Gestión avanzada de inventario.

Machine Learning aplicado a la predicción de la demanda.

Javier Hernández Hernández

Gestión avanzada del inventario

Área 4

Tutor/a de TFM

Lorena Polo Navarro

Profesor/a responsable de la asignatura

Antonio Lozano Bagen

Fecha Entrega

14/01/2023

Universitat Oberta de Catalunya

Ficha del Trabajo Final

Título del trabajo:	Gestión avanzada del inventario. Machine Learning aplicado a la predicción de la demanda.
Nombre del autor/a:	Javier Hernández Hernández
Nombre del Tutor/a de TFM:	Lorena Polo Navarro
Nombre del/de la PRA:	Antonio Lozano Bagen
Fecha de entrega:	02/2023
Titulación o programa:	Máster en ciencia de datos
Área del Trabajo Final:	Área 4
Idioma del trabajo:	Castellano
Palabras clave	Stock, demand, forecasting

Resumen del Trabajo

Nos encontramos en una era de continuos avances tecnológicos. Esta situación, trasladada al mundo empresarial, obliga a una continua adaptación y mejora de la eficiencia para asegurar la subsistencia y crecimiento de la empresa.

Dentro de las diferentes áreas de la empresa, el área de la logística encargada de la gestión del inventario es un pilar del que dependen otras áreas de la empresa como son producción y ventas. Una gestión eficiente del inventario llevará consigo mejoras en estas áreas.

El inventario pretende satisfacer la demanda tanto interna como externa de la empresa. Este trabajo se enfoca en la gestión del inventario desde la predicción de la demanda externa, en el marco de una empresa comercial.

Se aplicará metodología machine learning, bajo entorno R o Python para elaborar modelos predictivos de la demanda externa. La aplicación del modelo permitirá una mejor gestión del inventario existente en la empresa para satisfacer dicha demanda de una manera más eficiente.

Este trabajo fin de máster pretende elaborar una metodología para elaborar modelos de predicción de la demanda externa, a partir de herramientas de machine learning.

Abstract

We are in an era of continuous technological advances. This situation, transferred to the business world, requires a continuous adaptation and improvement of efficiency to ensure the survival and growth of the company.

Within the different areas of the company, the logistics area in charge of inventory management is a pillar on which other areas of the company depend, such as production and sales. Efficient inventory management will lead to improvements in these areas.

The inventory aims to satisfy both internal and external demand of the company. This work focuses on inventory management from the prediction of external demand, within the framework of an commerce company.

Machine learning methodology will be applied, under the R or Python environment, to develop predictive models of external demand. The application of the model will allow a better management of the existing inventory in the company to satisfy said demand in a more efficient way.

This master's thesis aims to develop a methodology to develop prediction models of external demand, based on machine learning tools.

Índice

Lista de tablas.

Lista de Figuras

1. Introducción

Vivimos en una sociedad en continua evolución. Esta es la clave de la subsistencia de la raza humana. Así es, si nos remontamos a los orígenes de la humanidad, podemos corroborarlo. Comenzamos siendo una sociedad de cazadores y recolectores caracterizados por una vida nómada. Seguidamente pasamos a una sociedad ganadera y hortícola, consecuencia del sedentarismo. Pasamos a una tercera evolución hacia sociedad agrícola en la que se seleccionaba y perfeccionaba las variedades que se cultivaban. Pasaron unos siglos hasta la siguiente evolución hacia una sociedad industrial que supuso la aparición del capitalismo y un objetivo hacia el estado del bienestar. Nos encontramos ahora en una fase de sociedad postindustrial que ha tocado techo y busca hacia donde dirigir sus miras.

En todas estas evoluciones encontramos un denominador común, la búsqueda de la perfección y superación, en sus diferentes matices. La transición de la sociedad recolectora a la sociedad ganadera supuso el origen del hogar. Se abandona la vida nómada, y se construyen asentamientos que han ido mejorando hasta las construcciones actuales. En el paso a sociedad agrícola, se comienza a entender la recolección no para la subsistencia y autoconsumo sino como una producto comercial. Se cultiva a gran escala para trueque y comercialización. En la sociedad industrial, se incorpora una búsqueda hacia el estado del bienestar. La industria crea productos cada vez más perfeccionados con los que se busca facilitar y hacer más cómoda la vida de las personas.

Ahora nos encontramos en un momento de transición, hacia una sociedad que queda por definir. La rápida evolución de la tecnología y la computación, con ordenadores cada vez más potentes y capaces están marcando el rumbo de esta evolución. Cualquier cosa del pasado es ampliamente mejorable en la actualidad gracias a esta nueva tecnología. Se incorpora en esta fase el concepto de eficiencia y mejora continua. Si un producto puede fabricarse con un menor coste y ahorro de residuos es preferible a otro que no lo sea. De igual modo, si un servicio puede prestarse en un menor tiempo y con mayor grado de satisfacción para el clientes será preferible a otro que no lo sea.

Esta reflexión lleva al tema en el que se enfoca este trabajo de fin de máster. Se quiere mejorar la eficiencia en la gestión de inventario, con el apoyo en la tecnología de Machine Learning. La información existente es infinita. La tecnología y computación existente hacen posible su procesamiento en aras de conseguir respuesta a tantas cuestiones que nos podemos plantear. En el caso concreto de este trabajo, la cuestión que se plantea es entender y conocer el comportamiento del consumidor a la hora de tomar la decisión de compra.

Se inicia el trabajo con un conjunto de datos extraídos de una empresa retail, sobre las ventas realizadas de un producto en un espacio de tiempo, y el perfil de sus clientes. Se complementa con otra información que pueda ayudarnos a obtener este patrón que

buscamos como puede ser festividad, meteorología, ubicación geográfica, etc. La búsqueda no está limitada, salvo por las barreras que nos imponga la Ley orgánica de Protección de Datos (LOPD) y la tecnología. Si esto lo permite, se podría profundizar y complementar con búsquedas por redes sociales, estados de ánimo, sentimientos, etc. El objetivo es detectar aquellos factores de los que mayor dependencia tiene el hecho de comprar y construir un modelo que permita su predicción en un determinado espacio temporal.

Si esto se consiguiera, la empresa dispondría de un algoritmo sobre el que construir la cantidad de producto terminado que debe almacenar en cada momento, para satisfacer la demanda de sus clientes, sin demoras ni rotura de stock. Así mismo, se consigue ahorro en costes derivados del almacenaje de producto, compra de materia prima y producción programada. Para demostrarlo, se tomará de referencia una serie de magnitudes, sobre las que tomaremos valor antes de implantar el modelo, y después. Si nuestra hipótesis fuera cierta, debería obtenerse una mejora en estas magnitudes.

Este trabajo fin de máster pretende elaborar una metodología para elaborar modelos de predicción de la demanda externa, a partir de herramientas de machine learning.

1.1. Contexto y justificación del Trabajo

El sector comercial se enfrenta a un problema por resolver, como es ajustar y optimizar la dimensión de sus niveles de stock, de modo que permita abastecer satisfactoriamente la demanda de sus clientes y no suponga un coste excesivo en términos de inmovilizado. En la resolución de este problema, deben cumplirse ante todo, una serie de premisas;

- No reducir la calidad de servicio al cliente.
- Asegurar no rotura de stock.

Si la empresa consigue optimizar esta gestión, consigue una ventaja competitiva con respecto a la competencia. Pero además de esta ventaja, se consigue mejorar el conocimiento que se tiene de sus clientes y sus hábitos de consumo, con lo que podrán realizar acciones de marketing personalizadas.

Mi vida profesional no se ha desarrollado en un sector de mi vocación. Llevo 25 años trabajando en el sector financiero. Desde mi formación académica en Estadística, he tenido inclinación hacia la gestión de la información y extracción de conclusiones a partir de los datos. Tengo la oportunidad en este trabajo, y gracias a los conocimientos adquiridos en el Máster de Ciencia de Datos, de abordar una problemática existente en el sector industrial. Como resultado de este trabajo, me gustaría aportar valor y conocimiento hacia la optimización en los procesos industriales, concretamente, en la gestión del inventario. Me motiva poder aportar valor como resultado de este trabajo.

1.2. Objetivos del Trabajo

El objetivo general de este trabajo es conseguir un modelo que permita predecir la demanda de un producto en un espacio temporal.

Otros objetivos específicos;

- Optimizar la gestión del inventario de la empresa, con reducción en el coste asociado al stock de seguridad.

1.3. Impacto en sostenibilidad, ético-social y de diversidad

La competencia de compromiso ético y global (CCEG) está definido como;

"Actuar de manera honesta, ética, sostenible, socialmente responsable y respetuosa con los derechos humanos y la diversidad, tanto en la práctica académica como en la profesional, y diseñar soluciones para mejorar estas prácticas."

Se abordan por tanto en este trabajo, tres dimensiones del compromiso CCEG, alineadas con los Objetivos de desarrollo sostenible de la ONU para 2030 (ODS). Se citan a continuación, para cada una de las dimensiones, cuál de los objetivos se encuadra en este trabajo:

- Sostenibilidad.
  Dentro de la dimensión de sostenibilidad, este trabajo tiene impacto positivo en los siguientes ODS:
  - ODS 9 – Industry, innovation and infraestructure.
  - ODS 12 – Responsable consumption and production.

  Con la aplicación del resultado obtenido en este trabajo, se consigue dimensionar adecuadamente el stock de producto terminado, suficiente para atender su demanda. Se evita una producción en exceso que implicaría mayor desperdicio.

- Comportamiento ético y responsabilidad social (RS).
  El impacto positivo de este trabajo, dentro de la dimensión de comportamiento ético y responsabilidad social, son;
  - ODS 8 – Decent work and economic growth.

  Este trabajo pretende acompañar en el crecimiento económico, gracias a una mejor gestión del stock. Se ahorra en costes de almacenamiento y mejora la experiencia hacia el cliente.

- Diversidad (género entre otros) y derechos humanos.

Por último, y en la dimensión de diversidad y derechos humanos, este trabajo tiene impacto positivo en los siguientes:

- ODS 5 – Gender equality
- ODS 10 – Reduce inequalities

A la hora de tratar la información, no se ha tenido en consideración discriminantes por razón de sexo o raza. La información se trata por igual, independientemente de esta condición. El resultado obtenido no discrimina a ningún grupo por sexo o raza.

1.4. Enfoque y método seguido

Este trabajo pretende la aplicación de técnicas de Machine Learning para conseguir definir un modelo que permita la predicción de la demanda futura para un determinado producto y en un espacio de tiempo. Por tanto, la demanda entendida como cantidad de producto, será la variable dependiente en este trabajo.

Para alcanzar este objetivo, se parte de un histórico de la demanda de un producto. A partir de técnicas estadísticas se identificarán primero aquellas variables que mayor dependencia muestran con la variable dependiente. Se utilizarán diferentes modelos supervisados, o combinación de estos, hasta conseguir aquel que mejor predicción haga de la demanda, medida como error cuadrático medio entre el valor obtenido y el valor real.

Entre otros modelos, se harán pruebas con modelos de clasificación, regresión, o redes neuronales. Primero se diseñará un modelo a partir de un conjunto de datos de entrenamiento. Se seleccionará aquel modelo con el que se obtengan mejores resultados para un conjunto de testeo.

En la fase de preparación del data set y obtención de los estadísticos básicos, se utilizará como software el programa R, con apoyo de lenguaje de programación Python, bajo entorno Jupyter. En el diseño del modelo y redes neuronales, se utilizarán librerías específicas para lenguaje de programación Python. Algunas de las librerías a utilizar son; SciPy, NumPy, pandas, Scikit-Learn, TensorFlow, Keras, etc. En la ejecución de diseño en redes neuronales, se utilizará collab de Google que permite trabajar bajo entorno GPU.

Para alojar data set y los diferentes modelos utilizados, así como los resultados obtenidos, se ha creado un repositorio en GitHub.

https://github.com/vespi-github/TFM

Trabajar en este entorno permite también llevar mayor control de las versiones del trabajo.

1.5. Planificación del trabajo

Este trabajo tiene su punto de partida el 28/09/2022. Durante un espacio temporal de 117 días, se planifican las diferentes tareas que llevarán como resultado, la defensa pública en la semana del 23/01/2023 al 03/02/2023. Se ha elaborado un diagrama de Gantt para planificar en este espacio de tiempo, las diferentes tareas a realizar.

Se identifican 6 bloques principales, que coinciden con cada una de las PEC. Dentro de cada bloque, se han desarrollado las diferentes tareas a realizar, en relación con el objetivo de la PEC y dirigido hacia este trabajo.

PEC 1 Definición y planificación del TFM.

- Definición del tema
- Contexto y objetivos
- Enfoque y metodología
- Planificación temporal
- Impacto en sostenibilidad, ético-social y diversidad
- Redacción memoria

PEC 2 Estado del arte

- Lectura bibliografía
- Revisión objetivos
- Revisión bibliografía

PEC 3 Diseño e implementación del TFM

- Definir estado inicial en la gestión de inventario en la empresa
- Extraer data set, demanda de producto
- Completar data set con variables externas
- Limpieza fichero
- Análisis estadístico
- Reducción dimensionalidad. Aplicación algoritmos SVM.
- Definición conjuntos entrenamiento y testeo
- Construcción diferentes modelos predicción.
- Evaluación del modelo
- Selección modelo óptimo y afinamiento
- Evaluación del nuevo modelo
- Definir entregable a la empresa para predecir demanda
- Actuación sobre la gestión del inventario
- Comparativa entre el estado inicial y post-modelo predicción

PEC 4 Redacción memoria, versión previa

- Preparación memoria

- Preparación video
- Preparación presentación

PEC 5 Redacción memoria, versión final

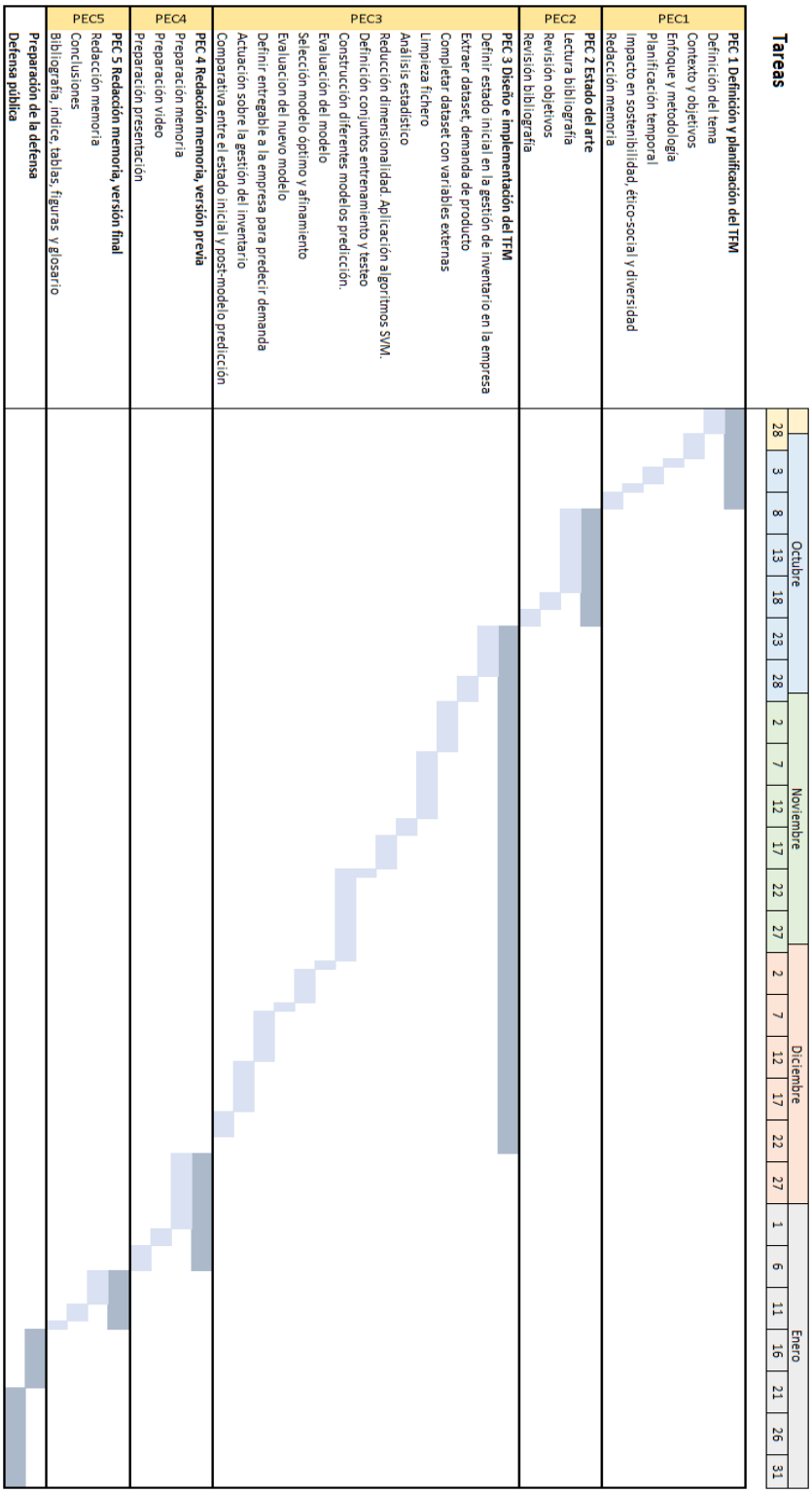
- Redacción memoria
- Conclusiones
- Bibliografía, índice, tablas, figuras y glosario

Fase final.

- Preparación de la defensa
- Defensa pública

La planificación se realiza sobre una carga de trabajo teórica, estimada al inicio. En su avance, pueden realizarse ajustes en función de los resultados objetivos. Puede alterarse la programación de las tareas dentro de cada bloque, no así las fechas de finalización de cada bloque que deben considerarse fijas, y deben respetarse. Se incorpora programación temporal mediante diagrama de Gantt, de las actividades y tareas a realizar.

Tabla 1 Diagrama de Gantt. Fuente; elaboración propia.



1.6. Breve sumario de productos obtenidos

Como resultado de este trabajo, se obtendrá un modelo que permitirá predecir la demanda futura de un artículo, en base a una serie de variables independientes. Con este modelo se pretende cuantificar el stock de seguridad necesario en inventario para poder satisfacer la demanda.

1.7. Breve descripción de otros capítulos de la memoria

Se incluirán en esta memoria otros capítulos con contenido de cómo se ha llegado hasta el resultado final. Se comenzará con un resumen de referencias bibliográficas y papers obtenidos sobre el tema de este TFM . Este apartado conformará lo que se conoce como Estado del Arte.

Se sigue con el desarrollo de trabajo en cuestión. Cada uno de los avances obtenidos y su aplicación al objetivo de este trabajo.

Por último se pasa al apartado de conclusiones donde se pondrá orden a todos los resultados obtenidos, su enfoque al problema en estudio. Hay que cuantificar el beneficio que supone en la gestión del inventario, con una comparativa entre ciertas magnitudes seleccionadas en el momento de partida, y el resultado de estas magnitudes después de implementar el modelo encontrado.

2. Estado del arte

Son muchos los trabajos que encontramos publicados en relación con la predicción de la demanda. Es una cuestión que interesa conocer a la empresa para una mejor gestión del inventario. Como ya se ha explicado en el apartado 1.1, predecir la demanda esperada de un producto en un determinado espacio temporal futuro, es clave para evitar la rotura de stock, y garantizar los plazo de entrega al consumidor.

Para una empresa manufacturera o comercial, la previsión de la demanda es esencial para organizar y planificar la producción, las compras, el transporte y la mano de obra. La predicción se clasifica por tipo y por técnica

Tipos de predicción:

- Cualitativos. Son de carácter subjetivo, basados en estimaciones y opiniones, o por programación de trabajos.
- Análisis de series temporales. A partir de datos históricos se analizan las componentes de tendencia, estacionalidad y ciclo para predecir un valor futuro.
- Relaciones causales. La demanda se predice a partir de su relación con una o más causas. Se aplican modelos de regresión lineal y multilineal.
- Simulación. Se utilizan modelos complejos, simulados por ordenador.

Técnicas y modelos.

- Cualitativos. A partir de método Delphi, por investigación de mercado, consenso de expertos.
- Series temporales. medias móviles, suavizamiento exponencial, técnica de Box Jenkins, descomposición de la estacionalidad, análisis de regresión.
- Relaciones causales. Análisis de regresión lineal y multilineal.
- Simulación computacional. Se aplican algoritmos de aprendizaje automático.

2.1. Modelos tradicionales frente a aprendizaje automático.

Los primeros modelos se diseñaron bajo modelos estadísticos tradicionales, del tipo medias móviles, regresión, probabilísticos y estocásticos. Se construían sobre Excel, a partir de bases de datos de reducida dimensión. En la actualidad, las bases de datos disponibles son de mayor tamaño. Además de los datos propios de la empresa (datos internos), interesa conocer la correspondencia con otras fuentes externas que enriquecen y amplían la información disponible.

Por otro lado, se dispone de ordenadores con mayor potencia computacional y nuevas herramientas de software basadas en Python y R. Por todo esto, y desde principio de este siglo se están probando nuevos modelos de predicción construidos a partir de Machine Learning (ML). El aprendizaje automático aplicado a la predicción de la demanda, a partir de datos reales, llega a alcanzar una precisión del 92,38%. (Adnan Khan, M. et al., 2020)

La prevision de la demanda es la estimación de las ventas de productos/servicios en el futuro sobre la base de los datos presentes y pasados, y de aspectos coyunturales del mercado. A medida que aumenta el tamaño de los datos, el modelo produce resultados más precisos.

