UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Innovación en la Gestión de Campañas Comerciales: Desarrollo de un Modelo Predictivo para la Eficiencia Comercial desde la Predicción Global hasta la Precisión Local en Inditex

Trabajo Fin de Grado

Grado en Ciencia de Datos

AUTOR/A: Ramón Tadghighi, Pablo

Tutor/a: Esteban Amaro, Rosa Inés

Cotutor/a: Hernández Orallo, José

Cotutor/a externo: Díaz Cortés, Javier

CURSO ACADÉMICO: 2023/2024

Agradecimientos

En primer lugar, quiero agradecer a mi familia por su apoyo durante todo mi periodo estudiantil y a mis amigos por estos cuatro años. Vuestro apoyo ha sido fundamental para llegar hasta aquí.

También quiero agradecer a Inditex por darme la oportunidad de realizar las prácticas con ellos y desarrollar mi TFG. Ha sido una experiencia inolvidable que ha elevado mi formación de manera significativa y me ha dado la oportunidad de vivir grandes experiencias.

Además, agradezco a mis tutores de la UPV por su guía a lo largo de todo el trabajo y a mi tutor de la empresa por enseñarme a desarrollar un proyecto de forma profesional.

Gracias a todos por vuestro apoyo y contribución a mi crecimiento académico y profesional.

Resumen

El Trabajo de Fin de Grado se enfoca en la innovación en la gestión de campañas comerciales en Inditex, con el objetivo de obtener un modelo capaz de prever las ventas en cada una de las tiendas de una prenda nueva clasificada según los parámetros modelo-calidad-color, que Inditex autodenomina MOCACO.

Para ello, empezamos entendiendo el comportamiento de las curvas de venta de los MOCACO en las tiendas, lo que nos permite clasificar los MOCACO en tres clases; fantasías, básicos y otros. A continuación, pasamos a la creación de los experimentos, en total haremos ocho; los dos primeros corresponden a los baseline, los tres siguientes a modelos de perceptrón multicapa y los tres últimos a modelos de XGBoost con entrenamiento distribuido. Mientras que los baselines tienen un preprocesamiento diferente al resto, los otros modelos tienen el mismo; los de perceptrón multicapa se llevarán a cabo con scikit-learn y los de XGBoost con pyspark.

El primer baseline consiste en desagregar las predicciones del modelo actual a nivel mundo por el peso de la tienda, calculado como el porcentaje de venta de la tienda. El segundo es similar al primero, con la diferencia de que cada tienda tiene un peso diferente por semana.

En cada uno de los experimentos vamos creando nuevas variables, cambiando la estructura del modelo o cubriendo las limitaciones del experimento anterior. Al comparar los modelos utilizamos dos métricas, una que mide el error semanal (WAPE) y otra que mide el error global del modelo (WWAPE), ambas se interpretan como porcentaje. Tras la evaluación, el mejor modelo es un XGBoost que lo encontramos en el último experimento. La información que tiene este modelo para realizar las predicciones es en relación con la descripción de los MOCACO; envío inicial y ventas de los MOCACO parecidos que han tenido una situación similar en la tienda, descripción de la situación de la tienda y venta de la familia en la tienda, principalmente. Para acabar realizamos la optimización de los hiperparámetros y su explicabilidad viendo la importancia de cada una de las siete variables a nivel global y semanal.

La principal limitación que hemos tenido a lo largo del desarrollo del proyecto es la memoria de los clústeres utilizados para ejecutar los modelos; que hemos conseguido resolver haciendo un muestreo de cara a garantizar la representatividad de la muestra.

Al obtener un modelo que cumple con el objetivo, hemos conseguido un gran impacto Puesto que la empresa consigue ahorrar costes al hacer una asignación correcta de los MOCACO en tienda durante el proceso de distribución. A su vez, permite obtener la máxima venta posible en tiendas que de otro modo hubieran recibido menos mercancía. En conclusión, se consigue ahorrar costes y aumentar los ingresos de la compañía al realizar una mejor gestión.

Palabras clave: forecasting de la demanda, productos nuevos, Fashion retail, aprendizaje automático, ingeniería de características

Abstract

The Bachelor's Thesis focuses on innovation in commercial campaign management at Inditex, with the aim of developing a model capable of predicting sales in each store for a new garment classified according to the model-quality-color parameters, which Inditex refers to as MOCACO.

To achieve this, we begin by understanding the behavior of the sales curves of MOCACO in the stores, which allows us to classify MOCACO into three categories: fantasies, basics, and others. Next, we proceed to the creation of the experiments, with a total of eight; the first two are baselines, the next three are multilayer perceptron models, and the last three are XGBoost models with distributed training. While the baselines have different preprocessing from the rest, the other models share the same preprocessing; the multilayer perceptron models are implemented with scikit-learn, and the XGBoost models with PySpark.

The first baseline involves disaggregating the current global model predictions by the store's weight, calculated as the store's sales percentage. The second is similar to the first, with the difference that each store has a different weight per week.

In each experiment, we create new variables, change the model structure, or address the limitations of the previous experiment. To compare the models, we use two metrics: one that measures the weekly error (WAPE) and another that measures the overall model error (WWAPE), both interpreted as percentages. After evaluation, the best model is an XGBoost found in the last experiment. The information this model uses for predictions includes the description of MOCACO; initial shipment and sales of similar MOCACO that had a similar situation in the store, the store's situation description, and the family sales in the store.

Finally, we optimize the hyperparameters and explain the model by analyzing the importance of each of the seven variables globally and weekly. The main limitation throughout the project development was the memory of the clusters used to run the models; we solved this by sampling to ensure the sample's representativeness.

By achieving a model that meets the objective, we have made a significant impact as the company can save costs by correctly allocating MOCACO in stores during the distribution process. At the same time, it allows for maximizing sales in stores that would otherwise have received less merchandise. In conclusion, we save costs and increase the company's revenue by improving management.

Keywords: demand forecasting, new products, retail sector, machine learning, feature engineering

Resum

El Treball de Fi de Grau se centra en la innovació en la gestió de campanyes comercials en Inditex, amb l'objectiu d'obtenir un model capaç de preveure les vendes en cadascuna de les tendes d'una peça nova classificada segons els paràmetres model-qualitat-color, que Inditex anomena MOCACO.

Per a això, comencem entenent el comportament de les corbes de venda dels MOCACO en les tendes, el que ens permet classificar els MOCACO en tres classes: fantasies, bàsics i altres. A continuació, passem a la creació dels experiments, en total farem huit; els dos primers corresponen als baseline, els tres següents a models de perceptró multicapa i els tres últims a models de XGBoost amb entrenament distribuït. Mentre que els baseline tenen un preprocessament diferent a la resta, els altres models tenen el mateix; els de perceptró multicapa es portaran a terme amb scikit-learn i els de XGBoost amb pyspark.

El primer baseline consisteix a desagregar les prediccions del model actual a nivell mundial pel pes de la tenda, calculat com el percentatge de venda de la tenda. El segon és similar al primer, amb la diferència que cada tenda té un pes diferent per setmana.

En cadascun dels experiments anem creant noves variables, canviant l'estructura del model o cobrint les limitacions de l'experiment anterior. En comparar els models utilitzem dues mètriques, una que mesura l'error setmanal (WAPE) i una altra que mesura l'error global del model (WWAPE), ambdues s'interpreten com a percentatge. Després de l'avaluació, el millor model és un XGBoost que trobem en l'últim experiment. La informació que té aquest model per a realitzar les prediccions està relacionada amb la descripció dels MOCACO; enviament inicial i vendes dels MOCACO semblants que han tingut una situació similar en la tenda, descripció de la situació de la tenda i venda de la família en la tenda, principalment. Per a acabar, realitzem l'optimització dels hiperparàmetres i la seua explicabilitat veient la importància de cadascuna de les set variables a nivell global i setmanal.

La principal limitació que hem tingut al llarg del desenvolupament del projecte és la memòria dels clústers utilitzats per a executar els models; que hem aconseguit resoldre fent un mostreig per a garantir la representativitat de la mostra.

En obtindre un model que compleix amb l'objectiu, hem aconseguit un gran impacte ja que l'empresa aconsegueix estalviar costos en fer una assignació correcta dels MOCACO en tenda durant el procés de distribució. Alhora, permet obtindre la màxima venda possible en tendes que d'una altra manera haurien rebut menys mercaderia. En conclusió, s'aconsegueix estalviar costos i augmentar els ingressos de la companyia en realitzar una millor gestió.

Paraules clau: forecasting de la demanda, productes nous, sector de retail, aprenentatge automàtic, enginyeria de característiques

Tabla de contenidos

Índice de figuras

Índice de tablas

Glosario

MOCACO: hace referencia a MOdelo - CAlidad - COlor, es una prenda sin incluir su talla, por ejemplo, una camiseta negra. De este modo, se puede englobar a varias prendas con una misma identificación, es decir, representa a todas las tallas.

TechDay: Evento en el que se concentra en el auditorio toda el área de tecnología, miembros importantes de la empresa como Óscar García Maceiras (CEO de Inditex) y dan información sobre cómo ha ido la empresa el último año.

Ciclo de vida: utilizamos la expresión cuando queremos hablar del tiempo que pasa desde que un MOCACO se empieza hasta que se acaba de comercializar.

MLP: “Multilayer perception” (Perceptrón multicapa)

XGBoost: “Extreme Gradient Boosting”

NDA: "Non-Disclosure Agreement" (Acuerdo de Confidencialidad), es un contrato legal compromete a no divulgar cierta información confidencial que se comparte entre ellas.

MOCACO – tienda: utilizamos el guion entre MOCACO y tienda para hacer referencia a que se está hablando de un MOCACO concreto en una tienda concreta.

CAPÍTULO 1

Introducción

El Trabajo de Fin de Grado (TFG) que abordamos a lo largo de la memoria se realiza en el marco de prácticas de empresa. Inditex es una de las empresas punteras en el sector del retail, específicamente en la moda, teniendo una facturación de 35.947 millones de euros en el 2023 (1). El grupo está formado por Zara, Pull&Bear, Massimo Dutti, Bershka, Stradivarius, Oysho, Lefties y Zara Home. Resulta complicado dar el número de tiendas que posee el grupo, debido a que cada día se abren y se cierran tiendas, pero a día 14-05-2024 tienen 5.632 tiendas (dato extraído del TechDay). Y en el año 2023 se han gestionado unos 2.000 millones de prendas, habiendo unos 5 millones de transacciones de clientes al día (dato extraído del TechDay). No obstante, el trabajo se va a centrar exclusivamente en la marca referente del grupo, Zara, que representa el 73,87% de la venta (1).

La planificación de campañas comerciales es crucial para las empresas de retail, porque permiten optimizar al máximo la cadena de suministro al realizar una buena gestión de la distribución. Cada producto lanzado en una nueva campaña comercial se convierte en una entidad independiente, con sus propias métricas de rendimiento y capacidad de evaluación. Por lo tanto, anticipar cómo se venderá un nuevo artículo es de suma importancia.

Todo el trabajo gira entorno a los MOCACO (modelo, calidad, color), por lo que es vital entender su significado. Todos vestimos con diferentes prendas, cada una de ellas tiene unas características, empezando por el modelo, luego el color y la calidad. Estas son las características más generales que pueden referenciar a una prenda, digamos que son las que describen a la prenda y luego le añadimos la talla. Un ejemplo podría ser una camiseta negra, un pantalón blanco, es decir, cualquier prenda que nos podamos imaginar excluyendo la característica de la talla. Por lo tanto, un MOCACO hace referencia a la prenda sin incluir la talla, de esta forma se puede hacer referencia a todas ellas.

Otro concepto fundamental para comprender el desarrollo es saber que es un ciclo de vida. Tenemos que diferenciar entre dos tipos de ciclo de vida, a nivel mundo y a nivel tienda. Cuando hablamos del ciclo de vida de un MOCACO a nivel mundo, nos referimos al tiempo que pasa desde que un MOCACO empieza a comercializarse hasta que se deja de comercializar. Y cuando tratamos sobre el ciclo de vida de un MOCACO a nivel tienda, hacemos referencia al tiempo que pasa desde que un MOCACO comienza a comercializarse en una tienda concreta hasta que deja de comercializarse en esa tienda. De esta forma, un MOCACO solo tiene un ciclo de vida a nivel mundo y tantos ciclos de vida a nivel tienda como tiendas en la que se esté comercializando.

Actualmente, la empresa dispone de un modelo que le permite conocer de antemano cuántas unidades va a vender un MOCACO nuevo a nivel mundial en cada semana de su ciclo de vida, con un porcentaje de error o WWAPE del 35%. Ser capaces de conocer cuántas unidades va a vender ayuda considerablemente a Inditex puesto que les permite llevar una correcta gestión de la mercancía, reduciendo los costes. No obstante, tener un modelo que únicamente hace las predicciones a nivel mundial es limitante, ya que nos da la información de cuánto venderemos en total, pero no sabemos cuánto destinar a cada una de las unidades.

Para solucionar el factor limitante del modelo actual, lo que planteamos es un modelo que, en vez de hacer predicciones a nivel mundial, sea capaz de decir cuántas unidades se venderán de un MOCACO nuevo en una tienda. Añadir esta nueva dimensión aumenta considerablemente la dificultad del modelo, ya que antes cada MOCACO tenía únicamente un ciclo de vida, ahora tendrá un ciclo de vida por cada tienda en la que se venda. Esto hace que el volumen de datos con el que trabajamos sea mucho mayor. Obtener este modelo provoca un gran cambio en la mejora de la gestión de las campañas, porque además de saber cuántas venderemos cada semana del ciclo de vida a nivel mundial, ahora sabremos cuántas venderemos cada semana del ciclo de vida de cada tienda, aportando una mayor precisión a la hora de distribuir los MOCACO.

1.1 Motivación

El TFG surge por la necesidad de INDITEX de mejora en la gestión de campañas comerciales, creando un modelo que permita realizar la predicción de la venta de un MOCACO nuevo en cada tienda de Zara.

Es una gran oportunidad poder colaborar con una gran empresa como INDITEX, líder en la innovación en el sector del retail, específicamente en la moda. Para poder mantenerse como líder, tiene un gran departamento tecnológico llamado INDITEX TECH, organizado con una estructura matricial formada por ocho verticales que son Ecommerce, Tienda, Logística, Producto y Sostenibilidad, Transporte, Distribución, Financiero y Personas. Cada vertical es un departamento independiente, que está servida por tres horizontales: UX & Design, Datos e IOP+ (2).

Este proyecto se encuentra dentro de la horizontal Datos, en el departamento de Analytics. Tal y como se dijo en la reunión del departamento del 28-03-2024, donde se hizo un balance del año 2023, se presentaron los proyectos en curso y se mostraron los objetivos del año. Obtener este modelo es uno de los objetivos principales para 2024 dado que, permite aumentar considerablemente el beneficio de la empresa reduciendo costes y aumentando la facturación al destinar correctamente los MOCACO a las tiendas.

Como estudiante de Ciencia de Datos de la UPV, tengo las capacidades necesarias para cubrir posibles necesidades del departamento. La perseverancia, disciplina y rápida resolución de problemas demostradas a la empresa en el workshop del 19-04-2023 en colaboración con la UPV, así como en un proyecto anterior realizado en Proyecto III (asignatura del 3er curso) donde desarrollé algoritmos de Machine

Learning, me llevó a un proceso de selección donde se me dio la oportunidad de poder formar parte de este proyecto.

Hay tres aspectos diferenciadores que aumentan mi motivación. El primero es que es un proyecto real que tiene un impacto en la empresa. El segundo es la gran experiencia que me aporta, tanto a nivel técnico como personal. A nivel técnico, porque utilizan la última tecnología en el mercado y tienen una gran cantidad de recursos para poder trabajar. A nivel personal, por todas las soft skills desarrolladas al adaptarme a un entorno nuevo y dinámico, en una empresa con miles de empleados. Por último, el uso del Marchine Learining para abordar el proyecto es una de las principales motivaciones, ya que es lo que más interés ha despertado en mí durante el grado y en lo que me he focalizado.

1.2 Objetivos

1.2.1 Objetivo general

El objetivo general es desarrollar un modelo capaz de predecir el número de unidades que un MOCACO nuevo va a vender en cada semana del ciclo de vida en la tienda.

1.2.2 Objetivos específicos y secundarios

Respecto a los objetivos específicos tenemos:

- **Preparar los datos.** Realizar el preprocesamiento de los datos para las variables numéricas, categóricas y las procedentes de fecha.
- **Generar nuevas variables que aporten valor.** Crear nuevas variables que permiten a los modelos minimizar el error.
- **Desarrollar modelos.** Generar modelos mediante diferentes técnicas de forma iterativa, incluyendo nuevas variables y supliendo limitaciones de modelos anteriores.
- **Evaluar modelos.** Evaluar los modelos desarrollados mediante las métricas establecidas.

De acuerdo con los objetivos específicos, desarrollamos los objetivos secundarios:

- **Entender el problema.** Familiarizarse bien con la forma de trabajar de la empresa, los términos que utilizamos a lo largo del proyecto, y fijar los objetivos que tenemos que alcanzar.
- **Comprender los datos.** Una visualización de los datos es importante para saber con qué estamos trabajando y entender el comportamiento de lo que queremos predecir. Para ello, tenemos que estudiar las curvas de venta de cada MOCACO en cada tienda. Además, es importante saber qué datos y cómo utilizarlos para obtener el máximo rendimiento. Este objetivo junto al de entender el problema son cruciales para entender cómo se va a hacer la muestra y qué métricas de evaluación vamos a utilizar.

- **Preparar los datos.** Una vez hemos estudiado los datos y entendemos su comportamiento, hay que limpiarlos y obtener nuevas variables para alcanzar el máximo rendimiento y tenerlos en un formato que acepte el modelo para poder entrenarlo.
- **Realizar las muestras.** Realizar una función de muestreo que nos permita tener una muestra que represente a toda la población y así poder entrenar el modelo teniendo en cuenta los recursos que tenemos, buscando el mayor equilibrio entre el gasto de recursos y el rendimiento. Al igual con el conjunto de prueba.
- **Establecer métricas de evaluación.** Tenemos que establecer cuáles son las métricas que vamos a utilizar para evaluar los diferentes modelos desarrollamos y así realizar una comparación siguiendo unas mismas métricas y un mismo criterio.
- **Crear baselines.** Establecer los baselines es crucial para luego tener modelos simples con los que poder comparar.
- **Crear modelos de Machine Learning.** Utilizar diferentes técnicas de Machine Learning diferentes que sean capaces de enfrentarse a los diferentes escenarios que van surgiendo. Y evaluarlos utilizando las métricas establecidas anteriormente mediante diferentes gráficas.
- **Comparar los modelos y seleccionar el mejor.** Comparar los modelos mediante las métricas de evaluación establecidas y seleccionar el que mayor rendimiento tenga.
- **Realizar la explicabilidad.** Obtener la explicabildiad global del modelo permitiendo entender la importancia de cada variable y su comportamiento.
- **Optimizar los hiperparámetros** Utilizar una técnica de optimización de hiperparámetros capaz de obtener la mejor combinación**.**

1.3 Metodología

Lo primero es comprender el problema que vamos a abordar y establecer la metodología que vamos a seguir. Lo hacemos mediante una reunión con el responsable del proyecto y las personas que han desarrollado el modelo a nivel mundial. Además, hacemos una introducción a la metodología que han llevado a cabo para obtener el modelo a nivel mundo.

Una vez que tenemos claro lo que debe hacer el modelo a nivel de tienda, comenzamos un proceso de formación para aprender a utilizar las tecnologías necesarias para llevar el proyecto a la fase de producción. Durante una semana, nos sumergimos en un proceso de formación donde exploramos las tecnologías y los procesos internos de la empresa, desde la producción de las prendas hasta su venta. Además, recibimos capacitación sobre la nube y la ciberseguridad.

Completado el proceso de formación, iniciamos la fase de desarrollo del TFG. Para comenzar, es necesario obtener el conjunto de datos principal. Una vez que lo tenemos, procedemos a realizar un procesamiento de datos y a comprender cómo se comportan los MOCACO en las tiendas. Esto incluye comparar las curvas de venta a nivel mundial con las curvas de venta a nivel de tienda para determinar si el

comportamiento de los MOCACO es similar y ver los diferentes tipos de curvas que tenemos.

El siguiente paso es crear los baselines que utilizaremos como modelos iniciales. Para construir estos modelos, primero realizamos el procesamiento de datos, que es igual para ambos, y luego creamos cada modelo por separado. Una vez construidos, los entrenamos y los evaluamos utilizando la misma muestra de evaluación para todos los modelos. Después de la evaluación, analizamos cómo se distribuye el error basándonos en diferentes agrupaciones de los datos.

Ya creados los baselines, procedemos a desarrollar los modelos de machine learning. Para estos modelos, exploramos diferentes escenarios. Esto implica entrenar un modelo con un conjunto específico de variables y evaluar su rendimiento. A partir de los resultados obtenidos, realizamos experimentos adicionales, como agregar más datos o abordar las limitaciones identificadas en el modelo anterior. Además, cada modelo se evalúa utilizando métricas de cálculo del error como el WWAPE y WAPE.

Finalmente, comparamos todos los modelos creados utilizando las métricas seleccionadas. Elegimos el mejor de ellos y analizamos la explicabilidad a nivel global para comprender su comportamiento. Por último, llevamos a cabo la optimización de hiperparámetros con la intención de mejorar el rendimiento del modelo seleccionado.

1.4 Impacto Esperado y ODS

Para entender bien el impacto en Inditex, es importante conocer el ciclo de vida del MOCACO. Una vez ya ha pasado por el proceso de diseño, creación, compras, producción y llega al almacén, llega un momento crucial que es determinar donde se manda el producto, ya que es muy importante para la venta que el MOCACO esté en el momento y lugar adecuado. Una vez el producto llega al almacén, el equipo de producción junto con los product managers deciden donde se tienen que enviar los MOCACO y cómo hacerlo, por lo que determinan a qué tienda tienen que mandarlos.

El proceso de distribución es fundamental, ya que puede generar ahorros en costes y mejorar el servicio al cliente. Los clientes buscan tener disponibles los productos que desean comprar y también esperan novedades constantemente. Para garantizar esta novedad, se envían dos camiones con mercancía nueva por semana, lo que subraya la importancia de la precisión en la distribución. Un enfoque preciso en la distribución no solo garantiza que los clientes tengan acceso a los productos deseados, sino que también contribuye a mantener su interés con la llegada regular de novedades. Esto no solo mejora la experiencia del cliente, sino que también puede impulsar las ventas y la lealtad a la marca. (3)

Para los equipos de producción y product managers resulta de vital importancia la información que les puede dar el modelo, les ahorra horas de trabajo y una mayor precisión en sus decisiones. Como consecuencia se reducen costes en el proceso de distribución desde las horas que invierten las personas que toman las decisiones, hasta los recursos empleados en el proceso del transporte y puesta de venta en las tiendas. Además, se evita tener los MOCACO en tienda que no se vayan a vender y que luego

haya que transportarlos a otro centro logístico. Adicionalmente a la parte logística, desde el enfoque de ventas, si se tiene un MOCACO en la tienda se puede vender, de lo contrario, si no está en tienda es posible que se pierda una venta.

Como vemos, tener una estimación de cuánto se va a vender cada MOCACO en cada tienda, lleva a la empresa a reducir en costes y en dar una mejor experiencia al cliente, por lo que tiene un gran impacto desde la empresa a la satisfacción del cliente.

Dejando a un lado el impacto que tiene en la empresa y en el cliente, veamos cuál es el impacto directo en el departamento. Este proyecto, es uno de los objetivos que quieren alcanzar para este año. Gracias a este trabajo hemos empezado un proyecto desde cero y disponen de unas primeras experimentaciones que dan como resultado un modelo con una perspectiva diferente a la que tenían pensada en un principio, además de disponer de variables con las que podrán generar nuevos modelos para aumentar el rendimiento.

Como estudiante de la UPV, tiene un gran impacto realizar el TFG en una empresa líder en su sector a nivel mundial. La UPV puede disponer, a través de este trabajo, de un proyecto innovador de especial relevancia para las empresas del mundo del retail que otros alumnos podrán tomar como referencia para crear nuevos proyectos.

En cuanto al impacto en los Objetivos de Desarrollo Sostenible (ODS) (4), consideramos que, de los objetivos globales establecidos por las Naciones Unidas, nuestro TFG tiene mayor impacto en los siguientes (5):

- **ODS 8. Trabajo decente y crecimiento económico**: mejorar la precisión en las previsiones de ventas puede llevar a una mejor planificación de la producción y distribución, evitando sobreproducción y reducción de inventarios, lo que puede aumentar la eficiencia operativa y la rentabilidad. Esto puede generar empleo estable y mejorar las condiciones laborales en toda la cadena de suministro. (6)
- **ODS 12. Producción y consumo responsable**: una previsión más precisa de la demanda puede ayudar a reducir el desperdicio de productos y materiales, ya que se producirán y distribuirán solo las cantidades necesarias. Esto puede minimizar los residuos textiles y la huella ecológica de la empresa. (7)
- **ODS 9. Industria, innovación e infraestructura**: el uso de tecnologías avanzadas para predecir ventas puede promover la innovación dentro de la industria de la moda, fomentando el desarrollo de infraestructuras tecnológicas más robustas y avanzadas. (8)
- **ODS 13. Acción por el clima**: mejorar la eficiencia en la cadena de suministro puede reducir las emisiones de carbono relacionadas con el transporte y la logística. Una mejor planificación puede significar menos viajes y más rutas eficientes. (9)

1.5 Estructura de la memoria

La estructura de la memoria es la siguiente:

- **Capítulo 1. Introducción**. Durante este capítulo explicamos el modelo actual de la empresa, cuáles son las limitaciones y cómo solucionarlas. Además, tenemos la motivación por parte de la empresa y personal cara al proyecto. Fijamos los objetivos, por un lado, el objetivo general, y por otro lado los objetivos específicos. Establecidos los objetivos, explicamos la metodología seguida, el impacto del proyecto en la empresa y las ODS. Y finalmente, acabamos explicando la estructura de la memoria.
- **Capítulo 2. Estado del arte.** Describimos cuál es la situación actual entorno al objetivo que queremos abordar, y acabamos con una crítica al estado del arte.
- **Capítulo 3. Análisis del problema.** Hacemos un análisis sobre el marco legal y ético y mostramos la planificación seguida para realizar el trabajo junto al presupuesto.
- **Capítulo 4. Fundamentos**. Explicamos cuáles son los modelos que proponemos y tratamos la parte más teórica de los modelos. Incluimos cuáles son las métricas de evaluación, dando una explicación sobre ellas. Por otra parte, explicamos cuál es la plataforma que hemos utilizado para el desarrollo del código, cuáles son los lenguajes de programación, y detallamos las librerías que hemos utilizado para crear los modelos, la representación de los datos, la explicabilidad y la optimización de hiperparámetros.
- **Capítulo 5. Análisis de datos.** Explicamos cuál es el dataset de partida, el procesamiento de datos, y vemos el comportamiento de las curvas de venta de los MOCACO en las tiendas, y se compara con las de a nivel mundo.
- **Capítulo 6. Experimentación**. Primero, explicamos lo que tienen en común todos los modelos, que es la muestra de evaluación del modelo. Luego, seguimos explicando los seis experimentos. Los dos primeros son los baselines y de los seis restantes, los tres primeros son MLP y los otros tres XGB con entrenamiento distribuido. Para cada experimento explicamos cuáles son las variables del dataset, el preprocesamiento, las variables que se han introducido para entrenar el modelo, la muestra de entrenamiento y su evaluación. Finalmente, comparamos los modelos y del mejor de ellos se hace la explicabilidad global y la optimización de hiperparámetros.
- **Capítulo 7. Conclusiones**. Vemos cuál es el trabajo realizado, el legado, la relación del trabajo desarrollado con el grado de Ciencia de Datos, las limitaciones y el trabajo futuro.
- **Bibliografía**
- **Anexos.** Tenemos tres anexos, en el primero encontramos la tabla de las ODS. Es el segundo todas las gráficas donde mostramos los clústeres de las curvas de venta a nivel tienda y a nivel mundo para compararlas y a su vez clasificar los MOCACO-tienda en fantasía, básicos y otros. En el tercero anexo, tenemos las gráficas que muestran la distribución del error por diferentes agrupaciones de cada uno de los experimentos realizados.

CAPÍTULO 2

Estado del arte

A lo largo de este capítulo, hacemos un estudio de otros proyectos que han abordado un problema similar, empezando por un problema más simple que ha evolucionado al problema nuevo que queremos abordar, vemos cuáles son los modelos utilizados y finalmente realizamos una crítica al estado del arte.

2.1 Análisis de la situación actual

El problema más común y simple con el que nos encontramos es realizar un forecasting de productos que ya teníamos en el mercado y queremos saber cuánto venderán en el futuro, por ello hemos investigado y tenemos estudios que abordan el problema de diferentes modos. En el trabajo (10) se utiliza los modelos Autoregressive and Integrated Moving Average (ARIMA) y Exponential Smoothing State (ETS) para afrontar el problema en el sector minorista, llegando a la conclusión de que tienen un rendimiento similar en términos de error de previsión impidiendo decantarse por uno de ellos.

Este mismo problema, en el trabajo (11) se resuelve aplicando ARIMA y Holt-Winter (HW) (modelos lineales), Wavelet Neural Networks (WNN) y Takagi-Sugeno Fuzzy System (TS) (modelos no lineales). Tras proponer diferentes modelos capaces de resolver el problema extraen la conclusión de que WNN fue el modelo más preciso, logrando una satisfacción mayor respecto a los otros modelos.

En el siguiente trabajo (12) proponen otra alternativa que consiste en utilizar redes neuronales feedforward (NN) y redes neuro-difusas (NF). El modelo NF ofrece un mejor rendimiento en comparación con el modelo NN. Este modelo combina redes neuronales y lógica difusa para manejar datos no lineales y proporcionar predicciones basadas en reglas.

