,4 %	10.723 ± 14.350
SARIMAX	180,2 % ± 205,5 %	249,5 % ± 384,2 %	15.444 ± 20.271

Tabla 4.6: Resultados de robustez agregación general, mensual.

[2]Solo ocupando datos desde el comienzo de la pandemia.

Rotación Normal

Al observar los resultados de las predicciones mensuales en el cluster de rotación normal (tabla 4.7) se observa que el algoritmo que mejor desempeño tuvo fue **Vanilla Recurrent Neural Network** aplicando el experimento de *temporalidad*. Este modelo mejora el rendimiento del baseline en 4,8 % con respecto a la métrica WMAPE y un 5,6 % respecto a la métrica MAPE.

Se observa, a su vez, que los modelos **Multi Layer Perceptron with Multiple Hidden Layers** y *Holt-Winters* también se evaluan mejor que al baseline, pues mejoran el WMAPE en un 1,0 % y 0,8 % respectivamente.

Como punto negativo, se debe mencionar al modelo **SARIMAX** se mantiene como el de peor desempeño de todos los modelos, teniendo un WMAPE de 158,3 % con desviación estándar 218,9 %. Lo que indica que el algoritmo no es capaz de realizar predicciones adecuadas en este nivel de agregación.

Modelo	**WMAPE**	**MAPE**	**MAE**
$\text{RNN}_{PANDEMIA}$	38,4 % ± 21,7 %	36,9 % ± 22,8 %	2.266 ± 3.648
MLP ML_{FOCUS}	42,2 % ± 24,7 %	42,7 % ± 24,4 %	2.156 ± 3.348
HW	42,4 % ± 32,2 %	44,7 % ± 36,2 %	2.250 ± 3.306
BASELINE	43,2 % ± 18,0 %	42,5 % ± 17,5 %	2.549 ± 3.672
GRU	43,3 % ± 20,6 %	40,9 % ± 20,1 %	2.693 ± 3.947
$\text{LGBM}_{HIBRIDO}$	45,1 % ± 25,9 %	45,2 % ± 27,6 %	2.301 ± 3.201
MLP 1L_{FOCUS}	46,4 % ± 36,0 %	49,0 % ± 43,2 %	2.392 ± 3.665
LSTM	47,6 % ± 24,9 %	45,8 % ± 26,8 %	2.915 ± 4.198
$\text{XGB}_{HIBRIDO}$	48,3 % ± 33,2 %	48,3 % ± 34,0 %	2.331 ± 2.880
$\text{RF}_{PANDEMIA}$	52,5 % ± 41,0 %	52,5 % ± 41,9 %	2.362 ± 3.178
$\text{SARIMA}_{PANDEMIA}$	69,6 % ± 78,1 %	78,9 % ± 101,7 %	3.184 ± 41.534
MA	79,5 % ± 67,3 %	88,1 % ± 82,8 %	3.175 ± 3.476
NAIVE	87,6 % ± 110,6 %	106,3 % ± 137,6 %	3.432 ± 3.704
$\text{ARIMA}_{PANDEMIA}$	94,6 % ± 430,2 %	112,3 % ± 165,3 %	4.069 ± 6.157
SARIMAX	158,3 % ± 218,9 %	180,6 % ± 264,9 %	6.007 ± 7.442

Tabla 4.7: Resultados de robustez cluster rotación normal, mensual.

Rotación Variable

Al observar los resultados de las predicciones mensuales en el cluster de rotación variable (tabla 4.8) se observa que el algoritmo que mejor desempeño tuvo fue **Vanilla Recurrent Neural Network** aplicando el experimento de *temporalidad*. Este modelo mejora el rendimiento del baseline en 1,6 % con respecto a la métrica WMAPE y un 1,5 % respecto a la métrica MAPE.

Se observa, a su vez, que ninguno de los otros modelos presentados obtienen mejor desempeño que el modelo de comparación. Lo cual indica que este es uno de los clusters en que el modelo base mejor desempeño obtiene.

Nuevamente se debe mencionar al modelo **SARIMAX** se mantiene como el de peor desempeño de todos los modelos, teniendo un WMAPE de 189,9 % con desviación estándar 162,3 %. Lo que indica que el algoritmo no es capaz de realizar predicciones adecuadas en este nivel de agregación.

Modelo	**WMAPE**	**MAPE**	**MAE**
$\text{RNN}_{PANDEMIA}$	30,8 % ± 13,6 %	29,7 % ± 13,3 %	5.141 ± 4.158
BASELINE	32,4 % ± 15,5 %	31,2 % ± 15,8 %	5.305 ± 3.930
HW	33,1 % ± 12,9 %	34,3 % ± 14,4 %	5.144 ± 3.369
$\text{GRU}_{PANDEMIA}$	32,7 % ± 21,5 %	34,9 % ± 24,3 %	4.382 ± 3.162
$\text{LGBM}_{PANDEMIA}$	36,0 % ± 20,9 %	36,8 % ± 25,2 %	5.049 ± 2.715
LSTM	37,6 % ± 17,7 %	35,8 % ± 16,8 %	6.578 ± 5.483
MLP 1L_{FOCUS}	38,0 % ± 25,7 %	39,9 % ± 32,6 %	5.140 ± 3.346
$\text{XGB}_{HIBRIDO}$	40,1 % ± 23,4 %	43,6 % ± 34,6 %	5.287 ± 2.544
$\text{RF}_{HIBRIDO}$	41,6 % ± 28,1 %	46,1 % ± 38,8 %	5.280 ± 2.447
MLP ML	58,5 % ± 23,3 %	59,2 % ± 26,9 %	8.057 ± 3.560
MA	68,2 % ± 59,8 %	76,9 % ± 74,0 %	7.658 ± 3.666
$\text{SARIMA}_{PANDEMIA}$	72,2 % ± 69,5 %	79,3 % ± 79,2 %	7.866 ± 42.957
$\text{ARIMA}_{PANDEMIA}$	92,9 % ± 99,4 %	108,0 % ± 113,9 %	9.770 ± 6.172
NAIVE	112,0 % ± 70,1 %	127,9 % ± 82,4 %	15.871 ± 10.783
SARIMAX	189,9 % ± 162,3 %	217,7 % ± 188,1 %	22.241 ± 13.751

Tabla 4.8: Resultados de robustez cluster rotación variable, mensual.

Baja rotación

Al observar los resultados de las predicciones mensuales en el cluster de baja rotación (tabla 4.9) se observa que el algoritmo que mejor desempeño tuvo fue **Vanilla Recurrent Neural Network** aplicando el experimento de *temporalidad*. Este modelo mejora el rendimiento del baseline en 15,9 % con respecto a la métrica WMAPE y un 23,9 % respecto a la métrica MAPE.

Se observa, a su vez, que 7 de los modelos presentados presentan mejor desempeño que el modelo de comparación. Lo cual indica que este es uno de los clusters en que el modelo base, peor desempeño obtiene.

Nuevamente se debe mencionar al modelo **SARIMAX** se mantiene como el de peor desempeño de todos los modelos, teniendo un WMAPE de 92,4 % con desviación estándar 63,2 %.

Modelo	**WMAPE**	**MAPE**	**MAE**
$\text{RNN}_{PANDEMIA}$	43,0 % ± 23,0 %	45,4 % ± 26,5 %	1.694 ± 2.285
$\text{GRU}_{PANDEMIA}$	45,8 % ± 22,1 %	52,6 % ± 33,7 %	1.843 ± 2.816
MLP ML_{FOCUS}	47,1 % ± 25,5 %	52,4 % ± 42,2 %	1.748 ± 2.429
$\text{LGBM}_{PANDEMIA}$	47,1 % ± 27,6 %	50,9 % ± 39,5 %	1.652 ± 2.480
$\text{RF}_{PANDEMIA}$	49,8 % ± 29,3 %	61,9 % ± 58,3 %	1.693 ± 2.475
$\text{XGB}_{PANDEMIA}$	51,7 % ± 23,0 %	58,3 % ± 34,7 %	1.652 ± 2.201
$\text{SARIMA}_{PANDEMIA}$	57,0 % ± 20,2 %	71,7 % ± 39,6 %	2.006 ± 27.942
$\text{LSTM}_{PANDEMIA}$	51,7 % ± 22,6 %	56,7 % ± 30,6 %	2.135 ± 3.421
BASELINE	58,9 % ± 30,5 %	69,3 % ± 44,9 %	1.889 ± 2.493
MLP 1L	60,2 % ± 39,2 %	72,9 % ± 68,5 %	1.864 ± 2.430
MA	60,6 % ± 62,2 %	75,8 % ± 70,6 %	2.202 ± 3.365
$\text{ARIMA}_{PANDEMIA}$	61,7 % ± 37,3 %	75,2 % ± 56,4 %	2.073 ± 2.712
NAIVE	63,9 % ± 33,2 %	84,1 % ± 74,3 %	2.139 ± 2.855
HW	68,5 % ± 53,2 %	79,7 % ± 70,7 %	2.295 ± 4.035
$\text{SARIMAX}_{PANDEMIA}$	92,4 % ± 63,2 %	113,7 % ± 107,3 %	2.331 ± 2.504

Tabla 4.9: Resultados de robustez cluster baja rotación, mensual.

Alta rotación

Al observar los resultados de las predicciones mensuales en el cluster de alta rotación (tabla 4.10) se observa que el algoritmo que mejor desempeño tuvo fue **Holt-Winters**. Este modelo mejora el rendimiento del baseline en 5,1 % con respecto a la métrica WMAPE y un 4,2 % respecto a la métrica MAPE.

Se observa, a su vez, que 5 de los modelos presentados presentan mejor desempeño que el modelo de comparación.

Nuevamente el modelo **SARIMAX** se mantiene como el de peor desempeño de todos los modelos, teniendo un WMAPE de 111,5 % con desviación estándar 86,1 %. Lo que indica que el algoritmo no es capaz de realizar predicciones adecuadas en este nivel de agregación.