Aunque la Machine Learning aplicado a predicciones ha sido objeto de estudios recientes, pocos se han dirigido a la predicción de la demanda. En un estudio reciente, (Ali, O. G. et al, 2009) aplica redes neuronales para predecir la demanda en presencia de promociones. Contribuciones de otros estudios, exploran otros métodos alternativos como los árboles de regresión, soporte vectores de regresión (SVM) y procesos gausianos (Bojer, C. S. & Meldgaard, J. P., 2021).

Con técnicas avanzadas de aprendizaje automático como redes neuronales (NN), redes neuronales recurrentes (RNN) y maquina de soporte vectorial (SVM) se consiguen mejores resultados de rendimiento para prever la demanda que con otros métodos de predicción más tradicionales como la tendencia, la media móvil o la regresión lineal, pero con una precisión en las previsiones que no son estadísticamente significativos (Carbonneau, R., Laframboise, K., & Vahidov, R., 2008).

Las RNN se ejecutan mejor que el método SVM, pero ninguna de las dos supera en precisión, la de otros enfoques tradicionales como las medias móviles o regresión lineal, en la predicción de demanda mensual (Carbonneau, R., Laframboise, K., & Vahidov, R., 2008).

La posibilidad de mejorar la precisión de las previsiones permite reducir los costes derivados de almacenamiento y un mayor grado de satisfacción del cliente gracias a la reducción en los plazos de entrega. El SVM y RNN han demostrado ser las técnicas de previsión más precisas, pero no ofrecen una gran mejora con respecto a los métodos más tradicionales, por su mayor coste computacional. Se cuestiona en el estudio si la mejora en la precisión del modelo, compensa el mayor coste computacional de cálculo, y complejidad del modelo.

El Machine Learning (ML), en particular las redes neuronales, son una propuesta alternativa a los métodos estadísticos tradicionales. La principal ventaja de estos métodos es la utilización de algoritmos no lineales, capaces de aprender por prueba y error, y mejorar su rendimiento con el tiempo mediante la observación de datos históricos. Digamos que esta metodología se retroalimenta continuamente y aprende con los nuevos datos que se van incorporando en su entrenamiento (Spiliotisa E. et al., 2020).

Existe discrepancia entre las bondades del ML frente a los modelos estadísticos tradicionales. En su estudio (Makridakis, S. et al, 2018), los autores evaluaron la precisión de diez métodos populares de ML y ocho métodos estadísticos, y concluyeron que estos últimos funcionan mejor. Sin embargo, en el resultado posterior de la M4 competition (Makridakis, S. et al, 2018), encontraron nuevos métodos basados en algoritmos ML que sí proporcionaban pronósticos precisos.

Los métodos de ML proveen mejores resultados en precisión y son más adecuados en pronóstico de demanda a gran escala (Huber, J. & Stuckenschmidt, H., 2020). Se introduce una nueva característica, el tamaño del data set. Cuando se procesa un fichero formado por gran número de observaciones, el ML ofrece mejores modelos de predicción.

En el estudio de (Spiliotisa E. et al., 2020) se evalúa el rendimiento de los métodos populares de ML para el pronóstico diario de la demanda y compara su precisión y sesgo con los de los métodos estadísticos estándar. Se parte de un conjunto de datos inicial, que se va incrementando continuamente con la aportación de nuevos datos y otros no relacionados. El estudio concluye que incluir la recursividad en el modelo de red neuronal, mejora la precisión de la predicción, pero incrementa el coste computacional.

Los modelos conseguidos bajo métodos de redes neuronales, mejoran a los modelos ARIMA y regresión múltiple, cuando en la serie temporal se observa una clara componente de tendencia y estacionalidad (Huber, J. & Stuckenschmidt, H., 2020). El ajuste previo de la estacionalidad puede mejorar significativamente el rendimiento del modelo de red neuronal (Chu, C.W. & Zhang, G.P., 2003). El mejor modelo es el obtenido sobre una base de datos desestacionalizada o no estacionaria..

En resumen, el estudio de (Spiliotisa E. et al., 2020) concluye que los 4 métodos con mayor precisión en términos de error cuadrático medio son del grupo de Machine Learning. En su trabajo evalúa diferentes modelo obtenidos a partir de métodos estadísticos tradicionales o machine learning, sobre un mismo data set.

Métodos estadísticos evaluados.

Para predicción de la demanda, se utilizan los siguientes modelos estadísticos.

- Naive. La predicción en el tiempo t es igual a la última observación conocida de la serie temporal y.
- Naive stacional. La predicción en el tiempo t es igual a la última observación conocida en el tiempo t-m donde m es la frecuencia de la serie temporal *y*. Se extrae la componente estacional.
- Simple Exponential Smoothing (SES). Se aplica directamente sobre los datos. Penaliza los periodos de cero demanda o demanda intermitente. Se aplica pesos ponderados que da mayor peso a los datos recientes, con respecto a los más antiguos. No detecta la tendencia ni estacionalidad significativa.

- Medias móviles (MA). Las predicciones son calculadas como la media de las últimas k observaciones donde k es el ciclo de la serie temporal *y*.
- Método de Croston (CRO). Se propone predecir la demanda separando la serie temporal en dos componentes; el tamaño de la demanda no nula, y los intervalos de Inter demanda. Ambos componentes son predichos por SES con un parámetro de 0,1 y valor inicial igual a la primera observación. Se considera el método estándar para predecir la demanda intermitente.

Métodos machine learning.

Los modelos fueron entrenados con el mismo conjunto de datos utilizado en los métodos estadísticos anteriores. Se fijan una primera capa inicial de entrada con número de nodos múltiplo de 6, correspondiente a los días de la semana, de lunes a sábado

El nodo de salida es único correspondiente a la predicción de demanda en un modelo supervisado de regresión. La precisión del modelo se valida usando el error cuadrático medio escalado (RMSSE). Los datos se normalizan en una escala entre 0 y 1.

- Perceptrón multi-capa (Multi-Layer Perceptron, MLP).
    - Red neuronal usando el paquete de R, RSNNS.
    - Se entrena con hasta 3 capas.
    - Los pesos iniciales se fijan aleatoriamente con la función learnFunc.
    - Número de interacciones entre 100 y 1.000.
    - La función de activación en las capas ocultas (hiddenActFunc) es logística.
    - La función de activación en la capa de salida (linOut) puede ser logística o lineal.
    - Se utiliza la mediana como valor representativo de las predicciones.
- Red neuronal bayesiana (Bayesian Neural Network, BNN).
    - Similar al modelo MLP.
    - Se optimizan los pesos iniciales según el concepto bayesiano
    - Paquete en R, BRN
    - Valor de parámetro μ, se selecciona entre 0,001 0,01 y 0,1.
- Random Forest (RF). Es una combinación de los modelos de árbol de decisión con regresión (Regression Tree, RF). La precisión del modelo depende del tamaño de las ramas, y la correlación de los árboles.
    - Paquete en R, randomForest.
    - El número de ramas (ntree) se selecciona entre 100, 250, 500 y 1.000
    - El tamaño mínimo de nodo (nodosize) se selecciona entre 5, 10, 100 y 500.
    - El número de variables muestreadas para cada clasificación (mtry) se elige entre x/2, x/3, x/5 y x/10.
    - El muestreo Bootstrap se realiza con reemplazo.

- Árbol de gradiente creciente (Gradient Boosting Trees, GBT). Modelo similar al RF, pero sólo construye un árbol. Es más propenso a caer en el overfitting, especialmente cuando los datos tienen ruido.
    - Usar el paquete de R, gbn.
    - Ratio de aprendizaje (shrinkage) entre 0,001, 0,01 y 0,1
    - Profundidad entre 1, 2, 4, 8 y 16.
    - Número total de árboles (n.trees) entre 100, 250, 500 y 1.000
    - Modelo de ajuste (distribution) Gaussian o Laplace.
- Regresión con vecinos (K-Nearest Neighbor Regression, KNNR). Distancia euclídea entre los puntos utilizados en el entrenamiento y el test.
    - K puntos cercanos, elegido entre 3 y 99.
    - Implantado paquete R, caret.
- Vector soporte regresión (Support Vector Regression, SVR). Genera predicciones identificados por el hiperplano que maximiza la distancia entre dos clases, y minimiza la distancia entre las observaciones de una misma clase.
    - Paquete en R, e1071
    - Núcleo usado (kernel) será entre lineal, polinomial, radial y sigmoid.
    - La tolerancia (tolerance) entre 0,001, 0,01 y 0,1.
    - El valor del parámetro μ entre 0,3 y 0,7.
- Proceso Gaussiano (Gaussian Processes, GP). El modelo asocia la variable dependiente con múltiples independientes con distribución normal aleatoria..
    - Paquete en R, kernlab
    - Las funciones kernel (kernel) entre radial basic, polinomial, lineal, hiperbólica tangente, Laplacian, Bessel y Anova radial basic.
    - La varianza del ruido (var) entre 0,001, 0,01 y 0,1.

Los 4 métodos con mayor precisión en términos de error cuadrático medio han sido (GBT, RF, SVR y KNNR). Las redes neuronales son superados por los modelos estadísticos. Estos resultados se obtienen para unos datos que sólo abarcan un espacio temporal de un año, con ausencia de estacionalidad, y sin incorporar otros datos externos que amplían la dimensionalidad del conjunto de datos.

El algoritmo de regresión de vectores de soporte (SVR) se ha demostrado eficaz en la predicción de demanda en la venta de viajes minorista. Se consiguen mejores resultados en términos de rendimiento predictivo en comparación con modelos ARIMA (Karmy, J.P. & Maldonado, S., 2019).

Existen diversos estudios sobre predicción con SVR entre los que destacamos (Levis, A.A., & Papageorgiou, L.G., 2020) que realizó predicciones de demanda con un algoritmo de tres pasos con regresores lineales y (Lu, Chi-Jie, 2014) que usó SVR para predecir ventas con regresión multivariante.

2.2. Fuentes internas y externas.

Después de la lectura del apartado anterior, se llega a la conclusión que los algoritmos basados en ML necesitan bases de datos de mayor tamaño, para mejorar en precisión a los modelos tradicionales.

Se dispone de fuentes de origen interno (histórico de ventas de un producto, punto de venta, fecha, promociones, etc.) que se complementan con fuentes de origen externo (festividad, climatología, demografía, socioeconómicos, etc.).

2.2.1. Fuentes internas

En el conjunto de datos, se incluyen dos características; coeficiente de variación de la demanda no nula (CV2), y la media del número de periodos de tiempo entre dos demandas sucesivas no zero (ADI), propuesto por (Syntetos, A. A., & Boylan, J. E., 2005).

Cuando se quiere predecir demanda en días especiales, uno de los principales problemas es el tamaño del data set de ventas, porque no se dispone de una serie histórica amplia. Se introducen nuevas variables para ampliar el data set:

- Mediana móvil del día de la semana
- la variación absoluta y relativa de la demanda en días especiales con respecto a la mediana móvil de cada día de la semana.
- Variación relativa de la demanda con respecto de la mediana móvil de cada día de la semana, para tiendas de una misma clase o tipología.

2.2.2. Fuentes externas.

Se hace un especial hincapié en el tratamiento de cierta información incorporada al fichero desde fuentes externas. La predicción en días festivos ha sido motivo de estudio (Huber, J. & Stuckenschmidt, H., 2020). Los días festivos no cumplen con un patrón habitual de ventas, que permita una predicción eficaz. Los modelos que mejor precisión demuestran son modelos de ML supervisado (redes neuronales y árbol decisión de gradiente positivo).

(Ramanathan, U. & Muyldermans, L., 2010) definen en su trabajo cómo puede afectar a la demanda, las acciones promocionales de ventas. Otros aspectos como los festivos nacionales ha sido objeto de estudio por (Soares, L. J., & Medeiros, M. C., 2008) que identifica en su trabajo un total de 15 tipos de día, incluyendo los fines de semana, días anteriores y posteriores a un festivo nacional, puentes festivos, etc.

(Taylor, J. W., 2007) introduce un método corrector para suavizar la curva de demanda en estos días, con respecto al resto de días de venta, que consiste en una transformación exponencial de la variable objetivo.

Se desarrolla modelo S-ARIMAX, donde la variable festividad se incluye como variable ficticia binaria que sólo presenta valor 1, ó 0 según sea día festivo o no (Arunra, N. S. & Ahrens, D., 2015). Se entiende por días especiales, aquellos en los que la demanda es ampliamente superior a la de días regulares. Los días especiales son días que caen con frecuencia en días laborales, aunque también puede coincidir con domingos. En el estudio se consideran días especiales los días festivos y los días colindantes (anteriores y posteriores) y la semana siguiente.

2.2.3. Otras fuentes en acciones promocionales

Existen factores particulares en la predicción de la demanda bajo acciones promocionales (Fildes, R. et al, 2022) que deben tenerse en cuenta. Esta evidencia lleva a la necesidad de completar el data set con otras fuentes de datos o transformaciones. Veamos a continuación qué factores se refiere.

Efecto látigo

Existe evidencia que en algunas categorías de producto, la demanda depende del volumen de existencias, en una relación directa. A mayor volumen de existencias, mayor interés en acciones para promocionar su venta y en consecuencia, mayor volumen de ventas. Es lo que se conoce como efecto látigo (Koschat, M. A., 2008).

Intermitencia

Otra característica de la demanda es la intermitencia que consiste en periodos de baja demanda o de falta de suministro. Es vital su estudio, especialmente en retail más que en industria. Para estudio de la intermitencia, las técnicas utilizadas son el método Croston (Croston, J. D., 1972), el boot-strap (Willemain, T. R., Smart, C. N., & Schwarz, H. F., 2004), el método de aproximación Syntetos-Boylan (SBA) (Syntetos, A. A., & Boylan, J. E., 2005) y el método TSB (Teunter, R. H. et al, 2011). El método Croston predice la demanda bajo la suposición de que la variable objetivo sigue una distribución binomial negativa (Kolassa, S., 2016). Las predicciones bajo el modelo de Croston resultan de gran utilidad por su buena precisión (Shenstone, L., & Hyndman, R. J., 2005).

Otra alternativa para corregir la intermitencia, es el uso de series temporales agregadas (MAPA) mejorado con suavizado exponencial en las variables explicativas (Kourentzes, N. et al, 2014).

Estacionalidad

La venta de producto minorista tiene una fuerte componente de estacionalidad en su evolución. Se identifican ciclos en el patrón de venta semanal y tendencia general a lo largo del periodo de un año. Un buen algoritmo de predicción debe contemplar esta casuística y detectar los patrones de estacionalidad en su modelo (Huang, T. et al, 2019).

Festividades y fechas especiales.

La predicción de la demanda puede verse afectado por otros aspectos como festividades, periodos vacacionales, etc. También se da esta situación en fechas especiales como puede ser competiciones deportivas de gran interés, y otros eventos de interés general. Algunos días festivos se repiten a intervalos regulares y pueden modelarse como estacionales con relativa facilidad (vacaciones estivales, navidades, domingos, etc.). Otros eventos no tiene un patrón de regularidad y resulta más difícil detectar estacionalidad. Es el caso de semana santa, competiciones deportivas como Olimpiadas, partidos relevantes de fútbol, carnavales, etc. Para estos casos, se recomienda construir previamente variables dummies (Cooper, L. G. et al, 1999) para incorporar al data set.

Climatología.

La temperatura o condiciones climatológicas pueden ser factores influyentes en el comportamiento de la demanda (Dubé, J. P., 2004). Se introduce la dificultad, que las condiciones meteorológicas son impredecibles, y requieren de predicción y pronóstico particulares. Por tanto, se introduce una componente de incertidumbre por el error debido a la predicción de climatología a una fecha o intervalo determinado.

Marketing Mix y promociones.

Los descuentos y acciones promocionales, deben tenerse en consideración a la hora de predecir la demanda a corto plazo, puesto que son factores que influyen en la misma. En especial, el uso y tipo de anuncios son también una componente promocional que se debe tomar en consideración (Gijsbrechts, E. et al, 2003).

Reseñas en redes sociales e internet.

Las reseñas en redes sociales o internet, también es una vía de anunciar un producto y publicitarlo. La frecuencia en que se cita un producto en internet, influye directamente en la demanda de dicho producto. Una forma de conocer el impacto de una campaña promocional en redes sociales, es mediante técnicas de análisis de sentimientos (Zhang, W. et al, 2018).

2.3. Modelo de regresión o de clasificación.

En los puntos anteriores se ha debatido sobre la conveniencia entre la utilización de modelos estadísticos a métodos de ML. Posteriormente se ha completado con la necesidad de ampliar la dimensionalidad del data set con información procedente de

fuentes externas. Llegados a este punto, se quiere aplicar un modelo supervisado que puede ser de regresión o de clasificación.,

Se aplican diferentes modelos de ML supervisado; NN y árbol decisión de gradiente positivo (XGBoost). Se consiguen mejores resultados con los modelos basados en clasificación, con respecto a los basados en regresión (Huber, J. & Stuckenschmidt, H., 2020). La aplicación de un modelo de clasificación, tiene ventajas. El enfoque como problema de regresión predice sólo un valor de tipo cuantitativo, mientras que en un problema de clasificación se predicen n clases (variable cualitativa). El modelo de clasificación define una función de distribución de probabilidad para todas las clases (discreto), y clasifica cada observación según sea mayor su probabilidad de pertenencia a una de las n clases posibles.

2.4. Principales algoritmos de predicción en Machine Learning.

Se tratan a continuación el procedimiento y metodología en los tres algoritmos seleccionados como idóneos para este trabajo.

2.4.1. Red neuronal (NN)

El algoritmo NN se compone de tres fases. En la primera de entrenamiento se configuran los hiperparámetros del modelo. Le sigue la fase de test donde se valida y obtiene la precisión del modelo. Por último, se añade una tercera fase de reentrenamiento, en la que volvemos a la fase uno con nuevas observaciones. Hay que fijar adecuadamente esta fase, para no incurrir en un sobrecoste computacional.

Previo a la implementación de la NN, los datos hay que escalarlos y estandarizarlos. Para la estacionalidad, pueden modelarse con funciones trigonométricas (Crone, S. F., Hibon, M., & Nikolopoulos, K., 2011). Estandarizar en un rango entre [-0,5, 0,5] es beneficioso para la backpropagation según (LeCun, Y. A., Bottou, L., Orr, G. B., & Müller, K. R., 2012).

Funciones de activación utilizadas en las capas ocultas;

Ilustración 1 Función activación capas ocultas red neuronal. Fuente; (Huber, J. & Stuckenschmidt, H., 2020).

- rectified linear activation: $\sigma_{relu}(x) = x^+ = \max(0, x)$
- exponential linear activation:
$$\sigma_{elu}(x) = \begin{cases} x & \text{if } x \geq 0 \\ e^x - 1 & \text{otherwise} \end{cases}$$

Función de activación en la última capa, en problema de regresión (linear) y en clasificación (softmax).

Ilustración 2 Función activación en última capa red neuronal. Fuente; (Huber, J. & Stuckenschmidt, H., 2020).

- a linear function: $\sigma_{linear}(x) = x$
- a softmax function: $\sigma_{softmax}(x) = [\frac{exp(x_1)}{\sum_c exp(x_c)} \cdots \frac{exp(x_C)}{\sum_c exp(x_c)}]$

Para optimizar los pesos en cada nodo, se utiliza el algoritmo gradiente estocástico ADAM (Kingma, D. P. & Ba, J. L., 2015).

2.4.2. Red neuronal recurrente (RNN).

Seguimos con el mismo procedimiento de tres fases descrito en el apartado anterior. En este caso, el algoritmo aplicado es RNN que procesa la información de entrada en un orden secuencial y aplica la misma red a cada capa de una misa secuencia (Huber, J. & Stuckenschmidt, H., 2020). Se utiliza la variante conocida como memoria a largo plazo (LSTM) (Hochreiter, S., & Schmidhuber, J., 1997) que se basa en puertas de entrada i_t, puertas de salida o_t, puertas de olvido f_t, dato en nodo intermedio c_t y dato en nodo de salida h_t.

Ilustración 3 Elementos en RNN, variante LSTM. Fuente; (Huber, J. & Stuckenschmidt, H., 2020)

$$f_t = \sigma_{sigmoid}(W_f x_t + U_f h_{t-1} + b_f)$$
$$i_t = \sigma_{sigmoid}(W_i x_t + U_i h_{t-1} + b_i)$$
$$o_t = \sigma_{sigmoid}(W_o x_t + U_o h_{t-1} + b_o)$$
$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_{tanh}(W_c x_t + U_c h_{t-1} + b_c)$$
$$h_t = o_t \circ \sigma_{tanh}(c_t).$$

Los parámetros son entrenados con un algoritmo gradiente estocástico ADAM.

2.4.3. Algoritmo de regresión de vectores de soporte (SVR).

El algoritmo de regresión de vectores de soporte (SVR) se ha demostrado eficaz en la predicción de demanda en la venta de viajes minorista. Se consiguen mejores resultados en términos de rendimiento predictivo en comparación con modelos ARIMA (Karmy, J.P. & Maldonado, S., 2019).