Este otro, está enfocado en la industria minorista, donde genera pronósticos de ventas basándose en el algoritmo Prophet de Facebook (13), desarrollado para predecir series temporales. En todos estos casos, ya hay datos históricos con los que trabajar, una mejora que se muestra en este estudio es que trabajar con productos que tengan pocos datos, e incluso nuevos. Esto se debe a que todas las empresas relacionadas con el mundo de la venta desearían saber de antemano que rendimiento va a tener el producto que van a sacar a la venta.

Como vemos, el problema de obtener la predicción de productos existentes se puede resolver utilizando diferentes tipos de modelos, bien sean lineales o no lineales. Sin embargo, la mejora que se propone en el estudio anterior nos introduce en nuestra problemática, las soluciones planteadas tienen una limitación indicada en el trabajo

anterior, y es que para los productos nuevos que no tenemos datos históricos no podemos emplear los algoritmos desarrollados en los trabajos anteriores. Este es un problema muy importante en el sector del retail, ya que prever la venta de un producto cuando se lanza por primera vez es muy importante sobre todo para la gestión del stock, debido a que no tenemos referencias pasadas de cómo ha ido el producto.

Se han investigado diferentes estudios sobre la problemática en otros sectores que aportan una solución adaptada al problema según su sector. Por esta razón, veamos los diferentes enfoques que se llevan a cabo.

En el sector de la alimentación, en el que es de suma importancia tener la información de venta, ya que se trabaja con alimentos perecederos (14) se propone una solución que utiliza un enfoque basado en el aprendizaje por transferencia mediante una MLP. De este modo, se transfiere el conocimiento de productos existentes a nuevos productos, mejorando la precisión del pronóstico. Además, también se utiliza la predicción a nivel semanal, obteniendo un modelo entrenado para todas las semanas, es decir, un único modelo para todo.

Otra solución adaptada, en este caso al sector del retail, para la venta de equipamiento gastronómico en una tienda, deciden utilizar un árbol de decisión (15). En este caso, se realiza un único modelo que obtiene la predicción de los productos nuevos a nivel anual, no a nivel semanal.

Otro enfoque utilizado en el sector del retail son los modelos MuQAR (16). Se trata de una arquitectura de aprendizaje profundo que combina las características visuales, textuales y temporales utilizando un perceptrón multicapa multimodal (FusionMLP) y una red neuronal quasi-autorregresiva (QAR). Esto está ligado a nuestro problema porque trabajamos con prendas de vestir que pueden ser caracterizadas mediante etiquetas. En nuestro caso no aplicaremos un modelo multimodal, aunque sí que utilizamos información de las prendas, tenemos características que describen como son las prendas y esto se obtiene con modelos implementados por la empresa. Esto nos enseña un enfoque muy interesante que es añadir la descripción de lo que queremos predecir, ya que no tendremos ventas históricas, pero sí que se puede extraer información de la propia prenda.

El estudio (17) está en el mismo marco que el nuestro; dónde el objetivo principal es desarrollar modelos predictivos que permitan prever la demanda de nuevos artículos en la moda. Este enfoque es crucial para el sector de la moda debido a la introducción constante de nuevos productos y estilos. Para ello, utilizan modelos basados en árboles, en concreto el XGBoost y el gradient boosting regression trees (GBRT) y modelos de aprendizaje profundo, long short-term memory (LSTM) y multi-layer perceptron (MLP). Al evaluar los modelos, se obtiene que ofrece un mejor rendimiento el XGBoost cuando se optimiza con la pérdida cuadrática media en una escala logarítmica, seguido del GBRT, entre los dos modelos de aprendizaje profundo, el LSTM mostró un mejor desempeño, aunque se queda detrás de los basados en árboles.

Por último, se pueden utilizar diferentes enfoques según la necesidad, que es el método que se emplea en el sector de la gestión de la cadena de suministro y la

previsión de la demanda de nuevos productos. Su nombre es el DemandForest (18).Este algoritmo combina K-means, Random Forest y Quantile Regression Forest.

Como vemos hay diferentes técnicas que se utilizan para resolver el problema con artículos nuevos, pero todas ellas utilizan algoritmos como MLP y árboles de decisión.

Tras revisar los estudios realizados por otras entidades, veamos cuál es el enfoque que ha adoptado Inditex. Actualmente, la compañía dispone de un modelo a nivel mundial que realiza la predicción semanal de los artículos nuevos. Utilizan una red neuronal artificial (MLP) donde se incorporan variables relacionadas con artículos comparables y otras características de las prendas. A diferencia de los demás modelos propuestos, desarrollan diversos modelos enfocados en diferentes agrupaciones de los MOCACO.

2.2 Crítica al estado del arte

La mayoría de los estudios obtienen modelos como los descritos en estos (14), (15), (16), (17) y (18) que les permite alcanzar el objetivo de saber cuánto venderá el producto nuevo, pero el detalle con el que obtienen la información no es tan especifico como el que se quiere abordar en este trabajo.

En los siguientes modelos tenemos un gap, no están habilitados para poder hacer la predicción teniendo en cuenta la tienda, hace la predicción mundial. Tenemos el modelo obtenido a partir del estudio (14), que utiliza una MLP para obtener la predicción, a pesar de que no haga la predicción para cada tienda, sí que es capaz de hacer la predicción a nivel semanal. Lo mismo sucede con los estudios (15), (17) y (18). Respecto al estudio (16), se implementa un modelo multimodal, en nuestro caso no necesitamos este tipo de modelos porque todos los datos que se podrían extraer de imágenes, texto o vídeo ya los tenemos recopilados en bases de datos, por lo tanto, no sería necesario volver a calcularlo. No obstante, es un buen enfoque utilizar un modelo multimodal para obtener información de las prendas, sin embargo, mantiene el mismo gap que el resto de los modelos.

Otro gap que tenemos es el horizonte temporal de los modelos, en nuestro caso queremos tener la predicción semanal. En los modelos (15), (16), (17) y (18) la predicción se obtiene con un horizonte temporal más amplio, es decir mensual o anual. Esto hace que los modelos no den la información con el detalle necesario para poder hacer una buena gestión de las campañas. Sin embargo, en el modelo obtenido en el estudio (14) sí que es semanal.

En conclusión, tenemos dos gaps en la literatura, por un lado, la dimensión geográfica de la predicción, esto puede darse porque en los estudios no tenían un problema donde tienen que obtener la predicción para varias tiendas. Por otro lado, el horizonte temporal que utilizan para obtener la predicción.

Con nuestra solución, resolvemos los dos principales gaps encontrados en el estado del arte para resolver este tipo de problemas. Por un lado, añadimos la dimensión tienda, permitiendo obtener una predicción más detallada, no siendo únicamente a nivel

mundial. Por otro lado, añadimos que la predicción no sea agregada, es decir, que la predicción se haga para el ciclo de vida del producto, esto permite ser más exactos en el momento de la gestión.

Si juntamos estos dos aspectos, obtenemos un modelo mucho más detallado que los modelos que se tienen hoy en día. Pasando de una predicción global y agregada a una predicción local y desagregada. Nuestro trabajo consiste en cubrir los gaps existentes dando lugar a un modelo que obtiene la predicción de las ventas semanales de productos nuevos en las tiendas.

CAPÍTULO 3

Análisis del problema

En este capítulo vamos a hacer un análisis del marco legal y ético viendo que el proyecto cumple los requisitos necesarios y qué situaciones se tienen que controlar. Además, mostramos el plan de trabajo que hemos seguido mediante un Gantt y se acaba mostrando cuáles son los recursos que hemos utilizado para el desarrollo del proyecto

.3.1 Análisis del marco legal y ético

El marco legal es un conjunto de normas y principios que regulan el comportamiento de profesionales y organizaciones en áreas como la propiedad intelectual, el uso de datos sensibles y la ética. Por otro lado, el marco ético se refiere a una guía que orienta sobre aspectos importantes para actuar de forma correcta ante diferentes situaciones. (19)

Todos los datos utilizados en este proyecto son proporcionados por la empresa y se consideran confidenciales. Es obligatorio que, al presentar los resultados del proyecto a personas ajenas a la empresa o que no hayan firmado un Acuerdo de Confidencialidad (NDA) (20), no mostremos ningún dato explícito de la empresa. Además, hemos realizado un convenio de prácticas que nos permite utilizar los datos y herramientas de la compañía durante el periodo establecido en el convenio. Por estas dos razones, estamos comprometidos a no difundir ningún dato relacionado con el proyecto a terceras personas. Además, solo podemos mostrar información relacionada con los modelos y resultados agregados que no revelen información específica o sensible sobre la empresa, que han pasado el filtro del co-tutor de Inditex que valida que toda la información que aparece en la memoria se puede difundir en los repositorios de la UPV.

El acceso a los datos estará restringido a los miembros del equipo del proyecto que hayan sido autorizados y que hayan firmado un NDA. Los datos solo se utilizarán para los fines específicos del proyecto y no los compartiremos con terceros sin el consentimiento expreso de la empresa.

En nuestro caso, no utilizamos ningún dato que pueda identificar a los clientes, ya que la información más detallada que podemos obtener es la venta que ha habido de un MOCACO en una tienda en un día concreto. Por lo que cumplimos la privacidad de los datos, ya que no utilizamos datos personales de los clientes.

El impacto en los empleados es un aspecto ético que considerar. Las predicciones pueden influir en decisiones sobre la dotación de personal y la asignación de recursos. Si estas decisiones no se manejan con cuidado, pueden afectar

negativamente a los empleados, como reducciones de personal o cargas de trabajo injustas. Es fundamental que las predicciones del modelo se utilicen de modo que no perjudiquen a los trabajadores, ya que los resultados vienen derivados de las simulaciones, pero no pretenden servir como único elemento en la toma de decisiones que pueda afectar a los empleados.

3.2 Plan de trabajo

En la metodología vemos los pasos que hemos seguido para alcanzar el resultado, ahora nos centramos en la planificación y el tiempo utilizado en cada uno de los pasos, teniendo en cuenta que el proyecto consta en total de tres meses, 12 semanas a jornada completa.

La primera semana la dedicaremos a comprender el problema y a formarnos desde el proceso de las prendas hasta cómo utilizar Databricks, esta tarea acaba en el tiempo planificado, no tiene muchas dificultades ya que los ritmos están marcados y no requieren de desarrollo de código en el que pueden aparecer problemas.

Tras acabar esta primera semana que nos sirve para situarnos en la empresa, tenemos planificado una semana en la que la dedicaremos a estudiar el comportamiento de las curvas de venta de los MOCACO en las tiendas. La duración fue de una semana más de lo planificado, esto ocurre porque es la primera tarea que requiere desarrollo de código en Databricks, utilizando pyspark. Además, empezamos a utilizar las tablas donde están alojados los datos, lo que implica comprender el significado de las variables y saber referenciar a los MOCACO, para ello es necesario recibir asesoramiento del equipo.

Entrando en la tercera semana, tenemos el preprocesamiento de los baselines, teníamos estimado que la duración iba a ser de una semana, pero requiere ampliar una semana más. Este retraso se debe a que la tarea anterior se retrasó, entonces tuvimos que paralelizar ambas tareas, con las reuniones y otras tareas.

En la cuarta semana, desarrollamos el primer baseline, cumplimos con su duración de una semana, además, durante esta semana llevamos paralelamente lo que queda de la tarea del preprocesamiento y el desarrollo del baseline uno. En la siguiente semana, desarrollamos el baseline dos que también cumplimos con su duración de una semana.

Ya cerrados los baselines empezamos la semana cinco con el preprocesamiento de los experimentos que utilizan una MLP. Esta tarea tiene un retraso de una semana por el evento del TechDay que duró todo el día y una actividad del departamento que duró una mañana, esto hizo que tuviésemos que ampliar el tiempo planificado.

De la semana seis a la ocho, creamos y evaluamos los modelos que utilizan una MLP, el primer experimento está planificado para hacerlo durante dos semanas, donde también realizamos la función de muestreo, creamos la estructura que utilizamos en los otros experimentos y nos enfrentamos por primera vez a las limitaciones de memoria, por estas razones el experimento tiene dos semanas. El experimento donde se desarrolla la MLP dos, lo realizamos durante una semana, lo que hacemos es crear nuevas variables y mantenemos la misma estructura que en el experimento anterior. El

último experimento de las MLP dura dos semanas, tiene esta duración porque durante la semana ocho, paralelizamos el desarrollo de la MLP dos con el de la MLP tres, por esta razón dura una semana más.

Ya acabados los modelos que utilizan una MLP, nos centramos en los de XGBoost. En la semana nueve realizamos todo el preprocesamiento necesario para estos modelos, este proceso se realiza durante una semana. En la semana nueve desarrollamos simultáneamente el XGB 1 y el XGB 2, lo hacemos al mismo tiempo, porque reutilizamos la plantilla de los experimentos anteriores, pero adaptándola para un XGBoost. El XGB uno, lo hacemos más rápido porque no tenemos que desarrollar nuevas variables, ya que son las misma es que en el experimento de la MLP dos y MLP tres, esto nos permite estar casi a tiempo completo con el XGB dos.

Para el último modelo XGB, dedicamos dos semanas a su desarrollo debido a la necesidad de crear nuevas variables y ajustar algunos enfoques utilizados en modelos anteriores. No experimentamos retrasos en esta tarea, y era crucial cumplir con el tiempo estimado, ya que estábamos cerca del límite de los tres meses. Después de este periodo, perderíamos el acceso necesario para utilizar los datos.

En la última semana, donde estábamos acabando el XGB tres, también hacemos la comparación de los modelos, la explicabilidad del XGB tres y la optimización de los hiperparámetros.

Finalmente, logramos acabar el proyecto en el tiempo planificado, aunque tuvimos que realizar algunas tareas paralelamente que no estaban planificadas hacerlas de esta manera. No obstante, era necesario para que el proyecto estuviese hecho al final de las 12 semanas.

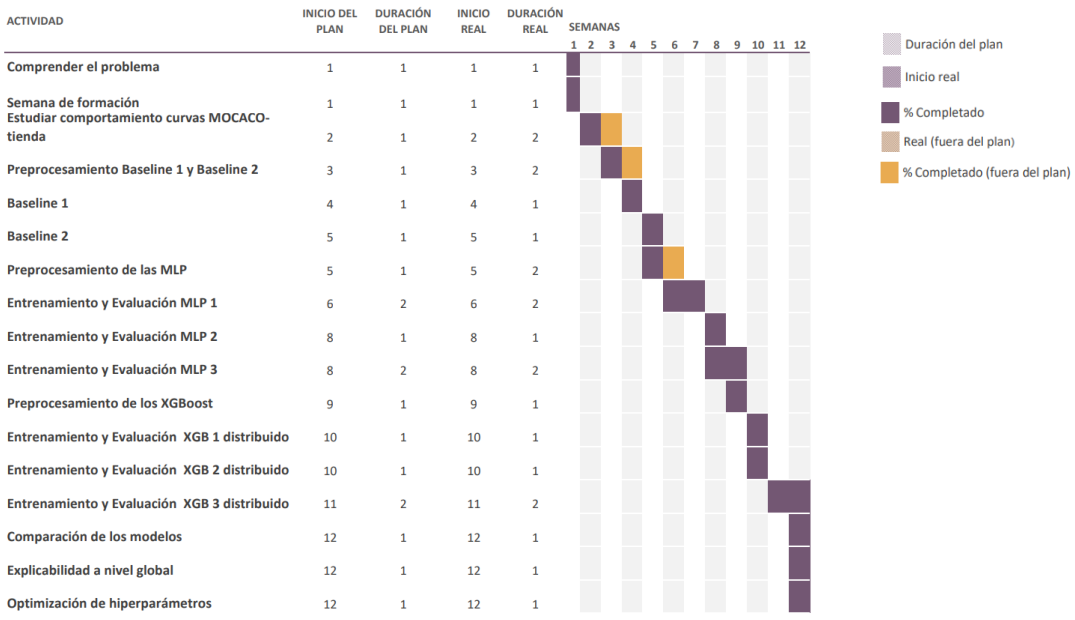


Ilustración 1. Diagrama de Gantt

3.3 Recursos

Para desarrollar el trabajo en tres meses hemos utilizado Databricks, que requiere una licencia, además dentro de la plataforma puedes utilizar un clúster para ejecutar los notebooks. En nuestro caso, para agilizar utilizamos dos clústeres que tienen las características de la ilustración dos. El precio es por hora, por lo que, para hacer una medida aproximada, diremos que está activado durante toda la jornada laboral, ocho horas al día.

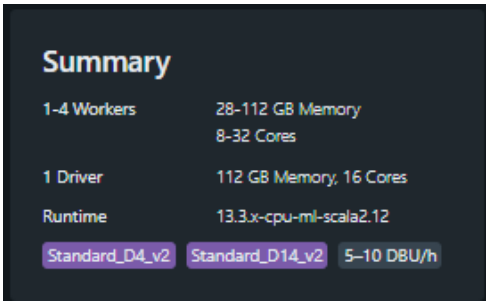


Ilustración 2. Características clúster 13.3 LTS ML

A parte de estos recursos, la empresa también nos facilitó todos los materiales necesarios ordenador, ratón y cascos. En concreto, el ordenador es el Lenovo 21 SBD700.

En cuanto a los datos, los tienen alojados en Snowflake y luego algunas de las tablas que se van utilizando las tienen alojadas en el workspace y se accede mediante pyspark. Calcular el coste que implica tener los datos almacenados en estas plataformas es imposible de calcular porque cada empresa tiene una tarifa y va en función de la cantidad de datos que almacenan. Además, como los datos ya los tienen almacenados y no se hace exclusivamente para este proyecto, no tendremos en cuenta el coste.

Respecto a los costes del personal, solo tendremos en cuenta el sueldo de las prácticas, ya que se ha llevado a cabo durante las prácticas y no tenemos colaboradores que desarrollen código.

Todos los precios que se indican son orientativos basándonos en los precios de mercado, esto es un coste aproximado ya que la empresa tiene acuerdos con las diferentes empresas que le dan servicios y el coste diferente para ellos. Por razones de privacidad, no pueden decir cuáles son los precios de cada uno de los productos.

Recursos	Duración	Precio	Coste
Clúster 1 de Databricks	480 h	2,5 €/h	1200 € (21)
Clúster 2 de Databricks	480 h	2,5 €/h	1200 € (21)
Sueldo prácticas		1500 €/mes	4500 €
Ordenador			1499,99 € (22)
Total			8399,99 €

Tabla 1. Coste del proyecto

CAPÍTULO 4

Fundamentos

En este capítulo vemos la parte teórica de los modelos que utilizamos en los diferentes experimentos, incluyendo el modelo de la solución propuesta para el problema. Además, incluimos la parte teórica de las métricas de evaluación, la plataforma, los lenguajes de programación y las librerías utilizadas.

4.1 Fundamentos teóricos

Vamos a empezar con una explicación de los Perceptrones Multicapa (MLP), que es uno de los tipos de redes neuronales que existen. Y seguiremos con el modelo de Refuerzo de Gradientes Extremo (XGBoost). Ambos son modelos muy potentes que permiten obtener predicciones muy cercanas al valor real, además son modelos que como hemos visto anteriormente se utilizan para abordar problemas de predicción y clasificación.

4.1.1 Redes Neuronales Artificiales

Las redes neuronales artificiales (23) son modelos que se originan con base en el sistema nervioso central de las personas, que es el encargado de procesar información a través de neuronas. Esto permite que puedan aprender de forma automática mediante el entramiento supervisado, es decir, con datos donde ya se tiene la variable respuesta.

La estructura de la red sigue el mismo patrón que el de la ilustración tres, donde tenemos una capa de entrada, las capas ocultas y la capa de salida. La red está formada por neuronas artificiales interconectadas masivamente. Cada neurona es una unidad de procesamiento que procesa información, calculando la suma ponderada de las entradas utilizando unos pesos asignados (w_{ij}), donde el término bias (w_0) siempre se suma. Para evitar que la combinación de neuronas genere una operación lineal, se aplica una función de activación (φ) que introduce no linealidades entre las operaciones (24).

$$y_i = \varphi_i \left((\sum_{j=1}^{m_i} w_{ij}\, x_{ij}) + w_{i0} \right)$$

y_i: valor de salida de la neurona i.

φ_i: función de activación para la neurona i.

m_i: cantidad de entradas a la neurona i.

w_{ij}: peso para la entrada j de la neurona i, w_{i0} será el bias.

El aprendizaje automático es el encargado de ir ajustando los pesos del modelo para que se obtenga el máximo rendimiento. Para obtener el máximo rendimiento lo que hacemos es minimizar el error, por esta razón el entrenamiento se trata como si fuese un problema de minimización del error. El algoritmo realiza un proceso iterativo, donde cada iteración la llamamos epoch, y se busca mejorar los pesos (w_{ij}). Para controlar la convergencia, las mejoras se suavizan por la tasa de aprendizaje (n) (25).

$$Wij = Wij + n * get_upgrade(wij; train_data)$$

Para determinar las mejoras de los pesos, se utilizan algoritmos de descenso por gradiente y la retro-propagación del error, que propaga el error hacia atrás a través de la red, ajustando los pesos en cada capa para reducir el error global. Estos algoritmos emplean una muestra representativa de los datos para calcular el error del modelo y los gradientes de los pesos respecto al error.

Cuando se trabaja con grandes volúmenes de datos (26), es común utilizar algoritmos de optimización por lotes para introducir mayor aleatoriedad en el proceso. Una variante frecuente es el algoritmo de Descenso de Gradiente Estocástico (SGD), que también añade aleatoriedad al entrenamiento. Una mejora importante de este algoritmo es la implementación de tasas de aprendizaje adaptativas, que ajustan el tamaño del paso en función de la magnitud y el historial del gradiente. Un método popular que sigue esta idea es la Estimación de Momentos Adaptativos (ADAM), el cual combina las técnicas de RMSProp y Momentum, utilizando parámetros para controlar la tasa de decaimiento exponencial del primer y segundo momento de los gradientes. ADAMW, una versión mejorada de ADAM, corrige un problema en la implementación del decaimiento de pesos de ADAM, lo que ayuda a evitar una convergencia subóptima.

Para abordar nuestro problema, lo que utilizaremos es un tipo de RNA muy popular, la MLP. Donde las neuronas están estructuradas en capas y están totalmente conectadas entre capas adyacentes. De modo, que la salida de una capa será la entrada de la siguiente capa.

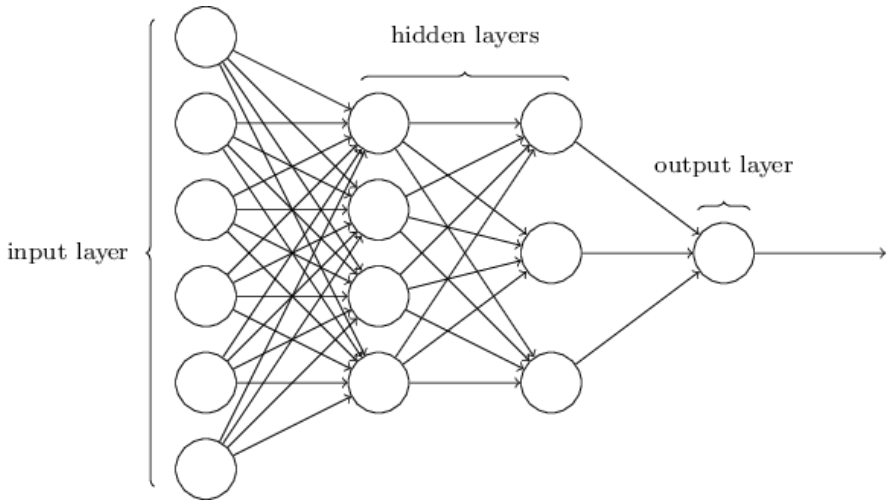


Ilustración 3. Estructura de una MLP

Fuente: (42)

4.1.2 XGBoost

XGBoost (27) es un algoritmo de aprendizaje automático de árboles de decisión que aumenta el gradiente. Estos árboles de decisión son algoritmos de ML supervisados que pueden utilizarse tanto para problemas de clasificación como de regresión. De modo que se predice el valor de una variable objetivo mediante el aprendizaje de unas reglas de decisión que se van aprendiendo en la fase de entrenamiento.

En este modelo se empieza con un árbol de decisión y se van agregando nuevos árboles para corregir los errores que han cometido los anteriores, este proceso se hace hasta que no se pueda realizar más mejoras. De ahí viene el aumento de gradiente, los modelos nuevos corrigen los modelos anteriores y luego se combinan para obtener la predicción final. Para ello se utiliza el algoritmo de descenso de gradiente, así se minimiza el error de predicción al agregar nuevos modelos.

Destaca por su buen rendimiento y la por su rapidez, esto ocurre debido a la capacidad de utilizar diversos procesadores para trabajar a la vez, realizando tareas más pequeñas que llevan a completar la tarea más grande.

El proceso de XGBoost (28) (29) comienza diferenciando el conjunto de entrenamiento y la variable objetivo. Se selecciona una función de pérdida para medir el error de predicción y se define la tasa de aprendizaje para controlar el ritmo de aprendizaje del modelo para evitar el sobreajuste.

$$\widehat{f_\theta}(x) = \arg\min_\theta \sum_{i=1}^{N} L(y_i, \theta)$$

El argmin se utiliza para obtener el valor que minimiza la función. La función de perdida mide la diferencia entre los valores reales y las predicciones del modelo, y tiene como objetivo encontrar los parámetros que minimicen esta función, es decir, busca los parámetros que reduzcan al máximo el error.

El objetivo de XGBoost es minimizar la función de pérdida $L(y, F(x))$ encontrando los parámetros óptimos del modelo. argmin se utiliza para describir este proceso de minimización. La estimación inicial θ sirve como punto de partida, y aunque comienza con un gran error de predicción, este error se reduce iterativamente mediante la adición de nuevos árboles que corrigen los errores de predicción anteriores. Cada iteración del algoritmo mejora el modelo y reduce el error de predicción, acercándose cada vez más a los valores óptimos que minimizan la función de pérdida.

Para todo $m \in \{1, 2, 3, ..., M\}$, se calcula los gradientes y las matrices de Hesse y así se aumenta el gradiente de los árboles.

$$\widehat{g_m}(x_i) = \left[\frac{\partial L\left(y_i, f(x_i)\right)}{\partial f(x_i)}\right]_{f(x)=\widehat{f_{(m-1)}}(x)}$$

$\widehat{g_m}(x_i)$: gradiente.

$L\left(y_i, f(x_i)\right)$: función de pérdida, mide la discrepancia entre la predicción del modelo $f(x_i)$ y el valor real y_i.

y_i: valor real del dato i-ésimo.

$f(x_i)$: predicción del modelo para la muestra i-ésima.

$\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}$: derivada parcial de la función de pérdida con respecto a la predicción $f(x_i)$. Mide cómo cambia la pérdida cuando cambia la predicción.

$\widehat{f_{(m-1)}}(x)$: predicción del modelo en la iteración $(m-1)$.

$$\widehat{h_m}(x_i) = \left[\frac{\partial^2 L\left(y_i, f(x_i)\right)}{\partial f(x_i)^2}\right]_{f(x)=\widehat{f_{(m-1)}}(x)}$$

$\widehat{h_m}(x_i)$: Hessiano.

$\frac{\partial^2 L(y_i, f(x_i))}{\partial f(x_i)^2}$: segunda derivada de la función de pérdida con respecto a la predicción $f(x\_i)$. Mide la curvatura de la pérdida en relación con la predicción, se usa para ajustar el modelo teniendo en cuenta cómo cambia la tasa de cambio de la pérdida.

Los gradientes indican cómo cambiar las predicciones del modelo para reducir la función de pérdida. El hessiano es la derivada del gradiente, proporciona información sobre la tasa de cambio del gradiente, permitiendo ajustes más precisos. Juntos, permiten que el algoritmo de XGBoost optimice el modelo de manera eficiente y efectiva, mejorando continuamente las predicciones en cada iteración.

Usando las nuevas matrices, se crea otro árbol que resuelve el siguiente problema de optimización para cada iteración del algoritmo.

$$\widehat{\phi_m} = \arg\min_{\phi \in \Phi} \sum_{i=1}^{N} \frac{1}{2} \widehat{h_m}(x_i) \left[\frac{\widehat{g_m}(x_i)}{\widehat{h_m}(x_i)} - \phi_m(x_i)\right]^2$$

$\widehat{\phi_m}$: Mejor aproximación del ajuste del modelo en la iteración m-ésima

$\arg\min \phi \in \Phi$: busca la función (ϕ) que minimiza la expresión del sumatorio.

(Φ): es el espacio de funciones posible.

$\hat{h}_m(x_i)$: valor Hessiano para la muestra i-ésima en la iteración m.

$\hat{g}_m(x_i)$: valor del gradiente para la muestra i-ésima en la iteración m.

$\left[\frac{\widehat{g_m}(x_i)}{\widehat{h_m}(x_i)} - \phi_m(x_i)\right]^2$: Es el término de error cuadrático que mide la discrepancia entre el valor ajustado $\frac{\widehat{g_m}(x_i)}{\widehat{h_m}(x_i)}$ y la función $\phi_m(x_i)$. Este término penaliza las grandes diferencias entre estos dos valores, asegurando que el modelo se ajuste adecuadamente.

La siguiente fórmula es par la actualización del modelo, veamos que es cada parámetro:

$$\widehat{p_m}(x) = \alpha\widehat{\phi_m}(x)$$

$\widehat{p_m}(x)$: predicción del modelo en la m-ésima iteración.

α: Es un parámetro de regularización que controla la contribución de la nueva función $\widehat{\phi_m}(x)$ al modelo global. Este parámetro puede ajustarse para evitar sobreajuste.