Modelo	**WMAPE**	**MAPE**	**MAE**
HW	25,1 % ± 11,6 %	24,8 % ± 11,6 %	10.313 ± 4.887
$\text{GRU}_{PANDEMIA}$	26,0 % ± 13,9 %	25,4 % ± 12,6 %	11.167 ± 8.198
$\text{LSTM}_{PANDEMIA}$	27,9 % ± 16,1 %	27,3 % ± 15,1 %	11.856 ± 8.516
LGBM_{FOCUS}	28,7 % ± 10,2 %	26,8 % ± 10,9 %	11.832 ± 4.331
MLP ML_{FOCUS}	30,4 % ± 11,7 %	29,0 % ± 12,5 %	12.705 ± 6.017
BASELINE	30,2 % ± 13,5 %	29,1 % ± 12,8 %	14.425 ± 10.306
MLP $\text{1L}_{PANDEMIA}$	32,1 % ± 12,3 %	33,5 % ± 12,9 %	13.408 ± 6.452
$\text{RF}_{PANDEMIA}$	34,6 % ± 15,3 %	35,5 % ± 15,7 %	13.651 ± 5.735
$\text{RNN}_{PANDEMIA}$	36,8 % ± 15,3 %	33,2 % ± 12,8 %	16.383 ± 9.667
$\text{XGB}_{PANDEMIA}$	36,1 % ± 16,7 %	34,8 % ± 18,4 %	14.821 ± 5.939
MA	46,0 % ± 23,2 %	52,4 % ± 26,6 %	18.212 ± 8.289
$\text{SARIMA}_{PANDEMIA}$	49,4 % ± 46,0 %	55,2 % ± 50,7 %	17.656 ± 95.318
ARIMA	76,2 % ± 49,7 %	92,0 % ± 53,6 %	30.104 ± 17.968
NAIVE	93,0 % ± 47,0 %	110,6 % ± 49,0 %	38.045 ± 17.517
SARIMAX	111,5 % ± 86,1 %	127,3 % ± 89,7 %	41.982 ± 28.859

Tabla 4.10: Resultados de robustez cluster alta rotación, mensual.

Rotación Itinerante

Al observar los resultados de las predicciones mensuales en el cluster de rotación itinerante (tabla 4.11) se observa que el algoritmo que mejor desempeño tuvo fue **Random Forest** aplicando el experimento *mixto*[3]. Este modelo mejora el rendimiento del baseline en 52,0 % con respecto a la métrica WMAPE y un 96,5 % respecto a la métrica MAPE.

Se observa, a su vez, que 5 de los modelos presentados presentan mejor desempeño que el modelo de comparación. Y que este obtiene su peor rendimiento en este cluster.

Como punto negativo, se debe mencionar al modelo **Gated Recurrent Unit** como el de peor desempeño de todos los modelos, teniendo un WMAPE de 900,0 % con desviación estándar 2.189,3 % siendo ambos valores los peores registros de todos los experimentos. Esto es interesante, pues significa que la posición que ocupa este algoritmo en la agregación general está perjudicado por el pobre desempeño que tiene en este cluster.

Modelo	**WMAPE**	**MAPE**	**MAE**
$RF_{HIBRIDO}$	59,6 % ± 20,8 %	118,5 % ± 89,1 %	179 ± 234
$XGB_{PANDEMIA}$	64,6 % ± 25,7 %	135,4 % ± 99,6 %	102 ± 198
MLP ML_{FOCUS}	66,1 % ± 24,9 %	127,8 % ± 137,3 %	175 ± 414
$LGBM_{FOCUS}$	107,8 % ± 38,6 %	241,6 % ± 274,1 %	108 ± 205
MLP $1L_{FOCUS}$	111,8 % ± 64,2 %	267,4 % ± 249,7 %	217 ± 478
BASELINE	111,6 % ± 90,2 %	215,0 % ± 303,1 %	268 ± 587
$SARIMA_{PANDEMIA}$	172,8 % ± 191,1 %	326,9 % ± 339,0 %	101 ± 1.623
NAIVE	176,0 % ± 103,5 %	513,5 % ± 729,6 %	179 ± 234
$ARIMA_{PANDEMIA}$	180,8 % ± 133,6 %	381,5 % ± 312,6 %	124 ± 161
$SARIMAX_{PANDEMIA}$	215,7 % ± 180,3 %	421,4 % ± 409,3 %	139 ± 168
MA	228,3 % ± 224,2 %	475,5 % ± 492,9 %	220 ± 408
$RNN_{PANDEMIA}$	271,8 % ± 484,0 %	408,7 % ± 859,6 %	228 ± 537
HW	316,3 % ± 263,7 %	569,1 % ± 450,1 %	173 ± 189
$LSTM_{PANDEMIA}$	338,6 % ± 524,6 %	569,3 % ± 861,1 %	218 ± 470
$GRU_{PANDEMIA}$	900,0 % ± 2.189,3 %	1.530,5 % ± 3.735,9 %	203 ± 427

Tabla 4.11: Resultados de robustez cluster rotación itinerante, mensual.

[3]Aplicando los experimentos de temporalidad y concentración.

Rotación Potenciada

Al observar los resultados de las predicciones mensuales en el cluster de rotación potenciada (tabla 4.12) se observa que el algoritmo que mejor desempeño tuvo fue **Random Forest** aplicando el experimento de *concentración*[4]. Este modelo mejora el rendimiento del baseline en 30,0 % con respecto a la métrica WMAPE y un 24,5 % respecto a la métrica MAPE.

Se observa, a su vez, que 11 de los modelos presentados presentan mejor desempeño que el modelo de comparación. Siendo este cluster en el que peor posicionado queda el modelo usado como comparación.

Como punto negativo, se debe mencionar nuevamente al modelo **SARIMAX** como el de peor desempeño de todos los modelos, teniendo un WMAPE de 80,9 % con desviación estándar 65,5 % siendo estas las mejores métricas que el algoritmo logra obtener en un cluster.

Modelo	**WMAPE**	**MAPE**	**MAE**
RF_{FOCUS}	28.7 % ± 14.3 %	32.2 % ± 17.3 %	2.457 ± 1.395
$LGBM_{PANDEMIA}$	29,0 % ± 10,0 %	31,8 % ± 11,7 %	2.669 ± 1.317
$LSTM_{PANDEMIA}$	30,1 % ± 10,4 %	31,8 % ± 12,5 %	2.859 ± 1.681
$GRU_{PANDEMIA}$	31,9 % ± 9,8 %	34,2 % ± 14,0 %	3.092 ± 1.953
HW	31,9 % ± 15,3 %	31,4 % ± 15,6 %	2.930 ± 1.324
$XGB_{HIBRIDO}$	32,8 % ± 17,2 %	36,0 % ± 22,5 %	2.913 ± 1.838
MLP $1L_{PANDEMIA}$	33,2 % ± 19,2 %	35,8 % ± 19,9 %	2.955 ± 2.077
MLP ML_{FOCUS}	41,1 % ± 8,7 %	40,9 % ± 10,6 %	4.017 ± 2.543
$RNN_{PANDEMIA}$	47,1 % ± 14,2 %	46,4 % ± 13,5 %	4.604 ± 2.742
$ARIMA_{PANDEMIA}$	49,6 % ± 27,9 %	55,1 % ± 34,8 %	4.464 ± 3.148
MA	57,3 % ± 31,1 %	64,5 % ± 35,6 %	5.129 ± 2.944
BASELINE	58,7 % ± 26,1 %	56,7 % ± 27,6 %	4.812 ± 2.045
NAIVE	65,8 % ± 30,2 %	73,9 % ± 34,4 %	6.453 ± 5.390
$SARIMA_{PANDEMIA}$	66,4 % ± 59,8 %	73,7 % ± 72,2 %	6.280 ± 52.880
$SARIMAX_{PANDEMIA}$	80,9 % ± 65,5 %	93,9 % ± 86,2 %	8.087 ± 6.801

Tabla 4.12: Resultados de robustez cluster rotación potenciada, mensual.

[4]Ajustando un modelo por cluster.

4.2.2. Predicción diario

General

Al observar los resultados de las predicciones diarias de manera general (tabla 4.13) se observa que el algoritmo que mejor desempeño tuvo fue **Multi Layer Perceptron with 1 Dense Hidden Layer** aplicando el experimento de *temporalidad*. Este modelo mejora el rendimiento del baseline en 11,3 % con respecto a la métrica WMAPE y un 11,5 % respecto a la métrica MAPE.

Se observa, a su vez, que 9 de los modelos presentados presentan mejor desempeño que el modelo de comparación. Lo que indica que el modelo de comparación no logra un buen desempeño general en este nivel de predicciones.

Como punto negativo, se debe mencionar al modelo **Vanilla Recurrent Neural Network** como el de peor desempeño de todos los modelos, teniendo un WMAPE de 70,6 % con desviación estándar 48,1 %. Lo que indica que el algoritmo no es capaz de realizar predicciones adecuadas con este nivel de granularidad. Lo cual es interesante, pues pese a ser el que mejor captura las tendencias mensuales, no logra igual desempeño al cambiar la granularidad.

Modelo	**WMAPE**	**MAPE**	**MAE**
MLP $1L_{PANDEMIA}$	36,6 % ± 14,4 %	37,2 % ± 39,9 %	171 ± 248
XGB_{FOCUS}	37,5 % ± 14,6 %	35,4 % ± 30,1 %	166 ± 218
$LGBM_{HIBRIDO}$	38,4 % ± 16,9 %	36,9 % ± 47,2 %	171 ± 226
$RF_{HIBRIDO}$	41,5 % ± 19,0 %	39,0 % ± 52,5 %	190 ± 259
MLP $ML_{PANDEMIA}$	42,9 % ± 13,2 %	40,3 % ± 40,7 %	209 ± 304
ARIMA	45,4 % ± 19,4 %	44,9 % ± 61,6 %	211 ± 281
SARIMA	45,6 % ± 19,4 %	45,2 % ± 61,7 %	211 ± 281
SARIMAX	45,6 % ± 19,9 %	45,1 % ± 62,7 %	212 ± 282
LSTM	47,3 % ± 20,6 %	50,1 % ± 44,9 %	213 ± 284
BASELINE	47,9 % ± 22,8 %	48,7 % ± 37,8 %	192 ± 250
HW	49,5 % ± 35,6 %	55,3 % ± 107,9 %	217 ± 281
GRU	54,2 % ± 15,2 %	51,7 % ± 35,6 %	264 ± 351
NAIVE	54,4 % ± 45,3 %	61,4 % ± 112,2 %	233 ± 304
MA	64,3 % ± 54,4 %	80,2 % ± 127,4 %	246 ± 309
$RNN_{PANDEMIA}$	70,6 % ± 48,1 %	91,8 % ± 80,5 %	290 ± 369

Tabla 4.13: Resultados de robustez agregación general, diario.