Existen diversos estudios sobre predicción con SVR entre los que destacamos (Levis, A.A., & Papageorgiou, L.G., 2020) que consiguió un modelo para predicción de la demanda basado en regresión lineal con buenos resultados de precisión y (Lu, Chi-Jie, 2014) que usó SVR para predecir ventas con regresión multivariante.

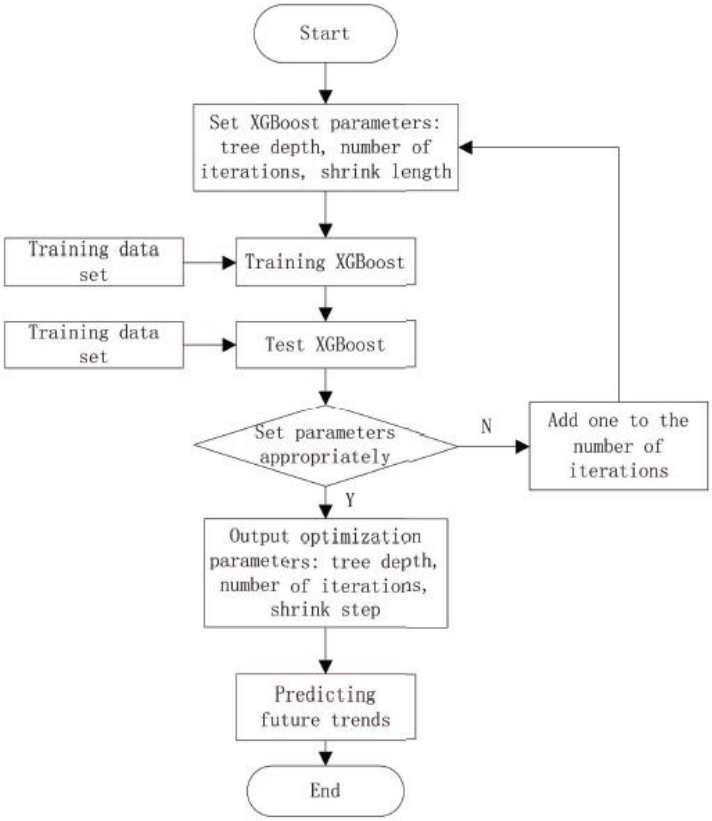
2.4.4. Árbol decisión gradiente creciente (BGRT).

Algoritmo con gran popularidad en los últimos años. Existen dos versiones, LightGBM (Ke, G. et al, 2017) y XGBoost (Chen, T., & Guestrin, C., 2016) .

De los dos algoritmos, hoy en día, XGBoost es el algoritmo de ML más usado por su sencillez y resultados. Es un algoritmo predictivo supervisado que utiliza el principio de boosting, que consiste en generar múltiples modelos de predicción sencillos, de manera secuencial, de modo que para cada modelo utiliza los resultados del modelo anterior. De este modo se llega a generar un modelo más fuerte con mejor poder predictivo y mayor estabilidad en sus resultados.

El criterio que utiliza este algoritmo para evolucionar en su transición secuencial de uno a otro modelo, es encontrar el mínimo de una función objetivo que puede ser la probabilidad de error en la clasificación, el área bajo la curva (AUC) o la raíz del error cuadrático medio (RMSE). Cada modelo es comparado con el anterior. Si con el nuevo modelo no se obtienen mejores resultados, se descarta y se vuelve de nuevo al modelo anterior, ajustando algunos parámetros. Si por el contrario, sí se obtiene mejores resultados con el nuevo modelo, se utiliza este como base, a partir del que continúa el proceso secuencial.

Ilustración 4 Proceso de predicción XGBoost. Fuente: (Xiaoqun, L. et al, 2019).



Este proceso continúa hasta llegar a un punto en el que la mejora entre dos modelo consecutivos es insignificante. Sobre su implementación y base teórica, puede consultarse (Xiaoqun, L. et al, 2019).

Es preferible utilizar XGBoost para data sets grandes, con más de 1.000 observaciones. También es apropiado cuando se mezclan variables categóricas y numéricas, o sólo numéricas. Es ideal en problemas supervisados de clasificación y regresión.

3. Diseño e implementación

La lectura de estudios realizados por investigadores, expertos y científicos, sobre esta materia, permite establecer una bases que es el punto de partida para este trabajo fin de máster (TFM). Como se verá a continuación, se dispone de un fichero de datos suficientemente grande, y amplio en dimensionalidad, para el que se aconseja el diseño de un modelo de predicción de la demanda, a través de algoritmos ML (Adnan Khan, M. et al., 2020).

Este apartado se estructura con una primera presentación del caso de estudio y descripción de las variables del fichero. Se sigue con la identificación de fechas clave por tratarse de festividad, periodo vacacional, promoción, etc. Se crean variables booleanas a partir de las fechas anteriores, de modo que 1 indica fecha festiva y 0 si no lo es. En el tercer apartado se realiza un análisis descriptivo del fichero, de componentes temporales de la serie y correlación entre variables. Con la información obtenida se propone una reducción de dimensionalidad.

Sobre el fichero depurado, se implementan diferentes algoritmos de ML. Como cita (Huber, J. & Stuckenschmidt, H., 2020), los algoritmos de supervisado para clasificación ofrecen mejores predicciones que los de regresión. Siguiendo esta recomendación, se obtienen modelos predicción de ventas futuras, aplicando algoritmo SVM, Árbol de decisión (XGBTO) y redes neuronales. Se selecciona aquel modelo que minore el coste de stock con respecto al modelo de cálculo de stock implantado por la compañía.

3.1. Presentación del caso de estudio.

Se dispone de un fichero de ventas, correspondiente a un caso real de distribución de retail a distintitos puntos de ventas. La información que se proporciona en los ficheros, es de 50 combinaciones producto-punto de venta (sku). Está anonizada para preservar el anonimato de modo que no se conoce la ubicación ni producto de cada sku.

Significado de las siglas SKU: Stock Keeping Unit o unidad de mantenimiento de existencias. Cada empresa define este término en base a características intrínsecas del producto, sus objetivos empresariales y necesidades de sus consumidores. En el caso concreto de este estudio, la empresa ha definido un SKU que combina producto y punto de venta, de modo que un mismo producto ubicado en dos tiendas diferentes tendrán un SKU distinto.

El objetivo del trabajo es diseñar un modelo de predicción de la demanda que permita una mejor estimación del stock de seguridad. Se compara el coste de stock de seguridad entre el modelo de previsión de la empresa, y el modelo de previsión que se obtiene a partir de ML en este trabajo. El criterio para seleccionar entre el modelo que utiliza la

empresa para calcular el stock, y el previsto según el modelo, será aquel que incurre en menor coste de stock, en términos monetarios.

- Coste stock de un día en stock = % Coste unitario * valor de las unidades en stock = 5% * (precio * Unidades en stock). El coste de stock es la suma del coste de almacenaje y el coste de oportunidad de inversión. Coste de almacenaje corresponde al coste del espacio, mantenimiento y operarios, para almacenaje del stock. El coste de oportunidad de inversión es el coste financiero asociado al lucro cesante (lo que se deja de ganar) por invertir capital en existencias, y no destinarlo a otras inversiones que reportarían beneficio. El importe de estos costes, el de almacenaje y el de oportunidad, son de difícil concreción. En su lugar, se estima con el 5% de su valor (precio medio * unidades en stock).

El stock en el periodo de estudio, es conocido y contenido en la hoja 04_Stock del fichero Datos7.

El stock de seguridad es la cantidad de inventario que se almacena como reserva en la instalación. Con este inventario, la compañía puede hacer frente a imprevistos como aumentos en la demanda, cambios en la rotación de un sku o retrasos con los proveedores. Se calcula el stock de seguridad en el modelo de predicción y para las ventas reales. Para el cálculo de stock de seguridad en las ventas previstas según el modelo, se realiza el siguiente cálculo:

- Stock de seguridad previsto = Factor servicio * RMSE * raíz del ciclo de aprovisionamiento. El factor servicio es el nivel de servicio deseado, obtenido como el valor para el que la probabilidad de satisfacer la demanda es de al menos el 95%. Suponiendo que la demanda se distribuye según una normal tipificada, para una probabilidad del 95% se corresponde un factor servicio = 1,64. RMSE (Root Mean Squared Error) o error cuadrático medio es la raíz cuadrada de la media del cuadrado de las diferencias entre el valor real y el valor estimado. $RMSE = \sqrt{(\Sigma \frac{(x_i - y_i)^2}{n}}$). El valor de ciclo de aprovisionamiento se facilitan en el fichero DatosCicloAprovisionamiento, variable *leadtime*.

Para el cálculo de stock de seguridad sobre ventas reales, se realiza el siguiente cálculo:

- Stock de seguridad real = (plazo máximo de entrega – plazo de entrega habitual) * demanda media del producto. Los valores de plazo máximo de entrega y plazo de entrega habitual, se pueden obtener del fichero DatosCicloAprovisionamiento, bajo el nombre de las variables *diasEntrePedidos* y *diasLeadtime*, respectivamente. La demanda media se calcula para cada sku, en el periodo de estudio.

El principal objetivo del stock de seguridad es evitar que se produzcan roturas de stock, es decir, se acepten pedidos que por volumen de existencias no pueden ser atendidos.

3.2. Identificación de los ficheros y variables.

Se dispone de tres ficheros de datos con la siguiente información:

Tabla 2 Resumen data set y variables. Fuente: elaboración propia

Nombre	Hoja	Dimensión	Variables	Fecha inicio	Fecha fin
Datos7.xls	01_Ventas	(34151, 3)	sku, fecha, udsVenta	14/03/2020	14/03/2022
	02_Calendarios	(58250, 4)	sku, fecha, bolOpen, bolHoliday	15/03/2020	23/05/2023
	03_Promociones	(2128, 3)	sku, fecha ini, fecha fin	04/01/2011	20/04/2022
	04_Stock	(36550, 3)	sku, fecha, udsStock	15/03/2020	15/03/2022
diasEntrePedidos		(50, 3)	sku, diasEntrePedidos, diasLeadtime		
DatosPrecioMedio		(50, 2)	idSkulta, eurPrecioMedio		

El fichero Datos7 en Excel, contiene 4 hojas:

- 01_Ventas:
  Histórico de ventas de los 50 sku, por fecha.

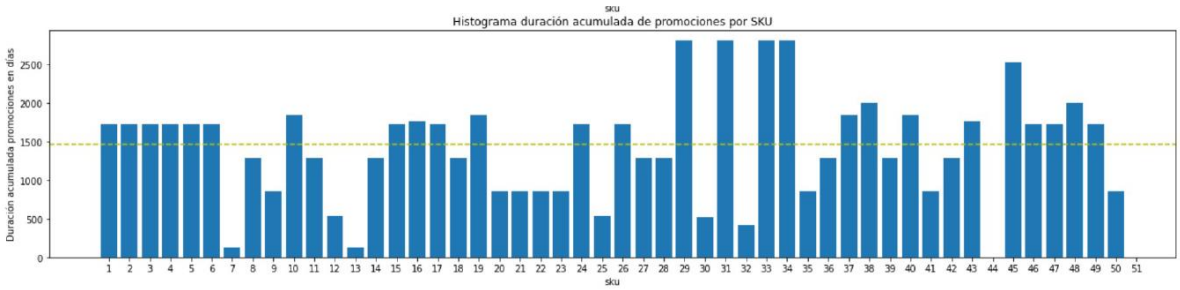
- 02_Calendarios:
  Contiene dos variables booleanas que toma valor 1 si la fecha es festivo (bolHoliday) y si el establecimiento está abierto (bolOpen). Existen 1165 observaciones por cada sku, que corresponde con el número de días entre el 15/03/2020 y el 23/05/2023, por tanto la serie es completa. Véase en la siguiente tabla, el agrupamiento de fecha/sku según las categorías de las variables bolOpen y bolHoliday.

Tabla 3 Agrupamiento por categorías de bolOpen y bolHoliday

Festivo	0	1	subtotal
Abierto			
0	604	4753	5357
1	51780	1113	52893
subtotal	52384	5866	58250

- 03_Promociones:
  Contiene las fechas de inicio y fin de las campañas promocionales.
  En un primer análisis, interesa conocer el número de promociones, expresado en días de duración, para cada uno de los sku. La siguiente tabla, puede visualizarse la distribución de las promociones, comparada con la duración media.

Ilustración 5 Duración en días de las promociones por cada sku. Fuente: elaboración propia.



La variable sku=44 no tiene ninguna promoción asignada. Puede afirmarse que las promociones son de una duración similar. Existen 6 valores de sku (7, 12, 13, 25, 30 y 32) cuya duración es muy inferior al resto. Así mismo, los valores de sku (29, 31, 33, 34 y 45) han tenido mayor días de promoción que el resto.

- 04_Stock:
  Es de utilidad para explicar ventas a cero en el pasado por rotura de stock. En estos casos conviene reconstruir las ventas para que los modelos no aprendan de estos periodos excepcionales de ventas a cero debido a roturas.

Los ficheros DatosCicloAprovisionamiento y DatosPrecioMedio sirven para estimar el stock de seguridad final y contienen la siguiente información;

El fichero DatosCicloAprovisionamiento contiene información relativa al stock

- diasEntrePedidos, es el periodo máximo que puede transcurrir para entregar un pedido.
- diasLeadtime, es el plazo de entrega habitual para entregar un pedido.
  En los siguientes histogramas puede visualizarse la distribución de estas variables por cada sku para un primer análisis visual.

Ilustración 6 Histograma días entre pedidos por sku. Fuente: elaboración propia.

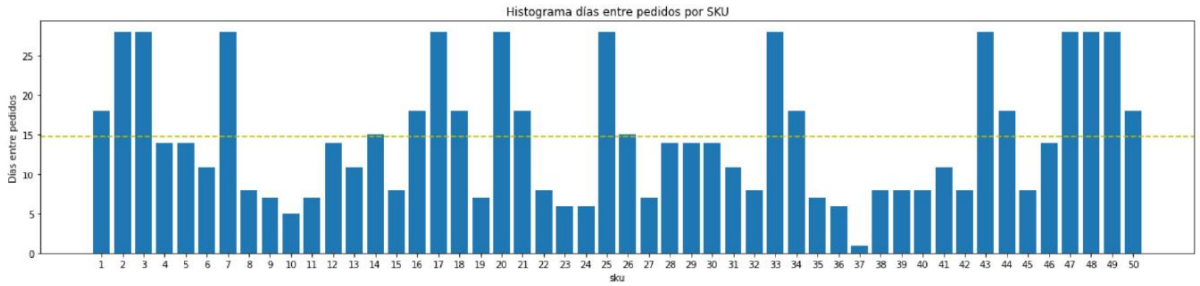


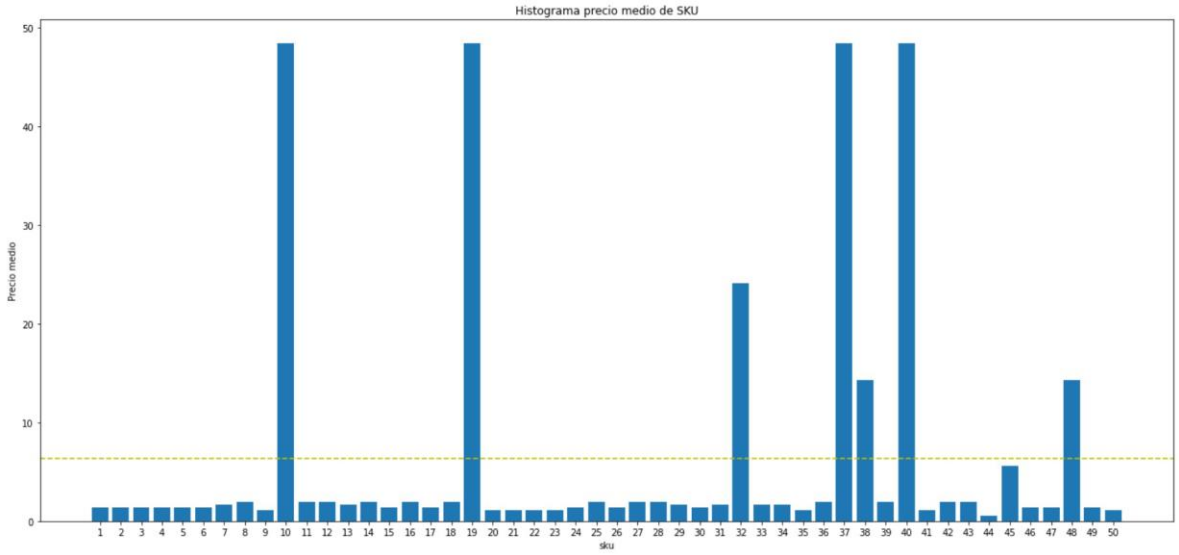
Ilustración 7 Histograma tiempo espera recibir pedido por sku

Se observa que el tiempo de espera habitual es muy similar, con tres niveles o categorías (2, 4 y 6). Los que mayor retraso tienen son los sku (10 y 37).
Destacar que precisamente los sku con mayor plazo de entrega, son los que menor plazo transcurre entre pedido, por tanto se puede intuir que para estos sku, el stock está mal dimensionado para su demanda.

El fichero DatosPrecioMedio contiene una única variable para cada sku.

- eurPrecioMedio, es el precio medio para cada sku.
  Existen 3 niveles de precio. [0.5 – 6). [6 – 25) y [25 – 50]. Véase en el siguiente histograma la distribución de casa sku, por su precio medio.

Ilustración 8 Histograma precio medio por sku. Fuente: elaboración propia.



Destacar que los sku con mayor retraso de entrega y menor espacio entre pedido (sku 10 y 37) tienen un precio medio en el tramo alto.

3.3. Limpieza y preparación del fichero.

En una primera fase, se procede a la limpieza y preparación del fichero venta y basic. El fichero venta será el data set para este trabajo. El fichero basic, contendrá información para el cálculo del stock de seguridad.

3.3.1. Unión de ficheros.

Se dispone de diferentes ficheros que deben unirse para conformar un único fichero que será el data set sobre el que se trabajará. Se toma como fichero principal el Datos7.xls, y dentro de éste, la hoja de ventas, sobre el que se irán incorporando las variables del resto de hojas, tomando como variable de unión, la variable sku.

- 01_Ventas + 02_Calendarios = venta
  Unión en modalidad 'inner' para que las variables sku y fecha coincidan en ambos ficheros. Se pierde información del 14/03/2020 por no estar en calendarios.
- venta + 03_Promociones = venta
  Se crea en promociones una nueva variable llamada *durac_promo* con el número de días de duración para cada promoción.
  Se traslada la información a venta, con la creación de una variables booleana llamad *bolPromo* que tendrá valor 1 si para cada sku y fecha, se encuentra en periodo de promoción. Véase la tabla con agrupamiento por fechas que existen promoción.

Tabla 4 Tabla agrupamiento por variable bolPromo. Fuente: elaboración propia

bolPromo	sku	fecha	udsVenta	bolOpen	bolHoliday	date
0	17965	17965	17965	17965	17965	17965
1	16173	16173	16173	16173	16173	16173

- venta + 04_Stock = venta
  Unión de los ficheros por la izquierda, dando prioridad al fichero venta.

Se crea la variable *date* al cambiar el tipo de dato de la variable *fecha*, a tipo datetime. Se graba este primer fichero, indexado por la variable *date.*

Los ficheros DatosCicloAprovisionamiento y DatosPrecioMedio se unen y dan lugar a un segundo fichero llamado basic, con información necesaria para calcular el stock de seguridad para cada sku, en función de la predicción de su demanda. Por tanto, y a modo de resumen, se ha unificado toda la información en dos ficheros.

Tabla 5 Descripción de los ficheros data set. Fuente: elaboración propia.

Nombre	Dimensión	Variables	Fecha inicio	Fecha fin
venta	(34138, 3)	sku, fecha, udsVenta, bolOpen, bolHoliday, bolPromo, durac_promo, udsStock	15/03/2020	14/03/2022
basic	(50, 2)	sku, diasEntrePedidos, diasLeadtime, eurPrecioMedio		

3.3.2. Estadísticos descriptivos.

Son variables cuantitativas (*udsVenta, durac_promo y udsStock*), categóricas (*sku*), booleanas (*bolOpen, bolHoliday y bolPromo*) y fecha es tipo numérica, pero debe transformarse a datetime.

Véase la tabla con los principales estadísticos descriptivos.

Tabla 6 Estadísticos descriptivos del fichero venta. Fuente: elaboración propia

	sku	fecha	udsVenta	bolOpen	bolHoliday	bolPromo	durac_promo	udsStock
count	34138.000000	3.413800e+04	34138.000000	34138.000000	34138.000000	34138.000000	34138.000000	34138.000000
mean	25.547689	2.020817e+07	13.635421	0.860361	0.165886	0.473754	14.034683	311.596989
std	14.480658	6.166929e+03	16.451301	0.346617	0.371983	0.499318	15.685832	269.632981
min	1.000000	2.020032e+07	-91.000000	0.000000	0.000000	0.000000	0.000000	-147.000000
25%	13.000000	2.020102e+07	0.000000	1.000000	0.000000	0.000000	0.000000	168.000000
50%	26.000000	2.021041e+07	7.000000	1.000000	0.000000	0.000000	0.000000	238.000000
75%	38.000000	2.021092e+07	21.000000	1.000000	0.000000	1.000000	27.000000	350.000000
max	50.000000	2.022031e+07	273.000000	1.000000	1.000000	1.000000	52.000000	2310.000000

3.3.3. Tratamiento outliers, nulos y anómalos.

Se considera outliers, a efectos de este estudio, aquellas observaciones que quedan fuera del rango [ẋ ± 3*σ]. Se calcula para las variables cuantitativas *udsVenta, durac_promo y udsStock.* Se identifican 667 observaciones con valor en *udsVenta* fuera del rango, y 992 observaciones con valor en *udsStock* fuera del rango. Ha de comprobarse si estos outliers pueden deberse a un error en la toma de datos, o están relacionados con el comportamiento de otra variable.