Para resolver este problema se utiliza una aproximación de Taylor permite simplificar el problema de optimización al usar derivadas de primer y segundo orden (gradientes y hessianos) en lugar de calcular la pérdida exacta en cada iteración. Esto facilita el uso de técnicas de optimización tradicionales. La tasa de cambio del gradiente indica cuánto debe ajustarse el modelo, gradientes pronunciados indican la necesidad de cambios significativos, mientras que gradientes planos indican que el modelo está cerca de la convergencia. Esta técnica permite a XGBoost optimizar de manera eficiente y mejorar continuamente el modelo durante el proceso de aumento de gradiente.

El siguiente paso es agregar los nuevos árboles al modelo anterior:

$$\widehat{f_m}(x) = \widehat{f_{m-1}}(x) + \widehat{p_m}(x)$$

$\widehat{f_m}(x)$: predicción del modelo combinado en la iteración m para la entrada x.

$\widehat{f_{m-1}}(x)$: predicción del modelo combinado en la iteración $m - 1$ para la entrada x.

$\widehat{p_m}(x)$: nuevo árbol de decisión ajustado en la iteración m que se va a añadir al modelo combinado.

Este proceso se va repitiendo, entrenando árboles de decisión. Donde en cada iteración se ajusta un nuevo árbol para minimizar la función de pérdida, corrigiendo los errores anteriores. Y al final el resultado es la suma de todos los árboles.

$$\hat{f}(x) = \widehat{f_M}(x) = \sum_{m=0}^{M} \widehat{f_m}(x)$$

$\hat{f}(x)$: predicción final del modelo para la entrada x.

$\widehat{f_M}(x)$: predicción del modelo en la iteración M para la entra x.

$\sum_{m=0}^{M} \widehat{f_m}(x)$: sumatorio de todas las predicciones de los modelos individuales desde $m = 0$ hasta $m = M$.

4.1.3 Métricas

Para la evaluación de los modelos vamos a utilizar dos métricas. La primera de ellas es el Weighted Absolute Percentage Error (WAPE) y luego utilizamos otra métrica que está basada en ella, el Weighted Weekly Absolute Percentage Error (WWAPE). Las métricas elegidas son estas, debido a que se tiene que seguir el criterio seguido en el modelo a nivel mundo realizado por la empresa.

El **WAPE** (30) es una métrica que pondera el error absoluto porcentual por las ventas reales. En nuestro caso, como tenemos un modelo por semana, obtendremos 21 WAPE, indicándonos el error por semana del modelo resultante al juntar los 21 modelos.

$$\mathrm{APE} = \left|\frac{y - \hat{y}}{y + 1}\right| \times 100$$

$$\mathrm{WAPE} = \frac{\sum_i (\mathrm{APE}_i \cdot y_i)}{\sum_i y_i}$$

y_i: valor real de ventas para el ítem i.

$\hat{y}_i$: valor predicho de ventas para el ítem i.

$\sum_{i \in w}$: la suma se realiza de todos los ítems i en la semana w.

El **WWAPE** viene derivada del WAPE, agrega un nivel de agregación adicional. Una vez tenemos el error a nivel semanal (error de cada uno de los modelos), se pondera el WAPE de cada semana según el total de ventas semanales, proporcionando una métrica más global que considera la variabilidad en las ventas de cada semana. De esta forma, calculamos el error global del modelo (modelo resultante de juntar los 21 modelos).

$$\mathrm{WWAPE} = \frac{\sum_w (\mathrm{WAPE}_w \times \sum_{i \in w} y_i)}{\sum_w \sum_{i \in w} y_i}$$

Modelos	
Multi-layer Perceptron	Es un tipo de red neuronal artificial compuesta por múltiples capas de neuronas que se utilizan para el aprendizaje supervisado.
XGBoost	Es un algoritmo de aprendizaje automático basado en árboles de decisión que utiliza técnicas de boosting para mejorar iterativamente el rendimiento del modelo al minimizar el error de predicción y aumentar su precisión.

Tabla 2. Resumen de los modelos

Métricas	
WAPE	Proporciona el error de cada uno de los modelos semanales que forman el modelo global.
WWAPE	Proporciona el error del modelo global (modelo resultante de juntar los 21 modelos)

Tabla 3. Resumen de las métricas de evaluación

4.2 Fundamentos tecnológicos

4.2.1 Plataforma y lenguajes de programación

Para desarrollar el código, utilizamos **Databricks** (31)**,** plataforma utilizada por la empresa para el desarrollo de código. Es una plataforma unificada y abierta que facilita la analítica de datos y está para crear, implementar, compartir y mantener soluciones empresariales de datos, análisis e inteligencia artificial a gran escala. Se conecta con el almacenamiento y la seguridad en la nube de la cuenta, gestionando y desplegando la infraestructura necesaria.

Combina la inteligencia artificial generativa y un data lakehouse para interpretar la semántica específica de los datos. Luego, ajusta automáticamente el rendimiento y administra la infraestructura conforme a los requisitos del usuario. Además, el procesamiento de lenguaje natural permite buscar y descubrir datos haciendo preguntas en lenguaje natural, y también ofrece ayuda para escribir código, solucionar errores y encontrar información en la documentación.

Ofrece herramientas que te ayudan a conectar tus fuentes de datos en una única plataforma, permitiendo procesar, almacenar, compartir, analizar, modelar conjuntos de datos. El entorno de trabajo proporciona una interfaz unificada y herramientas para diversas tareas relacionadas con los datos, como:

- Programación y gestión del procesamiento de datos
- Creación de visualizaciones
- Gestión de seguridad, gobernanza y recuperación ante desastres.
- Descubrimiento, anotación y exploración de datos.
- Modelado, seguimiento y despliegue de modelos de aprendizaje automático.
- Implementación de soluciones de inteligencia artificial generativa.

Facilita que los datos estén disponibles, limpios y organizados de manera eficiente, combinando Apache Spark, Delta Lake y herramientas personalizadas para ofrecer una experiencia ETL superior. Puedes usar SQL, Python y Scala para crear lógica ETL y programar trabajos de manera sencilla.

El lenguaje de programación que hemos utilizado para el tratamiento de los datos, el desarrollo de los modelos y la visualización de los datos es **Python** (32). Es un lenguaje de programación utilizado especialmente para el desarrollo de software y la ciencia de datos. En nuestro caso, lo utilizamos porque es el lenguaje de programación que hemos utilizado durante el grado y con el que hemos aprendido a desarrollar modelos de machine learning.

Guido Van Rossum, un programador de los Países Bajos creó Python en 1989. Inspirado en "Monty Python's Flying Circus", Python se lanzó en 1991. Sus versiones posteriores, como Python 1.0 en 1994 y Python 2.0 en 2000, introdujeron mejoras significativas. En 2008, Python 3.0 marcó un hito importante con características como la compatibilidad con Unicode y mejoras en la gestión de errores. Este lenguaje de programación se ha convertido en una herramienta poderosa y versátil utilizada en diversas aplicaciones y sectores.

Ofrece una serie de beneficios significativos para los programadores. Los programas escritos en este lenguaje son fácilmente legibles y comprensibles debido a su sintaxis simple y cercana al lenguaje natural. Además, generalmente se necesita menos líneas de código en comparación con otros lenguajes de programación.

Un aspecto que facilita su uso son las librerías que evita la necesidad de escribir código desde cero y agilizando el desarrollo. Python cuenta con una amplia biblioteca estándar que proporciona una gran cantidad de módulos y funciones listas para usar, permitiendo a los desarrolladores acceder a códigos reutilizables para casi cualquier tarea. A parte de esta biblioteca estándar, hay más de 137.000 bibliotecas disponibles.

Las principales características son:

- Lenguaje interpretado: esto se refiere a que el código se ejecuta línea por línea, por lo que, si hay algún error, en la ejecución se detiene y es más fácil detectar donde está y solucionarlo.
- Lenguaje tipeado dinámicamente: no hace falta especificar el tipo de variable, ya que lo detecta automáticamente
- Lenguaje de alto nivel: es similar al lenguaje natural, es decir, es similar a los idiomas que utilizamos las personas. Esto evita la preocupación de la arquitectura o la administración de la memoria.
- Lenguaje orientado a objetos: el código se organiza alrededor de objetos que tienen datos y funciones asociadas. Los objetos pueden interactuar entre sí, lo que permite una estructura más modular y reutilizable del código. Esto se logra a través de conceptos como encapsulación, herencia y polimorfismo, que ayudan a simplificar el desarrollo y el mantenimiento del software.

Por último, tenemos **SQL** (33), que lo utilizamos para la obtención de los datos, fusión de tablas y el filtrado. Seleccionamos este lenguaje porque Databricks permite utilizarlo al igual que Python, y permite combinar ambos lenguajes, lo que es idóneo para el desarrollo del proyecto. Es un lenguaje de programación diseñado para administrar y manipular bases de datos relacionales. Permite a los usuarios realizar diversas operaciones en una base de datos, como consultar datos, insertar nuevos registros, actualizar información existente y eliminar datos.

Se inventó en la década de 1970 con base en el modelo de datos relacional. Al inicio se conocía como el lenguaje de consultas estructuradas. Mas tarde, el término se abrevió a SQL. Oracle, antes conocido como Relational Software, se convirtió en el primer proveedor en ofrecer un sistema comercial de administración de bases de datos relacionales SQL.

Plataforma y Lenguajes de programación	
Databricks	Plataforma de análisis de datos y aprendizaje automático en la nube que facilita la colaboración, el procesamiento de grandes volúmenes de datos y el desarrollo de modelos de machine learning mediante el uso de Apache Spark.
Python	Lenguaje de programación de alto nivel, interpretado y de propósito general, conocido por su sintaxis clara y legible, que facilita el desarrollo rápido de aplicaciones en una amplia variedad de dominios, desde el desarrollo web hasta el análisis de datos y la inteligencia artificial.
SQL	Lenguaje de programación utilizado para gestionar y manipular bases de datos relacionales, permitiendo la ejecución de consultas para insertar, actualizar, eliminar y recuperar datos.

Tabla 4. Resumen de la plataforma y lenguajes de programación

4.2.2 Librerías

Pandas (34) es una biblioteca para la manipulación de datos. Es eficiente con el manejo de datos, y permite trabajar con volúmenes grandes de datos estructurados. Simplifica el proceso de limpieza, filtrado y transformación de datos

Una de las librerías más importantes es **pyspark** (35), gracias a ella hemos podido trabajar con grandes cantidades de datos. Es una API de Python para Apache Spark que permite hacer el procesamiento de datos a gran escala en tiempo real y en un entorno distribuido. Nos permite utilizar las funciones de Spark, como Spark SQL, que permite trabajar con datos estructurados, combinando consultas SQL y capacidades de Spark. Esto ofrece la ventaja de poder leer, escribir, transformar y analizar datos de manera eficiente utilizando tanto Python como SQL.

MLlib es una biblioteca de aprendizaje automático escalable construida sobre Spark (36). Proporciona un conjunto uniforme de API de alto nivel, lo que facilita la creación y el entrenamiento distribuido de modelos de aprendizaje automático. De esta forma podemos entrenar modelos como el XGBoost de forma distribuida.

Para crear las MLP utilizamos dos librerías, una de ellas en **TensorFlow** (37). Es una librería de código libre para el aprendizaje automático desarrollada por Google. El objetivo de esta librería es crear, entrenar y validar redes neuronales artificiales.

En conjunto con TensorFlow, empleamos **Keras** (34), una biblioteca de alto nivel para el aprendizaje profundo que funciona sobre TensorFlow. Con Keras, podemos centrarnos en diseñar las capas de las redes neuronales mientras delegamos los detalles técnicos de los tensores, sus dimensiones y sus operaciones matemáticas internas. Esta herramienta nos permite ejecutar aplicaciones de aprendizaje profundo sin tener que lidiar directamente con TensorFlow.

Scikit-learn es una biblioteca de Python que incluye una variedad de algoritmos de aprendizaje automático diseñados para abordar problemas supervisados y no supervisados de tamaño moderado (38). Su objetivo principal es hacer que el aprendizaje automático sea accesible para aquellos que no son expertos en el campo, mediante el uso de un lenguaje de programación de alto nivel. Se prioriza la facilidad de uso, el rendimiento, la calidad de la documentación y la consistencia de la interfaz de programación de aplicaciones.

Para optimizar los hiperparámetros, nos apoyamos en **Hyperopt** (39), una biblioteca diseñada para trabajar con algoritmos de entrenamiento distribuido. Esta herramienta genera múltiples pruebas con diversas configuraciones de hiperparámetros, siendo cada una ejecutada desde el nodo controlador. Esto garantiza acceso completo a todos los recursos disponibles en el clúster. Hyperopt es compatible con una amplia gama de algoritmos de aprendizaje automático distribuido, e incluye el seguimiento automático de MLflow y la clase SparkTrials para la sincronización distribuida

Por último, nos queda la librería **Dython** (40), es un conjunto de herramientas de análisis de datos en Python 3.x que permite obtener una comprensión de los datos de forma sencilla. Al proporcionar el conjunto de datos, identifica automáticamente cuales son categóricas y numéricas, calcula una medida de asociación relevante para cada una y genera un mapa de calor.

Librerías	
Pandas	Usada para la manipulación y análisis de datos, que proporciona estructuras de datos flexibles y potentes, como DataFrames, para trabajar de manera eficiente.
Pyspark	Interfaz de Python para Apache Spark, permite el procesamiento y análisis de grandes volúmenes de datos distribuidos de manera eficiente.
MLlib	Biblioteca de aprendizaje automático de Apache Spark, proporciona herramientas y algoritmos escalables diseñadas ejecutarse en entornos distribuidos.
TensorFlow	Biblioteca de código abierto desarrollada para la computación numérica y el aprendizaje automático, facilita la creación y el entrenamiento de modelos de redes neuronales.
Keras	API de alto nivel para el desarrollo de modelos de redes neuronales, facilita la implementación rápida de modelos de aprendizaje profundo.
Scikit-learn	Biblioteca de aprendizaje automático de código abierto que proporciona una amplia gama de algoritmos y herramientas para la evaluación de modelos y preprocesamiento de datos.
Hyperopt	Enfocada en la optimización de hiperparámetros, utiliza algoritmos de búsqueda adaptativa para encontrar automáticamente la mejor combinación de hiperparámetros para modelos de aprendizaje automático, minimizando una función objetivo.
Dython	Proporciona herramientas para simplificar y acelerar el proceso de análisis de datos.

Tabla 5. Resumen de librerías

CAPÍTULO 5

Análisis de datos

A lo largo del capítulo, veremos cuál es la tabla de partida y otras que hemos utilizado para obtener nuevas variables con las que trabajar y crear nuevos modelos. También veremos la explicación del preprocesamiento, diferenciando tres tipos de preprocesamiento: uno que se hace para los baseline, otro para los modelos basados en una MLP y el tercero que se utiliza en los modelos que emplean XGBoost. Por último, estudiaremos el comportamiento de las curvas de los MOCACO en las tiendas, comparándolas con las curvas de venta a nivel mundial y clasificando su comportamiento en fantasía, básico y otro.

5.1 El dataset

Como hemos comentado anteriormente, cada modelo está entrenado con una serie de variables, ya que van cubriendo limitaciones de los experimentos anteriores y buscamos que cada uno sea mejor que el anterior. El punto de partida es la tabla seis.

Para obtener el dataset de cada modelo, se utilizan una serie de tablas con la que podremos adquirir variables adjuntándolas directamente al dataset, o bien, hacer ingeniería de variables

Para comprender la información que contiene el dataset principal y el resto de las tablas vamos a hacer una descripción de cada una. Primero vamos a describir las tablas de spark:

En esta primera tabla, lo que encontramos en cada registro es cuántas unidades se han vendido de un MOCACO en una tienda. Esta información está asociada a una fecha, por lo que los datos se registran diariamente.

Tabla principal de ventas	
sku_artc	Identificador del MOCACO (se construye a partir de id_article e id_color)
Id_article	Identificador del articulo
Id_color	Identificador del color del artículo
Id_product	Identificador del tipo de producto
Id_campaign	Identificador de la campaña
Id_store	Identificador de la tienda
Id_sales_channel	Identificador del tipo de venta, online o en tienda física
Id_brand	Identificador de la marca
Id_section	Identificador de la sección a la que pertenece el MOCACO dentro de la marca
Date	Fecha en la que se realiza la venta del MOCACO en la tienda

sales	Unidades del MOCACO que se han vendido en la tienda en la fecha indicada

Tabla 6. Dataset inicial

En los modelos introducimos una serie de etiquetas como inputs para obtener una descripción de los MOCACO y así obtener una predicción más precisa. Veamos cuáles son las etiquetas que encontramos en dicha tabla para cada MOCACO:

Tabla de etiquetas	
sku_artc	Identificador del MOCACO
Id_article	Identificador del articulo
Id_color	Identificador del color del artículo
Id_product	Identificador del tipo de producto
Id_campaign	Identificador de la campaña
Id_store	Identificador de la tienda
Id_sales_channel	Identificador del tipo de venta, online o en tienda física
Id_brand	Identificador de la marca
Id_section	Identificador de la sección a la que pertenece el MOCACO en la marca
altos_final	Indica el tipo de alto
anchura_final	Indica el tipo de anchura
Cuellos_final	Indica el tipo de cuello
Mangas_final	Indica el tipo de manga
Estilo	Indica el estilo
Forma	Indica la forma
Grupo_prendas	Indica a que grupo de MOCACO pertenece

Tabla 7. Tabla de etiquetas

En esta tabla nos encontraremos una serie de información para cada MOCACO que nos ayuda con la creación del dataset.

Tabla de familia	
sku_artc	Identificador del MOCACO
id_article	Identificador del artículo
id_color	Identificador del color del artículo
id_brand	Identificador de la marca
id_section	Identificador de la sección dentro de cada marca
id_product	Identificador del tipo de producto
id_campaign	Identificador de la campaña
base_price	Precio base del MOCACO
cod_family	Código de familia del MOCACO
cod_subfamily	Código de la subfamilia del MOCACO
buyer_code	Código del comprador

Tabla 8- Tabla de familia

En la Tabla 9, cada registro hace referencia a un MOCACO donde se tienen diferentes datos a nivel diario. Es importante tener en cuenta que la información que se obtiene a partir de la tabla es a nivel mundial y no a nivel tienda. Es decir, cuando se describe una variable, se está hablando de su efecto a nivel mundial. Por ejemplo, "initial_purchase" es la compra inicial de stock de un MOCACO a nivel mundial, no indica que se ha comprado una cantidad específica de stock de un MOCACO en una tienda determinada.

Tabla de compra inicial	
sku_artc	Identificador del MOCACO
id_article	Identificador del artículo
id_color	Identificador del color del artículo
id_brand	Identificador de la marca
id_section	Identificador de la sección dentro de cada marca
id_product	Identificador del tipo de producto
id_campaign	Identificador de la campaña
date	Fecha de venta
id_sales_channel	Tipo de venta (online o tienda física)
first_day	Primer día de venta
day_indiex	Indica cuantos días han pasado desde el primer día que se vendió el MOCACO
sales	Ventas del MOCACO en la tienda en la fecha indicada
n_stores	Números de tiendas en la que se vende el MOCACO
n_countries	Número de países en el que se vende el MOCACO
cumulative_purchase_units	Compra cumulativa de stock del MOCACO
initial_purchase	Compra inicial de stock del MOCACO

Tabla 9. Tabla de compra inicial

Nos encontramos ante una tabla donde cada registro es un MOCACO y la información de interés está relacionada con el número de tiendas.

Tabla de número de tiendas	
sku_artc	Identificador del MOCACO
id_article	Identificador del articulo
id_color	Identificador del color del artículo
id_brand	Identificador de la marca
id_section	Identificación de la sección dentro de la marca
id_product	Identificación del tipo de articulo
id_campaign	Identificación de la campaña
id_sales_channel	Identificación del tipo de venta
n_stores_first_week	Número de tiendas en las que se vende el MOCACO a nivel mundial, en su primera semana de vida
n_stores_total	Número de tiendas en las que se ha vendido en total el MOCACO

Tabla 10. Tabla de número de tiendas

En esta última tabla nos indica a que país pertenece da tienda donde se venden los MOCACO:

Tabla de país	
id_store	Identificador de la tienda
country_iso	Abreviatura del país
id_brand	Marca a la que pertenece la tienda
cod_country	Código del país

Tabla 11. Tabla de país

Todas las tablas anteriores son tablas de Spark, ahora las siguientes tablas son de Snowflake, la única diferencia es el método de la carga de datos. Algunas de las tablas están en Spark y otras en Snowflake porque las que están almacenadas en el Catalog de Databricks, las de Spark, las utilizamos únicamente los miembros del departamento y son tablas que tienen un preprocesamiento hecho, las tablas de Snowflake son tablas que pueden acceder todas las personas de la empresa y obtener los datos.

En la siguiente tabla, cada registro hace referencia a un MOCACO donde se describe el tipo de la curva de venta, se diferencian tres tipos:

- Fantasía: MOCACO que las primeras semanas de vida tiene un volumen alto de venta y luego baja su venta. Es decir, artículos que en sus primeras semanas tienen un pico de venta muy alto
- Básicos: MOCACO que su venta es constante a lo largo de su ciclo de vida
- Otros: engloba a los MOCACO que tienen un comportamiento diferente al descrito

Respecto a los tipos de curva de venta, más adelante se verá un estudio sobre estos tipos de curva. Veamos que variables encontramos en la Tabla 12.

Tabla de tipo de curva	
sku_artc	Identificador del MOCACO
type	Descripción del tipo de curva de venta
tipo_venta	Tipo de venta de la descripción de la curva del MOCACO
seccion	Identificador de la sección
id_product	Identificador del tipo del artículo

Tabla 12. Tabla de tipo de curva de venta del MOCACO

Un dato muy interesante que permite obtener información sobre los MOCACO similares es la tabla en la que se registran, para cada MOCACO, sus MOCACO similares de las campañas anteriores y de la misma campaña, ordenados por similaridad.. Esta similaridad viene derivada del proyecto de Zaramatch, un proyecto del departamento que realiza el cálculo a partir de embedings de los MOCACO y calculan las distancias de estos para ver la similaridad.

Tabla de MOCACOS similares	
sku_artc	Identificador del MOCACO comparado
Id_campaign	Identificador de la campaña del MOCACO comparado
Sku_artc_comparable	Identificador del MOCACO similar
Id_campaign_comparable	Identificador de la campaña del MOCACO similar
Id_position	Orden de similaridad

Tabla 13. Tabla de similaridad

Ahora nos encontramos con la tabla de stock, donde podremos ver las unidades que se exportan a cada tienda:

Tabla de stock RFID agregado expuesto	
Producto	Descripción del tipo de producto
Cod_campana	Identificador de la campaña
Campana	Descripción de la campaña
Cod_seccion	Identificador de la sección
Seccion	Descripción de la sección
Cod_mercado	Identificador del país de la tienda
Mercado	Descripción del país de la tienda
Id_cadena	Identificador de la cadena
Cadena	Descripción de la cadena
cod_tienda	Identificador de la tienda
Tienda	Descripción de la tienda
Uds_expo	Unidades exportadas a la tienda
Fecha_hora	Fecha y hora en la que se ha realizado la exportación

Tabla 14. Tabla de envío inicial del MOCACO a tienda

Para todas las tablas en las que se recopila la información se establece una serie de criterios para reducir el análisis.

Nos vamos a centrar en ZARA en la sección de mujer, porque en los procesos experimentales siempre empiezan por esta sección. Dentro de todos los MOCACO que se venden, solo nos centraremos en aquellos que son prendas de vestir, como camisetas, pantalones, chaquetas, entre otros. No incluiremos los complementos, como joyas, ni los productos de belleza, como maquillaje y calzado. Además, solo tenemos datos para los días en los que hay ventas; si no ha habido venta, no tendremos datos registrados.

Teniendo claro en qué marca, sección y con qué tipo de productos vamos a trabajar, nos queda aclarar el último filtrado. Solo trabajaremos con las campañas de invierno 2023, verano 2023, invierno 2022 y verano 2022. La elección de estas campañas se basa en seleccionar la última campaña completa para hacer la inferencia de los modelos y poder evaluarlos. Luego, se seleccionan las tres campañas anteriores a esta para entrenar los modelos.

5.2 Comportamiento de las curvas de venta

Para ver el comportamiento de las curvas de venta utilizamos las semanas del ciclo de vida del MOCACO en la tienda. Para hacer el análisis, seleccionamos los MOCACO de todas las familias, y tiendas de todos los países, se calculan los percentiles respecto a la compra inicial del MOCACO a nivel mundial y al volumen de venta de las tiendas. De este modo obtenemos una muestra lo más equitativa posible.

Para la muestra generada aplicamos un "Functional Data Analysis" (FDA) proporcionado por los desarrolladores del modelo a nivel mundo, de modo que obtenemos 10 clústeres. Una vez que tenemos los clústeres, como queremos ver la comparación con las curvas de venta a nivel mundial, lo que hacemos es contar de cada clúster a nivel tienda cuántos MOCACO tenemos en cada uno de los clústeres a nivel mundo. De este modo, podemos graficar solamente aquellos clústeres que tienen un peso mínimo, este peso también tiene en cuenta el tamaño de cada clúster, así los clústeres pequeños tienen el mismo peso que los más grandes.

Para entender las curvas de venta recordemos que tipo de curvas tenemos.

Tipos de curva de venta	
Fantasía	MOCACO que tiene mucha venta en las primeras semanas de su ciclo de vida y luego baja para el resto de las semanas.
Básico	MOCACO que tiene una venta establa a lo largo de las semanas del ciclo de vida.
Otro	MOCACO que no sigue ninguna de las distribuciones anteriores.

Tabla 15. Tipos de curva de ventas

Tras este proceso, podemos ver el comportamiento que tienen los MOCACO en las tiendas y comprarlas con las curvas de venta que tienen a nivel mundial. Veamos algún ejemplo:

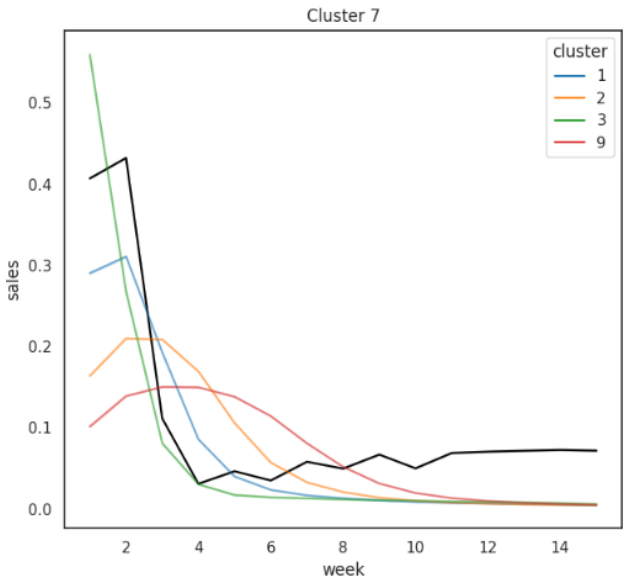


Ilustración 4. Comparación curva de venta del clúster 7 de nivel tienda con los clústers a nivel mundo que tienen al menos una representación del 5%

Clúster mundo	Peso
1	38,1%
2	17,61%
3	28,34%
9	6,1%

Tabla 16. Distribución de los pesos de los clústeres a nivel mundo en el clúster 7 de tienda

En negro tenemos la curva de venta del clúster a nivel tienda y en colores tenemos los clústeres a nivel mundo, pero solo visualizamos los que cumplan un peso mínimo del 5%. Podemos apreciar como la distribución es similar a la de los clústeres a nivel mundo, en concreto a los clústeres tres y uno, aunque no exactamente igual. Estos son un claro ejemplo de un clúster formado por fantasías, donde en las primeras semanas tienen una gran venta y luego disminuye hasta ser prácticamente nula.

Veamos otro caso donde la distribución es totalmente diferente:

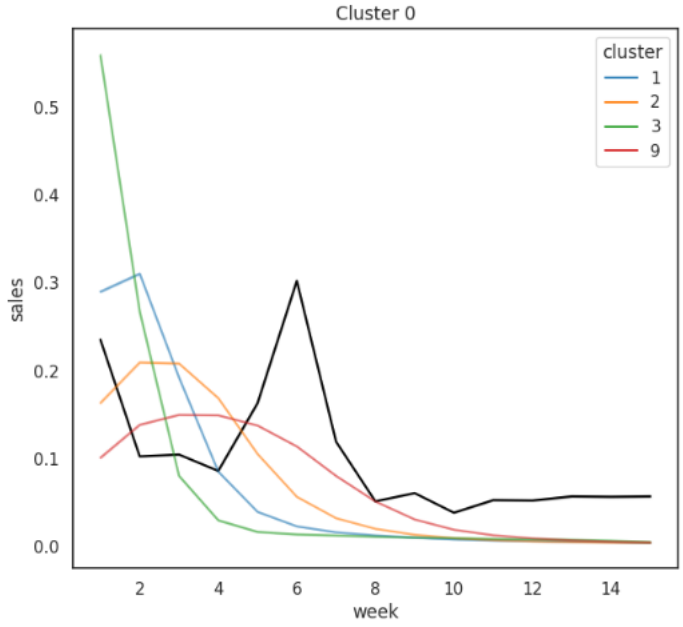


Ilustración 5. Comparación curva de venta del clúster 0 de nivel tienda con los clústeres a nivel mundo que tienen al menos una representación del 5%

Clúster mundo	Peso
1	38,99%
2	19,28%
3	23,38%
9	7,1%

Tabla 17. Distribución de los pesos de los clústeres a nivel mundo en el clúster 0 de tienda

En este caso vemos como la forma de la curva es totalmente diferente a la de los clústeres a nivel mundo que tienen un mayor peso. Este caso es el que sucede en la mayoría de los casos, mostrando que el comportamiento de los MOCACO en las tiendas es diferente que a nivel mundo. Este sería un caso de curva de la clase de otros, ya que no tiene una gran venta en las primeras semanas y tampoco tiene una venta estable. En el apéndice se pueden ver todas las gráficas.