Rotación Normal

Al observar los resultados de las predicciones diarias en el cluster de rotación normal (tabla 4.14) se observa que el algoritmo que mejor desempeño tuvo fue **Multi Layer Perceptron with 1 Dense Hidden Layer** aplicando el experimento de *temporalidad*. Este modelo mejora el rendimiento del baseline en 9,1 % con respecto a la métrica WMAPE y un 3,8 % respecto a la métrica MAPE.

Se observa, a su vez, que 6 de los modelos presentados presentan mejor desempeño que el modelo de comparación. Siendo el segundo de mejor desempeño el modelo **Extreme Gradient Boosting** que mejora el modelo base en 9,3 % con respecto a la métrica WMAPE y un 6,7 % respecto a la métrica MAPE.

Como punto negativo, se debe mencionar nuevamente al modelo **Vanilla Recurrent Neural Network** como el de peor desempeño de todos los modelos, teniendo un WMAPE de 71,1 % con desviación estándar 38,3 %. Lo que indica que el algoritmo no es capaz de realizar predicciones adecuadas en este cluster.

Modelo	**WMAPE**	**MAPE**	**MAE**
MLP $1L_{PANDEMIA}$	36,3 % ± 11,1 %	40,0 % ± 48,7 %	68 ± 62
$XGB_{HIBRIDO}$	36,1 % ± 15,7 %	37,1 % ± 41,4 %	62 ± 50
$LGBM_{HIBRIDO}$	38,9 % ± 21,3 %	43,6 % ± 76,1 %	68 ± 61
MLP $ML_{PANDEMIA}$	41,6 % ± 11,8 %	42,9 % ± 57,6 %	76 ± 65
LSTM	41,4 % ± 12,4 %	43,8 % ± 30,7 %	72 ± 54
$RF_{HIBRIDO}$	42,3 % ± 24,9 %	46,1 % ± 85,2 %	77 ± 74
BASELINE	45,4 % ± 16,0 %	43,8 % ± 23,8 %	80 ± 64
$GRU_{PANDEMIA}$	45,9 % ± 14,8 %	51,4 % ± 37,8 %	78 ± 62
ARIMA	46,3 % ± 24,9 %	52,5 % ± 99,6 %	83 ± 74
SARIMA	46,9 % ± 25,0 %	53,4 % ± 99,6 %	84 ± 73
SARIMAX	47,2 % ± 26,0 %	53,7 % ± 101,4 %	84 ± 75
HW	53,1 % ± 55,1 %	69,2 % ± 178,7 %	83 ± 65
NAIVE	61,1 % ± 70,9 %	79,1 % ± 183,1 %	90 ± 69
MA	63,1 % ± 55,3 %	83,5 % ± 165,2 %	107 ± 110
$RNN_{PANDEMIA}$	71,1 % ± 38,3 %	92,2 % ± 61,9 %	119 ± 97

Tabla 4.14: Resultados de robustez cluster rotación normal, diario.

Rotación Variable

Al observar los resultados de las predicciones diarias en el cluster de rotación variable (tabla 4.15) se observa que el algoritmo que mejor desempeño tuvo fue **Multi Layer Perceptron with 1 Dense Hidden Layer** aplicando el experimento de *concentración*. Este modelo mejora el rendimiento del baseline en 5,3 % con respecto a la métrica WMAPE y un 6,3 % respecto a la métrica MAPE.

Se observa, a su vez, que además del algoritmo ya señalado, los modelos basados en árboles también son capaces de mejorar el desempeño del modelo baseline tanto en la métrica WMAPE como en el MAPE, aunque por un margen menor al logrado por el modelo **MLP 1L**.

Como punto negativo, se debe mencionar nuevamente al modelo **Moving Average** como el de peor desempeño de todos los modelos, teniendo un WMAPE de 68,3 % con desviación estándar 53,4 %. Lo que indica que el algoritmo no es capaz de realizar predicciones adecuadas en este cluster.

Modelo	**WMAPE**	**MAPE**	**MAE**
MLP $1L_{FOCUS}$	32,8 % ± 8,7 %	28,5 % ± 8,1 %	217 ± 174
$XGB_{HIBRIDO}$	33,7 % ± 8,2 %	31,1 % ± 11,0 %	205 ± 133
$LGBM_{HIBRIDO}$	36,2 % ± 10,6 %	32,6 % ± 15,8 %	228 ± 175
$RF_{HIBRIDO}$	38,1 % ± 11,2 %	32,9 % ± 16,9 %	244 ± 188
BASELINE	38,1 % ± 13,2 %	34,8 % ± 13,6 %	245 ± 202
MLP $ML_{HIBRIDO}$	41,2 % ± 7,6 %	32,8 % ± 8,0 %	265 ± 185
$ARIMA_{PANDEMIA}$	42,3 % ± 10,3 %	35,9 % ± 15,7 %	268 ± 190
SARIMAX	42,7 % ± 11,3 %	37,7 % ± 17,9 %	269 ± 194
SARIMA	43,0 % ± 11,3 %	38,3 % ± 18,0 %	271 ± 194
GRU	51,0 % ± 11,0 %	42,9 % ± 12,0 %	341 ± 242
LSTM	44,5 % ± 11,4 %	42,6 % ± 18,0 %	284 ± 198
HW	48,8 % ± 17,6 %	52,0 % ± 33,8 %	292 ± 191
NAIVE	52,5 % ± 26,0 %	57,5 % ± 47,1 %	318 ± 235
$RNN_{PANDEMIA}$	66,1 % ± 26,1 %	80,0 % ± 46,3 %	381 ± 216
MA	68,3 % ± 53,4 %	78,2 % ± 91,1 %	341 ± 211

Tabla 4.15: Resultados de robustez cluster rotación variable, diario.

Baja rotación

Al observar los resultados de las predicciones diarias en el cluster de baja rotación (tabla 4.16) se observa que el algoritmo que mejor desempeño tuvo fue **Extreme Gradient Boosting** aplicando el experimento de *concentración*. Este modelo mejora el rendimiento del baseline en 20,3 % con respecto a la métrica WMAPE y un 31,2 % respecto a la métrica MAPE. Además, llama la atención que el top 3 de mejores modelos está compuesto exclusivamente por modelos basados en árboles, lo cual indica la facilidad de estos modelos para acomodarse a este cluster.

Se observa, a su vez, que el modelo usado actualmente por la empresa es el de peor rendimiento en este cluster, lo cual mezclado con lo comentado en las predicciones mensuales de este mismo cluster nos permiten intuir que existe un deficiencia consistente para predecir los valores de este cluster.

Como punto negativo, se debe mencionar nuevamente al modelo **Moving Average** como el de peor desempeño de todos los modelos implementados, teniendo un WMAPE de 62,2 % con desviación estándar 31,1 %.

Modelo	**WMAPE**	**MAPE**	**MAE**
XGB_{FOCUS}	43,3 % ± 15,0 %	40,9 % ± 22,3 %	53 ± 74
LGBM_{FOCUS}	44,0 % ± 15,1 %	42,5 % ± 23,5 %	61 ± 95
RF_{FOCUS}	45,1 % ± 16,6 %	43,2 % ± 29,9 %	64 ± 101
$\text{MLP 1L}_{PANDEMIA}$	46,7 % ± 19,5 %	53,4 % ± 51,1 %	70 ± 122
HW	47,6 % ± 19,8 %	45,6 % ± 31,7 %	77 ± 134
$\text{MLP ML}_{PANDEMIA}$	48,7 % ± 17,3 %	55,0 % ± 48,7 %	64 ± 99
$\text{SARIMA}_{PANDEMIA}$	49,5 % ± 19,8 %	47,3 % ± 29,2 %	80 ± 135
$\text{ARIMA}_{PANDEMIA}$	49,8 % ± 19,5 %	47,5 % ± 29,0 %	80 ± 135
SARIMAX	49,5 % ± 19,7 %	48,5 % ± 33,9 %	79 ± 135
NAIVE	51,5 % ± 18,2 %	49,0 % ± 28,0 %	79 ± 130
$\text{LSTM}_{PANDEMIA}$	52,7 % ± 17,6 %	63,8 % ± 47,8 %	60 ± 84
$\text{GRU}_{PANDEMIA}$	54,4 % ± 16,6 %	64,1 % ± 40,1 %	68 ± 102
$\text{RNN}_{PANDEMIA}$	57,0 % ± 21,5 %	67,3 % ± 53,0 %	80 ± 126
MA	62,2 % ± 31,1 %	84,1 % ± 112,7 %	98 ± 173
BASELINE	63,6 % ± 23,1 %	72,1 % ± 41,7 %	74 ± 99

Tabla 4.16: Resultados de robustez cluster baja rotación, diario.

Alta rotación

Al observar los resultados de las predicciones diarias en el cluster de alta rotación (tabla 4.17) se observa que el algoritmo que mejor desempeño tuvo fue **Extreme Gradient Boosting**. Este modelo mejora el rendimiento del baseline en 5,9 % con respecto a la métrica WMAPE y un 5,5 % respecto a la métrica MAPE.

Se observa, a su vez, que sólo los modelos *Light Gradient Boosting Machine* y *Multi Layer Perceptron with 1 Dense Hidden Layer* son capaces de mejorar el rendimiento del modelo base. Siendo este cluster donde el modelo usado actualmente mejor ranking obtiene.

Como punto negativo, se debe mencionar nuevamente al modelo **Vanilla Recurrent Neural Network** como el de peor desempeño de todos los modelos, teniendo un WMAPE de 60,4 % con desviación estándar 24,9 %. Lo que indica que el algoritmo no es capaz de realizar predicciones adecuadas en este cluster.