Al representar gráficamente en diagrama de puntos, los outliers frente a la variable *fecha*, se observa que los outliers están localizados en tres periodos concretos. Se considera que

estos outliers no deben eliminarse del dataset, por la información que aportan en el comportamiento de ventas y stock, en fechas concretas.

Ilustración 9 Diagrama puntos outlier udsVenta frente a fecha. Fuente; elaboración propia.

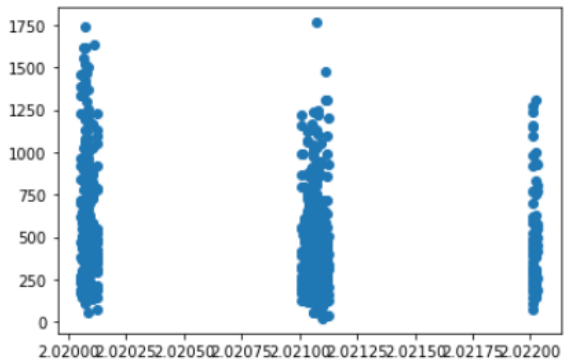
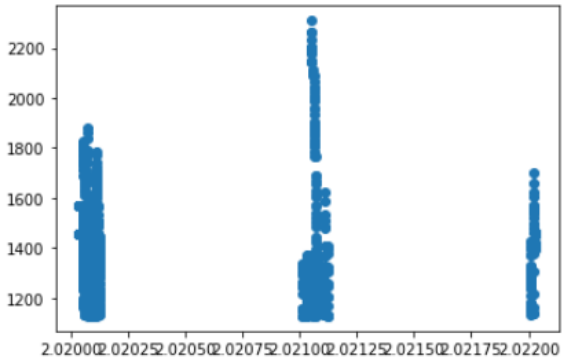


Ilustración 10 Diagrama puntos outlier udsStock frente a fecha. Fuente; elaboración propia.



Valores nulos o ausentes son aquellos con valor NaN o vacíos. No existen valores nulos a ausentes en el fichero venta.

Con respecto a valores anómalos, se considera anómalo un valor negativo en las variables *udsVenta* y *udsStock*, dado que estas variables sólo pueden tomar valor nulo o positivo. Existen 8 observaciones con *udsVenta* <0 y 190 observaciones con *udsStock* <0.

Todos los sku con valor negativo en *udsVenta*, corresponde con fecha de apertura del centro correspondiente y no es festivo. Se considera que ha existido rotura de stock y no se ha vendido nada. Por tanto, debe considerarse valor nulo en udsVenta y reemplazar el valor negativo por el valor 0.

Tabla 7 Observaciones con valor udsVenta <0. Fuente: elaboración propia.

	sku	fecha	udsVenta	bolOpen	bolHoliday	bolPromo	durac_promo	udsStock
7321	11	20211002	-7	1	0	0	0	-7
14326	21	20220114	-91	1	0	1	41	336
15582	23	20211103	-7	1	0	0	0	161
18135	27	20210723	-7	1	0	0	0	154
18227	27	20211023	-35	1	0	1	27	-35
23699	35	20210921	-70	1	0	0	0	140
25957	39	20200724	-7	1	0	0	0	231
26366	39	20210906	-7	1	0	1	21	21

En cuanto a las observaciones con valor negativo en la variable *udsStock* se considera que ha existido rotura de stock, y deben eliminarse del data set para que los modelos futuros, no aprendan de estos periodos excepcionales. Se eliminan por tanto 190 observaciones del fichero venta.

3.3.4. Transformación de variables.

En el fichero venta, la variable fecha que es tipo enero, debe transformase a tipo datetime, que se llamará *date*. Aprovechando la transformación se crean tres nuevas variables (*year, month y day*) que corresponde a la extracción de este dato, de la variable date.

3.3.4.1. Creación de variables dummies.

Siguiendo con la transformación, se crean dos nuevas variables que hacen referencia a la semana del año, y día de la semana (*week_year y week_day*). Se considera que podría aportar información adicional en caso que se detectara comportamiento distinto de las ventas, según semana del mes o día de la semana.

Se crean variables dummies a partir de las variables *year, month* y *week_day*. Con esto se pretende que a la hora de aplicar el algoritmo de ML, trate por igual estas variables, indistintamente de su valor. Es decir, no es mejor un mes 10 de un mes 2 por el hecho que su numeración sea mayor. Igual ocurre con el día de la semana y el año.

3.3.4.2. Creación variables festividad, vacaciones y otras.

Siguiendo la recomendación de (Arunra, N. S. & Ahrens, D., 2015), se crean nuevas variables categóricas en el fichero ventas para los siguientes festivos y fechas clave en el periodo comprendido entre 2020 y 2022. Estas variables son de tipo booleano, de modo que toma el valor 1 en las fechas indicadas y 0 en el resto.

Véase a continuación tabla con las variables creadas, grupo de pertenencia, nomenclatura y rango de fechas.

Tabla 8 Variables booleanas por festividades. Fuente: Elaboración propia.

Grupo	Subgrupo	variable	rango de fechas ?? (20, 21, 22)
Festividades	Semana santa	bol_ssanta??	[05/04/2020 al 12/04/2020], [28/03/2021 al 04/04/2021]
	Navidades	bol_nadal??	del 25 al 31 de diciembre
	Reyes Magos	bol_reyes??	del 1 al 6 de enero
	Año nuevo	bol_new_year	1 de enero
	Reyes Magos	bol_reyes	6 de enero
	viernes santo	bol_good_Friday	10/04/2020, 02/04/2021
	día del trabajador	bol_labor_day	1 de mayo
	hispanidad	bol_hispanic_day	12 de octubre
	todos los santos	bol_saints_day	1 de noviembre
	día de la constitución	bol_constitution_day	6 de diciembre
	Inmaculada	bol_virgin_day	8 de diciembre
	Navidad	bol_nadal_day	25 de diciembre
Otras fechas	Domingos	bol_Sunday	Día 7 de week_day
	Carnaval	bol_carnaval??	[12/02/2021 al 14/02/2021], [25/02/2022 al 27/02/2022]
	Halloween	bol_halloween	[31 de octubre]
	San Valentín	bol_valent	14 de febrero
vacaciones es	1ª quincena julio	bol_summer1q??	1 al 15 de julio
	2ª quincena julio	bol_summer2q??	16 al 31 de julio
	1ª quincena agosto	bol_summer3q??	1 al 15 de agosto
	2ª quincena agosto	bol_summer4q??	16 al 31 de agosto
Puentes	2020	bol_puente20	[01/05/2020 al 03/05/2020], [09/10/2020 al 12/10/2020], [05/12/2020 al 08/12/2020], [25/12/2020 al 27/12/2020]
	2021	bol_puente21	[01/01/2021 al 03/01/2021], [09/10/2021 al 12/10/2021], [30/11/2021 al 01/11/2021], [04/12/2021 al 08/12/2021]
	2022	bol_puente22	[06/01/2022 al 09/01/2022]
Anterior y posterior a festividades	1ª semana anterior	bol_w1prev_*****??	***** semana santa, navidades, reyes y carnaval.
	1ª semana posterior	bol_w1post_*****??	***** semana santa, navidades, reyes y carnaval.
	2ª semana anterior	bol_w2prev_*****??	***** semana santa, navidades, reyes y carnaval.
	2ª semana posterior	bol_w2post_*****??	***** semana santa, navidades, reyes y carnaval.
	1º día anterior	bol_d1prev_*****	***** San Valentín, halloween, año nuevo, reyes magos, viernes santo, día del trabajador, virgen de agosto, hispanidad, todos los santos, constitución, día Inmaculada, navidad
	1º día posterior	bol_d1post_*****	***** San Valentín, halloween, año nuevo, reyes magos, viernes santo, día del trabajador, virgen de agosto, hispanidad, todos los santos, constitución, día Inmaculada, navidad
	2º día anterior	bol_d2prev_*****	***** San Valentín, halloween, año nuevo, reyes magos, viernes santo, día del trabajador, virgen de agosto, hispanidad, todos los santos, constitución, día Inmaculada, navidad
	2º día posterior	bol_d2post_*****	***** San Valentín, halloween, año nuevo, reyes magos, viernes santo, día del trabajador, virgen de agosto, hispanidad, todos los santos, constitución, día Inmaculada, navidad
	primera semana mes	bol_weekfirst	2020(19, 23, 27, 32, 36, 40, 45, 49), 2021(1, 5, 9, 13, 18, 22, 26, 31, 35, 40, 44, 48) y 2022(1, 5, 9, 14, 18)
	ultima semana mes	bol_weeklast	2020(18, 22, 26, 31, 35, 39, 44, 48, 53), 2021(4, 8, 12, 17, 21, 25, 30, 34, 39, 43, 47, 52) y 2022(4, 8, 13, 17)

Como resultado final, después de estas transformaciones, se obtiene un data set con 33.948 observaciones y 136 variables.

3.3.4.3. Agrupamiento de variables festividad, vacaciones y otras.

Se quiere reducir el número de variables, para lo que se decide agrupar las nuevas variables creadas, según significado. La tabla siguiente muestra las nuevas variables creadas y las que agrupa.

Tabla 9 Nuevas variables por agrupamiento. Fuente: Elaboración propia.

Nueva variable	variables agrupadas
festivos	bol_ssanta20', 'bol_ssanta21', 'bol_nadal20', 'bol_nadal21', 'bol_reyes21', 'bol_reyes22', 'bol_carnaval21', 'bol_carnaval22', 'bol_halloween', 'bol_new_year', 'bol_reyes', 'bol_good_friday', 'bol_labor_day', 'bol_virgin_day', 'bol_hispanic_day', 'bol_saints_day', 'bol_constitution_day', 'bol_concept_day', 'bol_nadal_day', 'bol_valent'
puentes	bol_puente20', 'bol_puente21', 'bol_puente22'
semana_prev1	bol_w1prev_ssanta20', 'bol_w1prev_ssanta21', 'bol_w1prev_nadal20', 'bol_w1prev_nadal21', 'bol_w1prev_carnav21', 'bol_w1prev_carnav22', 'bol_w1prev_valent'
semana_prev2	bol_w2prev_ssanta20', 'bol_w2prev_ssanta21', 'bol_w2prev_nadal20', 'bol_w2prev_nadal21', 'bol_w2prev_carnav21', 'bol_w2prev_carnav22', 'bol_w2prev_valent'
semana_post1	bol_w1post_ssanta20', 'bol_w1post_ssanta21', 'bol_w1post_reyes21', 'bol_w1post_reyes22', 'bol_w1post_carnav21', 'bol_w1post_carnav22', 'bol_w1post_valent'
semana_post2	bol_w2post_ssanta20', 'bol_w2post_ssanta21', 'bol_w2post_reyes21', 'bol_w2post_reyes22', 'bol_w2post_carnav21', 'bol_w2post_carnav22', 'bol_w2post_valent'
dia_prev1	bol_d1prev_valent', 'bol_d1prev_halloween', 'bol_d1prev_newyear', 'bol_d1prev_reyes', 'bol_d1prev_goodfriday', 'bol_d1prev_laborday', 'bol_d1prev_virginday', 'bol_d1prev_hispanicday', 'bol_d1prev_saintsday', 'bol_d1prev_constitutionday', 'bol_d1prev_conceptday', 'bol_d1prev_nadalday'
dia_prev2	bol_d2prev_valent', 'bol_d2prev_halloween', 'bol_d2prev_newyear', 'bol_d2prev_reyes', 'bol_d2prev_goodfriday', 'bol_d2prev_laborday', 'bol_d2prev_virginday', 'bol_d2prev_hispanicday', 'bol_d2prev_saintsday', 'bol_d2prev_constitutionday', 'bol_d2prev_conceptday', 'bol_d2prev_nadalday'
dia_post1	bol_d1post_valent', 'bol_d1post_halloween', 'bol_d1post_newyear', 'bol_d1post_reyes', 'bol_d1post_goodfriday', 'bol_d1post_laborday', 'bol_d1post_virginday', 'bol_d1post_hispanicday', 'bol_d1post_saintsday', 'bol_d1post_constitutionday', 'bol_d1post_conceptday', 'bol_d1post_nadalday'
dia_post2	bol_d2post_valent', 'bol_d2post_halloween', 'bol_d2post_newyear', 'bol_d2post_reyes', 'bol_d2post_goodfriday', 'bol_d2post_laborday', 'bol_d2post_virginday', 'bol_d2post_hispanicday', 'bol_d2post_saintsday', 'bol_d2post_constitutionday', 'bol_d2post_conceptday', 'bol_d2post_nadalday'
summer_1q	'bol_summer1q20', 'bol_summer1q21'
summer_2q	'bol_summer2q20', 'bol_summer2q21'
summer_3q	'bol_summer3q20', 'bol_summer3q21'
summer_4q	'bol_summer4q20', 'bol_summer4q21'

Como resultado de este agrupamiento, se reduce la dimensión del fichero desde 136 variables, a 56 variables.

3.3.4.4. Categorización variable udsVenta.

Los algoritmos de clasificación basados en ML tienen un mejor rendimiento para variable objetivo de tipo categórico. En este trabajo, la variable objetivo udsVenta es cuantitativa de tipo entero con valores en el rango [0, 273].

Se definen tres modelos para categorizar la variable udsVenta;

- 7 intervalos de igual longitud (categoría 1)
- 14 intervalos de igual longitud (categoría 2)
- 27 clases por los valores que toma la variable udsVenta (categoría 3).

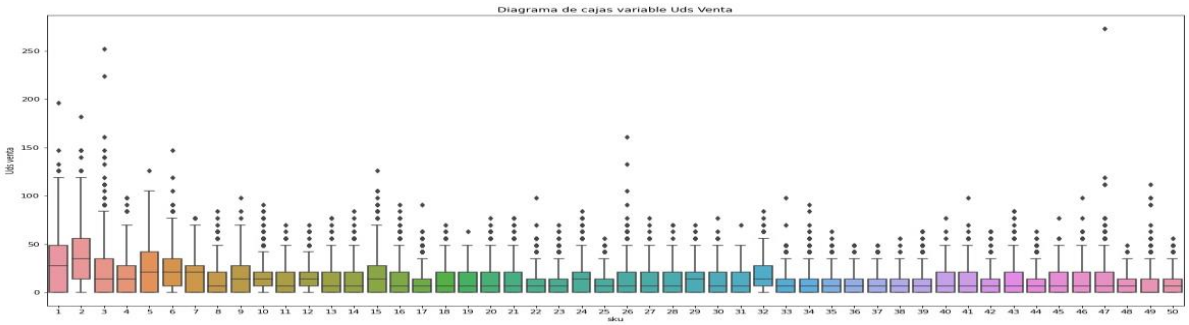
3.3.5. Visualización de las variables y análisis descriptivo.

La visualización de las variables permite identificar otros outliers o valores anómalos que hayan podido surgir durante la transformación. El diagrama de cajas es el ideal para esta visualización. Por otro lado, también interesa conocer si la distribución de frecuencias para cada variables se asemeja a la de una distribución normal. En caso contrario, se debería plantear una transformación para aproximar a una distribución normal.

3.3.5.1. Diagrama de cajas.

El diagrama de cajas permite identificar valores que pueden considerarse anómalos por situarse fuera de la nube de puntos entorno a su media. Estos valores son candidatos a ser eliminados.

Ilustración 11 Diagrama cajas para variable udsVenta. Fuente: elaboración propia

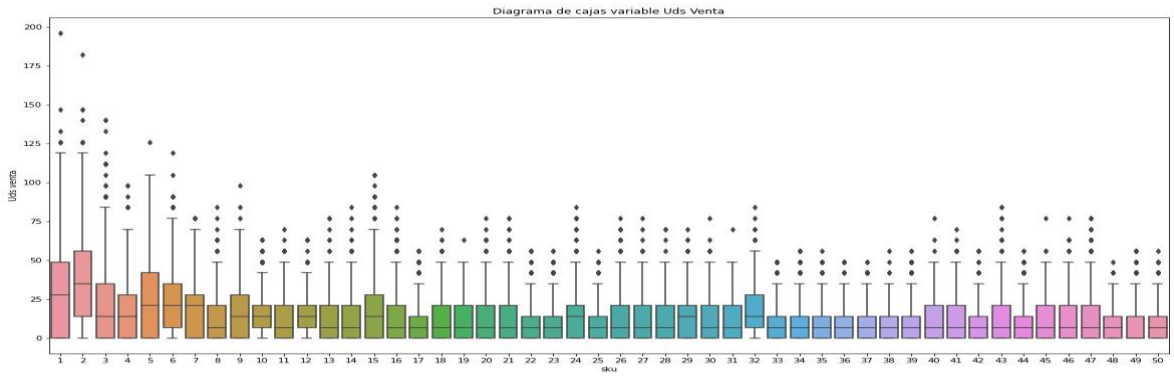


El diagrama de caja muestra valores outliers en cada uno de los agrupamientos realizados por cada valor de sku. Hay que ser cuidadoso a la hora de decidir la eliminación de outliers, dado que podría ser interesante conocer el volumen de ventas inusual de un sku en un momento determinado debido a factores de temporalidad como festividades, vacaciones o por factores promocionales.

Se considera outlier, a efectos de este análisis, aquellos valores que se encuentren disten más de 3 veces el rango intercuartílico (IQR). Por tanto, se identifican 56 valores que se encuentran fuera del rango [Q1-3*IQR, Q3+3*IQR], donde Q1 es el valor que ocupa el percentil del 25% y Q3 es el valor que ocupa el percentil del 75%. Fuera de este rango, existen 56 observaciones, que para un total de 33.948 observaciones, representa un 0,16% del data set. Se eliminan estos outliers del fichero.

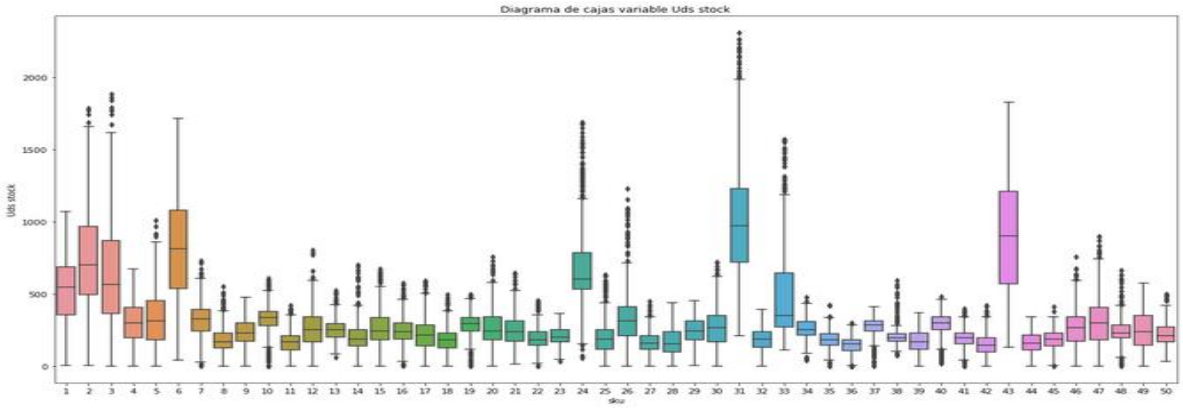
Véase el nuevo diagrama de cajas, tras eliminar los outliers.

Ilustración 12 Diagrama cajas para variable udsVenta. Fuente: elaboración propia



Se hace el mismo ejercicio, para la variable UdsStock, representación por diagrama de cajas.

Ilustración 13 Diagrama de cajas para variable udsStock. Fuente: elaboración propia.



No se detectan outliers para esta variable.

3.3.5.2. Histograma. Función de distribución.

Los algoritmos de ML requieren para un mejor modelado, que las variables se aproximen a una distribución normal. Al representar la frecuencia absoluta puede visualizarse su curva de distribución. Pongamos como ejemplo, el histograma para 6 valores de sku, seleccionados de manera aleatoria.

Ilustración 14 Histograma sku=11. Fuente; Elaboración propia.

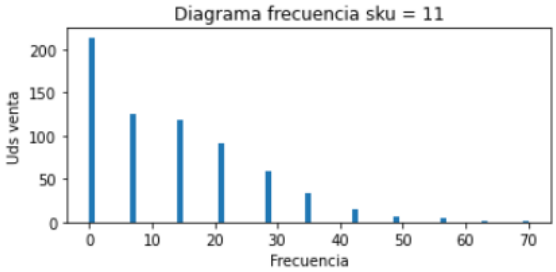


Ilustración 15 Histograma sku=23. Fuente; Elaboración propia.