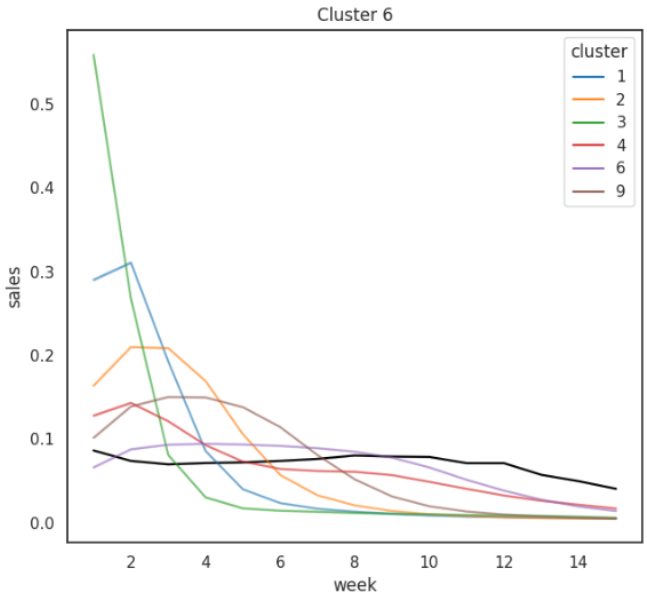


Ilustración 6. Comparación curva de venta del clúster 6 de nivel tienda con los clústeres a nivel mundo que tienen al menos una representación de 5%

Clúster mundo	Peso
1	22,97%
2	22,82%
3	5,5%
4	11,89%
6	11,4%
9	15,78%

Tabla 18. Distribución de los pesos de los clústeres a nivel mundo en el clúster 6 de tienda

Este es un claro ejemplo de una curva de un MOCACO básico, ya que tiene una venta estable a lo largo de todas las semanas y no tienen ningún pico de venta.

Con este análisis lo que queremos ver es que a nivel tienda, también podemos diferenciar los MOCACO en fantasías, básicos y otros. Pero, no todos los MOCACO se comportan igual a nivel mundo y a nivel tienda, pudiendo ser en algunos casos fantasías en una tienda y a nivel mundo un básico o uno de la clase otro. Sin embargo, hay muchos que sí que coinciden y lo que sí que se cumple es la forma de clasificar los productos en fantasía, básicos y otros. Tras contrastar la información obtenida del análisis con el encargado del proyecto, decidimos utilizar la clasificación de los MOCACO a nivel mundo y en un futuro se realizará la clasificación de los MOCACO en las tiendas.

5.3 Muestreo

Nuestro conjunto de datos está formado por millones de registros, en concreto para la primera semana tenemos unos 14 millones, es decir, 14 millones de combinaciones de MOCACO-tienda, por esta razón es importante generar una muestra para poder trabajar con los modelos.

Recordamos que para entrenar los modelos utilizamos las tres campañas anteriores a la última campaña completa disponible, es decir, utilizamos las campañas verano 2023, invierno 2022 y verano 2022 para entrenar los modelos, y la campaña de invierno 2023 para la inferencia.

Cuando generamos una muestra es esencial que esta sea representativa de la población. Para garantizar que se cumpla este requisito lo que hacemos es seleccionar los MOCACO de todas las tiendas, familias, países y campañas.

Vamos a presentar cuales son los experimentos que veremos más adelante, pero tenemos que hacerlo para entender el muestreo, no vamos a hacer una descripción de cada uno de ellos, solo vamos a ver cómo están formados:

- Experimento 1. Baseline 1: un único modelo
- Experimento 2. Baseline 2: un único modelo
- Experimento 3. MLP 1: un modelo formado por 21 modelos, uno por semana
- Experimento 4. MLP 2: un modelo formado por 21 modelos, uno por semana
- Experimento 5. MLP 3: un modelo formado por 21 modelos, uno por semana
- Experimento 6. XGB 1: un modelo formado por 21 modelos, uno por semana
- Experimento 7. XGB 2: un modelo formado por 21 modelos, uno por semana
- Experimento 8. XGB 3: un modelo formado por 21 modelos, uno por semana

Cada experimento predice las primeras 21 semanas del MOCACO en la tienda. Los dos primeros experimentos son un modelo único, pero los experimentos restantes son 21 modelos, uno por semana, que dan lugar al modelo. Es importante entenderlo porque no tendremos una única muestra de entrenamiento, lo que tendremos es una muestra de entrenamiento por semana.

Para calcular la muestra de entrenamiento dividimos los experimentos en tres secciones, en la primera tenemos los baselines, en la segunda los modelos que utilizan una MLP y en la tercera sección los modelos que utilizan una XGBoost. De este modo, los modelos que se encuentren en la misma sección tendrán la misma muestra de entrenamiento.

Para los baselines la muestra de entrenamiento será todo nuestro conjunto de datos, ya que estos permiten entrenar con todos los MOCACO-tienda debido a que son modelos deterministas y no tardan mucho en preparar los pesos que necesitan para obtener los pesos.

Respecto a la segunda sección que pertenece a los modelos que utilizan las MLP, obtenemos una muestra por semana que cada una tendrá como máximo 500.000 individuos, recordamos que cada individuo en las muestras de entrenamiento son MOCACO-tienda. Y para que sea representativa de la población obtenemos individuos MOCACO-tienda para que estén representadas todas las tiendas, familias de los MOCACO y las tres campañas de entrenamiento.

En relación con la tercera sección donde encontramos los experimentos que utilizan XGBoost, su muestra de entrenamiento es de 1.500.000 individuos MOCACO-tienda. La muestra es de un millón más que la de los experimentos de la segunda sección. En esta muestra se siguen los mismos pasos descritos anteriormente para que la muestra sea representativa.

Ya hemos explicado la muestra de entrenamiento de los experimentos, ahora nos queda explicar la muestra de prueba. En este caso, sí que tenemos una única muestra de prueba para todos los experimentos, ya que es necesario que sea igual para todos y así poder evaluar sobre la misma prueba, sino podríamos obtener inconsistencias a la hora de evaluar.

Para obtener la muestra de prueba únicamente utilizamos la última campaña completa, la de invierno 2023. Obtenemos una muestra representativa del 500.000 MOCACO-tienda. Obtenemos una muestra global porque los modelos hacen la predicción de las primeras 21 semanas.

Experimento	Muestra de entrenamiento	Muestra de prueba
Baseline 1	Todo el conjunto de datos	500.000 individuos
Baseline 2	Todo el conjunto de datos	500.000 individuos
MLP 1	500.000 individuos por semana	500.000 individuos
MLP 2	500.000 individuos por semana	500.000 individuos
MLP 3	500.000 individuos por semana	500.000 individuos
XGB 1	1.500.000 individuos por semana	500.000 individuos
XGB 2	1.500.000 individuos por semana	500.000 individuos
XGB3	1.500.000 individuos por semana	500.000 individuos

Tabla 19. Tamaño de muestra de entrenamiento y prueba por experimento

Para seleccionar el tamaño de las muestras lo que hemos hecho es entrenar los modelos con diferentes tamaños de muestra y escoger aquel que no da error de memoria y es el más grande posible-

5.4 Preprocesamiento

5.4.1 Baseline 1 y 2

Hasta el momento, tenemos el conjunto de datos a nivel diario, para realizar estos dos modelos lo que hacemos es calcularlo a nivel semanal, pero la semana hace referencia a la semana del año, los valores van de uno a 53. De este modo, en cada registro tenemos cuántas unidades hemos vendido del MOCACO en la tienda en una semana del año concreta.

Además, antes de hacer la suma de las ventas filtramos para que como mínimo la venta diaria sea 0, ya que no queremos que los modelos hagan predicciones negativas.

5.4.2 Perceptrón multicapa (MLP)

En el caso de los modelos que son una MLP, el preprocesamiento de datos es diferente al del baseline 1 y 2. En este caso, el modelo solo puede recibir como input un conjunto de datos que sea de pandas. Por esta razón, todo el preprocesado que describimos lo hacemos con la librería de scikit-learn.

En este conjunto de datos, no trabajamos con las semanas del año, ahora lo que hacemos es calcular la venta por semana, pero la semana hará referencia a la semana del ciclo de vida del MOCACO en la tienda. De este modo, para cada MOCACO en cada tienda tenemos datos desde su semana uno en la tienda hasta la última que se tenga registro.

En estos modelos tenemos variables categóricas y numéricas, sin embargo, dentro de las numéricas tenemos un subtipo, que son aquellas que vienen derivadas de

una fecha. De modo, que las variables que se introduzcan en el modelo y vengan de una fecha se les hará un tratamiento diferente.

Empezando por las categóricas, aplicamos el One Hot Encoding, es una técnica muy utilizada que se basa en construir columnas binarias. Para cada variable generamos tantas columnas como categorías tenga, poniendo uno si el valor corresponde al de la categoría y cero en el caso contrario.

Para las numéricas aplicamos el Standard Scaler, de este modo obtenemos todas las variables con valores de cero a uno, es decir, normalizamos haciendo que su media sea cero y su desviación típica uno. Lo que hace el scaler es calcular la media y la desviación estándar de cada variable y luego transformar cada valor restando la media y dividiendo por la desviación estándar.

Para las numéricas que vienen derivadas de una fecha, lo que aplicamos es el escalado cíclico. Es una técnica que se utiliza para transformar variables con valores cíclicos que tiene una naturaleza circular, donde el valor máximo se conecta con el valor mínimo, y, por lo tanto, se le aplica la transformación sinusoidal y cosinusoidal, evitando la periodicidad

5.4.3 Extreme Gradient Boosting (XGBoost)

En el caso de estos modelos, el preprocesado consiste en lo mismo que en el de los modelos que usan una MLP, pero se aplican con pyspark, esto hace que necesitemos aplicar más preprocesadores provenientes pyspark ML.

En este caso también trabajamos con un conjunto de datos igual al de las MLP, donde la semana hace referencia a la semana del ciclo de vida del MOCACO en la tienda. Además, tenemos variables categóricas, numéricas y cíclicas.

En el caso de las categóricas, antes de aplicar el One Hot Encoding tenemos que aplicar un String Indexer, que consiste en pasar de cadena a numérica. De este modo, cada categoría está representada por un número, se mantiene la misma estructura de categorías, pero estas ahora son números. Ahora ya podemos aplicar el One Hot Encoding. Al final, aplicamos un Vector assembler para tener todas las categóricas en un único vector.

Con las variables numéricas, primero tenemos que aplicar un Vector Assembler para poner todas en un vector y aplicamos el Standard Scaler. Respecto a las cíclicas se aplica la transformación sinusoidal y cosinusoidal.

El último paso, es aplicar un Vector Assembler para juntar todas las variables preprocesadas en un único vector para poder introducírselo al modelo como input.

CAPÍTULO 6

Experimentación

Vamos a ver ocho experimentos, donde los dos primeros corresponden a los baselines, los tres siguientes a experimentos que utilizan modelos MLP y los tres siguientes utilizan modelos XGBoost con entrenamiento distribuido. El propósito es que cada modelo sea mejor que el anterior, para ello creamos nuevas variables, cambiamos el enfoque de las variables, cambiamos la arquitectura del modelo o resolvemos cualquier limitación que pueda tener su modelo anterior.

Al final, comparamos los modelos, obtenemos el que mejor rendimiento tiene y realizamos la optimización de hiperparámetros junto a la explicabilidad global del modelo.

6.1 Experimento 1: baseline 1

6.1.1 Explicación del funcionamiento

Recordamos que el conjunto de datos con el que trabajamos está a nivel de semana del año, es decir, de la semana uno a la 53.

Este primer baseline consiste en calcular para cada tienda un peso constante, es decir, que cada tienda tendrá un peso asignado que es igual para todas las semanas del año. Este peso se calcula como el porcentaje de venta que representa la tienda. Una vez tenemos los pesos, lo que hacemos es desagregar la predicción a nivel mundo del MOCACO, mediante una multiplicación de los pesos de las tiendas por las predicciones que hace el modelo actual de la compañía.

Si tomamos un MOCACO, desagregamos la predicción de cada semana por los pesos de las tiendas y al final tenemos cuantas unidades se van a vender del MOCACO en cada tienda.

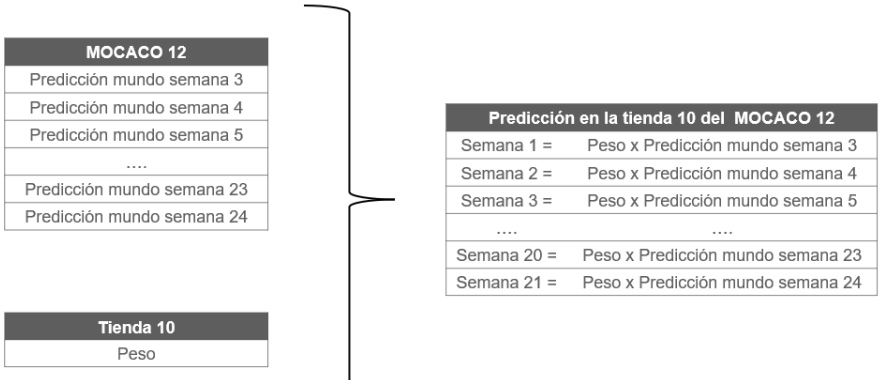


Ilustración 7. Explicación del funcionamiento del Experimento 1

6.1.2 Evaluación

Este primer experimento tiene un **WWAPE** de **63,92%**, esto significa que tiene un error de predicción de 63,92% del valor real. veamos su gráfica de WAPE:

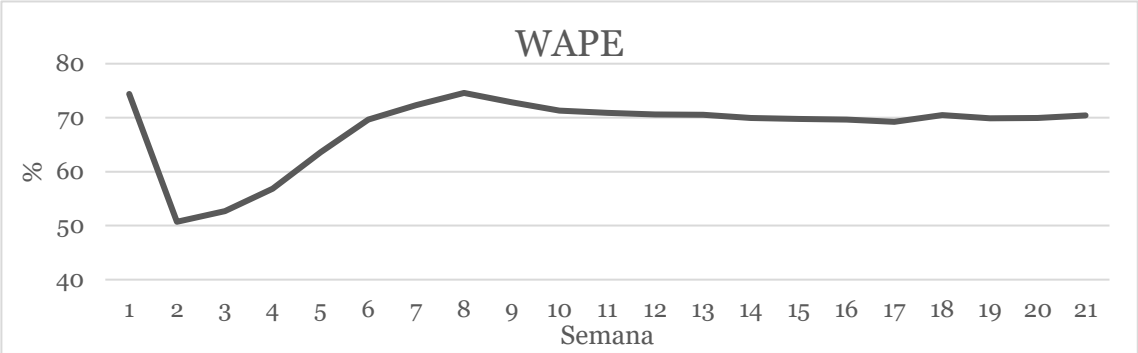


Ilustración 8. Gráfica del WAPE para el Experimento 1

El WAPE comienza en un valor alto, aproximadamente 75%. Este valor inicial indica que el modelo tenía un error significativo en la primera semana, sugiriendo una precisión baja en sus predicciones iniciales.

En la segunda semana se observa una caída, que desciende hasta aproximadamente al 50%. Esta disminución sugiere una mejora en la precisión del modelo entre la semana 1 y la semana 2.

Después de la semana 2, vuelve a aumentar nuevamente, alcanzando un pico de alrededor de 75% en la semana 8. A partir de la semana 9, se estabiliza en torno al 70%. Aunque hay pequeñas variaciones, en general, el error se mantiene constante cerca de este valor.

6.2 Experimento 2: baseline 2

6.2.1 Explicación del funcionamiento

Este segundo modelo, también trabaja con las semanas del año. Y el concepto es similar al anterior, pero en este caso en vez de tener pesos fijos para las tiendas, cada una tendrá un peso distinto en cada semana del año. Ahora para calcular el peso lo que hacemos es ver el porcentaje de venta representa la tienda en cada semana del año.

Entonces, se tiene para cada tienda un peso distinto según la semana de año y desagregamos la predicción del MOCACO a nivel mundo multiplicando los pesos. Al final el resultado es el mismo que el modelo anterior.

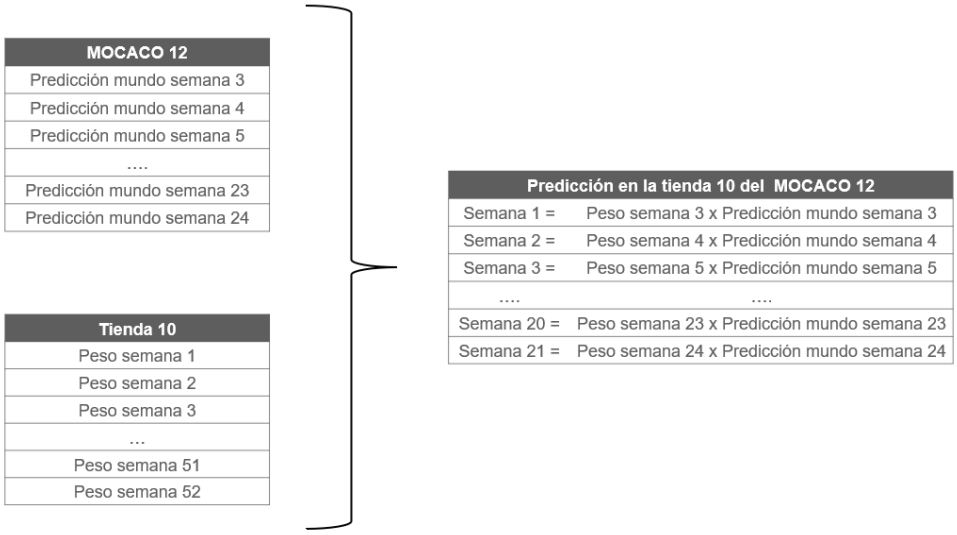


Ilustración 9. Explicación del funcionamiento del Experimento 2

6.2.2 Evaluación

Este modelo tiene un **WWAPE** de **73, 17%**, indica que el modelo tiene un error del 73,17% respecto al valor real. En un principio, se pensaba que este modelo iba a mejorar por tener en cuenta las diferentes semanas, sin embargo, vemos que el error aumenta en 9,25%.

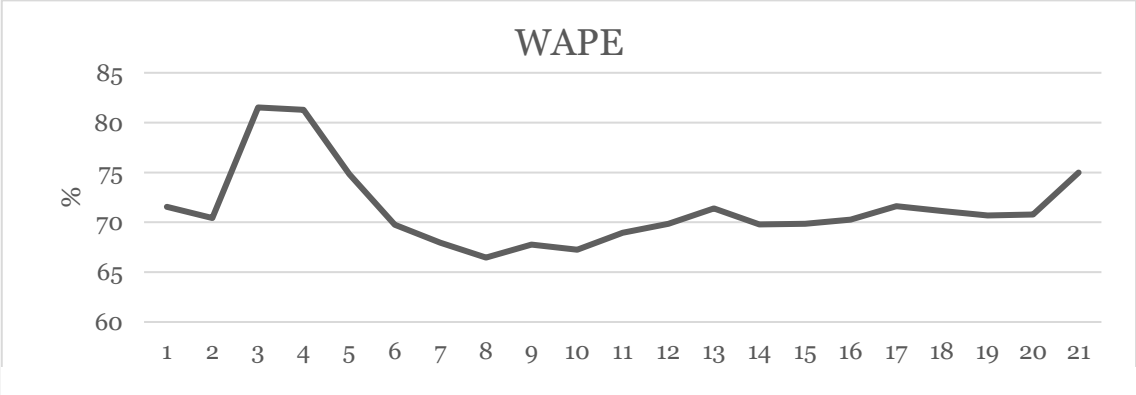


Ilustración 10. Gráfica del WAPE para el Experimento 2

Comienza en un valor cercano al 70% en la semana uno. Durante las primeras dos semanas, muestra una leve tendencia a la baja. Sin embargo, en la semana tres, se observa un aumento significativo, alcanzando un valor máximo de aproximadamente 81,5%. Esto indica que el modelo tuvo un desempeño particularmente malo en esta semana, con un error considerablemente mayor en sus predicciones.

Después del pico en la semana tres, comienza a disminuir de manera constante, alcanzando su punto más bajo alrededor de la semana ocho, con un valor de 66,,46%. Este descenso sugiere que el modelo mejoró su precisión durante este período.

A partir de la semana nueve, se estabiliza en un rango entre 65% y 70% hasta la semana 13, y a partir de la semana 13 ya no baja del 70% y el error gira en torno a este valor hasta la última semana.

6.3 Experimento 3: MLP 1

Este es el primer modelo de machine learning que creamos, consiste en una MLP. Para hacer la prueba de cómo funciona el modelo empezamos con una estructura simple.

Veamos cuáles son las variables que generamos, su estructura y los resultados en la evaluación.

6.3.1 Ingeniería de características

Primero tenemos la variable extraída de first_day_store, que es un dato de tipo fecha que se obtiene de la variable date seleccionando la primera fecha que se ha vendido el MOCACO en la tienda. De esta variable obtenemos First_day_weekofyaer_store.

Todas las variables que hacen referencia a las características de los MOCACO se extraen de una tabla alojada en Snowflake, y ponemos en nuestra base de datos aquellas características que corresponde a los MOCACO con los que estamos trabajando. Lo mismo sucede con las variables base_price, buyer_code e initial_purchase.

Solo nos quedan cuatro variables que están relacionadas con el número de tiendas, todas se obtienen de la misma tabla, solo que para dos de ellas filtramos para seleccionar que la venta sea en tienda física y para las otras dos que sea en venta online.

En la tabla de la siguiente página tenemos la explicación de cada variable y de que tipo es para el conjunto de datos que se utiliza en este modelo.

Columna	Tipo	Descripción
Id_color	Numérica	Identificador del color
Id_brand	Numérica	Identificador de la marca
Id_section	Numérica	Identificador de la sección
Id_product	Numérica	Identificador del producto
Id_campaign	Numérica	Identificador de la campaña
Id_store	Numérica	Identificador de la tienda
Id_sales_channel	Numérica	Identificador del tipo de venta
Sku_artc	Categórica	Identificador del MOCACO
Id_article	Numérica	Identificador del artículo
First_day_store	Fecha	Fecha del primer día de venta del MOCACO en la tienda
Sales_curve_types	Categórica	Tipo de curva de venta
Week_store	Numérica	Semana de vida del MOCACO en la tienda
Sales_week_store	Numérica	Unidades de venta
First_day_weekofyear_store	Numérica	Número de la semana del año que corresponde First_day_store
n_stores_first_week_fisica	Numérica	Número de tiendas en las que se ha hechos ventas físicas en la primera semana de vida del MOCACO
n_stores_total_fisica	Numérica	Número de tiendas en las que se han hecho ventas físicas a lo largo de toda la vida del MOCACO en la tienda
n_stores_first_week_online	Numérica	Número de tiendas en las que se ha hechos ventas online en la primera semana de vida del MOCACO
n_stores_total_online	Numérica	Número de tiendas en las que se han hecho ventas online a lo largo de toda la vida del MOCACO en la tienda
Altos_final	Categórica	Tipo de alto
Anchura final	Categórica	Tipo de anchura
Cuellos_final	Categórica	Tipo de cuello
Mangas_final	Categórica	Tipo de mangas
Estilo	Categórica	Tipo de estilo
Forma	Categórica	Tipo de forma
Grupo_prendas	Categórica	Grupo de prendas pertenece
Base_price	Numérica	Precio base
Cod_family	Numérica	Identificador de la familia
Cod_subfamily	Numérica	Identificador de la subfamilia
Buyer_code	Numérica	Identificador del comprador

Tabla 20. Conjunto de variables del Experimento 3

6.3.2 Entrenamiento

En la tabla X, tenemos en subrayado aquellas variables que se han utilizado para entrenar el modelo y a las que se les ha explicado el preprocesamiento explicado en el apartado de preprocesamiento.

En este tercer experimento, empezamos ya a introducir los modelos de machine learning, en este caso, vamos a utilizar una MLP. La estructura que hemos seguido es, una capa de entrada de tamaño del número de variables tras el preprocesado, en total son 2.007. Después tenemos una primera capa oculta densamente conectada con 256 neuronas y activación “LeakyReLU”, seguidamente tenemos una capa de dropout con una tasa del 30%. Le sigue otra capa de oculta densamente conectada con 128 neuronas y activación “LeakyReLU”, pasamos por una capa de normalización por lotes y acabamos con la capa de salida que es otra capa densamente conectada con activación “Relu” y una neurona. Respecto a la función de pérdida y el optimizador, utilizamos el error absoluto medio y Adam, respectivamente.

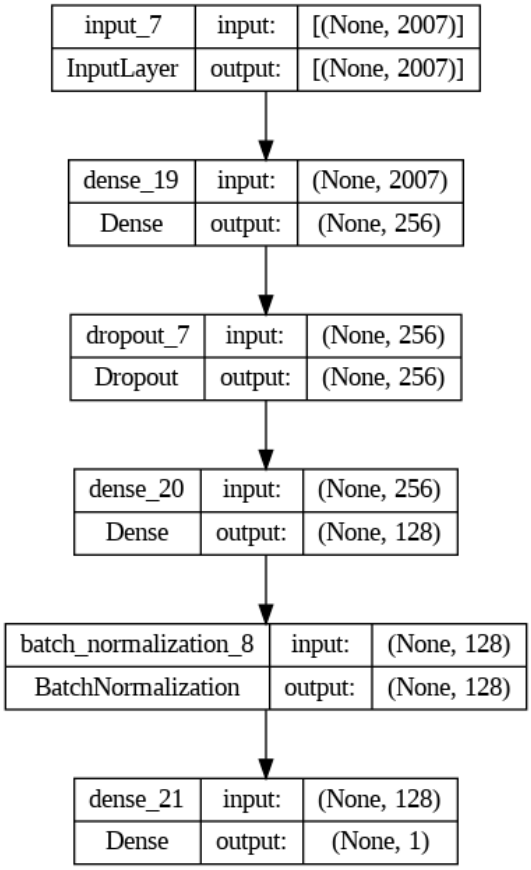


Ilustración 11. Estructura de las MLP del Experimento 3

6.3.3 Evaluación

Tras entrenar el modelo y evaluarlo con la muestra de prueba que estamos utilizando en todos los modelos, obtenemos un **WWAPE** de **52,84%**, lo que indica que el modelo tiene un error del 52,84% respecto al valor real.**.**

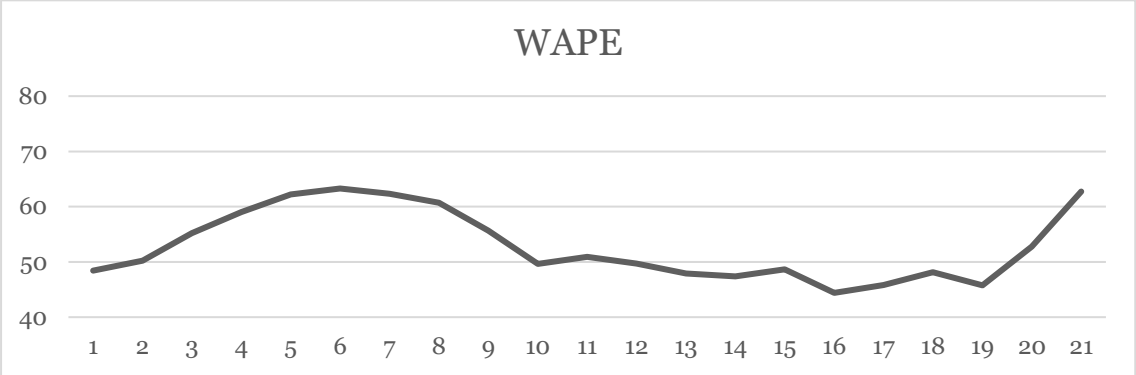


Ilustración 12. Gráfica del WAPE del Experimento 3

Este es el resultado tras calcular el WAPE, como vemos el error aumenta a partir de la semana dos hasta la seis y a partir de ahí decrece gradualmente hasta la semana 20 donde ya sube otra vez. El error en total oscila entre el 44% y el 64%.

6.4 Experimento 4: MLP 2

Antes de cambiar la estructura del modelo, vamos a probar a añadir más variables y ver si el modelo obtiene una mejor evaluación. Por lo que, utilizamos la misma estructura que en el experimento 3 y las variables descritas anteriormente más las nuevas que hemos creado.

6.4.1 Ingeniería de características

En relación con las variables temporales creamos nuevas para la tienda y añadimos la dimensión a nivel mundial, se extraen de first_day_store y first_day_world. Las nuevas que creamos son first_day_month_store, first_day_halfofmonth_store, first_day_month_world, first_day_halfofmonth_world, first_day_weekofyear_world. En la Tabla 21 tenemos la descripción de cada una.

Introducimos nuevas variables relacionadas con los MOCACO similares. Para obtener la información de los similares, lo que hacemos es obtener los 15 MOCACO más similares a cada MOCACO que tenemos en nuestra base de datos y a partir de ellas en cada caso hacemos una serie de cálculos. Con “avg_sales_week_store_comparables” lo que hacemos es calcular la media de venta que han tenido los similares en la tienda para esa semana, solo se tienen en cuenta aquellos MOCACO similares que han tenido venta en esa tienda esa semana. Con relación a avg_initial_purchase_comparables es similar a la variable anterior, pero en vez de calcular la media de las ventas, calculamos la media de la compra inicial.

En la siguiente página, tenemos la descripción de cada variable y el tipo de variable.