Modelo	**WMAPE**	**MAPE**	**MAE**
XGB	29,8 % ± 7,1 %	23,1 % ± 4,3 %	534 ± 330
$\text{LGBM}_{HIBRIDO}$	32,1 % ± 8,2 %	24,3 % ± 5,5 %	571 ± 340
$\text{MLP 1L}_{PANDEMIA}$	32,7 % ± 9,4 %	25,3 % ± 6,0 %	598 ± 429
BASELINE	35,7 % ± 9,6 %	28,6 % ± 9,7 %	639 ± 371
$\text{RF}_{HIBRIDO}$	37,1 % ± 8,9 %	28,2 % ± 8,4 %	662 ± 397
HW	40,9 % ± 10,1 %	32,1 % ± 13,3 %	725 ± 425
$\text{GRU}_{PANDEMIA}$	40,2 % ± 15,0 %	37,6 % ± 22,5 %	700 ± 409
$\text{ARIMA}_{PANDEMIA}$	41,8 % ± 9,8 %	31,1 % ± 7,6 %	743 ± 453
SARIMA	41,7 % ± 9,6 %	32,3 % ± 9,5 %	734 ± 421
SARIMAX	42,0 % ± 9,3 %	32,2 % ± 8,4 %	738 ± 421
$\text{MLP ML}_{PANDEMIA}$	42,9 % ± 9,9 %	31,3 % ± 7,1 %	777 ± 511
LSTM	43,2 % ± 12,9 %	42,7 % ± 20,9 %	747 ± 415
NAIVE	44,2 % ± 10,3 %	35,8 % ± 12,1 %	780 ± 440
MA	44,4 % ± 15,2 %	38,5 % ± 22,3 %	771 ± 480
$\text{RNN}_{PANDEMIA}$	60,4 % ± 24,9 %	69,1 % ± 39,3 %	998 ± 526

Tabla 4.17: Resultados de robustez cluster alta rotación, diario.

Rotación Itinerante

Al observar los resultados de las predicciones diarias en el cluster de rotación itinerante (tabla 4.18) se observa que el algoritmo que mejor desempeño tuvo fue **Multi Layer Perceptron with 1 Dense Hidden Layer** aplicando el experimento *mixto*. Este modelo mejora el rendimiento del baseline en 41,7 % con respecto a la métrica WMAPE y un 76,7 % respecto a la métrica MAPE.

Como punto negativo, se debe mencionar al modelo **Moving Average** como el de peor desempeño de todos los modelos, teniendo un WMAPE de 106,6 % con desviación estándar 121,1 %. Lo que indica que el algoritmo no es capaz de realizar predicciones adecuadas en este cluster.

Modelo	**WMAPE**	**MAPE**	**MAE**
MLP $1L_{HIBRIDO}$	38,2 % ± 27,3 %	35,8 % ± 42,3 %	4 ± 7
$LGBM_{FOCUS}$	39,2 % ± 26,1 %	32,2 % ± 29,4 %	4 ± 6
RF	39,2 % ± 25,9 %	34,8 % ± 29,8 %	4 ± 6
XGB	41,7 % ± 26,2 %	39,2 % ± 32,3 %	5 ± 9
MLP $ML_{PANDEMIA}$	41,7 % ± 28,7 %	44,9 % ± 45,5 %	4 ± 8
$SARIMAX_{PANDEMIA}$	44,4 % ± 28,5 %	39,6 % ± 32,3 %	5 ± 10
ARIMA	45,2 % ± 31,7 %	52,7 % ± 57,9 %	4 ± 6
SARIMA	45,2 % ± 31,7 %	52,7 % ± 57,9 %	4 ± 6
HW	52,8 % ± 40,5 %	68,9 % ± 88,9 %	5 ± 7
NAIVE	57,0 % ± 43,8 %	74,5 % ± 95,9 %	5 ± 9
RNN	58,5 % ± 33,6 %	86,0 % ± 92,7 %	5 ± 9
LSTM	61,4 % ± 16,0 %	63,6 % ± 28,5 %	5 ± 8
GRU	72,7 % ± 25,6 %	77,3 % ± 37,9 %	5 ± 9
BASELINE	79,9 % ± 45,2 %	112,5 % ± 89,0 %	8 ± 18
MA	106,6 % ± 121,1 %	171,0 % ± 197,2 %	6 ± 9

Tabla 4.18: Resultados de robustez cluster rotación itinerante, diario.

Rotación Potenciada

Al observar los resultados de las predicciones diarias en el cluster de rotación potenciada (tabla 4.19) se observa que el algoritmo que mejor desempeño tuvo fue **Multi Layer Perceptron with 1 Dense Hidden Layer** aplicando el experimento de *temporalidad*. Este modelo mejora el rendimiento del baseline en 25,7 % con respecto a la métrica WMAPE y un 27,0 % respecto a la métrica MAPE.

Se observa, a su vez, que el modelo usado como comparación es nuevamente el peor de los modelos evaluados.

Como punto negativo de los modelos implementados, se debe mencionar nuevamente al modelo **Vanilla Recurrent Neural Network** como el de peor desempeño, teniendo un WMAPE de 60,2 % con desviación estándar 6,8 %. Lo que indica que el algoritmo no es capaz de realizar predicciones adecuadas en este cluster.

Modelo	**WMAPE**	**MAPE**	**MAE**
MLP $1L_{PANDEMIA}$	34,4 % ± 10,9 %	28,1 % ± 6,5 %	134 ± 95
$XGB_{PANDEMIA}$	35,5 % ± 7,2 %	32,3 % ± 8,7 %	135 ± 81
$LGBM_{HIBRIDO}$	38,9 % ± 9,5 %	31,0 % ± 8,2 %	151 ± 111
$RF_{HIBRIDO}$	38,5 % ± 11,5 %	30,5 % ± 9,9 %	148 ± 104
MLP $ML_{PANDEMIA}$	44,7 % ± 9,8 %	33,6 % ± 6,3 %	171 ± 109
$ARIMA_{PANDEMIA}$	48,2 % ± 10,4 %	38,0 % ± 11,0 %	181 ± 111
$SARIMA_{PANDEMIA}$	48,2 % ± 10,4 %	38,0 % ± 11,0 %	181 ± 111
MA	48,7 % ± 9,7 %	39,6 % ± 14,8 %	185 ± 114
$SARIMAX_{PANDEMIA}$	48,5 % ± 10,2 %	40,6 % ± 10,6 %	183 ± 110
NAIVE	50,5 % ± 11,1 %	46,4 % ± 19,8 %	193 ± 127
HW	50,9 % ± 10,6 %	47,3 % ± 15,3 %	197 ± 128
$GRU_{PANDEMIA}$	52,4 % ± 7,0 %	52,9 % ± 10,8 %	195 ± 115
$LSTM_{PANDEMIA}$	52,4 % ± 11,3 %	54,9 % ± 19,2 %	194 ± 224
$RNN_{PANDEMIA}$	60,2 % ± 6,8 %	68,7 % ± 16,8 %	224 ± 142
BASELINE	60,1 % ± 26,3 %	55,1 % ± 28,1 %	198 ± 107

Tabla 4.19: Resultados de robustez cluster rotación potenciada, diario.

4.3. Análisis de resultados

A continuación se deglosarán los principales resultados de la implementación computacional de los modelos planteados. Para poder guiar de buena manera dichos principales aprendizajes, se muestran a continuación las preguntas que guiaran el relato propuesto.

- ¿Cual familia de modelos tuvo mejor desempeño?
- ¿Los experimentos aplicados tuvieron un impacto en las predicciones?
- De existir impacto ¿Cual experimento tiene mayor efecto sobre los resultados?

Es importante mencionar que este estudio se realizará tomando el desempeño general de los modelos, es decir, como los modelos pudieron hacer las predicciones correspondientes para las 128 sublíneas de estudio, sin diferenciar entre los segmentos identificados.

4.3.1. Análisis familia de modelos

Al comparar las familias de algoritmos probadas utilizando la métrica de referencia **WMAPE** (figura 4.2), se tiene que tanto para el caso mensual como para el caso diario, los algoritmos basados en árboles son los que tienen un mejor desempeño, obteniendo un error promedio de 39,1 % en el primer caso y un 46,5 % en el segundo.

Al comparar dichos resultados con las otras familias de modelos, se tiene que para el caso de las predicciones mensuales los modelos basados en árboles son 14,6 % mejores que los algoritmos de aprendizaje profundo y un 51,4 % mejores que los algoritmos estadísticos.

Para el caso de predicciones diarias, los algoritmos basados en árboles logran mejorar en un 11,2 % a los algoritmos basados en aprendizaje profundo y un 8,0 % a los estadísticos, lo cual es interesante pues nos indica que en promedio, los algoritmos más sencillos son capaces de generalizar mejor que los algoritmos que implica la calibración de un mayor número de parámetros.

Finalmente, respecto a los resultados de las familias de algoritmos, se señala que consistente con lo que se esperaba inicialmente, los modelos simples son los que peor desempeño muestran en los dos casos de estudio.