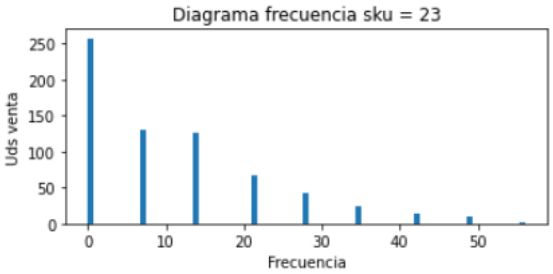


Ilustración 16 Histograma sku=29. Fuente; Elaboración propia

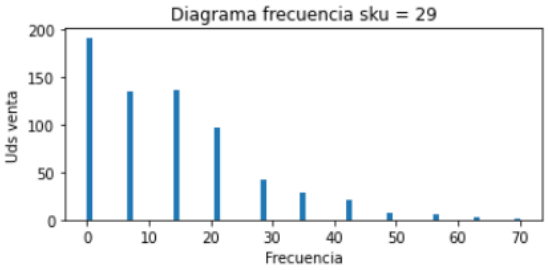


Ilustración 17 Histograma sku=40. Fuente; Elaboración propia

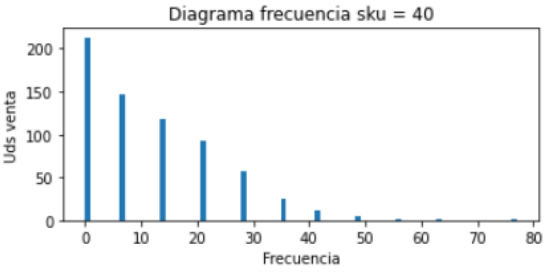


Ilustración 18 Histograma sku=44. Fuente; Elaboración propia.

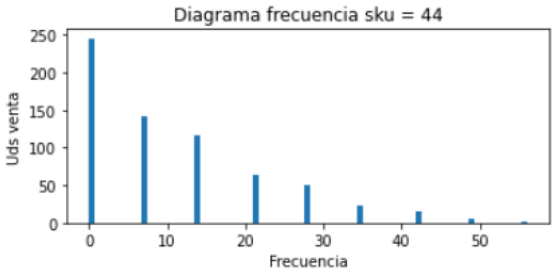
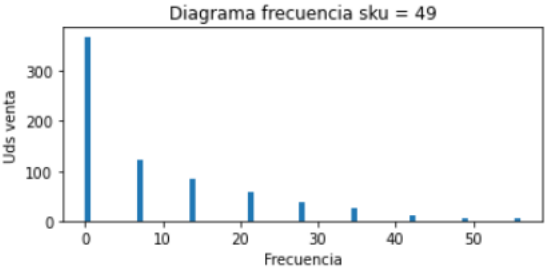


Ilustración 19 Histograma sku=49. Fuente; Elaboración propia.



La variable objetivo *udsVenta* no sigue una distribución de probabilidad normal. En todos los histogramas se observa un pronunciado sesgo hacia la derecha. Para corregir esta desviación, se obtiene una segunda variable objetivo *udsVentalog* como resultado de una transformación logarítmica sobre *udsVenta*. Con la variable transformada debería corregirse el sesgo, y aproximar la distribución a una distribución normal. Recordar, que

una vez finalizado el análisis, para interpretación del resultado, debe realizarse una transformación inversa de la variable.

Se representa histograma para la variable transformada sku, junto con su curva de distribución.

Ilustración 20 Histograma sku=11. Fuente: Elaboración propia

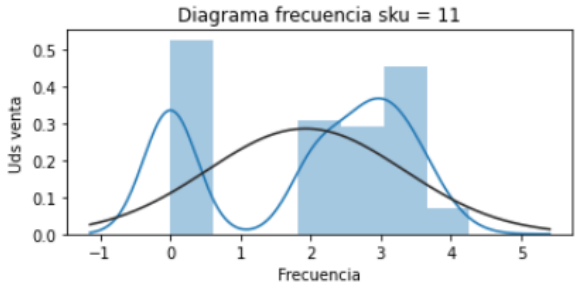


Ilustración 21 Histograma sku=23. Fuente; Elaboración propia.

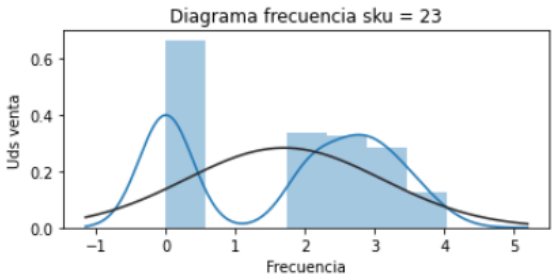


Ilustración 22 Histograma sku=29. Fuente; Elaboración propia

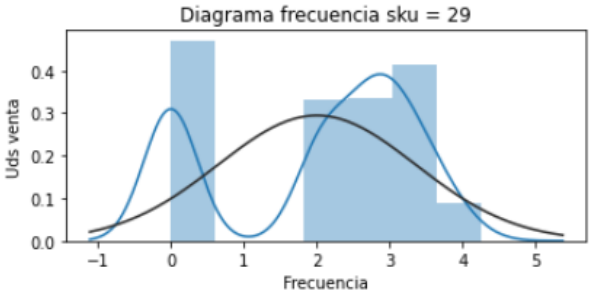


Ilustración 23 Histograma sku=40. Fuente; Elaboración propia.

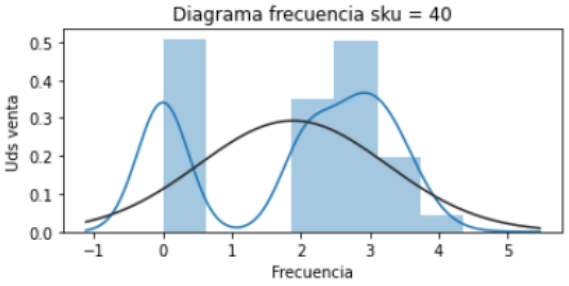


Ilustración 24 Histograma sku=44. Fuente; Elaboración propia.

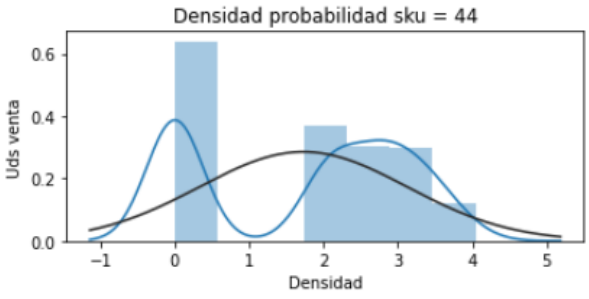
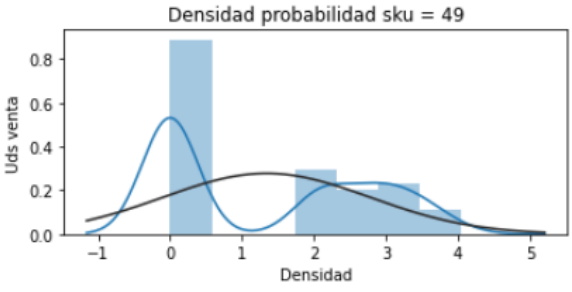


Ilustración 25 Histograma sku=49. Fuente; Elaboración propia.

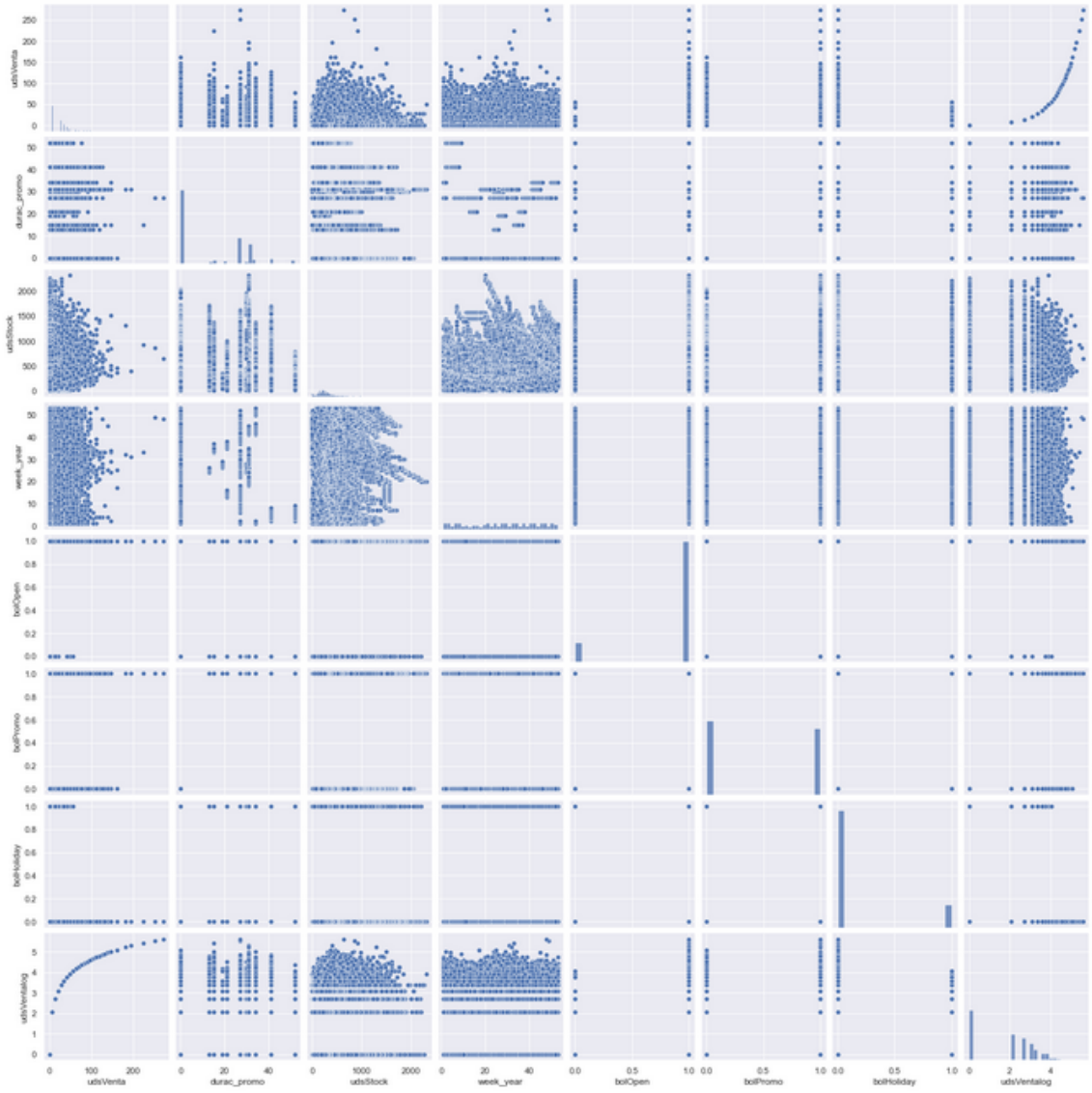


3.3.5.3. Análisis bivariante.

Se visualizan dos a dos las variables cuantitativas, en diagrama de dispersión, al objeto de identificar patrones de correlación o dependencia entre variables. Se realiza para las

variables que no son booleanas. Es decir, se analizan las variables (*udsVenta, durac_promo, udsStock, week_day, bolOpen, bolPromo, bolHoliday y udsVentalog).*

Ilustración 26 Representación bivariante. Fuente: elaboración propia

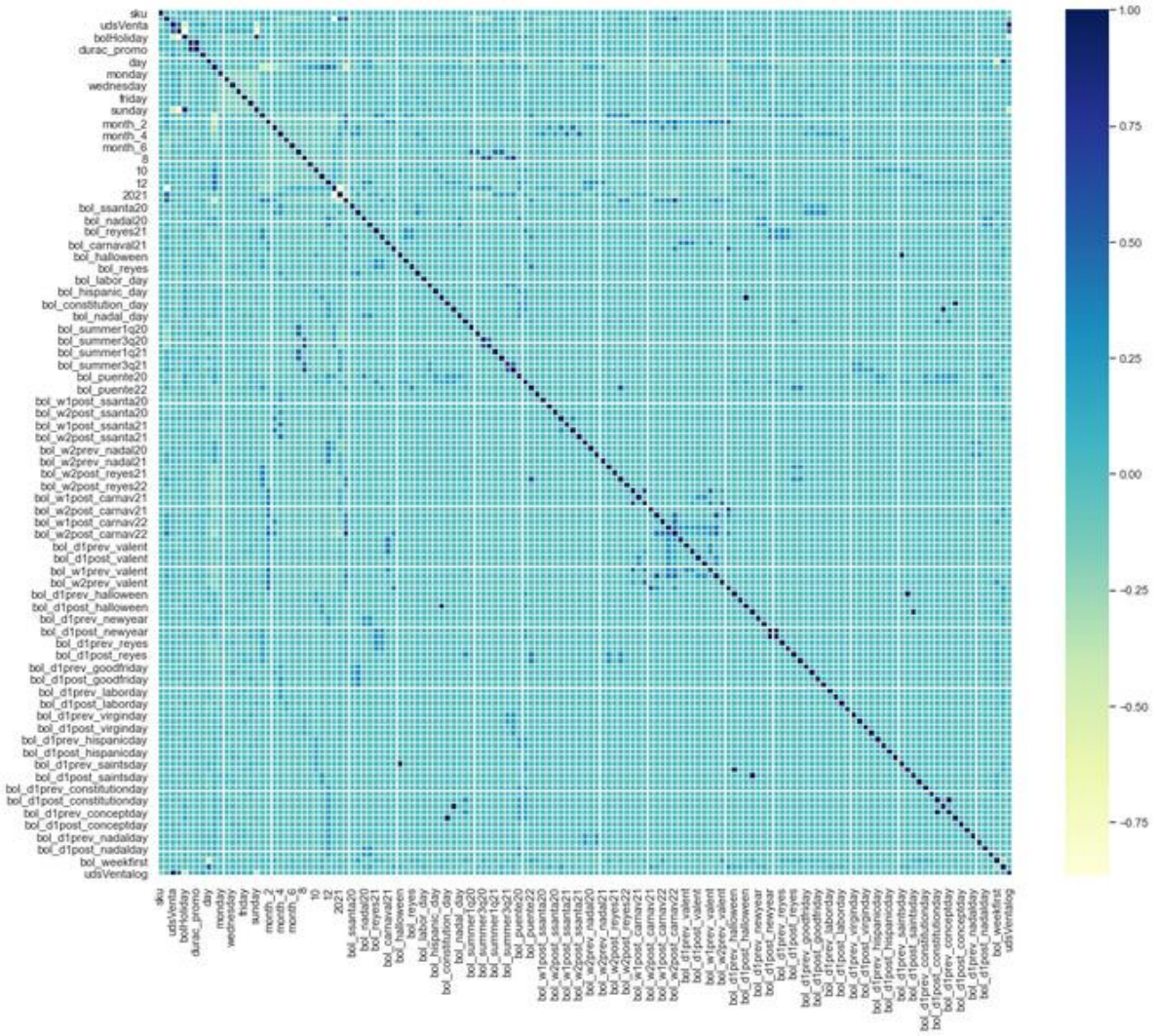


Sólo se observa dependencia de la variable udsVenta con las variables bolOpen y bolHoliday.

3.3.5.4. Análisis multivariante.

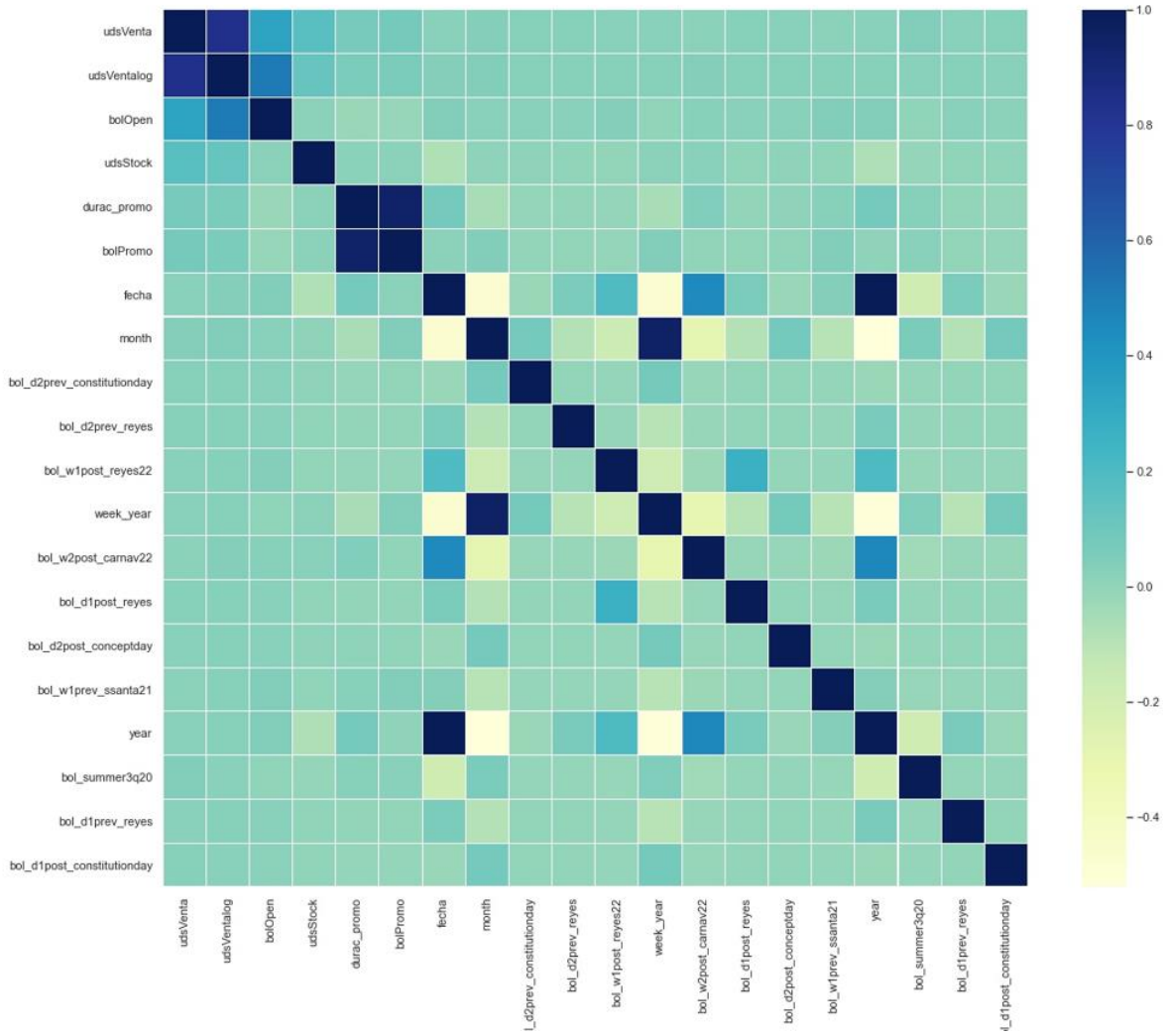
Se realiza análisis multivariante, por medio de matriz de correlaciones y mapa de color.

Ilustración 27 Mapa correlaciones. Fuente: elaboración propia.



No se observa una gran correlación entre variables. La mayoría de las correlaciones son nulas, con color tono azulado que corresponde a un valor de 0. Para un mejor análisis, véase ahora el mismo cuadro, pero para las 20 variables con mayor correlación.

Ilustración 28 Mapa correlaciones para las 20 mayores. Fuente: elaboración propia.



Entre las 20 variables representadas, las que mayor correlación tienen, son las de la siguiente tabla.

Tabla 10 Principales correlaciones entre variables. Fuente: elaboración propia.

Correlación negativa			**Correlación positiva**		
Variable	Variable	correlación	Variable	Variable	correlación
bolHoliday	bolOpen	-0,866	7_x	bolHoliday	0,836
2020	Fecha	-0,886	durac_promo	bolPromo	0,943
2021	2020	-0,800	bol_d1post_halloween	bol_saints_day	1

bol_d1prev_saintsday	bol_halloween	1
bol_d2prev_conceptday	bol_constitution_day	1
bol_d2post_constitutionday	bol_concept_day	1
bol_d2prev_saintsday	bol_d1prev_halloween	1
bol_d1post_saintsday	bol_d2post_halloween	1
bol_d2post_newyear	bol_d1post_newyear	1
bol_d1prev_conceptday	bol_d1post_constitutionday	1

Las variables con mayor correlación son booleanas. Para simplificar el data set, se elimina una de las variables, de los pares con correlación igual a 1. También se elimina la variable *durac_promo* por su elevada correlación con la variable *bolPromo.* El resto de pares de variables, se mantienen.