Columna	Tipo	Descripción
Id_color	Numérica	Identificador del color
Id_brand	Numérica	Identificador de la marca
Id_section	Numérica	Identificador de la sección
Id_product	Numérica	Identificador del producto
Id_campaign	Numérica	Identificador de la campaña
Id_store	Numérica	Identificador de la tienda
Id_sales_channel	Numérica	Identificador del tipo de venta
Sku_artc	Categórica	Identificador del MOCACO
Id_article	Numérica	Identificador del artículo
First_day_store	Fecha	Fecha del primer día de venta del MOCACO en la tienda
First_day_world	Fecha	Fecha del primer día de venta del MOCACO en cualquier tienda
Sales_curve_types	Categórica	Tipo de curva de venta
Week_store	Numérica	Semana de vida del MOCACO en la tienda
Sales_week_store	Numérica	Unidades de venta
First_day_weekofyear_store	Numérica	Número de la semana del año que corresponde First_day_store
First_day_month_store	Numérica	Número del mes que corresponde First_day_store
First_day_halfofmonth_store	Numérica -Binario	Indica si First_day_store se encuentra en la primera quincena del mes o en la segunda
First_day_weekofyear_world	Numérica	Número de la semana del año que corresponde First_day_world
First_day_month_world	Numérica -Binario	Número del mes que corresponde First_day_world
First_day_halfofmonth_store	Numérica -Binario	Indica si First_day_world se encuentra en la primera quincena del mes o en la segunda.
n_stores_first_week_fisica	Numérica	Número de tiendas en las que se ha hechos ventas físicas en la primera semana de vida del MOCACO
n_stores_total_fisica	Numérica	Número de tiendas en las que se han hecho ventas físicas a lo largo de toda la vida del MOCACO en la tienda
n_stores_first_week_online	Numérica	Número de tiendas en las que se ha hechos ventas online en la primera semana de vida del MOCACO
n_stores_total_online	Numérica	Número de tiendas en las que se han hecho ventas online a lo largo de toda la vida del MOCACO en la tienda
Altos_final	Categórica	Tipo de alto
Anchura final	Categórica	Tipo de anchura
Cuellos_final	Categórica	Tipo de cuello
Mangas_final	Categórica	Tipo de mangas
Estilo	Categórica	Tipo de estilo

Forma	Categórica	Tipo de forma
Grupo_prendas	Categórica	Grupo de prendas pertenece
Base_price	Numérica	Precio base
Cod_family	Numérica	Identificador de la familia
Cod_subfamily	Numérica	Identificador de la subfamilia
Buyer_code	Numérica	Identificador del comprador
Initial_purchase	Numérica	Compra inicial a nivel global
Avg_sales_week_store_comparables	Numérica	Media de ventas de los MOCACOS similares
Avg_inital_purchase_comparables	Numérica	Media de la compra inicial a nivel global de los MOCACOS similares

Tabla 21. Conjunto de variable del Experimento 4

6.4.2 Entrenamiento

En la tabla anterior tenemos las variables que hemos utilizado para entrenar el modelo en sombreado. Y sobre la estructura ya hemos comentado que es la misma que en el experimento anterior.

6.4.3 Evaluación

El **WWAPE** que obtenemos es del **51,71%**, esto significa que el modelo tiene un error del 51,71% respecto al valor real**.** Veamos cómo se distribuye el error en las semanas.

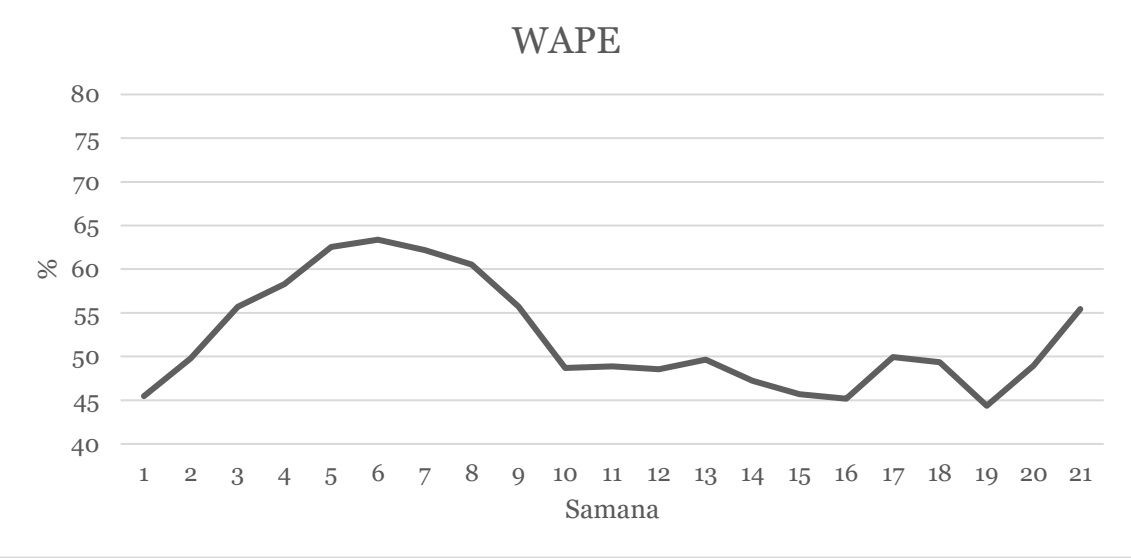


Ilustración 13. Gráfica del WAPE del Experimento 4

El error es más alto alrededor de las semanas tres y nueve, que es donde vemos la curva, esto se da por la distribución de los MOCACO en las tiendas. Visto que el error no es igual en todas las semanas, y que en las semanas iniciales el error es más grande, veamos cómo se distribuye el error según el país y si está relacionado con el porcentaje de venta.

6.5 Experimento 5: MLP 3

En este caso no tenemos nuevas variables para el modelo, lo que vamos a probar es si hacer una estructura más compleja del modelo reduce el error que tenemos

6.5.1 Entrenamiento

La estructura que seguiremos en este caso es una capa inicial con un tamaño de 2.014. Después tenemos una capa densamente conectada de 512 neuronas con función de activación “LeakyReLU”, le sique una capa de normalización por lotes y una capa de dropout con una tasa del 30%. Le sigue una capa oculta densa de 256 neuronas con una activación de “LeakyReLU” y una capa de normalización por lotes y otra de dropout con una tasa del 30%. Añadimos una última capa oculta de 64 neuronas con activación “LeakyReLU” y una última capa de normalización por lotes. Tras estas capas ocultas, tenemos la capa de salida, que es una capa densamente conectada con 1 neurona y activación “relu””. En este modelo utilizamos también una función de pérdida de error absoluto medio, y un optimizador Adam.

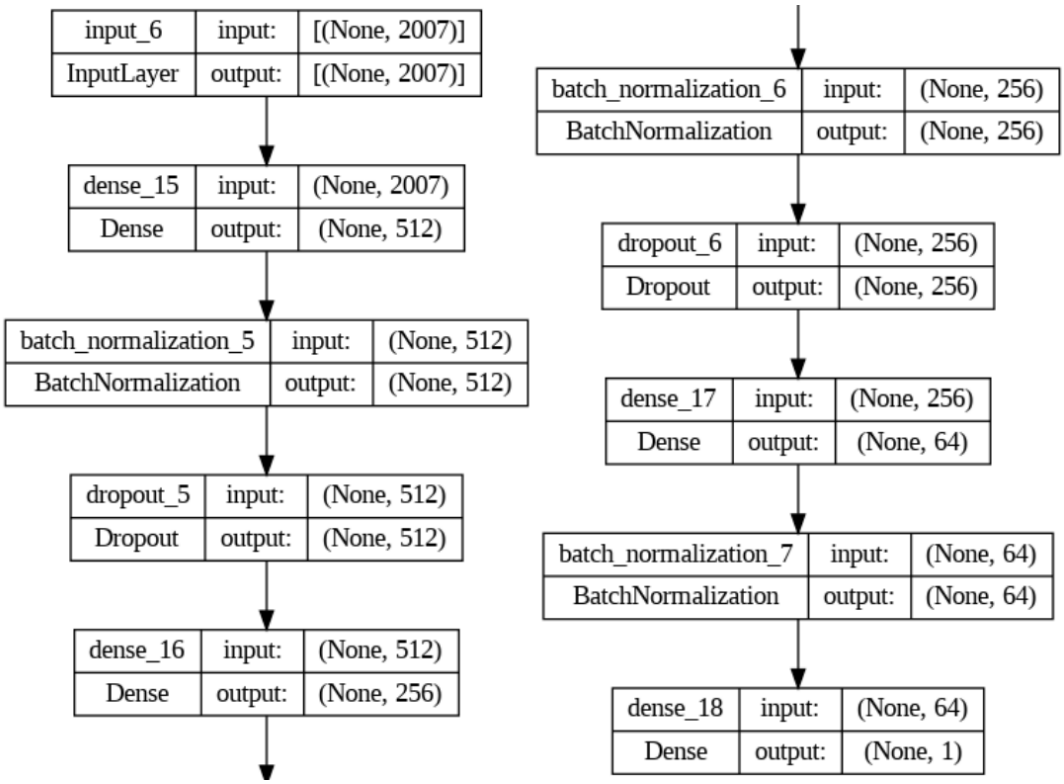


Ilustración 14. Estructura de las MLP del Experimento 5

6.5.2 Evaluación

Tras cambiar la estructura y mantener las variables vemos que impacto ha tenido en el error. El **WWAPE** es **52,09%**, con lo que obtenemos que, al aumentar la complejidad de la estructura de la MLP, el error ha aumentado en comparación con el anterior. Siendo ahora el error del modelo de un 52,09% respecto al valor real.

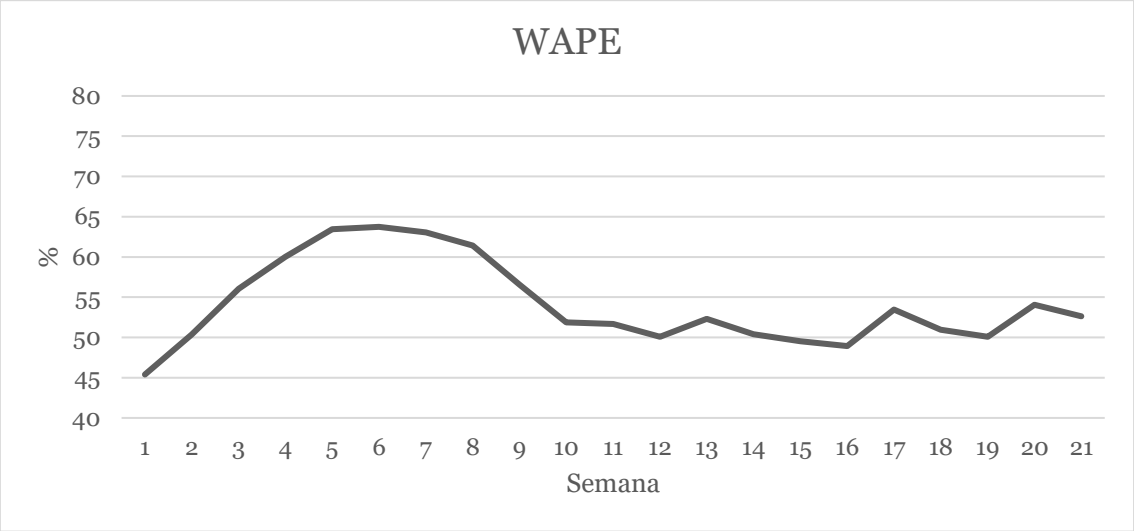


Ilustración 15. Gráfica del WAPE del Experimento 5

En las primeras semanas vemos que el error es bastante más alto que en el resto, teniendo una subida notable en las semanas de la tres a la nueve, luego baja y se mantiene para el resto de las semanas.

6.6 Experimento 6: XGB 1

Con las MLP, nos aparece un problema y es que vemos que al aumentar la complejidad de la estructura obtenemos un error mayor, y si generamos más variables tenemos que reducir la muestra de entrenamiento. Por esta razón, dejamos las MLP a un lado y empezamos con los XGB con entrenamiento distribuido. Esto nos permite añadir más variables, aumentar la muestra pasando de 500.000 registros a 1.500.000 registros, es decir, aumentamos en un millón la muestra. Además, el tiempo de entrenamiento del modelo pasa de tres horas a una hora.

Vamos a utiliza las mismas variables que el experimento cuatro y cinco para entrenar el modelo. Por lo que lo que cambia es el tipo modelo y la muestra del entrenamiento. Ahora para el procesamiento seguiremos los pasos descritos en el apartado de preprocesamiento.

En cuanto a los parámetros del XGB, dejamos los que están por defecto y si finalmente es el modelo escogido, realizaremos la optimización de los hiperparámetros.

6.6.2 Evaluación

Tras evaluar la predicción de la muestra de prueba, obtenemos que el **WWAPE** es de **47,6%**, esto nos indica que el modelo se desvía en promedio un 47,6% del valor real.. En la siguiente gráfica tenemos el error distribuido en las primeras 21 semanas del ciclo de vida de los MOCACO en las tiendas.

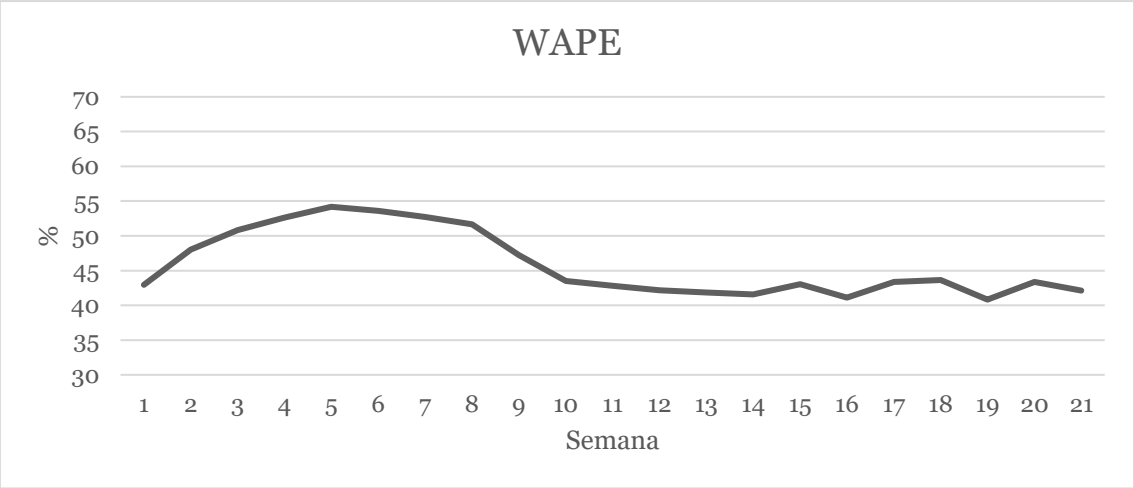


Ilustración 16. Gráfica del WAPE del Experimento 6

No sorprende, que como llevamos viendo en los experimentos anteriores, en las primeras semanas, en concreto, de la semana tres a la nueve el error sube bastante y luego se estabiliza para el resto de las semanas. Veamos cómo se distribuye en el siguiente gráfico.

6.7 Experimento 7: XGB 2

Ya hemos visto que el modelo anterior tiene un mejor rendimiento que la MLP, por lo que a partir de ahora utilizaremos el XGB con entrenamiento distribuido para seguir mejorando.

En este nuevo experimento lo que hacemos es crear nuevas variables para ver si mejoramos. La estructura, del XGB se mantiene igual que el anterior.

6.7.1 Ingeniería de características

Partimos de las mimas variables que el experimento predecesor, y añadimos nuevas variables una de ellas relacionadas con los similares y la otra con la venta de la familia. Ambas están focalizadas en describir la situación de la tienda.

Para la primera, lo que hacemos es calcular cuantos MOCACO similares hay en la tienda cuando el MOCACO a predecir va a llegar a la tienda. La siguiente variable, indica cuánto vende la familia del MOCACO en esa semana, de esta forma le damos un indicativo del funcionamiento de los MOCACO de la familia. A modo de experimento, lo que hacemos para calcular la venta de la familia es sumar la venta de todos los MOCACO que forman la familia incluido el de estudio, pero para un futuro lo que habría que hacer es un submodelo de predicción que haga la predicción de la familia.

6.7.2 Entrenamiento

Veamos la descripción de las variables y cuales utilizamos para entrenar el modelo.

Columna	Tipo	Descripción
Id_color	Numérica	Identificador del color
Id_brand	Numérica	Identificador de la marca
Id_section	Numérica	Identificador de la sección
Id_product	Numérica	Identificador del producto
Id_campaign	Numérica	Identificador de la campaña
Id_store	Numérica	Identificador de la tienda
Id_sales_channel	Numérica	Identificador del tipo de venta
Sku_artc	Categórica	Identificador del MOCACO
Id_article	Numérica	Identificador del artículo
First_day_store	Fecha	Fecha del primer día de venta del MOCACO en la tienda
First_day_world	Fecha	Fecha del primer día de venta del MOCACO en cualquier tienda
Sales_curve_types	Categórica	Tipo de curva de venta
Week_store	Numérica	Semana de vida del MOCACO en la tienda
Sales_week_store	Numérica	Unidades de venta
First_day_weekofyear_store	Numérica	Número de la semana del año que corresponde First_day_store
First_day_month_store	Numérica	Número del mes que corresponde First_day_store
First_day_halfofmonth_store	Numérica -Binario	Indica si First_day_store se encuentra en la primera quincena del mes o en la segunda
First_day_weekofyear_world	Numérica	Número de la semana del año que corresponde First_day_world
First_day_month_world	Numérica -Binario	Número del mes que corresponde First_day_world
First_day_halfofmonth_store	Numérica -Binario	Indica si First_day_world se encuentra en la primera quincena del mes o en la segunda.
n_stores_first_week_fisica	Numérica	Número de tiendas en las que se ha hechos ventas físicas en la primera semana de vida del MOCACO
n_stores_total_fisica	Numérica	Número de tiendas en las que se han hecho ventas físicas a lo largo de toda la vida del MOCACO en la tienda
n_stores_first_week_online	Numérica	Número de tiendas en las que se ha hechos ventas online en la primera semana de vida del MOCACO
n_stores_total_online	Numérica	Número de tiendas en las que se han hecho ventas online a lo largo de toda la vida del MOCACO en la tienda
Altos_final	Categórica	Tipo de alto
Anchura final	Categórica	Tipo de anchura
Cuellos_final	Categórica	Tipo de cuello
Mangas_final	Categórica	Tipo de mangas
Estilo	Categórica	Tipo de estilo

Forma	Categórica	Tipo de forma
Grupo_prendas	Categórica	Grupo de prendas pertenece
Base_price	Numérica	Precio base
Cod_family	Numérica	Identificador de la familia
Cod_subfamily	Numérica	Identificador de la subfamilia
Buyer_code	Numérica	Identificador del comprador
Initial_purchase	Numérica	Compra inicial a nivel global
Avg_sales_week_store_comparables	Numérica	Media de ventas de los MOCACOS similares
Avg_inital_purchase_comparables	Numérica	Media de la compra inicial a nivel global de los MOCACOS similares
Initial_purchase_store	Numérica	Envío inicial que se hace a la tienda
Similar_count	Numérica	Número de similares que hay cuando el MOCACO llega a tienda
Sales_week_store_family	Numérica	Ventas de la familia en la semana

Tabla 22. Conjunto de variables del Experimento 7

6.7.3 Evaluación

Tenemos un **WWAPE** de **47,26%**, por lo que el modelo se desvía un 47,26% del valor real. Veamos la distribución del error por semana.

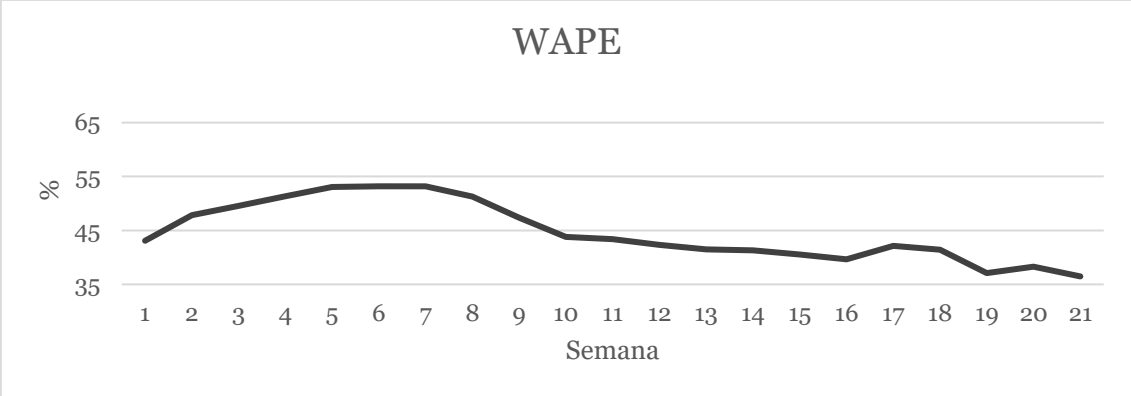


Ilustración 17. Gráfica del WAPE del Experimento 7

Al igual que en el anterior experimento, la forma de la curva es muy similar, con la diferencia de que el error que tenemos ahora por semana es menor.

6.8 Experimento 8: XGB 3

En este nuevo experimento, aumentamos el número de variables con el que entrenamos el modelo, además, modificamos la forma con la que trabajamos con los similares.

6.8.1 Ingeniería de características

Empecemos con las variables que añadimos al modelo, aparte de tener la compra inicial que se hace del MOCACO a nivel mundial, añadimos el envío inicial a la tienda en la primera semana, de esta forma especificamos cuantas unidades han llegado de ese MOCACO a la tienda.

Seguimos con las variables que hemos modificado, antes en los comprables lo que hacíamos era coger los 15 MOCACO más similares para cada uno y realizábamos los cálculos a partir de estos. Ahora cambiamos el modo de obtener los similares, para cada MOCACO tendremos los dos MOCACO más similares de la misma campaña que tengan un envío inicial en esa tienda similar, es decir que sea más o menos un 20%. De esta forma, obtenemos los dos más similares para esa tienda, de la misma campaña y que están en la misma situación. Seguimos el mismo proceso para seleccionar los dos más similares de la campaña anterior. De esta forma, tenemos datos de 4 similares, y no hacemos la media, sino que trabajamos de forma independiente con cada similar.

Tras el proceso que hemos descrito anteriormente con los similares, para cada uno de ellos creamos una variable que indique el envío inicial, cuantos similares había cuando llegaron, la venta de su familia, y su venta, todo esto haciendo referencia a la tienda.

6.8.2 Entrenamiento

En esta tabla tenemos la descripción de cada una de las variables y en sombreado indicamos cuales son las variables que se van a utilizar para entrenar el modelo. Además, recordemos que se le aplica el preprocesamiento descrito en el apartado X.

Columna	Tipo	Descripción
Id_color	Numérica	Identificador del color
Id_brand	Numérica	Identificador de la marca
Id_section	Numérica	Identificador de la sección
Id_product	Numérica	Identificador del producto
Id_campaign	Numérica	Identificador de la campaña
Id_store	Numérica	Identificador de la tienda
Id_sales_channel	Numérica	Identificador del tipo de venta
Sku_artc	Categórica	Identificador del MOCACO
Id_article	Numérica	Identificador del artículo
First_day_store	Fecha	Fecha del primer día de venta del MOCACO en la tienda
First_day_world	Fecha	Fecha del primer día de venta del MOCACO en cualquier tienda
Sales_curve_types	Categórica	Tipo de curva de venta
Week_store	Numérica	Semana de vida del MOCACO en la tienda
Sales_week_store	Numérica	Unidades de venta
First_day_weekofyear_store	Numérica	Número de la semana del año que corresponde First_day_store
First_day_month_store	Numérica	Número del mes que corresponde First_day_store
First_day_halfofmonth_store	Numérica -Binario	Indica si First_day_store se encuentra en la primera quincena del mes o en la segunda
First_day_weekofyear_world	Numérica	Número de la semana del año que corresponde First_day_world
First_day_month_world	Numérica -Binario	Número del mes que corresponde First_day_world
First_day_halfofmonth_store	Numérica -Binario	Indica si First_day_world se encuentra en la primera quincena del mes o en la segunda.
n_stores_first_week_fisica	Numérica	Número de tiendas en las que se ha hechos ventas físicas en la primera semana de vida del MOCACO
n_stores_total_fisica	Numérica	Número de tiendas en las que se han hecho ventas físicas a lo largo de toda la vida del MOCACO en la tienda
n_stores_first_week_online	Numérica	Número de tiendas en las que se ha hechos ventas online en la primera semana de vida del MOCACO
n_stores_total_online	Numérica	Número de tiendas en las que se han hecho ventas online a lo largo de toda la vida del MOCACO en la tienda
Altos_final	Categórica	Tipo de alto
Anchura final	Categórica	Tipo de anchura
Cuellos_final	Categórica	Tipo de cuello
Mangas_final	Categórica	Tipo de mangas
Estilo	Categórica	Tipo de estilo

Forma	Categórica	Tipo de forma
Grupo_prendas	Categórica	Grupo de prendas pertenece
Base_price	Numérica	Precio base
Cod_family	Numérica	Identificador de la familia
Cod_subfamily	Numérica	Identificador de la subfamilia
Buyer_code	Numérica	Identificador del comprador
Initial_purchase	Numérica	Compra inicial a nivel global
Initial_purchase_store	Numérica	Envío inicial que se hace a la tienda
Similar_count	Numérica	Número de similares que hay cuando el MOCACO llega a tienda
Sales_week_store_family	Numérica	Ventas de la familia en la semana
Initial_purchase_store_similar_same1	Numérica	Envío inicial en la tienda del similar más parecido en la misma situación de la misma campaña
Initial_purchase_store_similar_same2	Numérica	Envío inicial en la tienda del segundo similar más parecido en la misma situación de la misma campaña
Initial_purchase_store_similar_prev1	Numérica	Envío inicial en la tienda del similar más parecido en la misma situación de la campaña previa
Initial_purchase_store_similar_prev2	Numérica	Envío inicial en la tienda del segundo similar más parecido en la misma situación de la campaña previa
Sales_week_store_similar_same1	Numérica	Venta en la semana del más similar de la misma campaña, que tiene la misma situación en la tienda
Sales_week_store_similar_same2	Numérica	Venta en la semana del segundo más similar de la misma campaña, que tiene la misma situación en la tienda
Sales_week_store_similar_prev1	Numérica	Venta en la semana del más similar de la misma campaña, que tiene la misma situación en la tienda
Sales_week_store_similar_prev2	Numérica	Venta en la semana del segundo más similar de la misma campaña, que tiene la misma situación en la tienda
Similar_in_family_same1	Numérica	% de venta en la tienda que representa respecto a su familia el más similar de la misma campaña, que tiene la misma situación en la tienda
Similar_in_family_same2	Numérica	% de venta en la tienda que representa respecto a su familia el segundo más similar de la misma campaña, que tiene la misma situación en la tienda
Similar_in_family_prev1	Numérica	% de venta en la tienda que representa respecto a su

		familia el más similar de la campaña previa, que tiene la misma situación en la tienda
Similar_in_family_prev2	Numérica	% de venta en la tienda que representa respecto a su familia el segundo más similar de la campaña previa, que tiene la misma situación en la tienda

Tabla 23. Conjunto de variables del Experimento 8

6.8.3 Evaluación

En cuanto al **WWAPE**, obtenemos un valor del **45,17%**, que si recordamos el valor que teníamos a nivel mundo, aumenta un 10%. Aunque el error sea alto, realmente es un buen resultado, obteniendo una predicción que se difiere de un 45,17% del valor real.

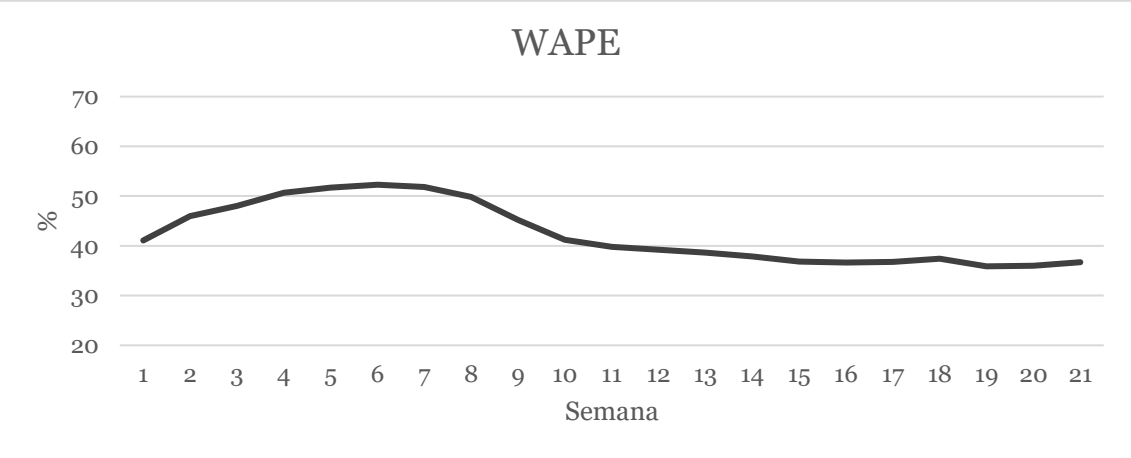


Ilustración 18. Gráfica del WAPE del Experimento 8

El gráfico muestra la evolución del WAPE a lo largo de 21 primeras semanas del ciclo de vida de los MOCACO en las tiendas.

De la semana uno a la seis, aumenta desde el 41,07% hasta el 52,28%, indicando que el error en las predicciones crece durante este periodo

De la semana siete a la diez, disminuye desde el 51,8% al 41,2% y empieza a estabilizarse otra vez tenido el mismo error que en la primera semana. Para las semanas que forman la cola el error se estabiliza sobre un 35%, siendo el error más bajo que alcanza el modelo.