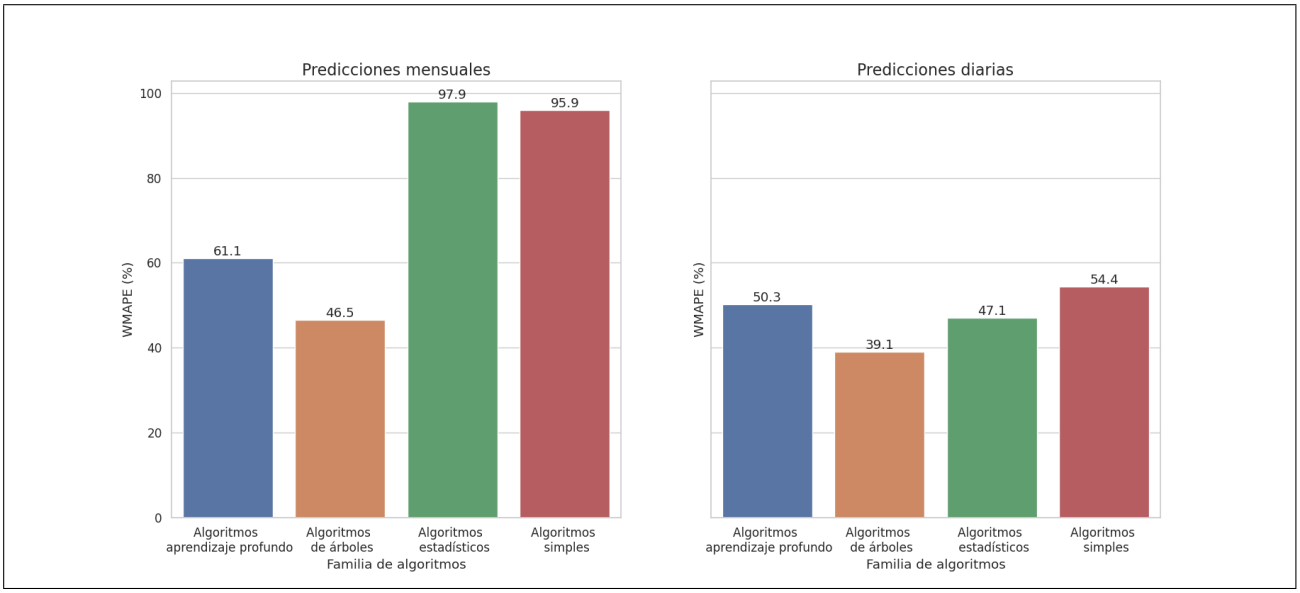

Figura 4.2: Métricas por familia de algoritmos

4.3.2. Análisis experimentos

Para analizar el efecto de los experimentos sobre la calidad de las predicciones, se compara el rendimiento de los modelos sin aplicar ningún experimento con el rendimiento de los modelos habiendo aplicado el experimento en cuestión. En particular, se tomará como indicador el promedio de los WMAPE de los modelos.

Cabe recalcar que los experimentos no fueron aplicados sobre todos los modelos, pues la aplicación de estos no es siempre posible (tabla 4.20).

Modelo/Experimento	Pandemia	Concentración	Híbrido
ARIMA	SI	NO	NO
SARIMA	SI	NO	NO
SARIMAX	SI	NO	NO
RF	SI	SI	SI
LGBM	SI	SI	SI
XGB	SI	SI	SI
MLP	SI	SI	SI
LSTM	SI	NO	NO
GRU	SI	NO	NO
RNN	SI	NO	NO

Tabla 4.20: Resumen aplicación de experimentos.

Experimento de pandemia

Al comparar los modelos en los cuales se aplicó este experimento (figura 4.3), se tiene que tanto para las predicciones mensuales como para las predicciones diarias, la aplicación de este procesamiento mejora, en promedio, el desempeño de los algoritmos. En concreto, las **predicciones mensuales se ven mejoradas en un 25 %** mientras que las **diarias en un 3 %**.

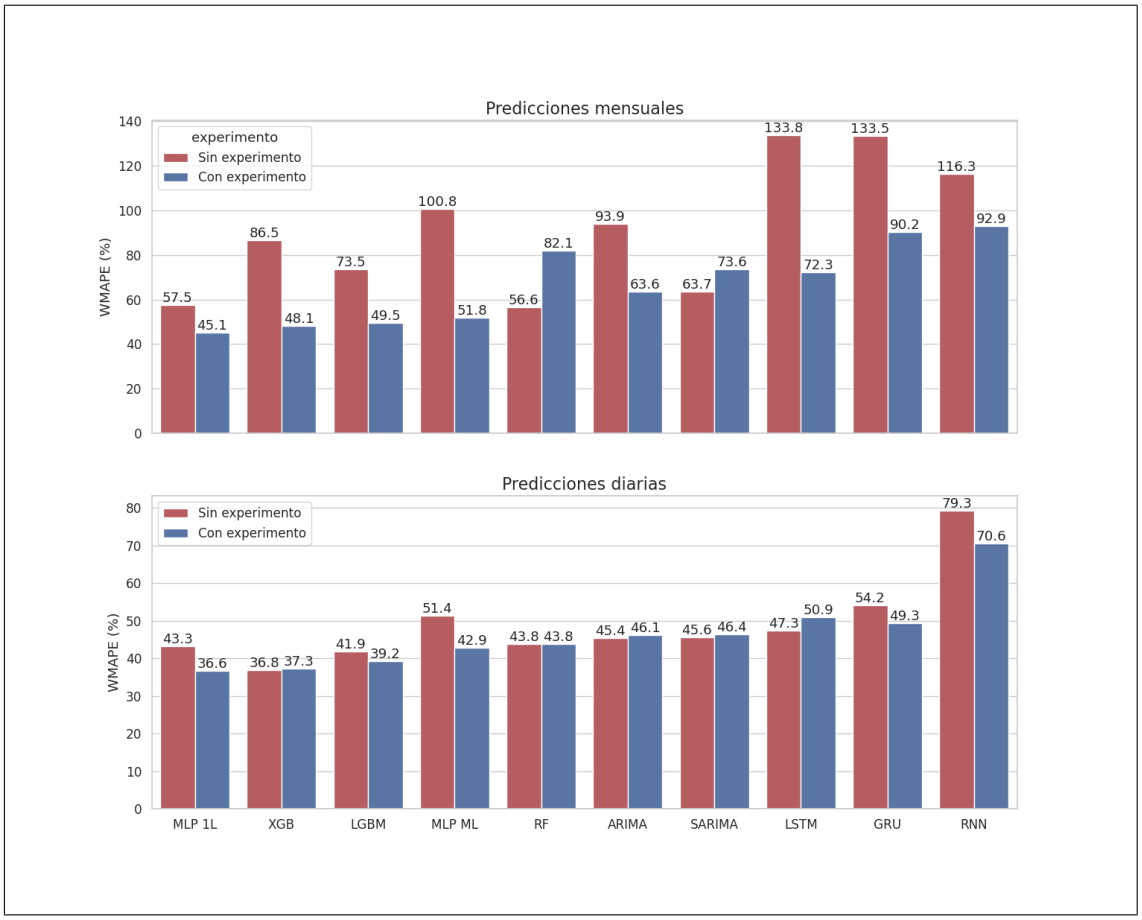

Figura 4.3: Efecto de experimento de pandemia

Lo anterior indica que, a modo general, los modelos logran tener un mejor desempeño cuando son entrenados utilizando la información generada a partir del comienzo de la pandemia en vez de utilizar toda la información disponible.

Pese a lo señalado previamente, se debe destacar que no todos los modelos se ven afectados positivamente en todos los escenarios, pues en el caso de las predicciones mensuales los 2 modelos de *Multi Layer Perceptron* empeoran su desempeño. Al mismo tiempo, en las predicciones diarias, 4 modelos diferentes se ven afectados de manera negativa y 1 es indiferente a la aplicación del mismo.

Experimento de concentración

Al comparar los modelos en los cuales se aplicó este experimento (figura 4.4), se tiene que, en promedio, solo mejora el desempeño de los algoritmos en el caso de las predicciones mensuales, las cuales se ven mejoradas, en promedio, en un 13 %. Por contraparte, para las predicciones diarias, no se observan patrones claramente identificables al aplicar este experimento.

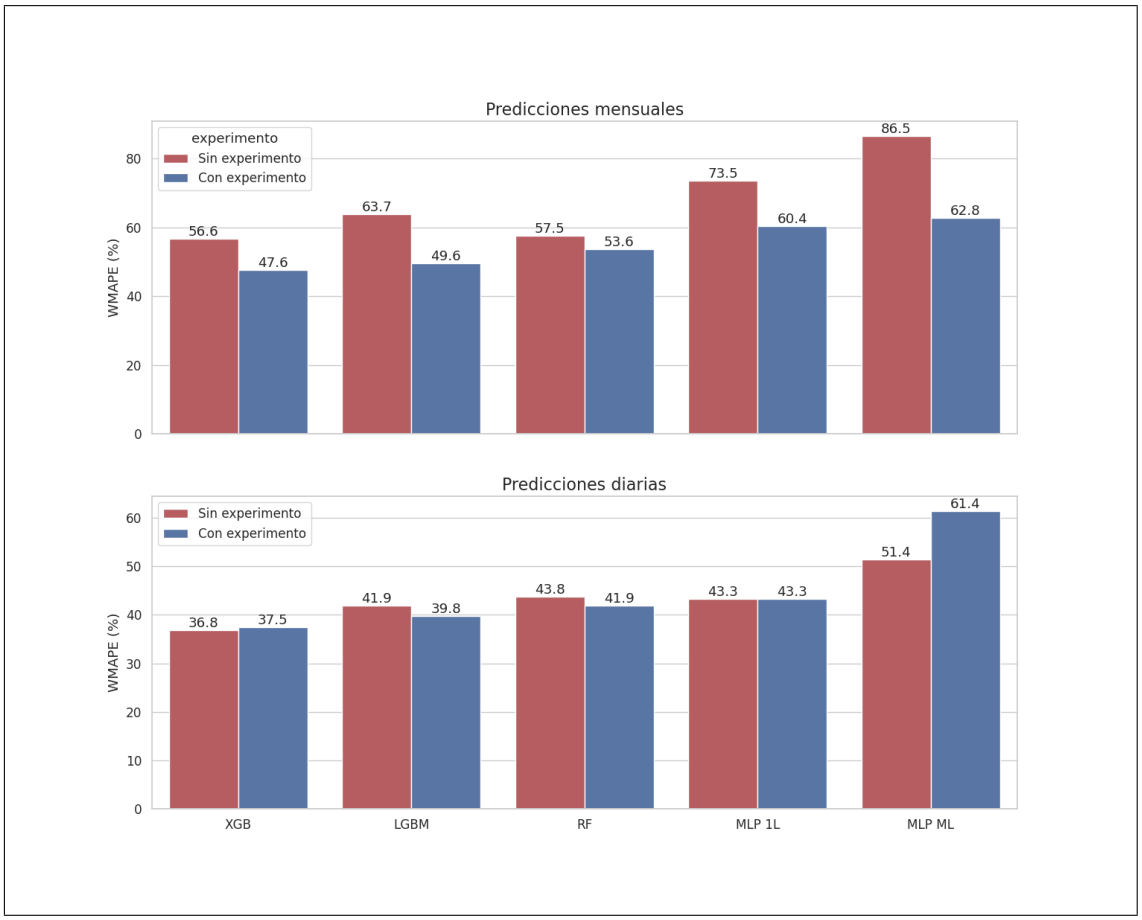

Figura 4.4: Efecto de experimento de concentración

Pese a lo señalado previamente, se debe destacar que los modelos *Ligth Gradient Boosting Machine* y *Random Forest* se ven afectados de manera positiva por este experimento, obteniendo mejoras de performance de 2,1 % y 1,9 % respectivamente.