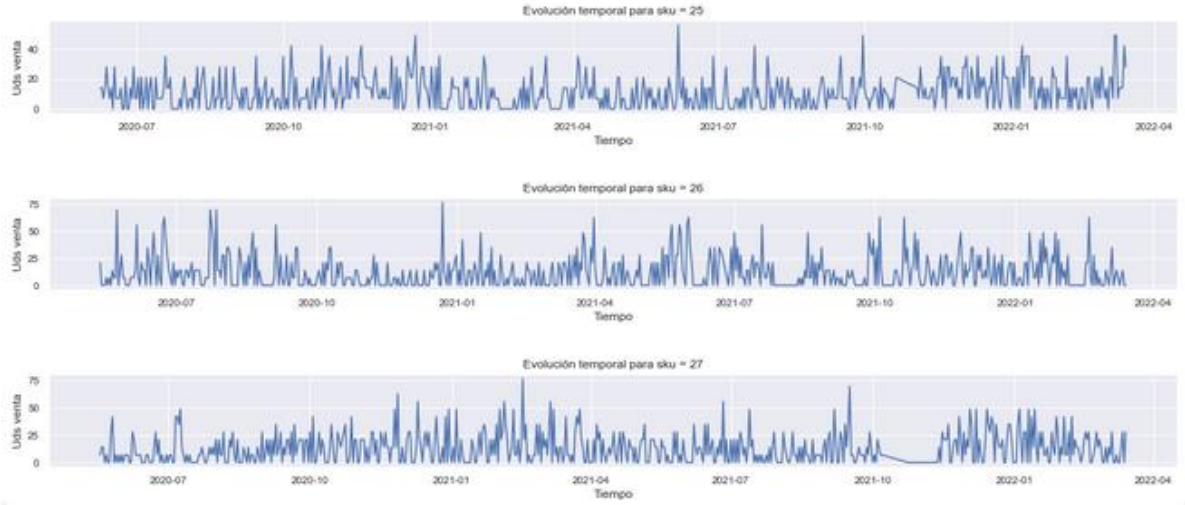
3.3.5.5. Análisis componentes de la serie temporal.

Se aborda en este apartado el análisis de la serie temporal. Una serie temporal es estacionaria cuando no se detecta estacionalidad. El resultado de predicción mejora cuando el data set es estacionario (Chu, C.W. & Zhang, G.P., 2003).

Véase gráfico distribución de la serie temporal para un conjunto aleatorio de valores de sku.

Ilustración 29 Representación serie temporal sku = [17, 18, 19]. Fuente: elaboración propia

Ilustración 30 Representación serie temporal sku = [25, 26, 27]. Fuente: elaboración propia.



No se observa existencia de estacionalidad en las series temporales. Véase también la descomposición de serie temporal en sus componentes (Casas Roma, J., febrero 2020). Las componentes de una serie temporal, son;

- Tendencia general se refiere al comportamiento general de los datos de una serie temporal, a aumentar o disminuir durante un largo periodo de tiempo.
- Estacionalidad, corresponde a patrones rítmicos o cíclicos que se observan de manera regular y periódica. Pueden ser estacionales o cíclicos. El primero es observable a corto plazo, mientras que los segundos se observan en periodos más largos.
- Error o residuo, es la componente que explica el comportamiento de la serie temporal que no se ha podido explicar por los componentes de tendencia y estacionalidad. Son imprevistas, incontrolables, impredecibles y erráticas.

Ilustración 31 Representación tendencia anual y estacionalidad mensual. Fuente: elaboración propia

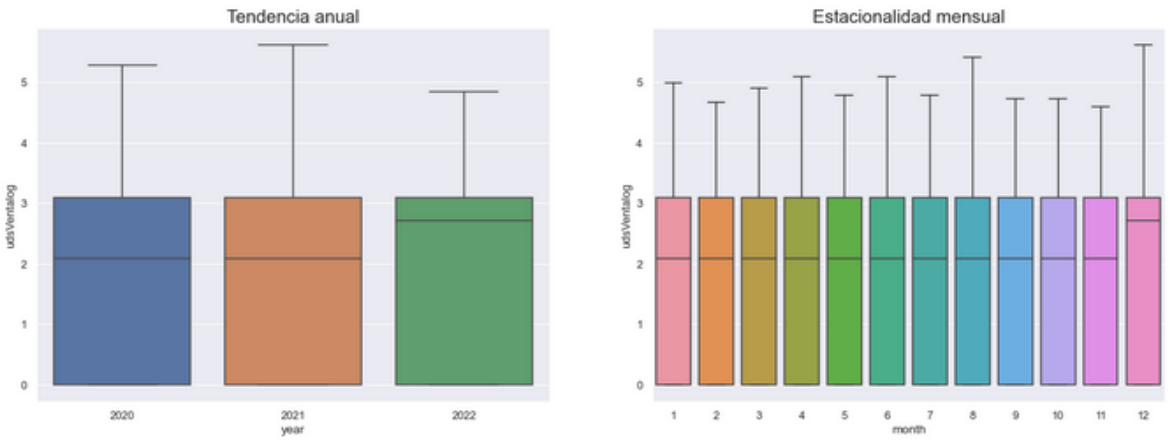
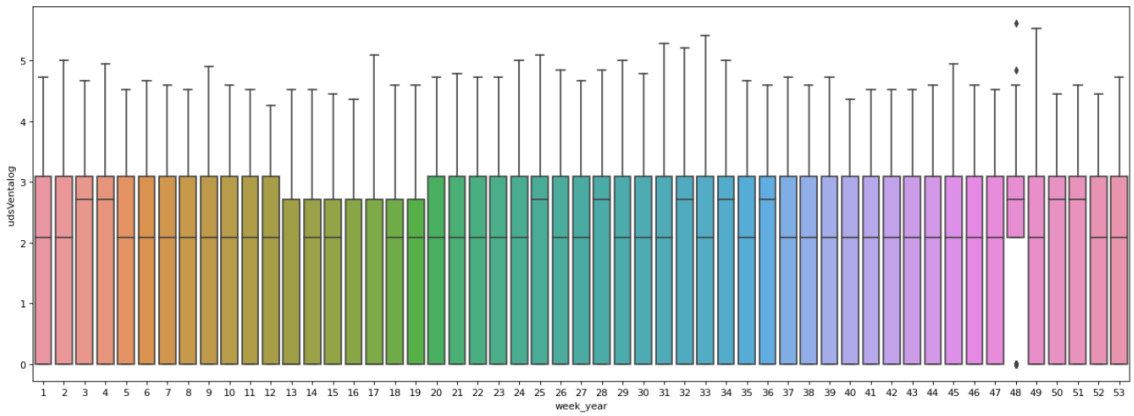


Ilustración 32 Representación estacionalidad semanal. Fuente: elaboración propia



Se representa en la ilustración 28 anterior, dos de las componentes (tendencia y estacionalidad mensual) para la variable objetivo *udsVenta*. La tendencia se representa por periodos anuales, mientras que la estacionalidad se representa por mensualidades. En la ilustración 29 se representa la estacionalidad semanal. Los diagramas de caja son planos para ambas componentes, sin variación en su media ni varianza.

El objetivo es descomponer la serie temporal en sus componentes, bien sea bajo un modelo aditivo o multiplicativo, y desarrollar modelos matemáticos para cada un de ellos. Se toman dos valores aleatorios de sku, para representar gráficamente sus componentes para un modelo aditivo (Ilustraciones 33 y 34).

Ilustración 33 Descomposición aditiva de las componentes de serie temporal sku=30. Fuente: elaboración propia

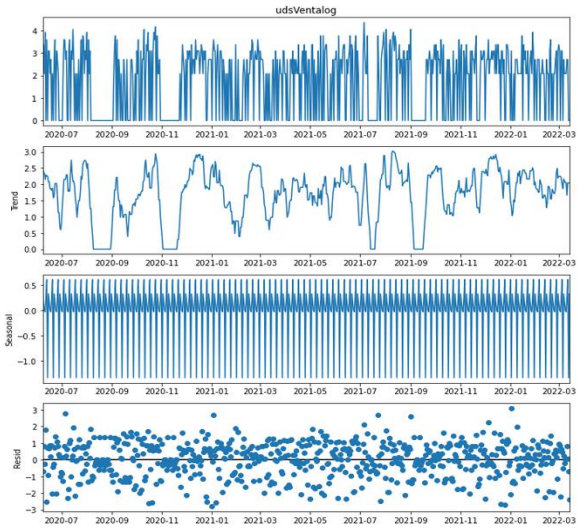
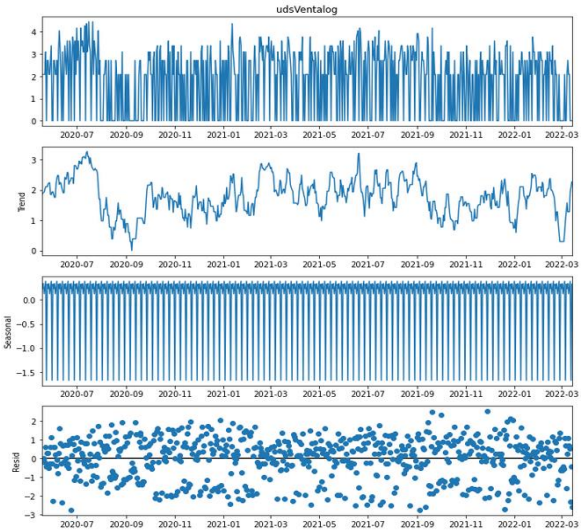


Ilustración 34 Descomposición aditiva de las componentes de serie temporal sku=43. Fuente: elaboración propia



No se observa es estas representaciones, tendencia alguna, ni patrón cíclico en la estacionalidad. Viene a confirmar lo que se está observando en todas las representaciones de este apartado que es la inexistencia de estacionalidad.

Véase por último, contraste de hipótesis para aceptar o rechazar la existencia de comportamiento estacionario en la serie temporal. Se aplica la prueba estacionaria de Dickey Fuller. la hipótesis nula es que la serie de tiempo posee una raíz unitaria y no es estacionaria. Si se rechazara la hipótesis nula afirmaríamos que la serie temporal es estacionaria. Se obtiene el estadístico ADF con valor -17,33.

```
ADF Statistic: -17.334928
p-value: 0.000000
Critical Values:
     1%: -3.431
     5%: -2.862
     10%: -2.567
```

El p-value es inferior a 0,05 (p=0), por tanto se rechaza la hipótesis nula H0 lo que significa que los datos no tiene una raíz unitaria y son estacionarios. El valor del estadístico ADF es. ADF = -17,33 que es inferior al valor crítico para una probabilidad de rechazo inferior al 1%.

Se concluye que la serie temporal de udsVenta con respecto a la fecha, es una serie estacionaria. La media y la varianza no cambian con el tiempo y tampoco siguen una tendencia.

Una serie estacionaria es mucho más fácil de predecir pues el comportamiento en el pasado puede trasladarse al futuro. Sus propiedades estadísticas se mantienen constantes a lo largo del tiempo.

3.4. Reducción de la dimensión del fichero.

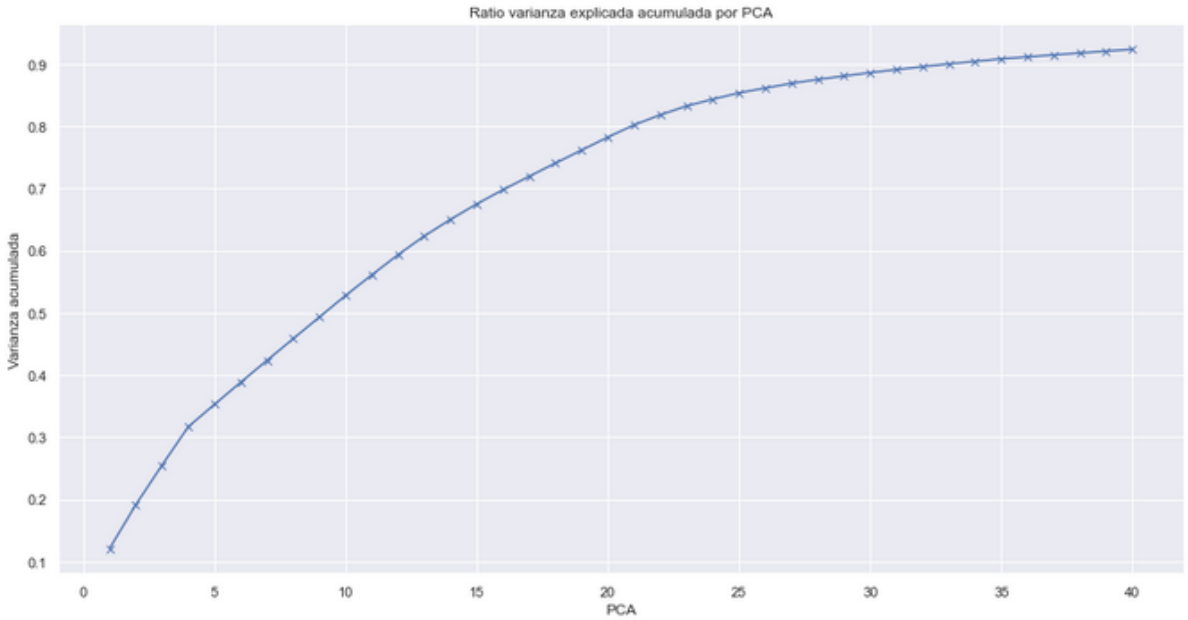
Recordemos que partimos de un data set con 142 variables, lo cual confiere cierta complejidad al algoritmo para diseñar un modelo de predicción. Se eliminan las 11 variables que se han identificado en el apartado de análisis de correlación. Se quiere ir un paso más, y reducir el número de variables aplicando algoritmo de componentes principales.

3.4.1. Análisis componentes principales.

Previo al algoritmo, se aplica normalización de las variables, al objeto de que todos los valores se encuentren en el rango [0, 1]. Para un total de 40 componentes principales, véase en el gráfico siguiente, la curva de varianza explicada que se acumula con cada componente principales (PCA).

Con las 4 primeras PCA se llega a explicar algo más del 30% de la variabilidad del modelo, y con 20 PCA se consigue explicar el 80%. A partir de 20 PCA, la aportación de cada nueva PCA a la variabilidad de la varianza es muy reducido. Por tanto, se eligen un modelo de 20 PCA. Puede comprobarse en la siguiente gráfica, la aportación de cada componentes a la varianza del modelo.

Ilustración 35 Curva varianza explicada acumulada por 40 componentes PCA. Fuente: elaboración propia



Se transforma el modelo de 142 variables independientes en un modelo de 20 variables. La complejidad del data ser se reduce considerablemente.

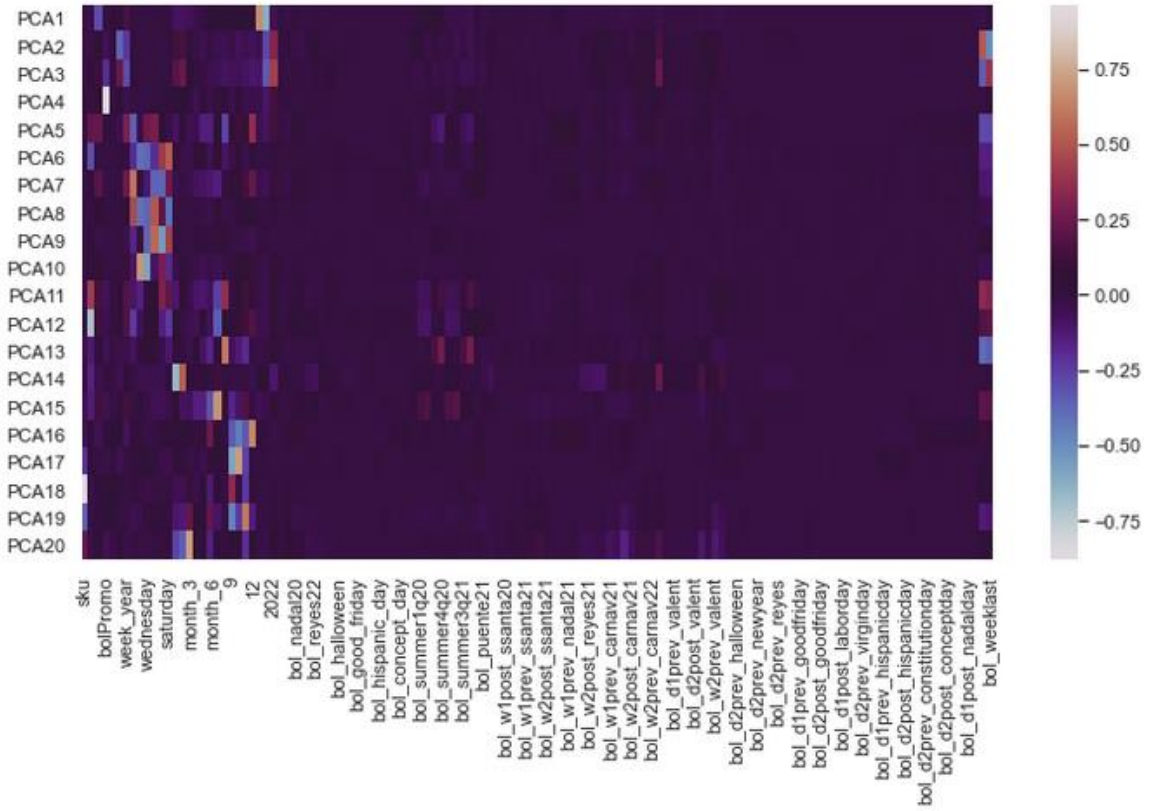
Véase a continuación la correlación existente de cada uno de los PCA con las variables originales. De entre todas, destacamos las siguientes componentes principales por su mayor correlación (superior o inferior a 0,80) con algunas variables originales.

Tabla 11 Correlación componentes principales. Fuente: elaboración propia

PCA	Variables dependientes positivas	Variables dependientes negativas
PCA4	bolPromo	
PCA11		bolOpen
PCA18		sku

El resto de PCA no tiene una correlación destacada con ninguna variable. En la siguiente ilustración, puede visualizarse el mapa de color, de correlaciones entre variables.

Ilustración 36 Mapa de color correlación de las PCA con las variables del modelo. Fuente: elaboración propia.

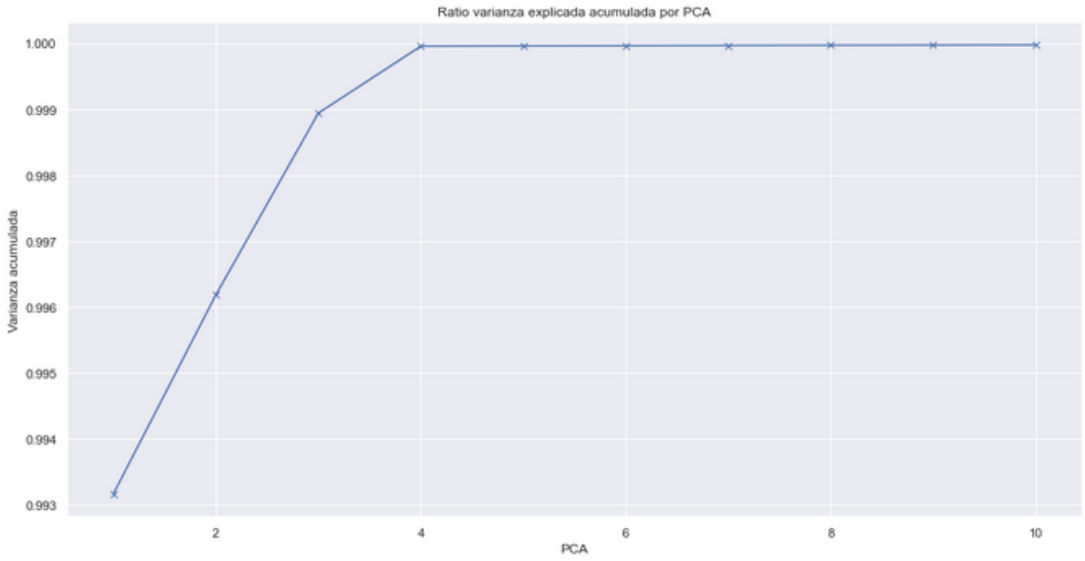


3.4.2. Análisis componentes principales con variables booleanas agrupadas.

En el apartado 3.3.4.3 Agrupamiento de variables festividad, vacaciones y otras. Se trata en este apartado un nuevo análisis de componentes principales para nuevo fichero con 47 variables, en lugar de las 142 variables iniciales.

Con 4 componentes principales se consigue explicar el 100% de la variabilidad del modelo, y con 3 componentes se alcanza explicar el 98% de la variabilidad del modelo. En la ilustración siguiente, puede observarse la aportación acumulada de cada componente a la varianza explicada del modelo. Es un buen resultado, el obtenido con este análisis. Sólo 3 componentes son capaces de explicar la variabilidad de un modelo de 47 variables.

Ilustración 37 Ratio de varianza explicada por PCA. Fuente: elaboración propia



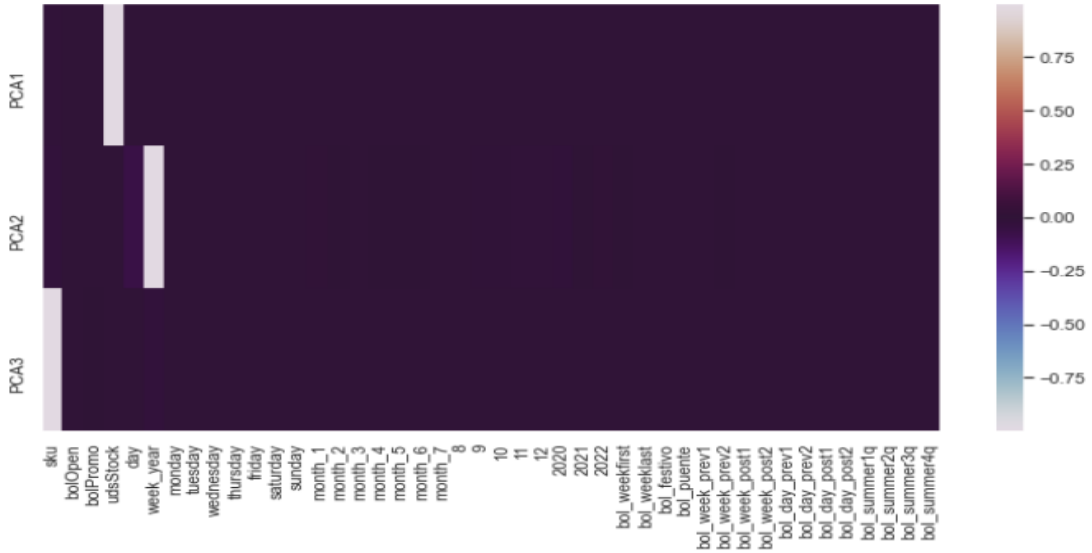
Se identifican las variables que tienen mayor contribución a la construcción de cada componente principal. No existen variables que contribuyen en negativo de manera significativa.