6.9 Comparación de modelos

Ya hemos analizado los ocho experimentos realizados, empezando por los baselines, las tres MLP y los tres XGB. Donde cada uno de ellos incluía nuevas variables para mejorar el modelo anterior o bien suplían una carencia. Primero veamos la comparativa de los WWAPE y cuál ha sido el mejor de ellos.

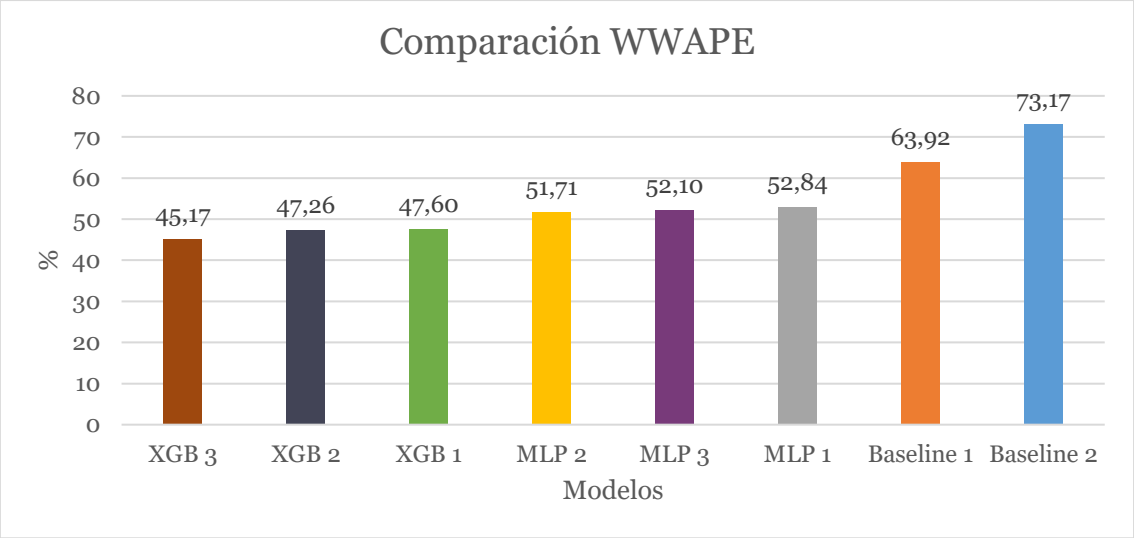


Ilustración 19. Comparación de los WWAPE de todos los experimentos

Vamos a analizar el grafico según hemos ido generando los modelos. El valor del WWAPE para el primer baseline es de 63,92%, es elevado, pero se esperaba mucho mayor, ya que el modelo es muy simple, donde cada tienda tiene un peso fijo y se desagrega la predicción a nivel mundo mediante una multiplicación. El segundo baseline tiene un error del 73,14%, mayor que el del primer baseline. Este resultado es sorprendente, ya que pensábamos que este modelo iba a tener un error menor porque tiene en cuenta la semana del año para calcular el peso, entonces se iba a poder ajustar mejor a las situaciones del mercado.

Ya dejamos los baselines a un lado y analizamos los errores de las MLP. En la MLP1 tenemos un error del 52,84% que si lo comparamos con el de las baselines tenemos una bajada del 11,11%. Esta bajada es considerable y vemos cómo utilizar este modelo de machine learning y las variables que hemos introducido nos ayuda a tener una mejor predicción. El siguiente modelo que hemos construido es la MLP 2, que tiene la misma arquitectura que el anterior, pero añadimos nuevas variables, gracias a esto, obtenemos una disminución del error del 1,13%. Ahora lo que hacemos es crear una estructura más compleja y así obtenemos la MLP 3, como vemos el error aumenta en 0.39% respecto al mismo modelo que tiene una estructura menos compleja. Ahora solo nos queda probar a entrenar con más datos, pero la MLP no nos lo permite debido a la capacidad de memoria, por esta razón empezamos con el entrenamiento distribuido y surgen los siguientes modelos.

El primero que creamos es el XGB 1, tiene las mismas variables que los modelos de MLP 2 y MLP 3, pero con un millón más de datos de entrenamiento. Gracias al entrenamiento distribuido conseguimos en disminuir el error en 4,11%, es una gran

bajada que nos lleva a dejar de entrenar modelos que no tengan entrenamiento distribuido, y nos abre una nueva ventana de estudio. A partir de ahora, solo se entrenarán los modelos con entrenamiento distribuido ya que tienen un mejor resultado y no tiene sentido entrenar modelos que sabemos que serán peores. Ya resuelto el límite de la capacidad de entrenamiento, nos queda añadir nuevas variables a los modelos. Tenemos el XGB 2 con un error del 47,26% disminuyendo un 0,34%, esto lo conseguimos gracias a introducir al modelo variables que lo que hacen es describir la situación de la tienda cuando llegan los MOCACO, además también añadimos una variable que actúa como una predicción de la venta de la familia. En el último experimento lo que hacemos es cambiar la forma de trabajar con los similares y añadimos variables nuevas como el envío que se hace a la tienda, gracias a esto conseguimos un error del 45,17%, disminuyendo un 2,09%, esto es una bajada considerable.

El mejor modelo de todos ha sido el XGB 3, logrando el menor error de todos, 45,17%. Ahora lo que vamos a hacer es ver el WAPE de los modelos a lo largo de las 21 semanas.

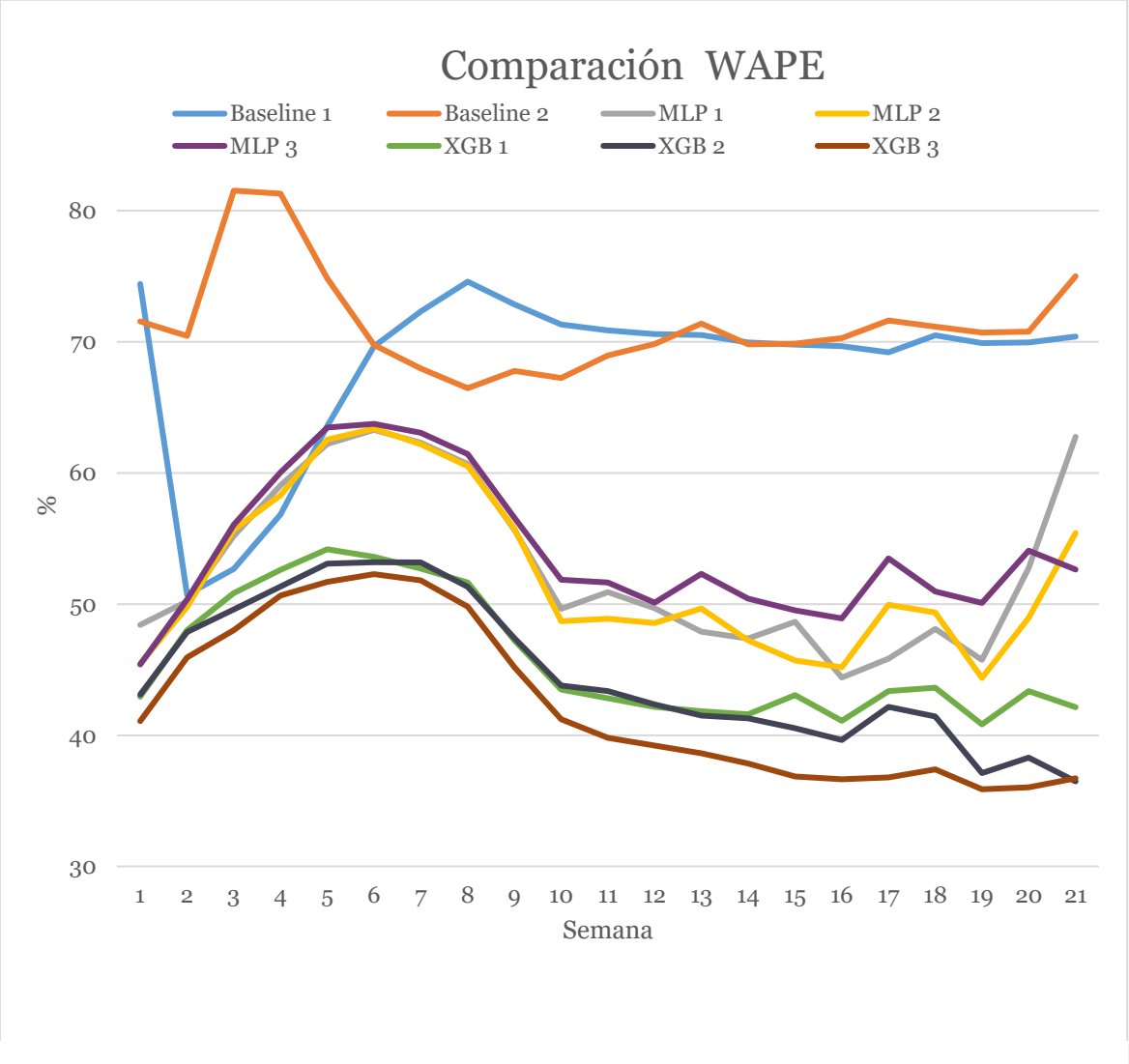


Ilustración 20. Comparación de los WAPE de todos los experimentos

El gráfico muestra la comparación del WAPE de los modelos a lo largo de 21 semanas. Nos va a permitir ver cómo se comportan los modelos según la semana del MOCACO en la tienda, y ver si hay modelos que funcionan mejor que otros en algunas semanas.

Podemos ver que las curvas del baseline y baseline 2 van independientemente, y las curvas de los modelos donde utilizamos una MLP o XGB tienen exactamente la misma forma.

Como vemos desde el experimento 3, MLP1, todos los modelos empiezan con un error más bajo en la primera semana y va ascendiendo hasta la semana 6 y luego desciendo hasta la semana 10 donde sigue bajando, pero levemente. Es curioso ver que estos modelos dibujan una curva cóncava muy similar desde la semana 1 a la 10, esto se da por el comportamiento de los MOCACO en las tiendas.

Si nos fijamos en los modelos XGB1 y XGB2 vemos como en la mayoría de los casos el XGB 2 es mejor en todas las semanas, sin embargo, hay semanas en las que el XGB 1 tiene un mejor rendimiento. Pero si nos fijamos en el XGB 3, esto ya no sucede, es siempre mejor que todos los modelos previos, demostrando una clara mejoría.

Como era de esperar, fijándonos en el WAPE también tenemos un claro ganador que es el XGB3, siendo el mejor modelo en todas las semanas.

Solo nos queda un aspecto para tener en cuenta, el tiempo que tarda cada modelo en realizar la inferencia de prueba. En este aspecto, no hay una diferencia 5considerable que haga decantarnos por un modelo que tenga un error peor debido a que el modelo con un menor error tenga un tiempo muy elevado. Por esta razón, no lo tendremos en cuenta para la elección del mejor modelo. Todos los modelos están alrededor de 50 minutos para una muestra de 500.000 registros, por lo que para cada predicción tardamos 0.006 segundos.

En definitiva, tras ver el **WWAPE** y **WAPE** de los modelos y ver que no hay una diferencia notoria en los **tiempos de inferencia**, nos decantamos por elegir el modelo **XGB 3** como mejor modelo.

6.10 Optimización de hiperparámetros

Para la optimización de hiperparámetros tenemos que hacer una serie de aclaraciones. Al trabajar con un XGB de pyspark, hay ciertos parámetros que el XGB de la librería Scikit-learn tiene, pero que con pyspark no se puede. Utilizaremos la librería de Hyperopt, que para cada modelo realiza un número de combinaciones de hiperparámetros determinado. Nosotros nos vamos a focalizar en optimizar los siguientes parámetros:

Hiperparámetro	Descripción	Valores probados
Max_depth	Profundidad máxima que puede alcanzar el árbol de decisión, donde cada división hace referencia a una división de los datos. A mayor profundidad, más riesgo de sobreajuste.	10 valores de 5 a 15
Eta	Controla la velocidad de aprendizaje del modelo. Un valor más bajo hace que el modelo aprenda más lentamente, un valor más alto hace que el modelo aprenda más rápidamente.	10 valores de 0 a 0.3

Tabla 24. Hiperparámetros a optimizar

Solo lo vamos a realizar para las diez primeras semanas, y si obtenemos mejores resultados lo hacemos para las 21 semanas. Esto lo hacemos porque cada semana es un modelo diferente, por lo que estamos entrenando 21 modelos. Hacer el proceso de optimización es un proceso largo, ya que requiere de varias iteraciones para obtener un buen resultado. Nosotros solo vamos a poder realizar 10 iteraciones por semana, probando 10 posibles combinaciones de los hiperparámetros establecidos. Para comprar el modelo optimizado y el XGB 3, no tendrmeos en cuenta el WWAPE, ya que el del XGB 3 lo tenemos para las tres semanas y para el optimizado unicamente tenemos las diez primeras semnas. Por ello, nos vamos a fijar solamente en el WAPE.

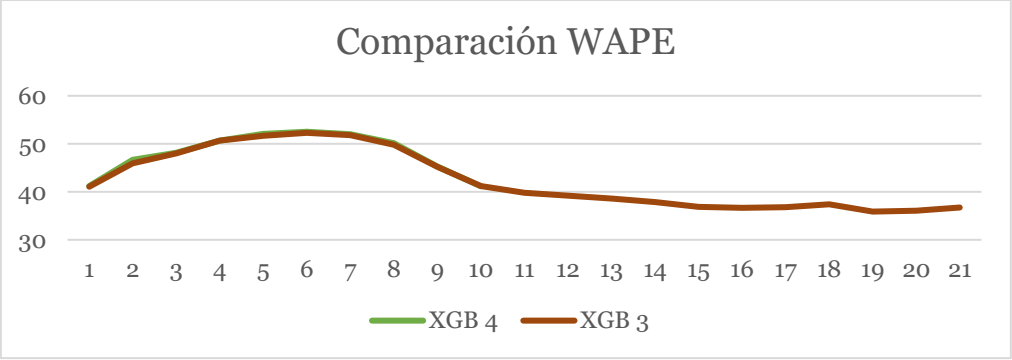


Ilustración 21. Comparación del WAPE del Experimento 8 y del mismo modelo con la optimización de hiperparámetros

El modelo sin optimizar es igual o mejor en todas las semanas, es posible que aún se pueda optimizar más el modelo, pero no podemos poner a entrenar el modelo con muchas combinaciones por limitaciones de memoria y económicas. Además, la mejora que podríamos alcanzar no sería muy significativa. Por lo que no es rentable la mejora que podemos tener respecto al coste que nos llevaría, ya que habría que crear

un clúster con más memoria y hacer más horas de ejecución, y todo ello conlleva un coste económico que no compensa.

6.11 Explicabilidad

6.11.1 Análisis de las variables

Para ver la explicación de las variables tenemos la Tabla 23, donde encontramos el nombre de la variable, el tipo de dato y su descripción. Hemos utilizado la librería Dython para obtener la matriz de las correlaciones de todas las variables que utilizamos en el modelo. Utilizamos esta librería porque nos permite hacerlo de las variables numéricas y categóricas. A continuación, mostramos las técnicas utilizadas por la librería Dython para comparar correlaciones entre distintos tipos de variables (41):

- Numérica y numérica: utiliza la “correlación de Pearson”. El valor oscila entre menos uno y uno. El valor uno indica una correlación positiva perfecta, el valor menos uno una correlación negativa perfecta y el valor cero indica que no hay correlación lineal.
- Categórica y categórica: la “correlación de Cramer’s V” es una medida de asociación entre dos variables categóricas que se basa en la prueba chi-cuadrado. El valor oscila entre cero y uno, donde 0 supone la ausencia de asociación y 1 es una asociación perfecta.
- Categórica y numérica: se utiliza la “razón de correlación”, el valor oscila entre cero y uno. Es una medida de la relación curvilínea entre la dispersión estadística dentro de las categorías individuales y la dispersión en toda la población o muestra.

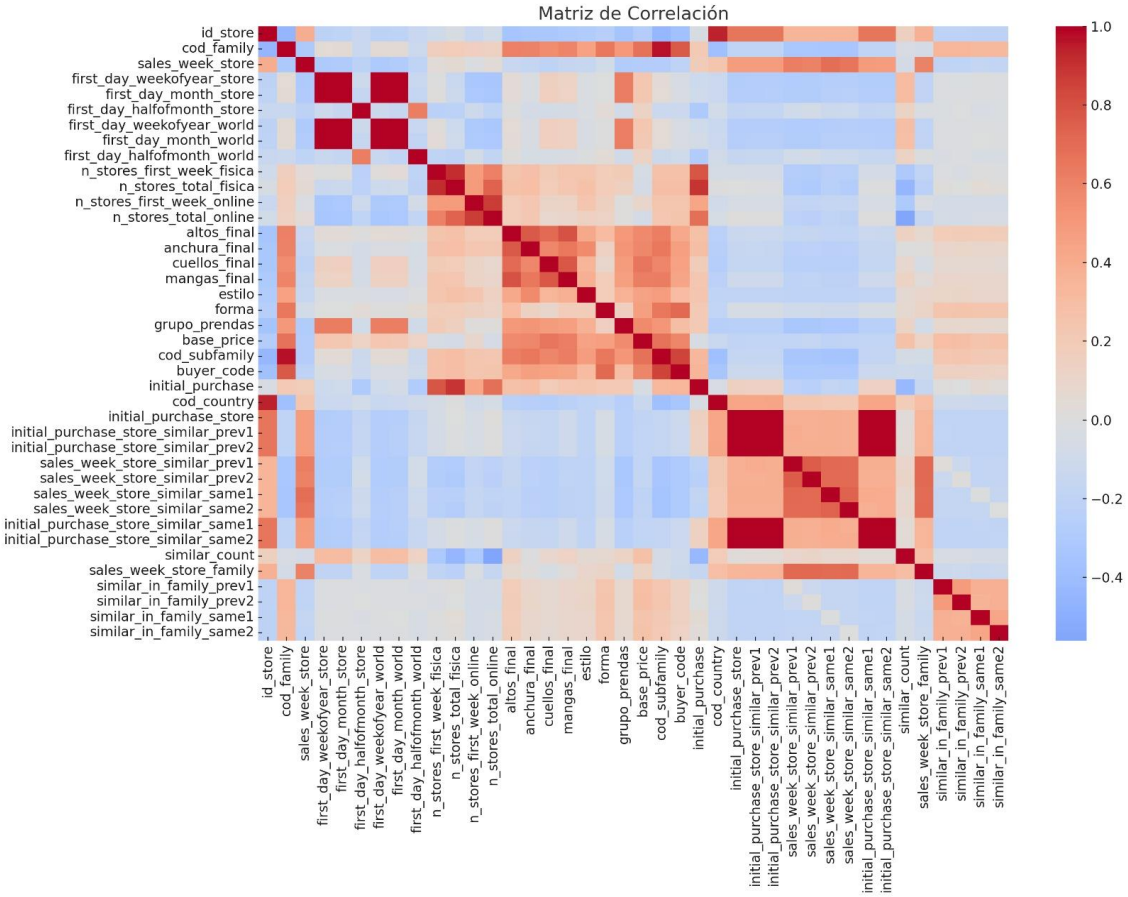


Ilustración 22. Matriz de correlaciones

No vamos a interpretar la matriz de correlación en detalle porque puede ser extenso debida a la cantidad de variables que tenemos. Sin embargo, podemos resaltar algunos de los patrones más interesantes y significativos asociados con la variable de estudio, sales_week_store.

En cuanto a las correlaciones positivas fuertes tenemos algunas variables como initial_purchase_store, sales_week_store_similar_same1, y sales_week_store_family que muestran correlaciones positivas con sales_week_store. Esto sugiere que las ventas semanales en una tienda están influenciadas significativamente por el envío inicial a la tienda, la similitud con su más similar en la tienda que está en la misma situación y la venta de la familia en la tienda.

En relación con las correlaciones negativas, no encontramos ninguna variable que tenga una correlación negativa tan significativa con sales_week_store, pero sí que hay algunas que muestran ligeras correlaciones negativas, pero son débiles.

Respecto a los patrones temporales, como first_day_weekofyear_store y first_day_month_store muestran correlaciones moderadas, lo que podría indicar que ciertas épocas del año influyen en las ventas debido a tendencias estacionales.

6.11.2 Importancia de las variables

Para los 21 modelos que forman el XGB 3, lo que vamos a hacer es seleccionar las siete variables más importantes para los 21 modelos y veamos qué tan importantes es en cada uno de ellos. Estamos seleccionando siete variables de 40, por loque, aunque sean las más importantes, en algunas ocasiones sus valores no serán tan altos, ya que hay muchas variables.

Para seleccionar estas siete variables, lo que hacemos es obtener la importancia de cada variable en cada uno de los modelos que forman cada semana y calculamos la media, de esta forma podemos calcular las 7 variables que tienen una mayor importancia. Utilizamos la media porque de este modo cogemos las más importantes de cada semana.

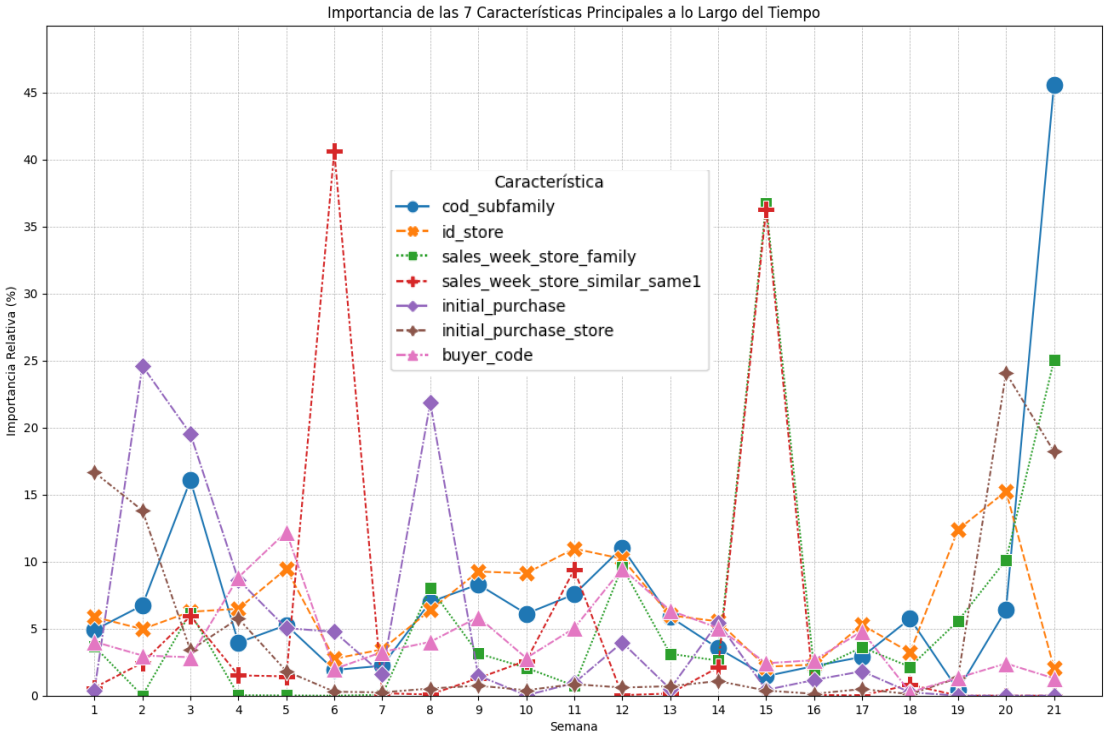


Ilustración 23. Importancia de las 7 variables principales a lo largo de las 21 semanas

Empecemos por cod_subfamily (círculo azul), esta variable tiene fluctuaciones a lo largo del tiempo. Es notablemente alta en la semana 21, donde alcanza su máxima importancia relativa. También tiene picos significativos en las semanas tres y 12. En general, para los 21 modelos, es una variable que está siempre entre las más importantes.

La característica id_store (x naranja) es importante en todos los modelos, pero su relevancia aumenta en las semanas cinco, de la nueva a la doce y de la 19 a la 20.

La variable sales_week_store_family (cuadro verde) tiene importancia en todas las semanas menos en las semanas dos, cuatro y cinco. Para la mayoría de las semanas la importancia se mantiene entre un 2,5% y 5%, pero en las semanas, ocho, 12,15, 20 y 21 sobre pasa del 10% llegando a un máximo en la semana 15 de un 37%.

En cuanto a sales_week_store_similar_same1 (cruz roja), tienes dos picos muy pronunciados en la semana seis y 15, llegando a más de un 40% en la semana seis y a más de un 35% en la semana 15. Para el resto de las semanas, hay ocasiones en las que tiene una importancia entre un 0% y un 5%. Y tiene poca importancia en las semanas de la seis a la ocho, doce, trece, y de la dieciséis a la 21.

En relación con initial_purchase (rombo morado), vemos que pasa de una muy baja importancia en la semana uno a una importancia del 25% en la semana dos, luego tenemos otro pico de más de un 20% en la semana ocho y para el resto de las semanas se mantiene entre un 2,5% y un 5%, aunque en las tres últimas semanas esta variable tiene una importancia bastante baja.

Llegamos a la penúltima variable, initial_purchase_store (rombo marrón), tiene una gran importancia en las dos primeras semanas estando alrededor de un 15%, para el resto de las semanas tiene una importancia entre un 0% y un 2,5%, hasta que llega la semana 20 y 21, para la semana 20 tiene una importancia de un 25% y para la semana 21 tiene un valor cercano a 20%.

Esta es la última de las siete características, buyer_code (triángulo rosa), esta variable tiene una importancia notoria a lo largo de las 21 semanas, aunque pierde importancia a partir de la semana 18. Su mayor pico es en la semana cinco con un 12,5%. No es una variable que tenga picos tan altos como otras, pero sí que mantiene su importancia a lo largo de las semanas.

La importancia de las características no es constante y varía significativamente semana a semana. Esto sugiere que la relevancia de las características puede depender de factores específicos de cada semana. Además, en ciertas semanas hay picos muy altos para algunas variables indicando que tienen mucha importancia en algunas semanas

6.11.3 Distribución del error

Lo primero que vamos a ver es que WWAPE tiene cada uno de los compradores junto a su porcentaje de venta respecto a la campaña que se está haciendo la inferencia, es decir, la campaña de invierno 23.

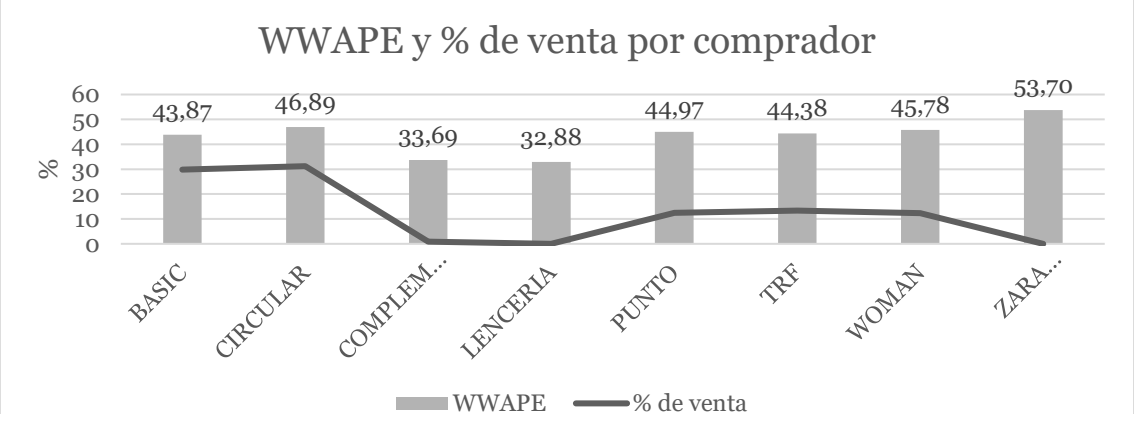


Ilustración 24. Distribución del WWAPE y % de venta por comprador del experimento con mejor rendimiento

Los compradores que representan el mayor porcentaje de venta tienen un WWAPE en torno a 45%. Vemos que su error no se difiere mucho de aquellos que tienen un porcentaje de venta menor, menos Complementos y Lencería que tienen un WWAPE de un 10% menos.

Ya visto por compradores, vamos a visualizar el error por familia, esto puede ser interesante para ver si hay alguna familia que tenga un error muy elevado.

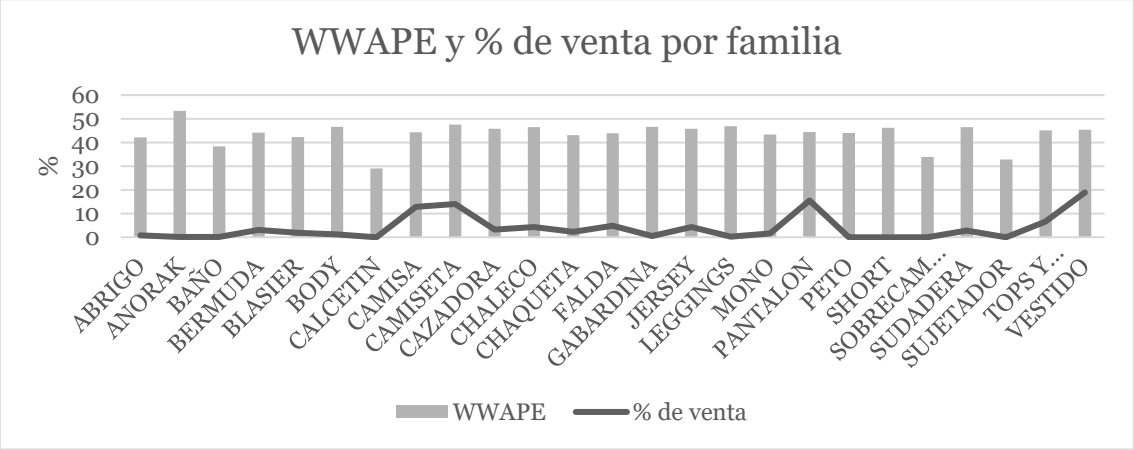


Ilustración 25. Distribución del WWAPE y % de venta por familia del experimento con mejor rendimiento

Fijándonos en las familias que tienen un mayor porcentaje de venta tenemos pantalón, vestido, camisa y camiseta. Todas ellas tienen WWAPE similar entorno a un 45%. Podemos ver que hay algunas familias que tiene un error bastante más bajo que otras, como es el caso de calcetín o sobrecamisa. Hay algunas familias que tienen un mayor o menor error, pero más o menos todas son similares.