Experimento híbrido

Al comparar los modelos en los cuales se aplicó este experimento (figura 4.5), se tiene que, en promedio, solo mejora el desempeño de los algoritmos en el caso de las predicciones mensuales, las cuales se ven mejoradas, en promedio, en un 14 %. Por contraparte, para las predicciones diarias, no se observan patrones claramente identificables al aplicar este experimento.

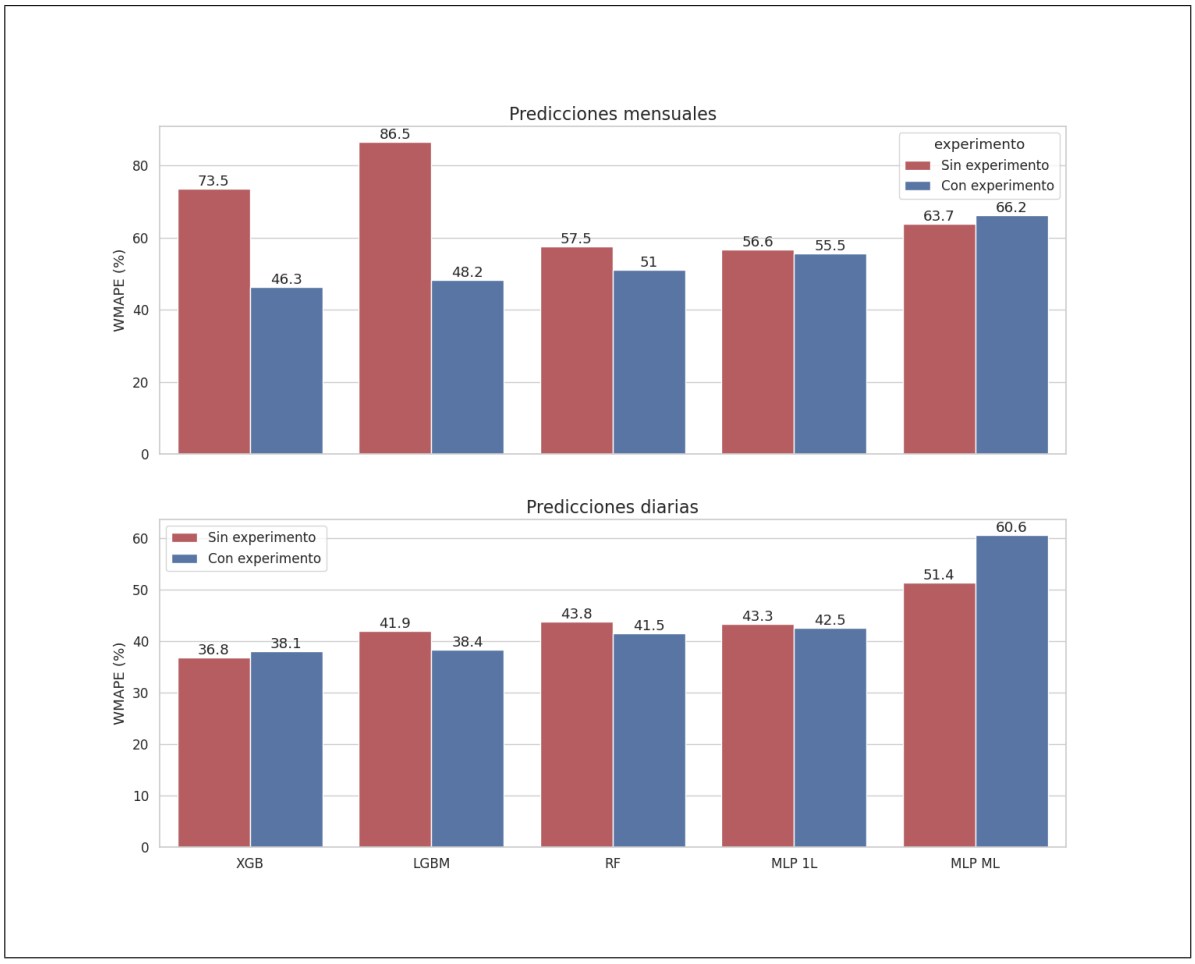

Figura 4.5: Efecto de experimento híbrido

Pese a lo señalado previamente, se debe destacar que los modelos *Ligth Gradient Boosting Machine* y *Random Forest* se ven afectados de manera positiva por este experimento, obteniendo mejoras de performance de 3,5 % y 2,3 % respectivamente.

4.3.3. Comparación de experimentos

Una vez establecido que, en general, los experimentos propuestos mejoran el rendimiento de los modelos, cabe preguntarse cual de los experimentos es más conveniente según el modelo y tipo de predicción que se quiera hacer. Para ello se compara el efecto que tuvo la aplicación de un experimento sobre el caso base, ocupando la métrica WMAPE (tablas 4.21 y 4.22)

Modelo/Experimento	Pandemia	Concentración	Híbrido
ARIMA	−43,3 %	NO APLICA	NO APLICA
SARIMA	−61,5 %	NO APLICA	NO APLICA
SARIMAX	−9,5 %	NO APLICA	NO APLICA
RF	−38,4 %	−23,7 %	−38,3 %
LGBM	−12,4 %	−3,9 %	−6,5 %
XGB	−24,0 %	−13,1 %	−27,2 %
MLP 1L	9,9 %	−14,1 %	2,5 %
MLP ML	25,5 %	−9,0 %	−1,1 %
LSTM	−30,3 %	NO APLICA	NO APLICA
GRU	−23,4 %	NO APLICA	NO APLICA
RNN	−49,0 %	NO APLICA	NO APLICA

Tabla 4.21: Resumen efectos de aplicación de experimentos, mensual.

Modelo/Experimento	Pandemia	Concentración	Híbrido
ARIMA	0, 7 %	NO APLICA	NO APLICA
SARIMA	0, 8 %	NO APLICA	NO APLICA
RF	0, 0 %	−1, 9 %	−2, 3 %
LGBM	−2, 7 %	−2, 1 %	−3, 5 %
XGB	0, 5 %	0, 7 %	1, 3 %
MLP 1L	−6, 7 %	0, 0 %	−0, 8 %
MLP ML	−8, 5 %	10, 0 %	9, 2 %
LSTM	3, 6 %	NO APLICA	NO APLICA
GRU	−4, 9 %	NO APLICA	NO APLICA
RNN	−8, 7 %	NO APLICA	NO APLICA

Tabla 4.22: Resumen efectos de aplicación de experimentos, diario.

De lo anterior, se pueden extraer las siguientes conclusiones:

- Para realizar las predicciones mensuales utilizando los métodos estadísticos, es conveniente aplicar el experimento de temporalidad en todos los casos. No obstante para las predicciones diarias, no se cumple dicha regla.
- Para realizar las predicciones mensuales utilizando los métodos basados en arboles, es conveniente aplicar el experimento de temporalidad, excepto para el modelo **XGB** para el cual es conveniente aplicar el experimento mixto. No obstante para las predicciones diarias, no se cumple dicha regla ya que aquí sería conveniente no aplicar ningún experimento sobre el modelo **XGB** y para los otros es conveniente aplicar el experimento mixto.

- Para realizar las predicciones mensuales utilizando los algoritmos de *Multi Layer Perceptron* es conveniente aplicar el experimento de concentración. No obstante para las predicciones diarias, no se cumple dicha regla y los indicadores señalan que es más conveniente utilizar el experimento de temporalidad en este escenario.
- Para realizar las predicciones mensuales utilizando los algoritmos basados en redes recurrentes, es conveniente aplicar el experimento de temporalidad. No obstante para las predicciones diarias, no se cumple, exactamente, dicha regla y los indicadores señalan se debiese aplicar este experimento sobre los modelos **RNN** y **GRU**.

4.4. Evaluación económica

Tal como se comenta en la etapa de entendimiento del negocio, se propone una forma de traducir el error de los modelos a coste económico para la empresa. Para poder realizar aquello se comparan escenarios de operación (pedidos de camiones) en base a los resultados de los modelos. Se presenta a continuación los pasos necesarios para realizar la evaluación económica, tomando como ejemplo las predicciones ocupadas como *baseline*.

1. Transformar la serie de tiempo pronosticada por el modelo[5] (figura 4.6) para computar el máximo de unidades vendidas por semana (tabla 4.23).

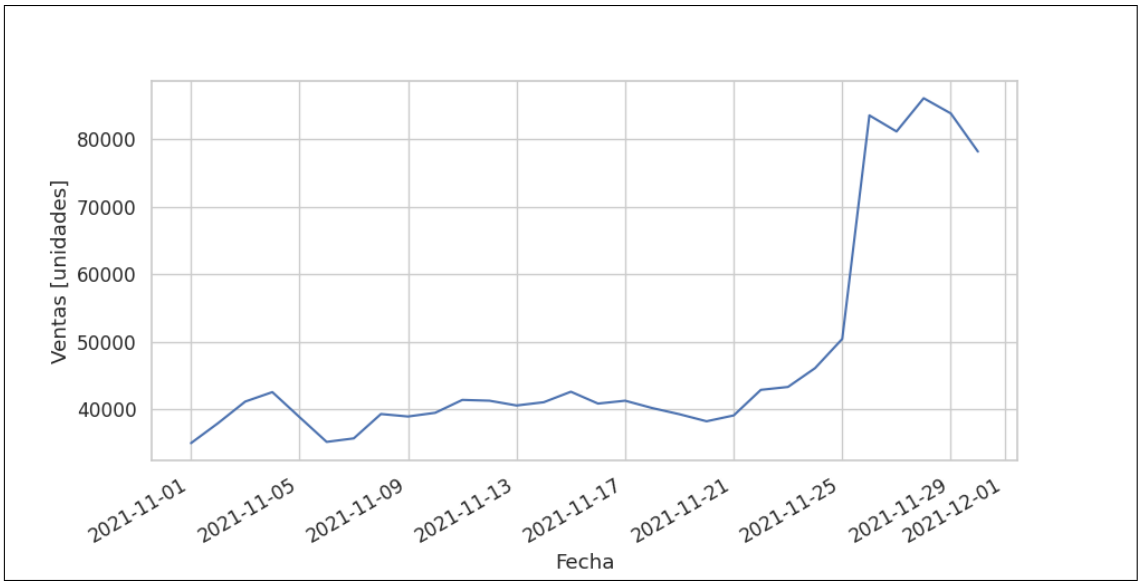

Figura 4.6: Ventas pronosticadas por modelo baseline

Modelo	Semana 1	Semana 2	Semana 3	Semana 4	Semana 5
Baseline	35.609	34.694	35.424	74.631	72.009

Tabla 4.23: Máximo semanal para modelo baseline.