Tabla 12 Correlación componentes principales en modelo de variables booleanas agrupadas. Fuente: elaboración propia

PCA	Variables dependientes positivas	Variables dependientes negativas
PCA1	udsStock	
PCA2		week_year
PCA3	sku	

En la siguiente gráfica puede visualizarse por colores, el peso de cada variable en su contribución a la construcción de cada componente principal. Como conclusión de este apartado, el modelo construido a partir de agrupamiento de las variables de festividad, vacaciones, puentes, domingos, etc. permite mejor separación de componentes principales, que el modelo completo de 142 variables. Por tanto, se continúa en este trabajo con el fichero de variables agrupadas a 48 variables.

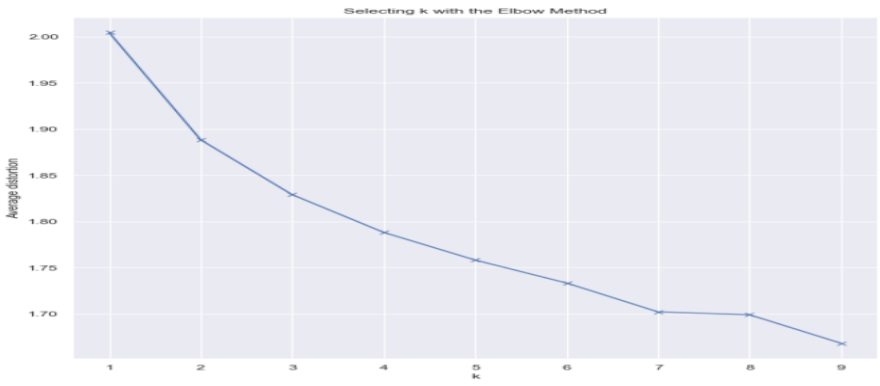
Ilustración 38 Mapa de color correlación de las PCA para modelo de variables booleanas agrupadas. Fuente; elaboración propia



3.5. Clusterización.

Se aborda ahora la posibilidad de agrupamiento de observaciones por clúster, mediante algoritmo no supervisado basado en K-Means. Basado en distancia euclídea, se calcula la distorsión media para cada centroide.

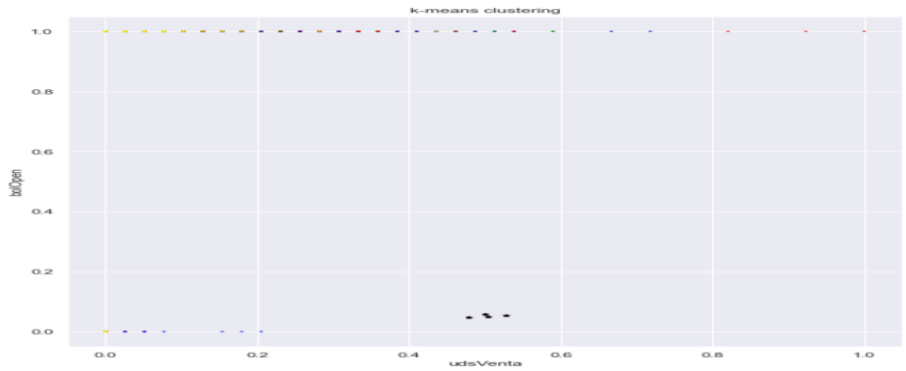
Ilustración 39 Curva método Elbow para distorsión media. Fuente; elaboración propia



No se observa un codo en la gráfica, pero sí una corrección progresiva de la curva a partir del 7º clúster. Se seleccionan 4 clúster para estudio del agrupamiento.

Se representan los centroides para algunos pares de variables. No se observa una clara separación de las observaciones para cada centroide. Quedan muy separados del centroide y los clúster muy juntos.

Ilustración 40 Representación variables bolOpen y bolHoliday frente a clúster 1 y 2. Fuente; elaboración propia



Se descarta la posibilidad de separación de observaciones por clúster. No se incorpora al fichero, nueva variable con referencias al clúster de pertenencia.

3.6. Separación fichero en conjuntos train y test.

Para separación del fichero en train y test, se considera como criterio una fecha concreta del rango total, de modo que en el conjunto train contenga el 80% de las observaciones y se reserva el 20% restante para el conjunto test. La fecha que actúa como filtro, es el 14/11/2021. En el conjunto train se tienen 25.567 observaciones, y en el conjunto test las 6.614 observaciones restantes.

3.7. Modelos de predicción.

Se ha depurado el data set y la siguiente fase trata el diseño un modelo a partir de algoritmos basados en ML. Se comparan los diferentes algoritmos, para diferentes opciones de categorizar la variables objetivo, visto en el apartado 3.3.4.4 Categorización variable udsVenta.

El nivel de precisión es mayor en modelos de clasificación supervisado con variables objetivo de tipo categórica. En un entrenamiento previo, con algoritmo SVR (versión del algoritmo SVM para variable objetivo cuantitativa), el nivel de precisión alcanzado es del -1,99% para el conjunto train, y del -12,7% para el conjunto test. Por tanto, se descarta el diseño de modelos de clasificación para variable objetivo de tipo cuantitativo, y se entrenan diferentes algoritmos para la variable objetivo de tipo categórico.

Se entrenan diferentes modelos para las diferentes clasificaciones realizadas para la variable objetivo udsVenta. Los mejores algoritmos, para prever cómo se modifica la demanda, son el soporte vectorial (SVM) (Carbonneau, R., Laframboise, K., & Vahidov, R.,

2008), redes neuronales con hasta 3 capas (Spiliotisa E. et al., 2020), XGBoost (Chen, T., & Guestrin, C., 2016) y árbol decisión de gradiente creciente (GBRT) (Ke, G. et al, 2017).

En la siguiente tabla se resume las principales magnitudes alcanzadas con el entrenamiento de cada uno de los algoritmos, tanto para el conjunto train como test. En la categoría udsVenta se indica el tipo de categorización utilizado para discretizar la variable udsVenta.

- 7 intervalos de igual longitud (categoría 1)
- 14 intervalos de igual longitud (categoría 2)
- 25 clases por los valores que toma la variable udsVenta (categoría 3).

Tabla 13 Cuadro resumen resultados diferentes algoritmos ML. Fuente; elaboración propia.

Algoritmo	Categoría udsVenta	Conjunto	precisión	MSE	RMSE
Gausiano	2	train	27,10%	1,76	2,19
		test	46,66%	1,02	1,55
SVM	1	train	92,48%	0,08	0,32
		test	91,07%	0,10	0,34
	2	train	69,65%	0,41	0,85
		test	63,67%	0,48	0,92
	3	train	35,62%	1,93	3,04
		test	29,78%	2,20	3,23
Arbol decisión	1	train	92,99%	0,07	0,29
		test	90,30%	0,10	0,35
	2	train	71,80%	0,34	0,72
		test	62,45%	0,47	0,87
	3	train	38,44%	1,45	2,38
		test	30,50%	1,77	2,68
Boosting Regres	1	train	18,24%	0,12	0,28
		test	5,80%	0,15	0,32
XGBoost	1	train	92,48%	0,08	0,32
		test	91,07%	0,10	0,34
Red Neuronal	1	train	91,69%	0,02	0,14
		test	90,09%	0,02	0,16
	3	train	15,30%	0,03	0,17
		test	14,21%	0,03	0,18

De la tabla adjunta se desprenden dos conclusiones:

- La categoría de udsVenta, con la que mejor precisión se obtiene es la categoría 1, consistente en 7 intervalos de igual tamaño.
- El algoritmo que mejor resultado da en términos de precisión es el árbol de decisión, aunque todos los cuatro algoritmos se consigue similar porcentaje precisión para los conjuntos train y test.

véase a continuación, los mejores resultados obtenidos con cada uno de estos cuatro algoritmos.

3.7.1. Algoritmo SVM

Se aplica GridSearchCV para obtener los parámetros que mayor precisión otorgan al modelo. Se aplica kernel = poly, C = 2 y cache_size = 100. Sólo clasifica los valores para el primer intervalo, que comprende los valores udsVenta = [0, 28]. Véase a continuación las matrices de confusión para los conjuntos train y test.

Ilustración 41 Matriz confusión conjunto train, algoritmo SVM, categoría 1. Fuente; elaboración propia.

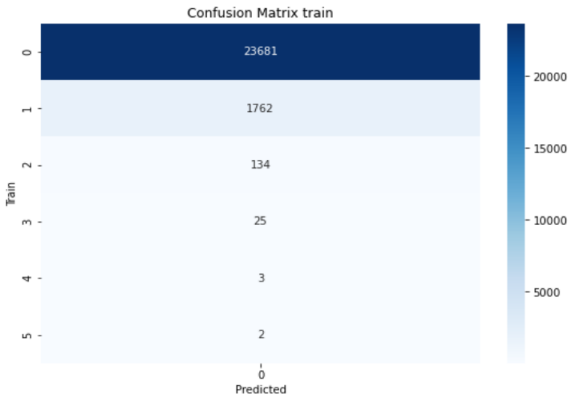
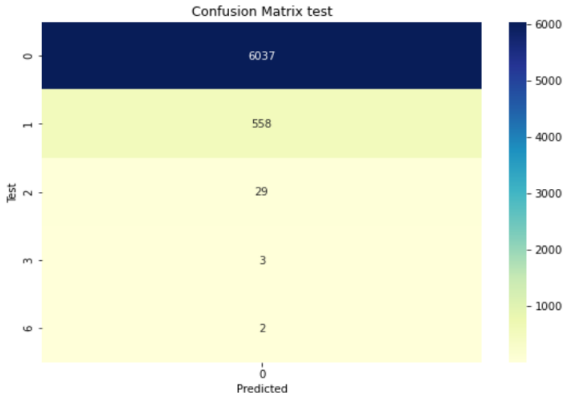


Ilustración 42 Matriz confusión conjunto test algoritmo SVM, categoría 1. Fuente; elaboración propia



3.7.2. Árbol de decisión.

Se aplica GridSearchCV para obtener los parámetros que mayor precisión otorgan al modelo. Los mejores parámetros son; max_depth = 8, min_samples_split = 100, 7 clases de clasificación. Véase a continuación las matrices de confusión para los conjuntos train y test.

Ilustración 43 Matriz confusión conjunto train en algoritmo Árbol Decisión, categoría 1. Fuente; elaboración propia

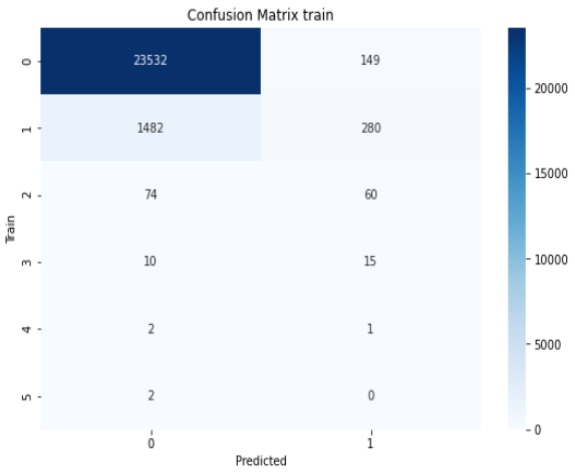
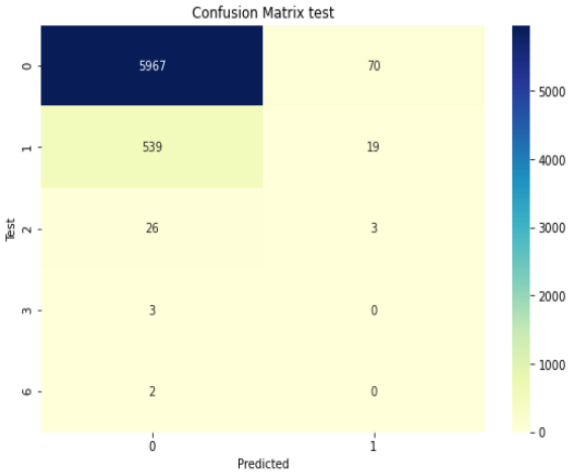


Ilustración 44 Matriz confusión conjunto test en algoritmo Árbol Decisión, categoría 1. Fuente; elaboración propia



Clasifica la variable objetivo udsVenta en 2 categorías, que corresponde con los intervalos; [0, 28] y [29, 56].

3.7.3. Algoritmo XGBoost

Se aplica GridSearchCV para obtener los parámetros que mayor precisión otorgan al modelo. Los mejores parámetros son; max_depth = 5, C = 2 y cache_size = 100. Sólo clasifica los valores para el primer intervalo, que comprende los valores udsVenta = [0, 28]. Véase a continuación las matrices de confusión para los conjuntos train y test.

Ilustración 45 Matriz confusión train, algoritmo XGBoost, cat. 1. Fuente; elaboración propia.

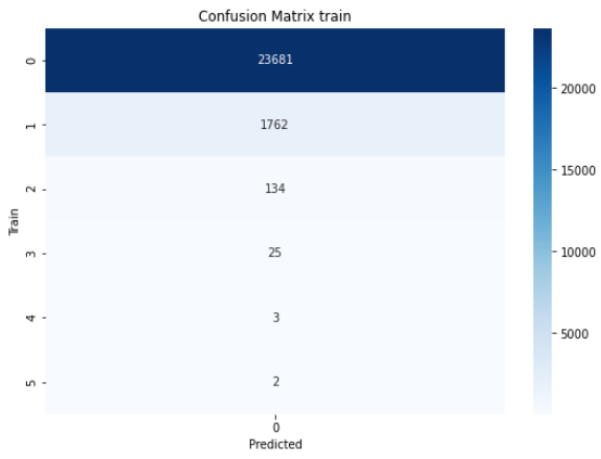
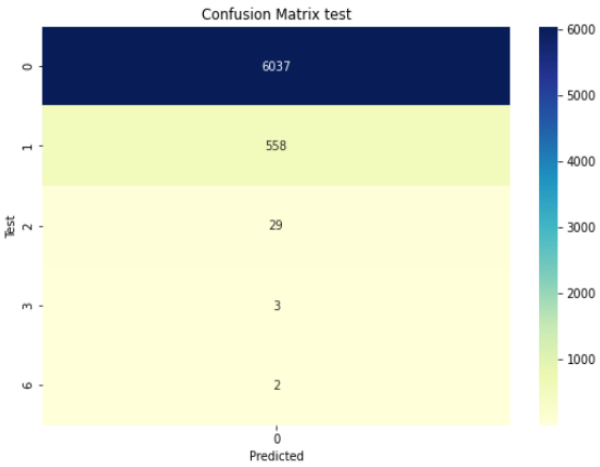


Ilustración 46 Matriz confusión test algoritmo XGBoost, cat. 1. Fuente; elaboración propia,



3.7.4. Redes Neuronales

Se entrena red neuronal con 4 capas, para clasificar 7 clases de la variable objetivo *udsVenta*. Se aplican funciones de activación elu y relu. En la última capa, la función de activación es softmax que pondera la predicción de la variable objetiva, entre los posibles valores, de modo que la categoría asignada será aquella con mayor valor.

Ilustración 47 Modelo algoritmo red neuronal. Fuente; elaboración propia.

```
# Número de clases
num_classes = 7

model_neuronal2 = Sequential()

# Añadir las capas indicadas
model_neuronal2.add(Dense(64, input_shape=(44,), activation="relu",
                          kernel_initializer="random_normal", bias_initializer="zeros"))
model_neuronal2.add(Dense(32, activation="elu"))
model_neuronal2.add(Dense(10, activation="relu"))
model_neuronal2.add(Dense(8, activation="relu"))
model_neuronal2.add(Dense(num_classes, activation="softmax"))
```

El modelo predice la variable udsVenta en las dos primeras categorías, que corresponde con los intervalos [0, 28] y [29, 56]. Véase a continuación las matrices de confusión para los conjuntos train y test.

Ilustración 48 Matriz confusión conjunto train en algoritmo Redes neuronales, categoría 1. Fuente; elaboración propia

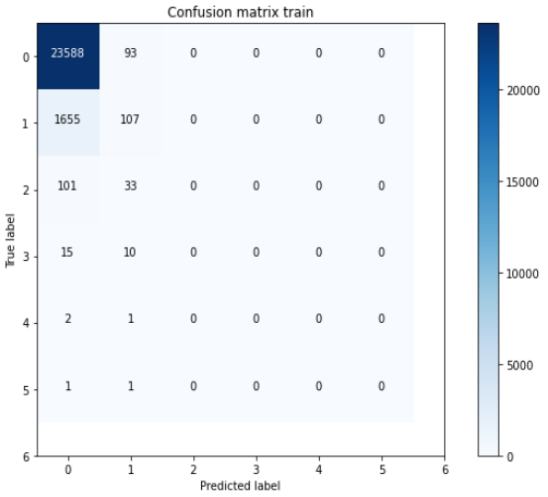
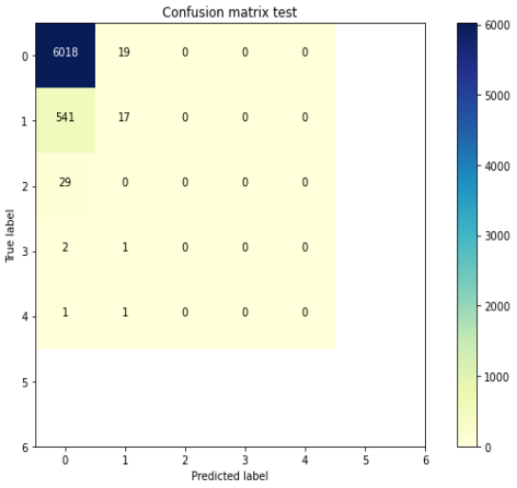


Ilustración 49 Matriz confusión conjunto test en algoritmo Redes Neuronales, categoría 1. Fuente; elaboración propia



3.8. Cálculo del coste stock seguridad.

Una vez seleccionados los modelos para predicción de la demanda con mayor porcentaje de predicción, se elige aquel que reduce el coste del stock de seguridad con respecto al modelo utilizado por la empresa. El modelo entrenado para predicción de la demanda será válido y reemplazará al modelo de gestión de stock utilizado por la empresa, si además de predecir la demanda en un nivel de precisión superior al 90%, reduce el coste de stock de seguridad asociado al modelo.

Se calcula el stock de seguridad para el modelo de predicción entrenado y para las ventas reales, en el periodo de test (desde el 15/11/2021 hasta el 14/03/2022, inclusive). Se ha considerado el conjunto test_1 que recordemos discretiza la variable objetivo *udsVenta* en 7 intervalos de igual tamaño, con lo que cada intervalo tiene una amplitud de 39 unidades de venta.

Para el cálculo de stock de seguridad en las ventas previstas en el periodo de test, se realiza el siguiente cálculo (Heizer, J. & Render, B., 2001):

- Stock de seguridad previsto = Factor servicio * RMSE * raíz del ciclo de aprovisionamiento. El factor servicio es el nivel de servicio deseado, obtenido como el valor para el que la probabilidad de satisfacer la demanda es de al menos el 98%. Suponiendo que la demanda se distribuye según una normal tipificada, para una probabilidad del 98% se corresponde un factor servicio = 2,05. RMSE (Root Mean Squared Error) o error cuadrático medio es la raíz cuadrada de la media del cuadrado de las diferencias entre el valor real y el valor estimado. $RMSE = \sqrt{(\Sigma \frac{(x_i - y_i)^2}{n}}$). El valor de ciclo de aprovisionamiento se facilitan en el fichero DatosCicloAprovisionamiento, variable *díasLeadtime* (plazo habitual de reposición por el proveedor).

Para el cálculo de stock de seguridad sobre ventas reales, se realiza el siguiente cálculo:

- Stock de seguridad real = (plazo máximo de entrega – plazo de entrega habitual) * demanda media del producto. Los valores de plazo máximo de entrega y plazo de entrega habitual, se pueden obtener del fichero DatosCicloAprovisionamiento, bajo el nombre de las variables *diasEntrePedidos* y *diasLeadtime*, respectivamente. La demanda media se calcula para cada sku en el periodo de estudio.
  Se ha considerado que si el periodo entre pedidos es superior al plazo máximo de entrega por parte del proveedor, no es necesario disponer de stock de seguridad para dicho producto. Resulta un stock de seguridad para el periodo del conjunto test, de 9.801 unidades.

Una vez conocido o estimado el stock de seguridad, se obtiene su coste que se calcula para un valor del 5% del precio de venta. Se aplica la fórmula; valor del stock = 5% * precio

medio de venta * stock. El valor del stock de seguridad real para el modelo utilizado por la empresa es de 1.148,71 euros. Véase en el apéndice, tabla con los datos utilizados en la obtención del stock de seguridad y su coste, para el modelo real utilizado por la empresa y la simulación para los modelos de predicción seleccionados.