CAPÍTULO 7

Conclusiones

En este último capítulo vemos como se ha alcanzado el objetivo general y los objetivos específicos, además tratamos la implicación de los resultados obtenidos. Por otra parte, tenemos el legado donde mostramos cual es el trabajo realizado. Seguimos viendo la relación del TFG con el grado cursado. Y para acabar, presentamos las limitaciones del proyecto y el trabajo futuro.

7.1 Trabajo realizado

El objetivo principal consiste en, obtener un modelo capaz de predecir las unidades que se van a vender de un MOCACO en una tienda durante su ciclo de vida, Este objetivo se ha alcanzado con un modelo XGboost con entrenamiento distribuido. Se llega a él gracias a una serie de experimentos donde la intención es que cada uno incluya nuevas variables para descubrir una nueva mejora o cubra alguna limitación del anterior.

De la misma manera, los objetivos específicos se han ido alcanzando a medida que se desarrolla el proyecto. En primer lugar, hemos comprendido bien los datos que teníamos para ver como podíamos abordar el problema. Para ello, lo que hacemos es ver el comportamiento de las curvas de los MOCACO en las tiendas. De este modo, podemos hacernos una idea visual del comportamiento de lo que queremos predecir. Tras el análisis vemos que tenemos tres grandes grupos, fantasía, básicos y otros, esto implica tener que hacer todo el proceso para cada uno de los grupos, y no es posible hacerlo en tres meses. Por ello, junto a la compañía, decidimos que vamos a enfocarnos en los fantasía, ya que para maximizar el beneficio es importante tener bien controlado los MOCACO que venden mucho en sus primeras semanas.

Ya realizado el análisis y focalizado el proyecto, seguimos con el desarrollo y abordamos el objetivo de preparar los datos de los baselines, los modelos que utilizan una MLP y los que utilizan un XGBoost. En este caso, no encontramos ningún problema más allá de utilizar una nueva herramienta, pyspark, que no la hemos utilizado hasta el momento, por lo que la principal dificultad se centra en la familiarización de la librería.

El siguiente objetivo es crear los baselines. El primero de ellos consiste en obtener unos pesos que representan el porcentaje de venta de cada tienda y desagregamos la predicción del modelo a nivel mundo. El segundo baseline consiste en que cada tienda tiene un peso por semana del año y se desagrega la predicción a nivel mundo en función de la semana. Con esto aprendemos que es importante crear una primera aproximación que no esté relacionado con modelos de aprendizaje automático

y así poder tomarlos como referencia. Hasta ahora, no hemos tenido ninguna limitación por parte del sistema, por lo que entrenamos con todos los datos que disponemos de las tres campañas anteriores a la seleccionada para la inferencia.

Para poder abordar el objetivo de comparar los modelos, primero tenemos que cumplir dos objetivos, uno que es establecer las métricas de evaluación, se utilizan las mismas que en el modelo a nivel mundo porque hay que seguir el mismo criterio. Por otro lado, creamos los modelos de Machine Learning donde encontramos un problema, la limitación de la memoria a la hora de entrenar el pipeline que engloba los preprocesadores y el propio modelo. Para solucionar este problema, creamos una función de muestreo capaz de generar una muestra representativa garantizando la selección de una muestra de tamaño determinado. Este tamaño lo obtenemos entrenando los modelos con diferentes tamaños escogiendo la muestra más grande posible. Generamos una de 500.000 MOCACO-tienda por semana para los experimentos que utilizan una MLP y una muestra de 1.500.000 MOCACO-tienda por semana para los experimentos de XGBoost.

Un aspecto diferenciador del trabajo es el entrenamiento distribuido con el XGboost, ya que esto nos permite tener una muestra mucho mayor que si hubiésemos utilizado un entrenamiento normal. La diferencia entre un entrenamiento distribuido o no, es si se utiliza un worker o más de uno, en nuestro caso utilizamos todos los workers disponibles, en concreto cuatro, nos permite dividir el trabajo entre los nodos para entrenar el modelo más rápido y eficientemente. Combinando el entrenamiento distribuido y la función de muestreo conseguimos una bajada considerable del error que nos abre una nueva ventana de estudio y cambia el enfoque inicial

Nuestro objetivo de optimización de hiperparámetros debía servirnos para obtener un modelo mejor. Sin embargo, no ha sido el caso. Esto puede pasar por dos razones, bien sea porque lo valores de serie sean lo óptimos, o que necesitemos probar más combinaciones de hiperparámetros. Los hiperparámetros explorados son la profundidad máxima y la tasa de aprendizaje. Por temas de recursos, no podemos hacer una búsqueda muy exhaustiva de estos valores, ya que el modelo está formado por 21 modelos y encontrar los valores óptimos tiene un coste de memoria y de tiempo muy alto. No obstante, se recomienda a Inditex continuar este ejercicio de optimización en el futuro para mejorar el resultado.

El último objetivo que nos queda es la explicabilidad del modelo, para desarrollarlo tenemos tres partes. La primera corresponde a la correlación de las variables que las ventas semanales en las tiendas, tras investigar encontramos la librería Dython capaz de obtener la correlación tanto de las numéricas y categóricas en una matriz, esto facilita mucho el trabajo al ver visualmente todas las correlaciones. Por otra parte, vemos como se distribuye la importancia de las siete variables más importantes, Por último, tenemos la distribución del error a lo largo de diferentes agrupaciones.

Tras cumplir todos los objetivos obtenemos un modelo que minimiza el error de la distribución de los MOCACO en las tiendas, lo que implica que hagamos una gestión correcta maximizando las ventas y minimizando los costes de distribución al hacer una asignación correcta. El proyecto se lleva a cabo en una empresa que factura millones

de euros al día y en el 2023 gestionó 2.000 millones de prendas, por lo que su impacto será mayor, lo que destaca lo crucial que es realizar una buena gestión. El modelo propuesto tiene un gran impacto en INDITEX, su uso permitirá ahorrar costes y permitirá vender más al controlar el stock.

7.2 Legado

Este proyecto es uno de los principales objetivos del departamento, en nuestro caso hemos resuelto el problema para ZARA mujer los MOCACO que son fantasía. Gracias a realizar estos experimentos podemos ver cuáles son las técnicas óptimas que tenemos que utilizar en el resto de las secciones de ZARA y en las otras marcas. Además, no solo sabemos que técnicas son las correctas, sino cuales son las variables que debemos utilizar para obtener unos buenos modelos.

Lo primero de todo al realizar el análisis de las curvas de venta y compararlas a nivel mundo, pueden ver que también se puede dividir el problema para fantasía, básicos y otros, lo que permite seguir la misma estructura que a nivel mundo.

Empezando por el conjunto de datos, aunque el trabajo se enfoque en los MOCACO fantasía el proceso de obtención de datos y el tratamiento se realiza para todos los MOCACO, de forma que este proceso ya está hecho. Además, hemos desarrollado una función de muestreo que permite obtener de forma sencilla el número de MOCACO-tienda que queramos tratar y garantiza que sea representativa.

Uno de los pasos cruciales es ver cuáles son las variables que son de utilidad para minimizar el error, ahora sabrán cuales son las que deben introducir y lo harán de forma sencilla, ya que hemos creado un notebook donde están las funciones hechas para obtener las variables únicamente pasándole los argumentos adecuados y dándole a ejecutar. Esto ayuda mucho porque no tendrán que crear código y las funciones no contienen errores que puedan generar duplicados o cualquier error que entorpezca el entrenamiento de los modelos.

Cada uno de los experimentos está formado por tres notebooks, uno donde tenemos todas las funciones que se utilizarán en la fase de entrenamiento y evaluación, de modo que hace que sea mucho más accesible para luego aprovechar las funciones que nos interesen. También tenemos el notebook donde se entrena el modelo y otro donde tenemos el proceso de inferencia. Gracias a esto la estructura de los modelos es muy clara lo que permite a la persona que vaya a continuar el proyecto entender rápidamente la estructura y poder utilizar las partes del código que sean de interés.

Además, como hemos visto durante los experimentos, los procesos de preprocesamiento de los datos como el One-Hot-Encoding los realizamos mediante Scikit-learn y Pyspark, esto permite que para los futuros modelos no se requiera una investigación profunda sobre su desarrollo mediante estas dos técnicas.

Aunque se haya abordado los MOCACO fantasía, el modelo es igual para el resto de MOCACO, por lo que prácticamente quedaría entrenar los modelos con los otros dos tipos y se obtendría los modelos necesarios para ZARA mujer. Por lo que dejamos una estructura y modelos hechos para completar el proceso.

En resumen, dejamos todo el código de los diferentes preprocesamientos, y de los experimentos que permiten entrenar nuevos modelos para otros MOCACO con cambios mínimos, incluso están preparados para que añadan cuales son los tipos de MOCACO que quieren entrenar y se entrenen todos los modelos guardando ordenadamente los modelos y sus preprocesadores para un uso futuro. Lo mismo ocurre con la inferencia donde con un click se llamará a todos los modelos y el resultado será la tabla final con todas las predicciones e información adicional. Todo esto facilita el proceso de creación de modelos y que se puedan utilizar todos a la vez para obtener la inferencia de todos los tipos de MOCACO. Además, incluimos el entrenamiento distribuido que hasta el momento no lo habían tenido en cuenta.

7.3 Relación del trabajo desarrollado con los estudios cursados

Los pasos seguidos en la fase de desarrollo están totalmente relacionados con el grado de Ciencia de datos, utilizamos el conocimiento adquirido desde el primer curso hasta el último. Además, ponemos en práctica las competencias transversales desarrollados durante los cuatro años del grado.

En cuanto a las asignaturas que nos han servido como apoyo para el desarrollo del proyecto tenemos, las relacionadas con el lenguaje de programación Python “Fundamentos de programación”, “Programación”, “Estructura de datos” y “Algorítmica” donde aprendimos a utilizar Python y nos permiten el desarrollo del código, gracias a estas asignaturas nos ha sido más sencillo hacer el preprocesamiento de los datos y la creación de los baselines, y en general, todo el desarrollo del proyecto, ya que se ha desarrollado en Python. Otro lenguaje que hemos utilizado es SQL, aprendido en la asignatura de “Bases de Datos”.

Por otra parte, tenemos las asignaturas enfocadas en la parte más estadísticas y de conocimiento de Machine Learning y Deep Learning. Gracias a las asignaturas de “Análisis exploratorio de datos”, “Modelos estadísticos para la toma de decisiones I”, “Modelos estadísticos para la toma de decisiones II”, “Modelos predictivos y descriptivos I”, “Modelos predictivos y descriptivos II”, “Evaluación, despliegue y monitorización” y “Técnicas escalables en aprendizaje automático” hemos tenido el conocimiento suficiente para agilizar la tarea de exploración de las curvas de venta, preprocesamiento y de desarrollo de modelos.

Una parte muy importante es la visualización de los datos, para ello hemos utilizado el conocimiento adquirido en la asignatura de “Visualización”, hemos graficado las curvas de venta de los MOCACO en las tiendas, las métricas de los modelos, la comparación, la matriz de correlaciones y la evolución de las siete variables más importantes.

Un aspecto muy importante de cualquier proyecto es la dinámica de trabajo y la gestión del tiempo. Para la gestión del tiempo utilizamos el conocimiento adquirido en “Gestión de proyectos”. Además, tenemos tres asignaturas donde también aprendimos a gestionar el tiempo y a tener una buena dinámica del trabajo, estas son “Proyecto I”, “Proyecto II” y “Proyecto II”, además han servido como entrenamiento para realizar el

TFG, ya que se sigue la misma estructura donde se desarrolla el trabajo y se entrega una memoria similar a la del TFG.

Por otro lado, tenemos las competencias transversales, las que utilizamos principalmente son:

- “Comprensión e integración”: Durante la formación en el grado adquirimos conocimiento sobre las técnicas y herramientas utilizadas en el análisis de datos y la construcción de modelos predictivos. Un aspecto esencial para todo el desarrollo del proyecto, especialmente en el análisis de las curvas de ventas y en el desarrollo de los experimentos.
- “Análisis y resolución de problemas”: Hemos partido de un problema inicial que necesita un tiempo más largo del disponible, entonces hemos hecho un análisis del problema y hemos resuelto el problema para un grupo de los MOCACO. Una vez analizado y acotado el problema, hemos ido desarrollando experimentos que cada uno resolvía un área de mejora y tras ir resolviendo subproblemas obtenemos la solución del problema inicial.
- “Comunicación efectiva”: comunicamos de forma efectiva utilizando los recursos necesarios tanto en la memoria del TFG como en la defensa. Además, llevamos a cabo una buena comunicación durante todas las reuniones realizadas con el equipo.
- “Planificación y gestión del tiempo”: hemos seguido un plan para realizar las tareas en el tiempo especificado y si hemos tenido retrasos los hemos resulto de efectivamente para obtener el resultado final en la fecha indicada.
- “Instrumental especifico”: hemos seleccionado las tecnologías adecuadas para llevar a cabo el proyecto, tenemos herramientas de análisis, visualización y machine learning.
- “Aprendizaje permanente”: hemos buscado las tecnologías necesarias para realizar el proyecto, las herramientas de trabajo y la teoría detrás de lo aplicado. Para ello, ha sido necesario un aprendizaje continuo a lo largo de todo el trabajo.
- “Aplicación y pensamiento crítico”: Aplicamos el conocimiento teórico adquirido y establecemos un proceso para alcanzar los objetivos propuestos, desde el procesamiento de datos hasta la creación del mejor modelo.

7.4 Limitaciones

La principal limitación del proyecto está enfocada en los recursos, específicamente en la memoria del clúster de Databricks que utilizamos para entrenar los modelos.

Por esta limitación aparece la necesidad de entrenar un modelo distribuido que permita entrenar con una muestra de entrenamiento mayor. Si tuviésemos más memoria se podrían entrenar otros tipos de modelos o incluso el mismo modelo, pero sin entrenamiento distribuido, lo que hubiese ahorrado tiempo, ya que no teníamos conocimientos previos sobre modelos distribuidos y hubiese permitido aprovechar el código realizado en los experimentos anteriores.

Además, tener más memoria permitiría poder entrenar los 21 modelos de cada experimento a la vez y no tener que ir entrenando por grupos de modelos. De esta forma se podría dejar entrenando los modelos fuera del horario laboral y al día siguiente tendríamos todos los modelos entrenados listos para continuar con el trabajo.

Otro problema que nos ha ocasionado la falta de memoria es a la hora de realizar la optimización de hiperparámetros, ya que, si hacemos muchas iteraciones para probar distintos valores, tenemos un problema de “out of memory”, ya que es como entrenar muchos modelos seguidos y coger el que minimice el error.

Esta ha sido la única limitación que hemos tenido, ya que la empresa nos ha proporcionado todos los recursos necesarios para realizar el trabajo lo más rápida y cómodamente posible.

7.5 Trabajos futuros

Como hemos comentado anteriormente, el trabajo se ha realizado para ZARA, la sección de mujeres, los MOCACO que son fantasías. Ahora, lo que quedaría para completar la sección de mujeres es completar los tipos de MOCACO, por lo que tendríamos que hacer los modelos para los básicos y la clase otros.

Una vez completada la sección de mujer, tendríamos que realizar el trabajo para las otras secciones, que son hombre y niño. Ya completados todos los modelos para ZARA se realizaría el mismo procedimiento para el resto de las marcas que forman el grupo, es decir, se tendría que hacer para Pull&Bear, Massimo Dutti, Bershka, Stradivarius, Oysho, Lefties y Zara Home.

De esto forma, obtendríamos el modelo que ayuda a la gestión de las campañas comerciales de todas las marcas del grupo Inditex, permitiendo ahorrar y facturar millones de euros al mejorar la gestión de los MOCACO.

Además, sería conveniente explorar otros planteamientos de modelos, es decir, en nuestro caso tenemos un modelo que está hecho para una semana, tipo de MOCACO e incluye a todas las tiendas, se podría explorar la opción de crear un modelo por semana, tienda y tipo de MOCACO. Esto permitiría al modelo ajustarse mejor a cada tienda y podría minimizar el error.

En cuanto a las variables, hay una de ellas que hace referencia a la predicción e la familia, en el proyecto calculamos la venta de todos los MOCACO que forman la familia, esto incluye sumar la venta del MOCACO que queremos predecir. Lo hacemos porque queremos experimentar si esta variable es de ayuda, visto el resultado de que sí que es importante, deberán realizar un submodelo de regresión que sea capaz de obtener este valor.

Otra tarea que se debería realizar es crear modelos que actúen según el día del MOCACO en la tienda. Por ejemplo, el modelo propuesto es para el día cero porque no tiene información de los MOCACO. Sin embargo, una vez que llega a la tienda, se deberían crear modelos para el día uno, día dos, etc. Estos modelos permitirían utilizar

la información disponible del MOCACO en la tienda, lo cual ayudaría a minimizar el error con el paso del tiempo y haría que la predicción sea más fiable.

Bibliografía

1. **INDITEX.** Resultados consolidados Ejercicio 2023. [En línea] 13 de 03 de 2024. https://www.inditex.com/itxcomweb/es/prensa/detalle-noticias?contentId=10da31b6-0c12-43e4-9e33-103766d27821.

2.**INDITEX**. techWelcome. [En línea] 17 de 2 de 2024. https://techwelcome.inditex.com/who-we-are.

3. **INDITEX.** Tra!n. [En línea] NETEX, 23 de 01 de 2024. [Citado el: 25 de 05 de 2024.] https://inditex-learnerportal.learningcloud.me/.

4. **Naciones Unidas.** Objetivos de Desarrollo Sostenible. [En línea] [Citado el: 1 de 6 de 2024.] https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/.

5. **Castiñeira, Ángel, y otros.** *La contribución de las empresas españolas a los Objetivos de Desarrollo Sostenible.* Barcelona : Observatorio de los ODS, 2024.

6. **Naciones Unidas.** Objetivo 8: Promover el crecimiento económico inclusivo y sostenible, el empleo y el trabajo decente para todos. [En línea] [Citado el: 1 de 6 de 2024.] https://www.un.org/sustainabledevelopment/es/economic-growth/.

7. **Naciones Unidas.** Objetivo 12: Garantizar modalidades de consumo y producción sostenibles. [En línea] [Citado el: 1 de 6 de 2024.] https://www.un.org/sustainabledevelopment/es/sustainable-consumption-production/.

8. **Naciones Unidas**. Objetivo 9: Construir infraestructuras resilientes, promover la industrialización sostenible y fomentar la innovación. [En línea] [Citado el: 1 de 6 de 2024.] https://www.un.org/sustainabledevelopment/es/infrastructure/.

9. **Naciones Unidas**. Objetivo 13: Adoptar medidas urgentes para combatir el cambio climático y sus efectos. [En línea] [Citado el: 1 de 6 de 2024.] https://www.un.org/sustainabledevelopment/es/climate-change-2/.

10. **Ramos, Patricia, Santos, Nicolau y Rebelo, Rui.** *Performance of state space and ARIMA models for consumer retail sales forecasting.* ElSevier, 2015, Vol. 34.

11. **Pereira da Veiga, Claudimar, y otros**. *Journal of retailing and consumer services.* Elsevier, 2016.

12. **Nunnari, Giuseppe y Nunnari, Valeria.** *Forecasting Monthly Sales Retail Time Series: A Case Study.* Thessaloniki : CBI, 2017.

13. **Zunic, Emir, y otros.** *Application of facebook's prophet algorithm for successful sales forecasting based on real-world data.* 2, Bosnia : International Journal of Computer Science & Information Technology, 2020, Vol. 12.

14. **Karb, Tristan, y otros.** *A network-based transfer learning approach to improve sales forecasting of new products.* Machine Learning, 2020.

15. **Pavez Cofré, Daniela Alejandra.** *Modelo de predicción de desempeño para productos nuevos en una empresa de retail.* Santiago de chile : Universidad de Chile Facultad de Ciencias Físicas y Matáticas, 2015.

16. **Papadopoulos, Stefanos, y otros.** *Multimodal quasi-autoregression: forecasting the visual popularity of new fashion products.* International Journal of Multimedia Infromation retrieval, 2022.

17. **Singh, Pawan Kumar, y otros.** *Fashion Retail: Forecasting Demand for New Items.* Alaska, 2019.

18. **Van Steenbergen, R.M y Mes, M.R.K.** *Forecasting demand profiles of new products.* ElSelvier, 2020.

19. **Melet Padrón, Alejandro.** *La investigación cualitativa en el marco de la ciencia jurídica.* Anuario, 2018, Vol. 41.

20. **Cotino Hueso, Lorenzo.** *Confidencialidad y protección de datos en la mediación en la Unión Europea.* 41, Valencia : IUS, 2018, Vol. 12.

21. **Microsoft.** Precios de Azure Databricks. [En línea] 3 de 4 de 2024. [Citado el: 4 de 5 de 2024.] https://azure.microsoft.com/es-es/pricing/details/databricks/.

22. **Lenovo.** Productos. [En línea] 29 de 3 de 2024. https://www.lenovo.com/es/es/laptops/?orgRef=https%253A%252F%252Fwww.google.com%252F&utf8Encode=function()%7Breturn%20unescape(encodeURIComponent(this))%7D&utf8Decode=function()%7Btry%7Breturn%20decodeURIComponent(escape(this))%7Dcatch(e)%7Breturn%20this%.

23. **Izaureta, Fernando y Saavedra, Carlos.** *Redes neuronales artificiales.* Departamento de Física, Universidad de Concepción Chile, 2000.

24. **Salas, Rodrigo.** Redes neuronales artificiales. Universidad de Valparaıso. Departamento de Computación, 2004, Vol. 1.

25. **López, Raquel y Fernández, José Miguel.** *Las redes neuronales artificiales.* Netbiblo, 2008.

26. **Guijas, Pedro.** *Complementary outfit generation using diffusion models.* A Coruña : Universidade da Coruña, 2024.

27. **Wirth, Sam.** XGBoost: Theory and Application. [En línea] Medium, 18 de 4 de 2023. [Citado el: 2 de 5 de 2024.] https://medium.com/hoyalytics/xgboost-theory-and-application-4801a5dba4fb#:~:text=The%20Theory%20Behind%20XGBoost&text=The%20first%20step%20is%20identifying,value%20of%20the%20target%20variable..

28. **Meng Y, Yang N, Qian Z, Zhang G.** *What Makes an Online Review More Helpful: An Interpretation Framework Using XGBoost and SHAP Values.* Journal of Theoretical and Applied Electronic Commerce, 2021.

29. **Zhang, Junyi** . *Insights into geospatial heterogeneity of landslide susceptibility based on the SHAP-XGBoost model.* Journal of environmental management, 2023, Vol. 332.

30. **Kim, Sungil.** *A new metric of absolute percentage error for intermittent demand forecasts.* ElSelvier, 2016, Vol. 32.

31. **Databricks.** What is Databricks? [En línea] Databricks, 22 de 5 de 2024. [Citado el: 25 de 5 de 2024.] https://docs.databricks.com/en/introduction/index.html.

32. **Amazon Web Service.** What is Python? [En línea] Amazon Web Service, 12 de 4 de 2024. [Citado el: 28 de 5 de 2024.] https://aws.amazon.com/es/what-is/python/#:~:text=Python%20es%20un%20lenguaje%20de,ejecutar%20en%20muchas%20plataformas%20diferentes.

33. **Microsoft.** Access SQL: conceptos básicos, vocabulario y sintaxis. [En línea] 3 de 2 de 2024. [Citado el: 25 de 4 de 2024.] https://support.microsoft.com/es-es/topic/access-sql-conceptos-b%C3%A1sicos-vocabulario-y-sintaxis-444d0303-cde1-424e-9a74-e8dc3e460671.

34. **Balanza, Raúl.** *Use of deep learning generative models for Monete Carlo event similuation in the context of LHC experiments.* Valencia, 2022.

35. **Apache Spark.** PySpark Overview. [En línea] 24 de 2 de 2024. https://spark.apache.org/docs/latest/api/python/index.html.

36. **Apache Spark**. *MLlib.* [En línea] 27 de 3 de 2024. [Citado el: 30 de 5 de 2024.] https://spark.apache.org/mllib/.

37. **TensorFlow.** Introducción a TensorFlow. [En línea] TensorFlow newsletter, 22 de 3 de 2024. [Citado el: 29 de 5 de 2024.] https://www.tensorflow.org/learn?hl=es-419.

38. **Pedragosa, Fabian, y otros.** *Scikit-learn: Machine Learning in Python.* Machine Learning Research, 2011, Vol. 12.

39. **mssaperla, jaseidman y irinaskaya.** Uso de algoritmos de aprendizaje distribuido con Hyperopt. [En línea] Azure Databricks, 1 de 3 de 2024. https://learn.microsoft.com/es-es/azure/databricks/machine-learning/automl-hyperparam-tuning/hyperopt-distributed-ml.

40. **Dython.** Dython. [En línea] 1 de 4 de 2024. [Citado el: 1 de 5 de 2024.] https://shakedzy.xyz/dython/.

41. **Prieto, Diego.** *Visualización y uso de técnicas de Aprendizaje Máquina Supervisado para la prediccion de resultados de la Fórmula 1.* A coruña : Universidade Da Coruña, 2023.

42. **Cassani, Raymundo.** Multilayer perceptron example. *Multilayer perceptron example.* [En línea] Github, 25 de 10 de 2020. [Citado el: 12 de 4 de 2024.] https://github.com/rcassani/mlp-example.