2. Transformar las unidades pronosticadas a unidades despachadas desde los CD (tabla 4.24). Lo cual se realiza multiplicando por 18 %[6].

Modelo	Semana 1	Semana 2	Semana 3	Semana 4	Semana 5
Baseline	6.125	5.967	6.093	12.837	12.386

Tabla 4.24: Estimación de unidades despachadas desde los CD, modelo baseline.

3. Calcular el número de camiones requerido (tabla 4.25), dividiendo el número calculado en el paso previo por el número de productos que se pueden cargar en el camión[7].

[5]Se utiliza la serie agregada de ventas, es decir, la suma de todos los productos vendidos por día.
[6]Dato proveniente de la operación, mostrado en *entendimiento de negocio*.
[7]Este número oscila entre 60 y 200 productos por camión, para el ejemplo se considera que se cargan 60

Modelo	Semana 1	Semana 2	Semana 3	Semana 4	Semana 5
Baseline	103	100	102	214	207

Tabla 4.25: Estimación de camiones necesarios para despachos, modelo baseline.

4. Contrastar el número de camiones solicitados con el número de camiones que debieron ser solicitados (tabla 4.26), al utilizar la demanda real[8].

Modelo	Semana 1	Semana 2	Semana 3	Semana 4	Semana 5
Baseline	50	61	60	70	80

Tabla 4.26: Diferencia de camiones solicitados con demanda real, modelo baseline.

5. Calcular sobrecoste que el error del modelo genera (tabla 4.27). Se debe considerar que si la diferencia de camiones es positiva, hubo un exceso en la contratación de camiones, lo que implica un sobrecoste de $140.000 por camion extra que se contrató demás. Por contraparte, si la diferencia de camiones es negativa, entonces hubo una falta en la contratación de camiones, lo que implica un sobrecoste de $210.000 por camión extra que se tuvo que contratar de manera extraordinaria.

Modelo	Sobrecoste total
Baseline	$45.080.000

Tabla 4.27: Sobrecoste total, modelo baseline.

Finalmente, se debe mencionar que el número de productos con el que se puede cargar un camión será una variable que sirve para sensibilizar el beneficio económico de los modelos, ya que si se considera un mayor número de productos por camión el número de camiones será menor y por tanto el impacto de la solución será menor.

Hecha esa consideración, se realiza el cálculo economico para todos los modelos implementados (tabla 4.28), considerando 3 tipos de escenario:

- Escenario Conservador: Cada camión carga 200 productos.
- Escenario Moderado: Cada camión carga 120 productos.
- Escenario Arriesgado: Cada camión carga 60 productos.

[8]La demanda real en este caso sería de 53, 39, 42, 144 y 127 camiones respectivamente

Modelo / Escenario	Arriesgado	Moderado	Conservador
$XGB_{PANDEMIA}$	$ 53.410.000	$26.810.000	$15.960.000
XGB	$ 74.620.000	$37.310.000	$22.680.000
$XGB_{HIBRIDO}$	$ 84.210.000	$42.210.000	$24.920.000
MLP $1L_{PANDEMIA}$	$ 88.620.000	$44.240.000	$26.180.000
$LGBM_{PANDEMIA}$	$ 98.420.000	$49.140.000	$29.330.000
XGB_{FOCUS}	$ 98.560.000	$49.280.000	$29.470.000
$LGBM_{HIBRIDO}$	$105.350.000	$52.710.000	$31.500.000
$LGBM_{FOCUS}$	$111.440.000	$55.650.000	$33.320.000
LGBM	$114.240.000	$56.840.000	$34.160.000
RNN	$115.360.000	$57.960.000	$34.440.000
$LSTM_{PANDEMIA}$	$117.950.000	$59.010.000	$35.350.000
LSTM	$120.470.000	$60.060.000	$36.120.000
$GRU_{PANDEMIA}$	$124.880.000	$62.440.000	$37.240.000
MLP $ML_{PANDEMIA}$	$128.240.000	$64.120.000	$38.150.000
BASELINE	$129.570.000	$64.890.000	$38.780.000
MLP ML	$129.850.000	$65.030.000	$38.850.000
$RNN_{PANDEMIA}$	$130.200.000	$65.310.000	$39.060.000
HW	$133.630.000	$66.780.000	$40.040.000
RF_{FOCUS}	$134.820.000	$67.410.000	$40.530.000
$RF_{HIBRIDO}$	$136.710.000	$68.320.000	$40.950.000
MLP 1L	$136.990.000	$68.460.000	$41.090.000
MLP ML_{FOCUS}	$139.020.000	$69.510.000	$41.650.000
MLP $ML_{HIBRIDO}$	$139.580.000	$69.790.000	$41.720.000
NAIVE	$140.980.000	$70.630.000	$42.000.000
$RF_{PANDEMIA}$	$143.640.000	$71.820.000	$43.050.000
RF	$143.850.000	$71.820.000	$43.260.000
MA	$147.210.000	$73.430.000	$43.960.000
MLP $1L_{HIBRIDO}$	$147.210.000	$73.710.000	$44.310.000
$SARIMA_{PANDEMIA}$	$149.940.000	$74.970.000	$45.150.000
$SARIMAX_{PANDEMIA}$	$149.940.000	$74.970.000	$45.150.000
SARIMA	$150.780.000	$75.390.000	$45.360.000
ARIMA	$150.780.000	$75.390.000	$45.360.000
MLP $1L_{FOCUS}$	$153.090.000	$76.650.000	$46.200.000
$ARIMA_{PANDEMIA}$	$156.030.000	$78.120.000	$46.830.000
GRU	$174.860.000	$87.570.000	$52.290.000

Tabla 4.28: Sobrecoste total, por escenario.

Al respecto, se evidencia que el modelo *Extreme Gradient Boosting Machine* con el experimento "*pandemia*" es el que genera el mayor impacto económico sobre la operación, con una reducción mensual del costo de entre $22.820.000 y $76.160.000 al comparar con el modelo usado actualmente. Es por lo anterior que se recomienda la adopción de este modelo para este caso de uso.

Además de ello, se destaca que todas las variantes de los modelo *XGB* y *LGBM* logran un mejor rendimiento económico que el modelo de comparación. Ante lo cual se evidencia que el modelamiento desarrollado mejora a la implementación actual que se le da a este caso de uso.

Se destaca al mismo tiempo, que ninguno de los modelos estadísticos es capaz de generar una reducción en costo con respecto al baseline ocupado por la empresa. Lo anterior llama la atención pues estos algoritmos tienen un menor error en las predicciones diarias de manera general (tabla 4.13). Sin embargo, la explicación para este comportamiento se evidencia en que en los segmentos de mayor volumen[9] los modelos estadísticos tienen peor perfomance que el modelo usado de comparación (tablas 4.14, 4.15 y 4.17).

Finalmente, se destaca que pese a que al evaluar el error promedio de los modelos de manera general (tabla 4.13) el modelo *Multi Layer Perceptron with 1 Dense Hidden Layer* es el que tiene menor error (36,6 %) a la hora de hacer la comparación económica es superado por un margen de entre $10.220.000 y $35.210.000 al comparar con el modelo *XGB*. Esto se debe a que el modelo *XGB* es capaz de entregar mejores predicciones para el cluster de alta rotación (tabla 4.17).

[9]Segmento de rotación normal, rotación variable y alta rotación.

Capítulo 5

Conclusiones y trabajo futuro

En este capítulo se deglosarán las principales conclusiones obtenidas a partir de los resultados obtenidos de la implementación computacional de los modelos planteados. Para poder guiar de buena manera los principales aprendizajes y trabajos propuestos, se muestran a continuación las preguntas que guiaran el relato propuesto.

- ¿Tuvo algún efecto la segmentación aplicada sobre las series de tiempo?
- ¿Cuál es el modelo que mejor resuelve la problemática planteada? ¿Cuánto mejora al modelo actual? ¿Cumple con los objetivos propuestos por el estudio?
- ¿Como se podrían mejorar las predicciones?

5.1. Efecto del clustering

Un paso importante dentro de este estudio fue la segmentación de las series de tiempo en 6 diferentes clusters. Por lo que cabe preguntarse respecto a la utilidad que este tuvo en el proceso presentado. Para ello, se vislumbran dos motivos por los cuales esta segmentación generó valor en la investigación.

1. Mejora en implementación : Generar los clusters permitió aplicar el experimento de concentración y el experimento híbrido, lo cual mejoró el rendimiento de los modelos en bastantes casos, tal como se comentó en el capítulo anterior.

   En particular, para el caso mensual esta técnica permite una disminución del error de entre 3,9 % y 27,2 % para los modelos de árboles y de entre 1,1 % y 14,1 % para los modelos de aprendizaje profundo (tabla 4.21). Por otra parte para las predicciones diarias esta dsminución es de entre 1,9 % y 3,5 % para los modelos $LGBM$ y RF (tabla 4.22).

2. Enriquecimiento de análisis : El desglose de resultados por cluster, añade más variables a tomar en cuenta a la hora de evaluar los modelos. Lo anterior nos permite tener un

conocimiento más integral de los productos que se están vendiendo, y previene algunos errores en la interpretación de resultados.