A partir de RMSE de los 4 algoritmos seleccionados por su mejor precisión, se seleccionará aquel para el que el coste del stock de seguridad mejore con respecto al coste del stock de seguridad del modelo utilizado por la empresa.

Tabla 14 Comparativa coste stock de seguridad de los diferentes modelos vs el modelo de la empresa. Fuente; elaboración propia.

Algoritmo	precisión	RMSE	coste stock seguridad real	coste stock seguridad modelo
SVM	91,07%	0,34	1.148,71	1.044,87
Árbol decisión	90,30%	0,35	1.148,71	1.230,39
XGBoost	91,07%	0,34	1.148,71	1.044,87
Red Neuronal	90,09%	0,16	1.148,71	615,20

El modelo obtenido a partir de red neuronal a pesar de no ser el de mejor precisión, sí permite una gestión del inventario con un coste de stock de seguridad inferior al resto de algoritmos, y mejor al utilizado por la empresa.

3.9. Consecución de los objetivos planteados.

Al inicio de este trabajo se fijó un objetivo principal;

- conseguir un modelo que permita predecir la demanda de un producto en un espacio temporal.

Los modelos obtenidos alcanzan una precisión superior al 90%. Puede decirse que el objetivo principal se ha alcanzado.

Se establecieron otros dos objetivos específicos;

- Definir el espacio temporal en el que la predicción de la demanda es estadísticamente significativa.
- Contrastar la eficiencia en la gestión del inventario de la empresa, gracias a la predicción de la demanda externa esperada, con una reducción de costes por almacenamiento, entre antes y después de aplicar el modelo.

No se han llegado a definir el espacio temporal en el que la predicción de la demanda es estadísticamente significativa. Sin embargo, sí puede afirmarse que el espacio temporal

sobre el que se ha entrenado el modelo es suficientemente grande como para que las predicciones obtenidas sean estadísticamente significativas.

El modelo para predicción de la demanda ha permitido rediseñar el stock de seguridad necesario para evitar roturas de stock. Por tanto, se ha optimizado la gestión del inventario con un ahorro en costes. Por tanto, sí se ha cumplido con el segundo de los objetivos específicos.

4. Conclusiones y trabajos futuros

Hasta el momento, se han entrenado diferentes modelos para predicción de la demanda, bajo diferentes algoritmos de ML, y con diferentes criterios de discretización de la variable objetivo. De cada uno de ellos, se ha obtenido su nivel de precisión, error cuadrático medio y matrices de confusión. Los niveles de precisión alcanzados han sido superiores al 90%. Se representa a continuación las predicciones obtenidas por cada uno de los modelos y las ventas reales para el periodo test, para los valores de sku = 3 y sku = 4.

Ilustración 50 Representación ventas reales y predicción mejores modelos. Fuente; elaboración propia.

En el apartado 3.1 Presentación del caso de estudio.l objetivo del trabajo; *"diseñar un modelo de predicción de la demanda que permita una mejor estimación del stock de seguridad".* Con los resultados obtenidos de los diferentes algoritmos entrenados con los datos de este estudio, se seleccionará aquel para el que el coste de stock de seguridad mejore con respecto al coste del modelo de previsión de la empresa, en términos monetarios.

El stock de seguridad es la cantidad de inventario que se almacena como reserva en la instalación. Con este inventario, la compañía puede hacer frente a imprevistos como aumentos en la demanda, cambios en la rotación de un sku o retrasos con los proveedores.

4.1. Elección del mejor modelo.

Finalmente, el modelo que permite una mejor gestión del inventario, bajo el criterio de optimizar el coste del stock de seguridad, es el modelo obtenido a partir de algoritmos de Redes Neuronales. Para una precisión similar al del resto de modelos, de entorno al 90%, se puede gestionar un stock de seguridad con un coste muy inferior al que tiene la empresa en su actual modelo de gestión de inventario. Concretamente, pasaría de un coste de 1.148,71 u.m. a 615,20 u.m. que supone un ahorro de 533,51 u.m.

En términos de precisión, todos los modelos coinciden en que se obtiene mejor precisión cuando se trabaja con la variable objetivo convertida en variable discreta con respecto a la variable tipo continuo. Concretamente, entre los diferentes tipo de discretizar la variable objetivo, la que mejor funciona es la discretización en 7 intervalos de igual tamaño.

4.2. Seguimiento de la planificación y metodología.

La planificación temporal definida al inicio de este trabajo se ha cumplido con total rigurosidad. Cada una de las tareas definidas se ha completado en el plazo de tiempo y fechas fijadas.

La metodología seguida para alcanzar los objetivos planteados en este trabajo ha sido adecuada. Se han establecido tres fases de trabajo, identificada cada una de ellas con un notebook de python;

- Preparación_fichero_TFM. En este notebook se han realizado las primeras tareas de lectura y depuración de los diferentes dataset, eliminación de outliers e incorporación de nuevas variables sobre festividad.
- Reducción_dimensión_TFM. Se procede en este notebook a una análisis descriptivo y temporal del dataset. Se estudia correlación de variables, componentes principales para acabar con una reducción de la dimensionalidad del dataset por agrupación o eliminación de variables.
- Algoritmo_ML. Por último, se entrenan diferentes modelos para predicción de la demanda, bajo diferentes algoritmos de ML. Previo a ello, se deciden varios criterios para discretizar la variable objetivo *udsVenta*. De cada uno de los modelos, interesa conocer el grado de precisión, los valores de MSE y RMSE, y las matrices de confusión.

El primero de los notebook ha sido el de mayor carga de trabajo. La preparación, limpieza y depuración del fichero ha llevado más tiempo que en el resto de fases. En la última de las fases, se materializan los resultados obtenidos, y el cumplimiento de los objetivos establecidos.

4.3. Impactos en sostenibilidad, ético-social y de diversidad.

Se ha logrado cumplir con todos los impactos previstos en el apartado 1.3 de este trabajo. No han aparecido impactos no previstos.

4.4. Línea de trabajo futuro.

Con este trabajo se fija una posible línea de trabajo futuro. El diseño de modelos de predicción bajo algoritmos basados en redes neuronales es increíblemente amplio. El modelo utilizado ha sido excesivamente sencillo, y los resultados obtenidos han sido extremadamente sorprendentes. El nivel de precisión en las predicciones supera el 90% y el ahorro en coste por stock de seguridad es considerablemente inferior al que incurre actualmente la empresa en su modelo de gestión de stock.

Ha quedado pendiente una nueva línea de trabajo para el entrenamiento de nuevos modelos de predicción basados únicamente en redes neuronales, con arquitecturas más amplias y complejas. Explorar la componente cíclica de la serie temporal con el entrenamiento de un nuevo dataset que incluya las ventas realizadas en el día, y los seis inmediatamente anteriores (Bagnato, J. I., 2020). El patrón de semana podría aportar una mejora en la precisión del modelo.

5. Bibliografía

Adnan Khan, M. et al. (2020). Effective Demand Forecasting Model Using Business Intelligence Empowered With Machine Learning. *IEEE Access*.

Ali, O. G. et al. (2009). SKU demand forecasting in the presence of promotions. *Expert Systems with Applications, 36*, 12340-12348.

Arunra, N. S. & Ahrens, D. (2015). A hybrid seasonal autoregressive integrated moving average and quantile regression for daily food sales forecasting. *International Journal of Production Economics, 170*, 321-335.

Bagnato, J. I. (2020). *Aprende machine learning. Teoría + práctica Python.*

Bojer, C. S. & Meldgaard, J. P. (2021). Kaggle forecasting competitions: An overlooked learning opportunity. *International Journal of Forecasting 37*, 587–603.

Carbonneau, R., Laframboise, K., & Vahidov, R. (2008). Application of machine learning techniques for supply chain demand forecasting. *European Journal of Operational Research 184*, 1140-1154.

Casas Roma, J. (febrero 2020). *Introducción al análisis de series temporales.* UOC.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. 785-794.

Chu, C.W. & Zhang, G.P. (2003). A comparative study of linear and nonlinear models for aggregate retail sales forecasting. *International Journal of Production Economics, 86*, 217-231.

Cooper, L. G. et al. (1999). Promocast: A new forecasting method for promotion planning. *Marketing Science, 18*, 301–316.

Crone, S. F., Hibon, M., & Nikolopoulos, K. (2011). Advances in forecasting with neural networks? Empirical evidence from the NN3 competition on time series prediction. *International Journal of Forecasting, 27*, 635-660.

Croston, J. D. (1972). Forecasting and stock control for intermittent demands. *Journal of the Operational Research Society, 23*, 289-303.

Dubé, J. P. (2004). Multiple discreteness and product differentiation: Demand for carbonated soft drinks. *Marketing Science, 23*, 66–81.

Fildes, R. et al. (2022). Retail forecasting: Research and practice. *International Journal of Forecasting 38*, 1283-1318.

Gijsbrechts, E. et al. (2003). The impact of store flyers on store traffic and store sales: A geo-marketing approach. *Journal of Retailing, 79*, 1–16.

Heizer, J. & Render, B. (2001). *Dirección de la Producción. Decisiones Tácticas.* Madrid: Prentice Hall.

Huang, T. et al. (2019). Forecastin retailer product sales in the presence of structural breaks. *European Journal of Operational Research, 279,*, 459–470.

Huber, J. & Stuckenschmidt, H. (2020). Daily retail demand forecasting using machine learning with emphasis on calendric special days. *Elsevier*.

Karmy, J.P. & Maldonado, S. (2019). Hierarchical time series forecasting via Support Vector Regression in the European Travel Retail Industry. *Expert Systems With Applications 137*, 59-73.

Ke, G. et al. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems, vol. 30*, 3146-3154.

Kingma, D. P. & Ba, J. L. (2015). A method for stochastic optimization. *In International conference on learning representations*.

Kolassa, S. (2016). Evaluating predictive count data distributions in retail sales forecasting. *International Journal of Forecasting, 32*, 788-803.

Koschat, M. A. (2008). Store inventory can affect demand: Empirical evidence from magazine retailing. *Journal of Retailing*(84), 165–179.

Kourentzes, N., Petropoulos, F., & Trapero, J. R. (2014). Improving forecasting by estimating time series structural component across multiple frequencies. *International Journal of Forecasting, 30*, 291-302.

LeCun, Y. A., Bottou, L., Orr, G. B., & Müller, K. R. (2012). BackProp. In Lecture notes in computer science, Neural networks: Tricks of the trade. *Springer, Berlin, Heidelberg.*, 9-48.

Levis, A.A., & Papageorgiou, L.G. (2020). Customer demand forecasting via support vector regression analysis. *Chemical Engineering Research and Design, 83*, 1009-1018.

Lu, Chi-Jie. (2014). Sales forecasting of computer products based on variable selection scheme and support vector regression. *Neurocomputing, 128*, 491-499.

Makridakis, S. et al. (2018). Statistical and machine learning forecasting methods: Concerns and ways forward. *PLOS ONE, 13*, 1-26.

Ramanathan, U. & Muyldermans, L. (2010). Identifying demand factors for promotional planning and forecasting: A case of a soft drink company in the UK. *Int. J. Production Economics 128*, 538-545.

Shenstone, L., & Hyndman, R. J. (2005). Stochastic models underlying Croston's method for intermittent demand forecasting. *Journal of Forecasting 24*, 389-402.

Soares, L. J., & Medeiros, M. C. (2008). Modeling and forecasting short-term electricity load: a comparison of methods with an application to brazilian data. *International Journal of Forecasting, 24*, 630-644.

Spiliotisa E. et al. (2020). Comparison of statistical and machine learning methods for daily SKU demand forecasting. *European Journal of Operational Research*.

Syntetos, A. A., & Boylan, J. E. (2005). The accuracy of intermitent demand estimates. *International Journal of Forecasting, 21*, 303-314.

Taylor, J. W. (2007). Forecasting daily supermarket sales using exponentially weighted quantile regression. *European Journal of Operational*, 154-167.

Teunter, R. H., Syntetos, A. A., & Babai, M. Z. (2011). Intermittent demand: Linking forecasting to inventory obsolescence. *European Journal of Operational Research, 214*, 606–615.

Willemain, T. R., Smart, C. N., & Schwarz, H. F. (2004). A new approack to forecasting intermittent demand for service parts inventories. *International Journal of Forecasting, 20*, 375–387.

Xiaoqun, L. et al. (2019). Research on Short-term Load Forecasting Using XGBoost Based on Similar Days. *International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*, 1-4.

Zhang, W. et al. (2018). Parallel aspect-oriented sentiment analysis for sales forecasting with big data. *Production and Operations Management, 27*, 1775–1794.

6. Anexos

Tabla 15 Stock seguridad y coste stock para el modelo de la empresa. Fuente: elaboración propia.

Modelo					Empresa	
sku	Plazo máximo entrega	Plazo habitual entrega	Precio Medio	Pedido medio	Stock Seguridad	Coste Stock
1	18	2	1,37	31	496	33,98
2	28	4	1,37	36	864	59,18
3	28	2	1,37	23	598	40,96
4	14	2	1,37	17	204	13,97
5	14	4	1,37	26	260	17,81
6	11	2	1,37	24	216	14,80
7	28	2	1,62	18	468	37,91
8	8	4	1,92	16	64	6,14
9	7	2	1,08	30	150	8,10
10	5	6	48,43	12	0	0,00
11	7	4	1,92	19	57	5,47
12	14	4	1,92	19	190	18,24
13	11	2	1,62	12	108	8,75
14	15	2	1,92	22	286	27,46
15	8	4	1,37	26	104	7,12
16	18	2	1,97	13	208	20,49
17	28	4	1,37	10	240	16,44
18	18	2	1,92	15	240	23,04
19	7	2	48,43	10	50	121,07
20	28	2	1,08	20	520	28,08
21	18	2	1,08	19	304	16,42
22	8	4	1,08	10	40	2,16
23	6	2	1,08	15	60	3,24
24	6	2	1,37	17	68	4,66
25	28	4	1,92	16	384	36,87
26	15	4	1,37	14	154	10,55
27	7	4	1,92	17	51	4,90
28	14	4	1,92	18	180	17,28
29	14	4	1,71	14	140	11,97
30	14	2	1,37	13	156	10,69
31	11	2	1,71	13	117	10,00
32	8	4	24,10	17	68	81,93

Modelo					Empresa	
sku	Plazo máximo entrega	Plazo habitual entrega	Precio Medio	Pedido medio	Stock Seguridad	Coste Stock
33	28	2	1,71	10	260	22,23
34	18	2	1,71	12	192	16,42
35	7	2	1,08	12	60	3,24
36	6	2	1,92	12	48	4,61
37	1	6	48,43	10	0	0,00
38	8	4	14,29	4	16	11,43
39	8	4	1,92	15	60	5,76
40	8	4	48,43	10	40	96,86
41	11	2	1,08	20	180	9,72
42	8	4	1,92	11	44	4,22
43	28	4	1,97	9	216	21,28
44	18	2	0,54	12	192	5,18
45	8	4	5,59	17	68	19,01
46	14	4	1,37	15	150	10,28
47	28	4	1,37	23	552	37,81
48	28	2	14,29	7	182	130,03
49	28	4	1,37	12	288	19,73
50	18	2	1,08	13	208	11,23
				Total	**9.801**	**1.148,71**

Tabla 16 Stock seguridad y coste stock para modelos algoritmos ML. Fuente; elaboración propia.

Modelo					SVM		Árbol Decisión		XGBoost		Red Neuronal	
sku	Plazo máximo entrega	Plazo habitual entrega	Precio Medio	Pedido medio	Stock Segur.	Coste Stock	Stock Segur.	Coste stock	Stock Segur.	Coste stock	Stock Segur.	Coste stock
1	18	2	1,37	31	39,00	2,67	78,00	5,34	39,00	2,67	39,00	2,67
2	28	4	1,37	36	78,00	5,34	78,00	5,34	78,00	5,34	39,00	2,67
3	28	2	1,37	23	39,00	2,67	78,00	5,34	39,00	2,67	39,00	2,67
4	14	2	1,37	17	39,00	2,67	78,00	5,34	39,00	2,67	39,00	2,67
5	14	4	1,37	26	78,00	5,34	78,00	5,34	78,00	5,34	39,00	2,67
6	11	2	1,37	24	39,00	2,67	78,00	5,34	39,00	2,67	39,00	2,67
7	28	2	1,62	18	39,00	3,16	78,00	6,32	39,00	3,16	39,00	3,16
8	8	4	1,92	16	78,00	7,49	78,00	7,49	78,00	7,49	39,00	3,74
9	7	2	1,08	30	39,00	2,11	78,00	4,21	39,00	2,11	39,00	2,11
10	5	6	48,43	12	78,00	188,87	78,00	188,87	78,00	188,87	39,00	94,44
11	7	4	1,92	19	78,00	7,49	78,00	7,49	78,00	7,49	39,00	3,74
12	14	4	1,92	19	78,00	7,49	78,00	7,49	78,00	7,49	39,00	3,74
13	11	2	1,62	12	39,00	3,16	78,00	6,32	39,00	3,16	39,00	3,16
14	15	2	1,92	22	39,00	3,74	78,00	7,49	39,00	3,74	39,00	3,74
15	8	4	1,37	26	78,00	5,34	78,00	5,34	78,00	5,34	39,00	2,67
16	18	2	1,97	13	39,00	3,84	78,00	7,68	39,00	3,84	39,00	3,84
17	28	4	1,37	10	78,00	5,34	78,00	5,34	78,00	5,34	39,00	2,67
18	18	2	1,92	15	39,00	3,74	78,00	7,49	39,00	3,74	39,00	3,74
19	7	2	48,43	10	39,00	94,44	78,00	188,87	39,00	94,44	39,00	94,44
20	28	2	1,08	20	39,00	2,11	78,00	4,21	39,00	2,11	39,00	2,11
21	18	2	1,08	19	39,00	2,11	78,00	4,21	39,00	2,11	39,00	2,11
22	8	4	1,08	10	78,00	4,21	78,00	4,21	78,00	4,21	39,00	2,11
23	6	2	1,08	15	39,00	2,11	78,00	4,21	39,00	2,11	39,00	2,11
24	6	2	1,37	17	39,00	2,67	78,00	5,34	39,00	2,67	39,00	2,67
25	28	4	1,92	16	78,00	7,49	78,00	7,49	78,00	7,49	39,00	3,74
26	15	4	1,37	14	78,00	5,34	78,00	5,34	78,00	5,34	39,00	2,67
27	7	4	1,92	17	78,00	7,49	78,00	7,49	78,00	7,49	39,00	3,74
28	14	4	1,92	18	78,00	7,49	78,00	7,49	78,00	7,49	39,00	3,74
29	14	4	1,71	14	78,00	6,67	78,00	6,67	78,00	6,67	39,00	3,33
30	14	2	1,37	13	39,00	2,67	78,00	5,34	39,00	2,67	39,00	2,67
31	11	2	1,71	13	39,00	3,33	78,00	6,67	39,00	3,33	39,00	3,33
32	8	4	24,10	17	78,00	93,98	78,00	93,98	78,00	93,98	39,00	46,99
33	28	2	1,71	10	39,00	3,33	78,00	6,67	39,00	3,33	39,00	3,33
34	18	2	1,71	12	39,00	3,33	78,00	6,67	39,00	3,33	39,00	3,33

Modelo					SVM		Árbol Decisión		XGBoost		Red Neuronal	
sku	Plazo máximo entrega	Plazo habitual entrega	Precio Medio	Pedido medio	Stock Segur.	Coste Stock	Stock Segur.	Coste stock	Stock Segur.	Coste stock	Stock Segur.	Coste stock
35	7	2	1,08	12	39,00	2,11	78,00	4,21	39,00	2,11	39,00	2,11
36	6	2	1,92	12	39,00	3,74	78,00	7,49	39,00	3,74	39,00	3,74
37	1	6	48,43	10	78,00	188,87	78,00	188,87	78,00	188,87	39,00	94,44
38	8	4	14,29	4	78,00	55,73	78,00	55,73	78,00	55,73	39,00	27,86
39	8	4	1,92	15	78,00	7,49	78,00	7,49	78,00	7,49	39,00	3,74
40	8	4	48,43	10	78,00	188,87	78,00	188,87	78,00	188,87	39,00	94,44
41	11	2	1,08	20	39,00	2,11	78,00	4,21	39,00	2,11	39,00	2,11
42	8	4	1,92	11	78,00	7,49	78,00	7,49	78,00	7,49	39,00	3,74
43	28	4	1,97	9	78,00	7,68	78,00	7,68	78,00	7,68	39,00	3,84
44	18	2	0,54	12	39,00	1,05	78,00	2,11	39,00	1,05	39,00	1,05
45	8	4	5,59	17	78,00	21,80	78,00	21,80	78,00	21,80	39,00	10,90
46	14	4	1,37	15	78,00	5,34	78,00	5,34	78,00	5,34	39,00	2,67
47	28	4	1,37	23	78,00	5,34	78,00	5,34	78,00	5,34	39,00	2,67
48	28	2	14,29	7	39,00	27,86	78,00	55,73	39,00	27,86	39,00	27,86
49	28	4	1,37	12	78,00	5,34	78,00	5,34	78,00	5,34	39,00	2,67
50	18	2	1,08	13	39,00	2,11	78,00	4,21	39,00	2,11	39,00	2,11
				Total	**2.925,00**	**1.044,87**	**3.900,00**	**1.230,39**	**2.925,00**	**1.044,87**	**1.950,00**	**615,20**