ANEXO 1: Objetivos de Desarrollo Sostenible

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

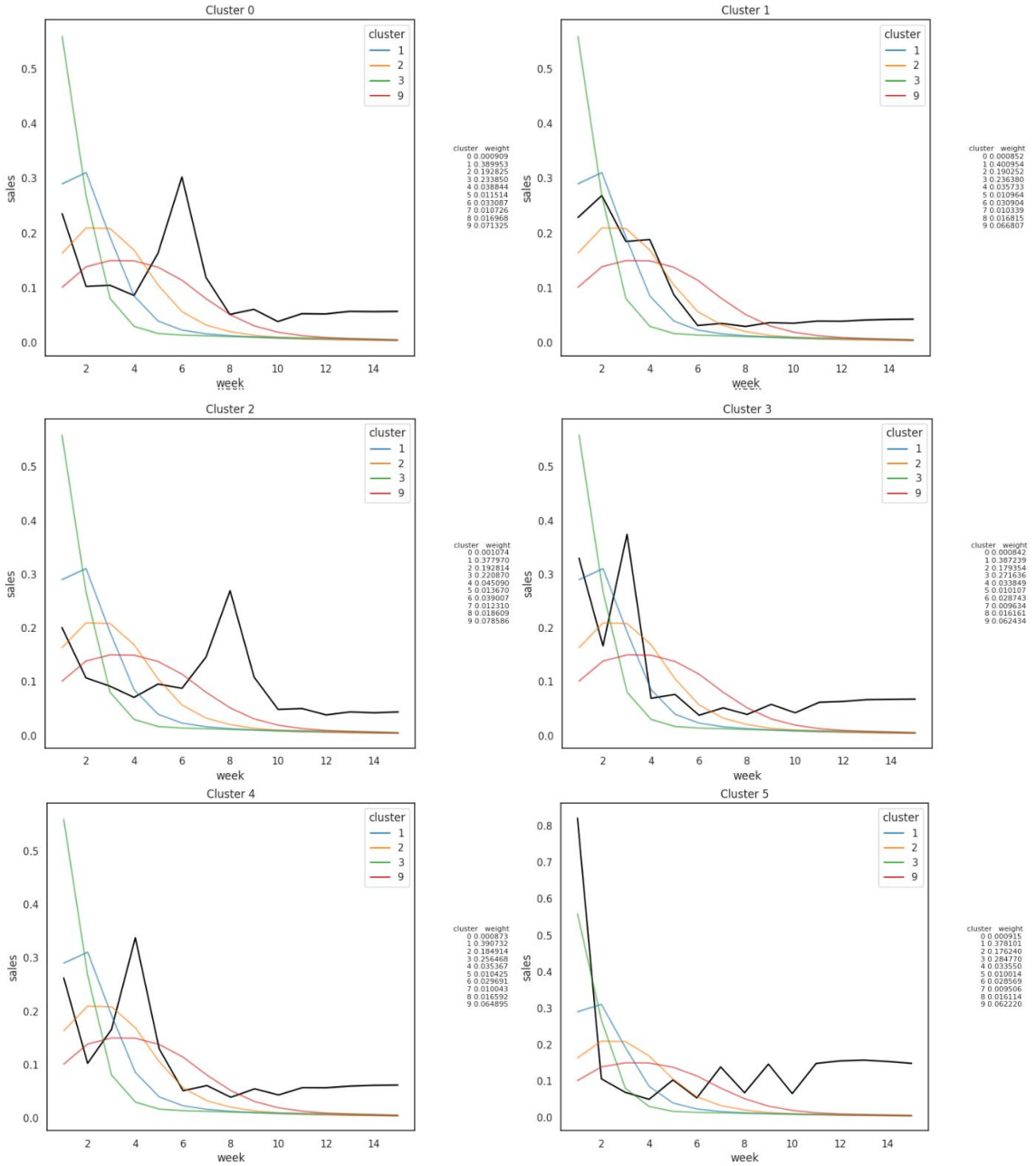
Objetivos de Desarrollo Sostenible	Alto	Medio	Bajo	No procede
ODS 1. Fin de la pobreza.				X
ODS 2. Hambre cero.				X
ODS 3. Salud y bienestar.				X
ODS 4. Educación de calidad.				X
ODS 5. Igualdad de género.				X
ODS 6. Agua limpia y saneamiento.			X	
ODS 7. Energía asequible y no contaminante.				X
ODS 8. Trabajo decente y crecimiento económico.	X			
ODS 9. Industria, innovación e infraestructura.	X			
ODS 10. Reducción de las desigualdades.				X
ODS 11. Ciudades y comunidades sostenibles.			X	
ODS 12. Producción y consumo responsables.	X			
ODS 13. Acción por el clima.	X			
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.			X	
ODS 16. Paz, justicia e instituciones sólidas.				X
ODS 17. Alianzas para lograr objetivos.				X

Tabla 25. Objetivos de desarrollo sostenible

En cuanto al impacto en los Objetivos de Desarrollo Sostenible (ODS) (4), consideramos que, de los objetivos globales establecidos por las Naciones Unidas, nuestro TFG tiene mayor impacto en los siguientes (5):

- **ODS 8. Trabajo decente y crecimiento económico**: mejorar la precisión en las previsiones de ventas puede llevar a una mejor planificación de la producción y distribución, evitando sobreproducción y reducción de inventarios, lo que puede aumentar la eficiencia operativa y la rentabilidad. Esto puede generar empleo estable y mejorar las condiciones laborales en toda la cadena de suministro. (6)
- **ODS 12. Producción y consumo responsable**: una previsión más precisa de la demanda puede ayudar a reducir el desperdicio de productos y materiales, ya que se producirán y distribuirán solo las cantidades necesarias. Esto puede minimizar los residuos textiles y la huella ecológica de la empresa. (7)
- **ODS 9. Industria, innovación e infraestructura**: el uso de tecnologías avanzadas para predecir ventas puede promover la innovación dentro de la industria de la moda, fomentando el desarrollo de infraestructuras tecnológicas más robustas y avanzadas. (8)
- **ODS 13. Acción por el clima**: mejorar la eficiencia en la cadena de suministro puede reducir las emisiones de carbono relacionadas con el transporte y la logística. Una mejor planificación puede significar menos viajes y más rutas eficientes. (9)

ANEXO 2: Curvas de venta a nivel tienda



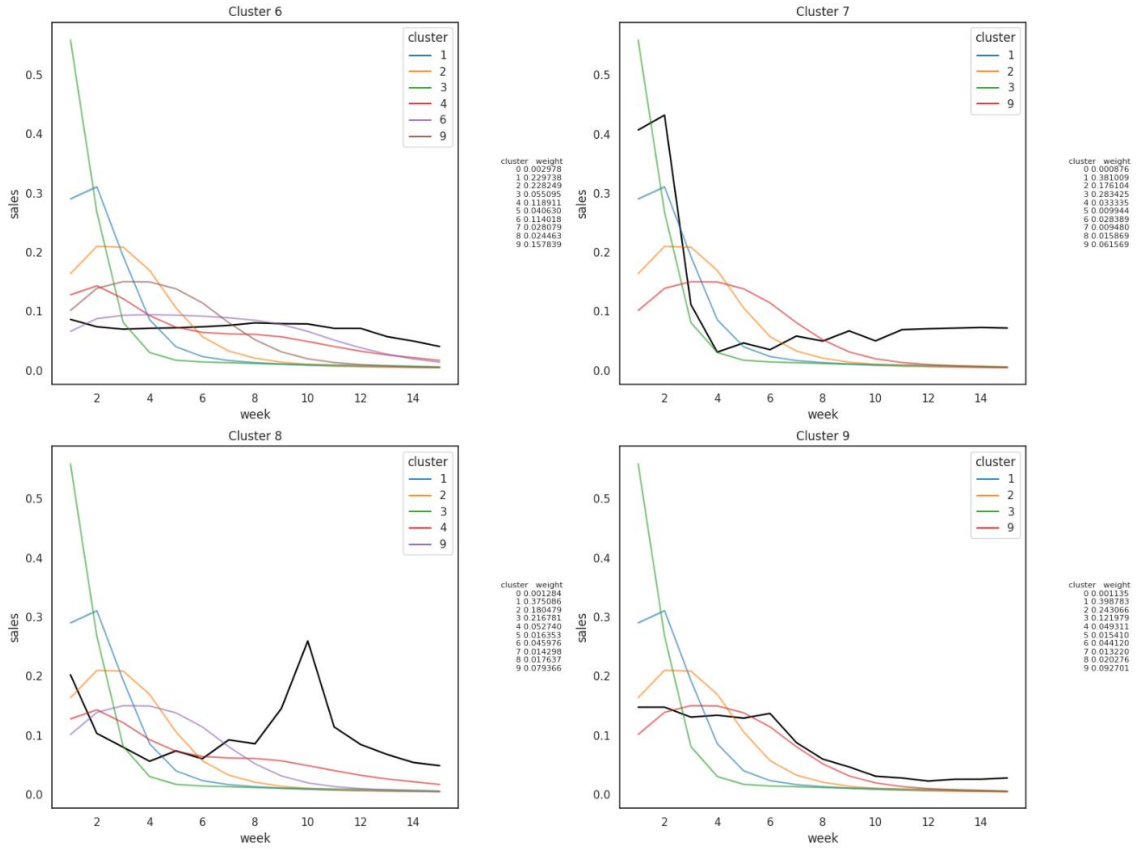


Ilustración 26. Comparación curva de venta de los clústeres de nivel tienda con los clústeres a nivel mundo que tienen al menos una representación de 5%

ANEXO 3: Distribución del error

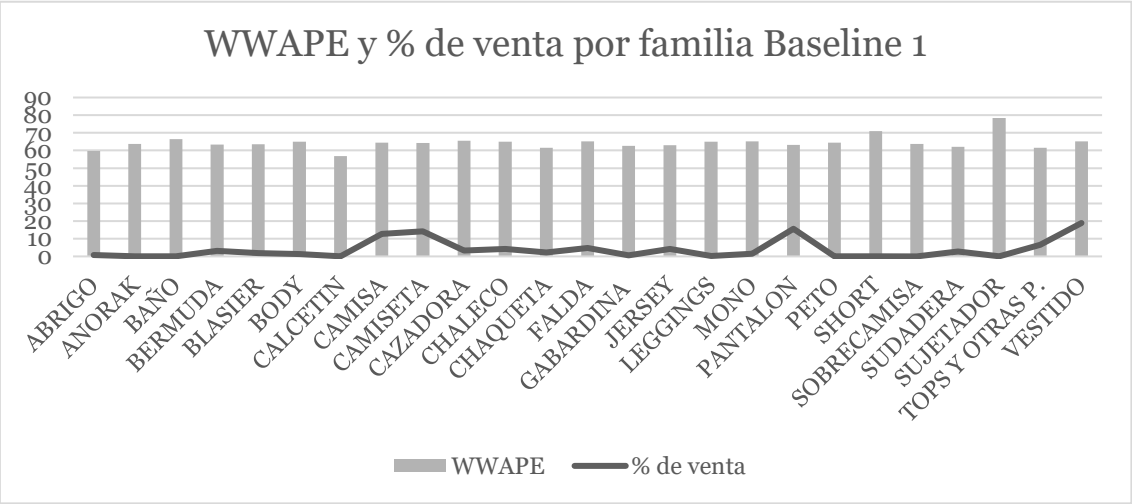


Ilustración 28. Distribución del WWAPE y % de venta por familia del experimento 1

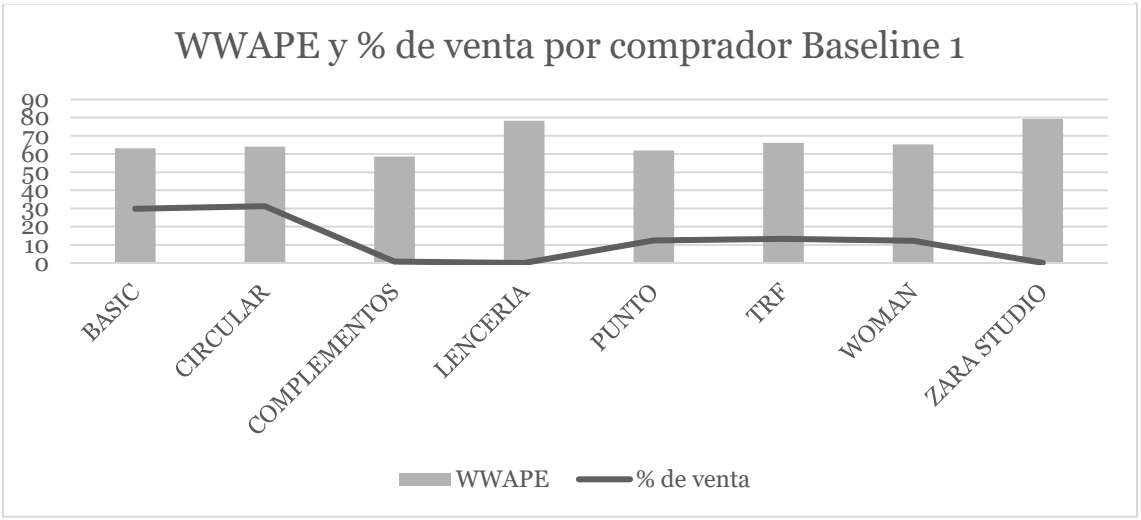


Ilustración 27. Distribución del WWAPE y % de venta por comprador del experimento 1

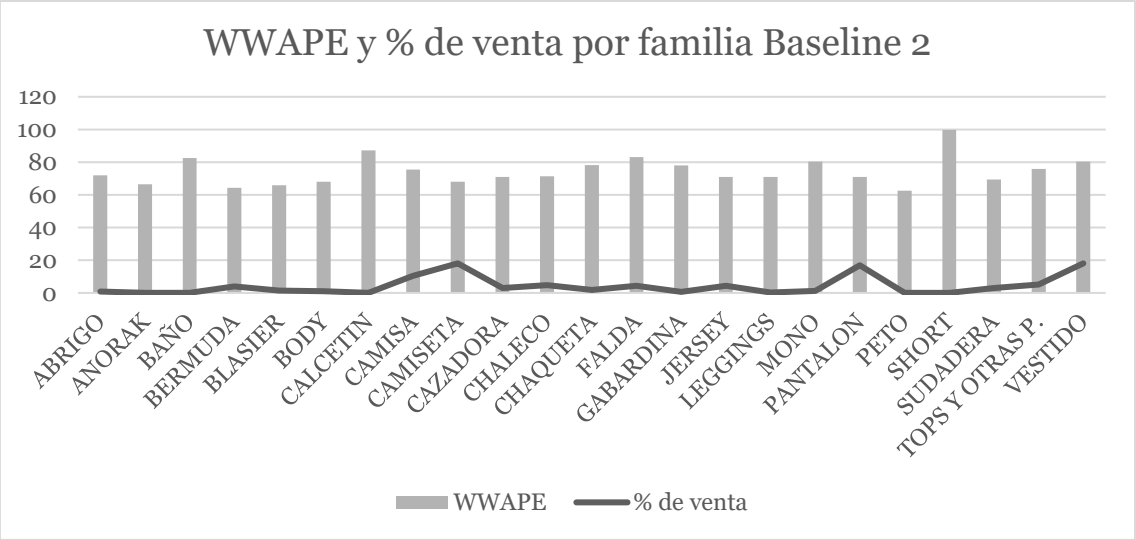


Ilustración 30. Distribución del WWAPE y % de venta por familia del experimento 2

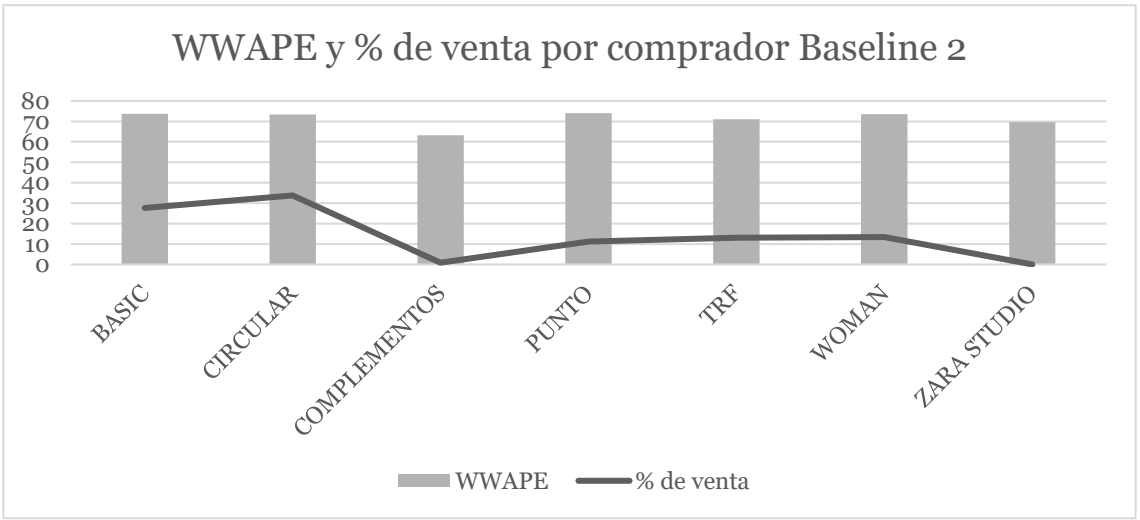


Ilustración 29. Distribución del WWAPE y % de venta por comprador del experimento 2

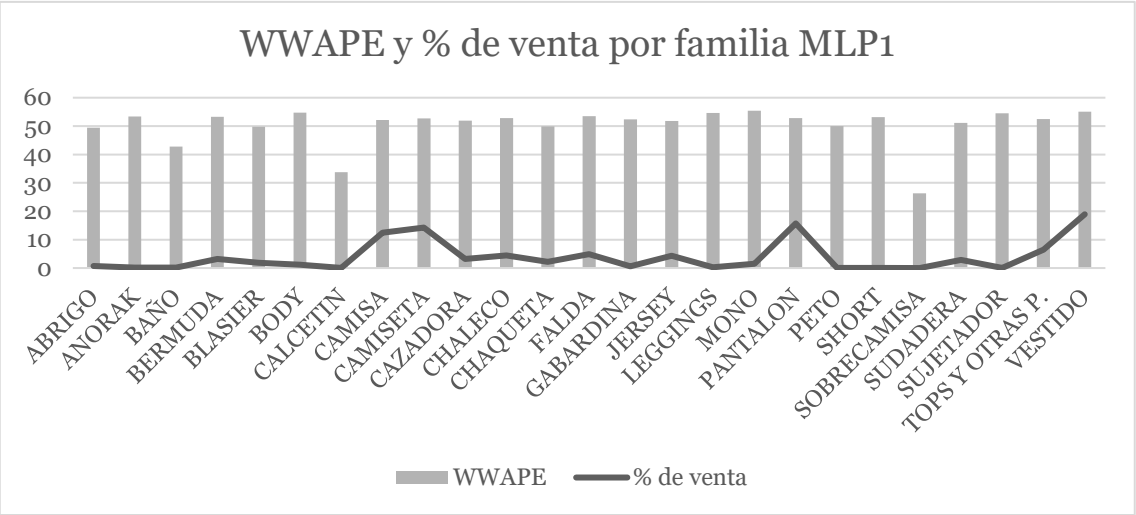


Ilustración 32. Distribución del WWAPE y % de venta por familia del experimento 3

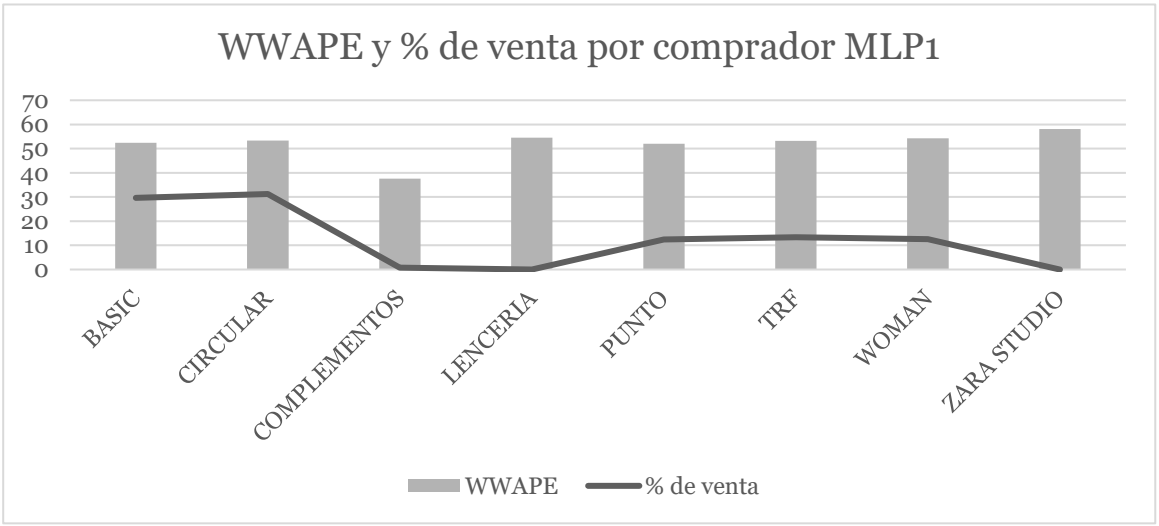


Ilustración 31.Distribución del WWAPE y % de venta por comprador del experimento 3

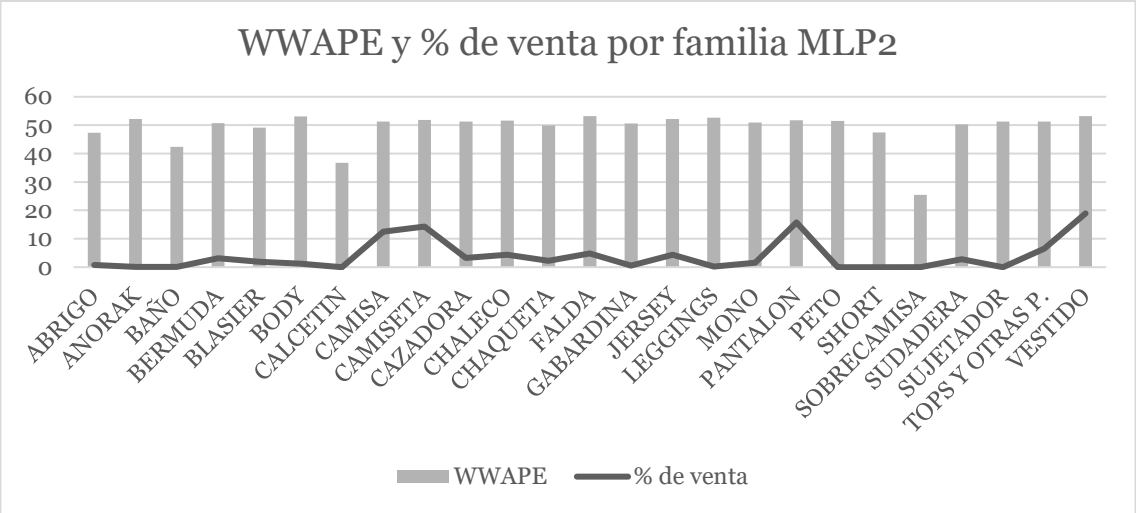


Ilustración 34. Distribución del WWAPE y % de venta por familia del experimento 4

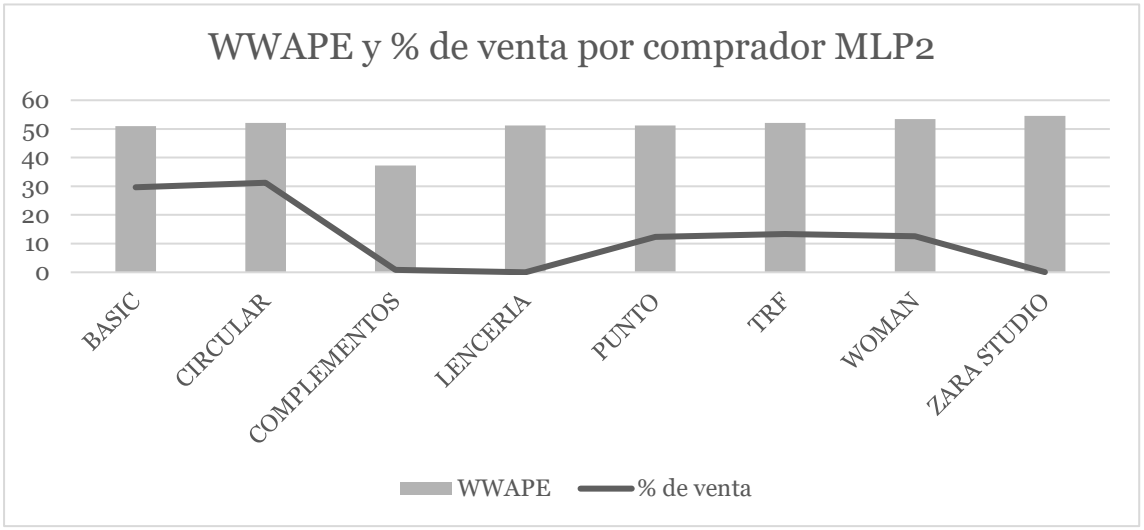


Ilustración 33. Distribución del WWAPE y % de venta por comprador del experimento 4

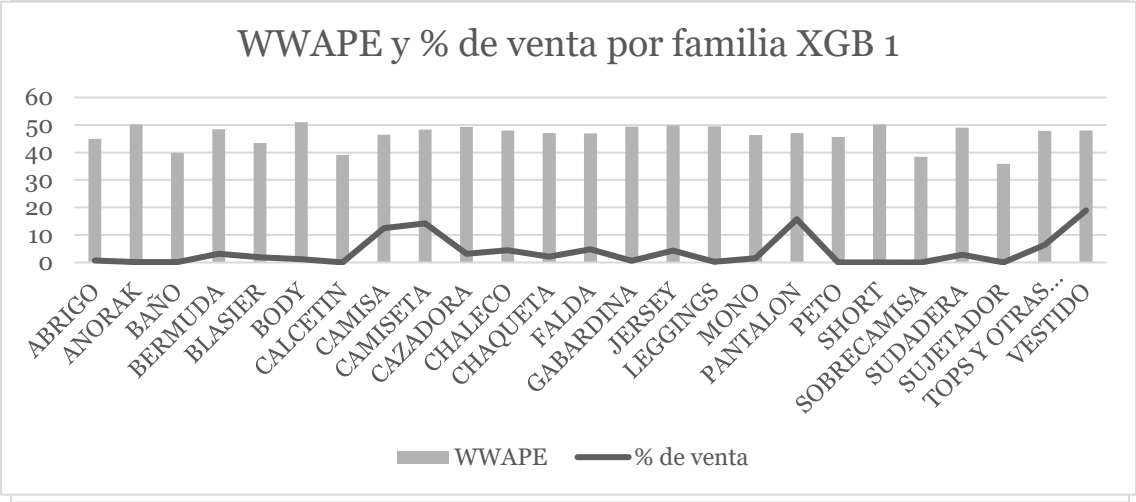


Ilustración 38.Distribución del WWAPE y % de venta por familia del experimento 6

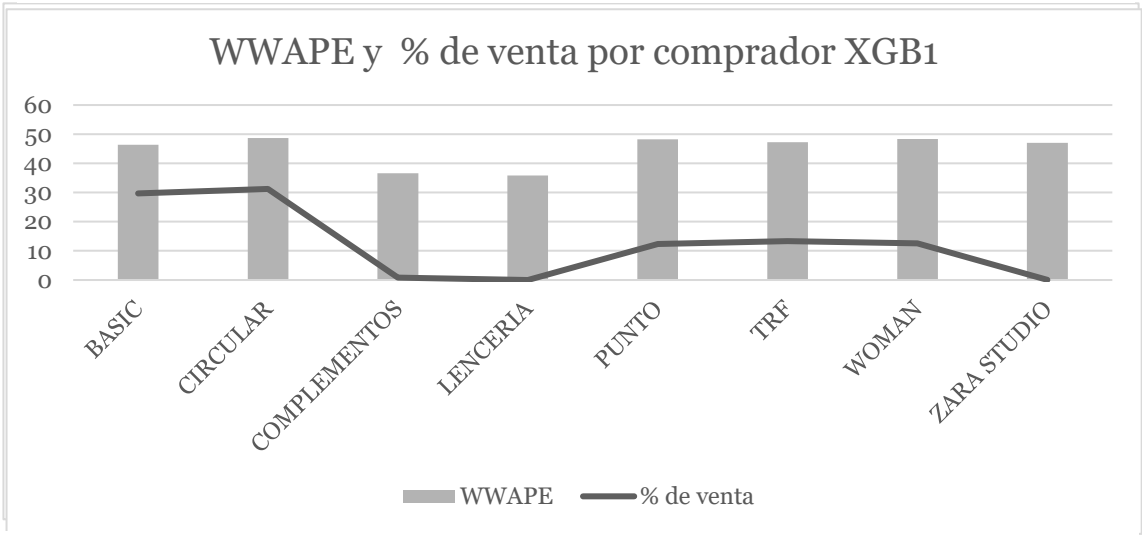


Ilustración 37. Distribución del WWAPE y % de venta por comprador del experimento 6

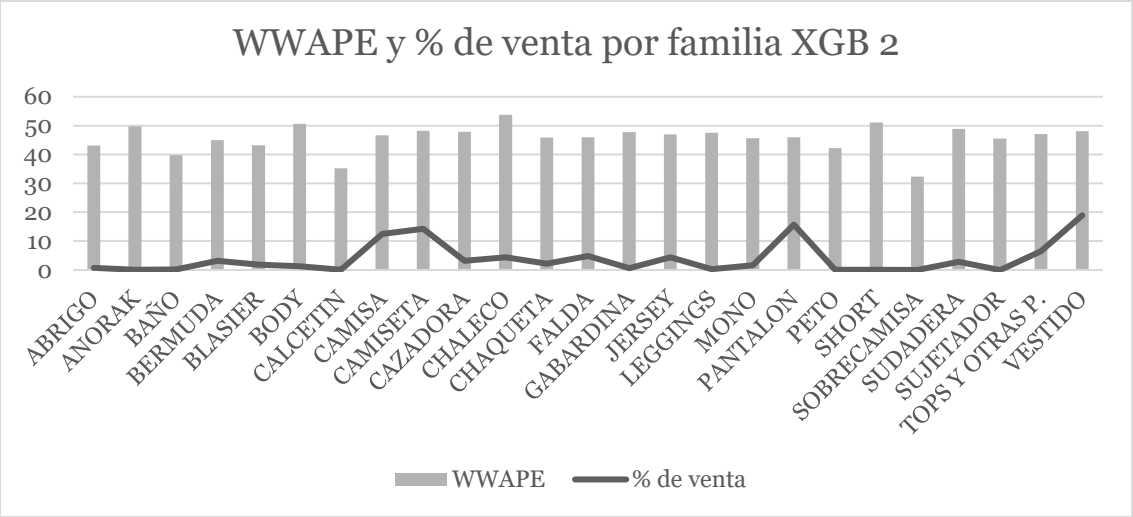


Ilustración 40. Distribución del WWAPE y % de venta por familia del experimento 7

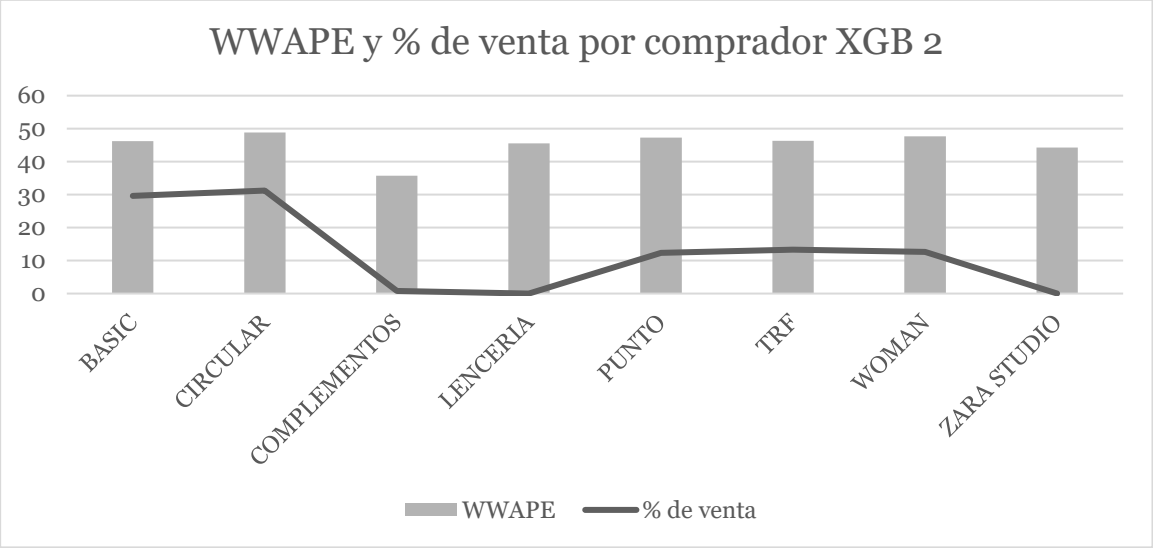


Ilustración 39. Distribución del WWAPE y % de venta por comprador del experimento 7