Respecto a este punto existe una serie de ejemplos que se pueden mencionar, como por ejemplo lo expuesto al final del capítulo anterior (página 67) en el cual se encuentra una explicación al porque modelos que en su rendimiento general tenían un bajo error tienen una evaluación económica que entrega un mayor sobre coste al comparar con modelos que tienen un error de predicción mayor. Dicho ejercicio, si bien fue aplicado sobre las predicciones diarias también se replica para el caso de predicciones mensuales, por ejemplo, al observar la performance de los modelos de manera agregada (tabla 4.6) el modelo **MLP ML** pareciera tener peor desempeño general que el **baseline** usado como comparación. No obstante al observar su desempeño en los diferentes clusters, vemos que este modelo mejora en 5 de los 6 clusters la performance de la predicción usada como punto de comparación.

En conclusión, la técnica de clustering ha demostrado ser una herramienta valiosa en el estudio presentado, ya que ha permitido mejorar la implementación de los modelos y reducir los errores de predicción en diferentes casos, lo que ha llevado a un aumento en el rendimiento de los mismos. Además, el desglose de resultados por cluster ha enriquecido el análisis al proporcionar información adicional sobre los productos y prevenir errores en la interpretación de los resultados. En general, la técnica de clustering ha sido un factor clave para obtener una visión más completa y detallada del comportamiento de las series de tiempo analizadas en este estudio.

5.2. Selección de modelo

Para poder seleccionar el modelo que mejor performance tiene en las series de tiempo estudiadas, y con ello definir una estrategia de cuales modelos deben ser utilizados, se enlistan los modelos que mejor rendimiento tienen, en función del cluster y nivel de agregación en las predicciones (tabla 5.1)

Segmento/Granularidad	Diario	Mensual
General	**MLP 1L**	**LGBM**
Rotación Normal	**MLP 1L**	**RNN**
Rotación Variable	**MLP 1L**	**RNN**
Baja Rotación	**XGB**	**RNN**
Alta Rotación	**XGB**	**HW**
Rotación Itinerante	**MLP 1L**	**RF**
Rotación Potenciada	**MLP 1L**	**RF**

Tabla 5.1: Resumen mejor modelo por nivel de agregación.

Se observa de lo anterior, que existen patrones claramente identificables respecto al comportamiento de los mejores modelos.

- Predicciones mensuales : Se observa que los modelos basados en *Redes Recurrentes* son aquellos que muestran mejor performance en los segmentos cuya rotación tiene un comportamiento estable. No obstante, para los segmentos que tienen un cambio de comportamiento pierden su efectividad, pasando a ser los algoritmos basados en árboles los que mejor son capaces de captar estos comportamientos.
- Predicciones diarias : Se observa que los modelos basados en árboles y los modelos *Multi Layer Perceptron with 1 Dense Hidden Layer* son aquellos que muestran mejor performance en los diferentes segmentos.

Con lo expresado previamente, se entregan las conclusiones de la experiencia y entregar las recomendaciones pertinentes respecto a los casos de uso en que deberían usarse los diferentes modelos.

- Para la extrapolación de tendencias de largo plazo, como lo son las predicciones mensuales, los modelos más efectivos son los basados en redes recurrentes. Especialmente el modelo *Vanilla Recurrent Neural Network*. No obstante, el mejor desempeño de este algoritmo se logra en series relativamente estables y que no presentan comportamientos *ruidosos*[1]. En particular, en estos casos el modelo *RNN* mejora al baseline entre 1,6 % para el caso de rotación variable (tabla 4.8) y un 15,9 % para el caso de baja rotación (tabla 4.9).
- Para la predicción de series de tiempo que presentan mucha volatilidad[2], ya sea por un cambio de tendencia, por un efecto de la estacionalidad o un evento externo, es más conveniente aplicar modelos basados en árboles o modelos basados en *Multi Layer Perceptron*. Lo anterior es válido tanto para predicciones mensuales como diarias. En particular, en las predicciones diarias el modelo *MLP 1L* disminuye en 25,7 % el error en el segmento de rotación potenciada (tabla 4.19) mientras que para el caso de predicción mensual en el segmento de rotación itinerante el modelo *RF* mejora el error en un 52,0 % (tabla 4.11).
- Para captar la volatilidad diaria de las ventas, conviene aplicar algoritmos basados en árboles o alguna de las variantes de *Multi-Layer Perceptron*, pues estas logran predecir con un menor error las ventas, sin importar el cluster en el cual se vean aplicados. En particular, para las predicciones del segmento de alta rotación, el modelo *XGB* mejora al modelo *baseline* en un 5,9 % (tabla 4.17) mientras que para el segmento de rotación itinerante con el modelo *MLP 1L* la mejora es de un 31,7 % (tabla 4.18).

Ahora bien, para poder determinar qué modelo es el mejor dado el caso de uso, también se debe tomar en consideración el impacto económico que cada uno acarrea. En particular, se tiene que el modelo que presenta un mayor beneficio para la operación es el modelo *Extreme Gradient Boosting Machine* el cual generaría un ahorro de entre $22.820.000 y $76.160.000 al compararlo con el modelo usado actualmente. Esto se debe a que este modelo es capaz de predecir especialmente bien al segmento de alta rotación, con lo cual logra un mayor impacto

[1]Segmentos de rotación normal, rotación variable y baja rotación
[2]Segmentos de rotación itinerante o rotación potenciada

en la planificación de la operación. En este apartado, también se destaca a los modelos *MLP 1L* y *LGBM* los cuales generan el segundo y tercer mayor ahorro respectivamente al comparar con el *baseline*.

Finalmente, considerando todos los factores evaluados previamente, se recomienda el uso del modelo *XGB* para realizar las predicciones de ahora en adelante para la planificación operacional. Debido a que con este se genera la mayor eficiencia económica y con ello se obtiene el mayor retorno para la organización. No obstante se recomienda que para mejorar en los objetivos no operacionales es conveniente aplicar el modelo *LGBM* pues es capaz de minimizar el error de mejor manera en las predicciones del mediano y largo plazo.

En conclusión, se ha cumplido con el objetivo planteado para este estudio. Se ha desarrollado un nuevo framework de pronóstico de demanda. El cual separa las predicciones en dos tipos: predicciones mensuales que responde a los objetivos de mediano y largo plazo para los cuales la compañia debe prepararse, y predicciones diarias las que responden a los objetivos operacionales a los que se debe hacer frente. La implementación computacional que se ha llevado a cabo, logró disminuir las métricas de performance actual, en 1,0 % en el caso mensual y un 11,3 % en el caso de predicciones diarias. Además, esta reducción en el error de las predicciones logra disminuir el coste de la operación en valores mucho mayores a los esperados al comenzar el estudio. Con lo cual se da este ejercicio como beneficioso.

5.3. Trabajo propuesto

Del estudio realizado, se proponen las siguientes líneas que podrían significar una mejora en el framework propuesto.

1. Se pueden introducir cambios sobre cómo los eventos y feriados son incluidos en el modelamiento del problema. Respecto a los feriados, se podría experimentar diferenciar cada feriado en vez de englobarlos en sólo una categoría. Respecto a los eventos, se podrían incluir campañas más específicas que sólo afecten a un número reducido de sublíneas[3].
2. Se pueden introducir variables macro económicas como inflación esperada o tasa de politica monetaria, de manera de introducir el efecto del entorno al problema.
3. Se podría introducir el precio de los productos al modelamiento del problema, entregando un grado de libertad adicional para la clusterización de sublíneas y una variable independiente para la predicción de la serie de tiempo, de manera de poder sensibilizar las ventas que se tendran de las sublíneas frente a variaciones del precio de los productos.
4. Dado que se ha identificado a los modelos basados en árboles como aquellos que mejor rendimiento entregan en este caso de uso, se propone continuar con la implementación de este tipo de algoritmos, incluyendo los modelos *CATBoost* o *ADABoost* de manera de poder comparar su desempeño tanto a nivel general como a nivel de cluster.

[3]Recordar que sólo se utilizaron los grandes macro eventos en este modelamiento

5. Se podría experimentar con modelos *hibridos*, los cuales consisten en el ensamblaje de dos modelos, el primero de ellos se encarga de predecir la tendencia de la serie de tiempo, mientras que el segundo de ellos se enfoca en predecir el error del primer modelo.

6. Si bien los resultados de las redes recurrentes no han sido muy satisfactorios, se podrían implementar algunas versiones de las mismas que en este trabajo no fueron considerados, como los *ENCODERS* o los *TRANSFORMERS*, los cuales se han mostrado prometedores a la hora de resolver otro tipo de problemas basados en secuencias.

Bibliografía

[1] Dr K.; Srivastava Siddharth Anoop Andrabi, Syed Hamad ul Haq; Alice. Sales forecasting using xgboost. 2022.

[2] Saurabh Chandra Ankur Jain, Manghat Nitish Menon. Sales forecasting for retail chains. 2015.

[3] C. Brooks. Introductory econometrics for finance. 2008.

[4] G.E. Hinton D.E. Rumelhart and R.J. Williams. Learning internal representations by error propagation. *MIT Press*, page 318–361, 1985.

[5] G. Dorffner. Neural networks for time series processing. 1996.

[6] N. Mastronardi G Dellino, T. Laudadio and C. Meloni. Sales forecasting models in the fresh food supply chain. pages 419–426, 2015.

[7] R.J. Hyndman and G. Athanasopoulos. Forecasting: Principles and practice. page 318–361, 2014.

[8] I. El Farissi I. Slimani and S. Achchab. Artificial neural networks for demand forecasting: Application using moroccan supermarket data. 2015.

[9] A. Orellana. Árboles de decisión y random forest. 2018.

[10] Rony Mitra Dyutimoy Das Sushmita Narayana Manoj K. Tiwari Priyam Saha, Nitesh Gudheniya. Demand forecasting of a multinational retail company using deep learning frameworks. 55(10):395–399, 2022.

[11] K. Laframboise R. Carbonneau and R. Vahidov. Application of machine learning techniques for supply chain demand forecasting. *European Journal of Operational Research*, pages 1140–1154, 2008.

[12] C. Shearer. The crisp-dm model: The new blueprint for data mining. *Journal of Data Warehousing*, 5(4):13–22, fall 2000.

[13] Patrick Brandtner Chibuzor Udokwu Taha Falatouri, Farzaneh Darbanian. Predictive analytics for demand forecasting – a comparison of sarima and lstm in retail scm. 2022.

[14] Logofatu D. Leon F. Muharemi F. Vairagade, N. Demand forecasting using random forest and artificial neural network for supply chain management. 2019.