UNIVERSIDAD EAFIT

MAESTRÍA EN CIENCIA DE DATOS Y ANALÍTICA

Equipo de trabajo:

Alba Miriam Gómez Arango
Angie Karina Barrera Ravelo
Jose Jorge Muñoz Mercado
Juan Felipe Burbano Zapata
Francisco Javier Moya Ortiz

Proyecto Integrador 2022-II

Estimación de ventas diarias para las categorías hobbies, alimentos y productos para el hogar en las tiendas de Walmart ubicadas en el estado de California

Noviembre 2022

Semestre II

UNIVERSIDAD EAFIT

Tabla de contenido

UNIVERSIDAD EAFIT

Tabla de ilustraciones

Tabla de gráficos

Índice de tablas

1. INTRODUCCIÓN

Planteamiento del problema

Desde la estadística clásica, existen los modelos SARIMA para el pronóstico de datos temporales, y en el caso particular de pronóstico de ventas es el modelo predominante. Sin embargo, desde la inteligencia artificial existen algoritmos que modelan el comportamiento de datos secuenciales. En este sentido, la propuesta de análisis del presente proyecto es determinar si para el caso particular de la predicción de ventas en las tiendas Walmart del estado de California se pueden predecir mejores resultados mediante modelos de Machine Learning o Deep Learning que con el modelo estadístico clásico.

Justificación:

Las técnicas de pronósticos tienen como objetivo general obtener un valor futuro bajo la incertidumbre, utilizando la información histórica como base y tratando de encontrar un patrón de comportamiento que permita predecir cómo será el comportamiento en el futuro. Esto puede ayudar a las empresas a anticipar el comportamiento del consumidor, tomar decisiones estratégicas, gestionar de forma inteligente los recursos, realizar inversiones inteligentes, evaluar el rendimiento de los vendedores, asignar recursos de manera eficiente e identificar y solucionar rápidamente los problemas potenciales. Además, los pronósticos pueden ser útiles para estimar las ventas futuras y planificar la producción y el inventario en consecuencia.

Objetivos

Objetivo general:

Comparar técnicas de predicción con series de tiempo para pronosticar las ventas en dólares de las tiendas Walmart en el Estado de California, Estados Unidos

Objetivos específicos:

- Explorar y analizar el conjunto de datos de las tiendas Walmart para evaluar la coherencia y consistencia con los objetivos de negocio y el problema de analítica a resolver.
- Establecer cuáles técnicas de analítica, vistas en el semestre, se pueden usar, para resolver problemas de predicción de series de tiempo.

- Preprocesar y preparar el conjunto de datos para la implementación de los modelos de series de tiempo seleccionados.
- Construir los modelos de serie de tiempo con el conjunto de datos que resuelva el problema de pronóstico de ventas en las tiendas Walmart del Estado de California
- Evaluar los resultados de los modelos de series de tiempo seleccionados y comparar las medidas de calidad.

2. MARCO TEÓRICO Y ESTADO DEL ARTE

2.1. Técnicas de analítica de datos para series de tiempo

2.1.1. SARIMA

El modelo SARIMA (Seasonal Autoregressive Integrated Moving Average) es una técnica de pronóstico utilizada para analizar y predecir datos temporales. Este es una extensión del modelo autorregresivo integrado de media móvil (ARIMA) que incluye términos de tendencia estacional para capturar la periodicidad en los datos [1]. Este modelo se basa en el análisis de las tendencias y patrones del pasado para predecir el comportamiento futuro, y se utiliza en diversos campos como la economía, la meteorología, la ingeniería, entre otros. Así como otros métodos tradicionales de análisis de series temporales, SARIMA se construye con la combinación de uno o más de los siguientes componentes [10]:

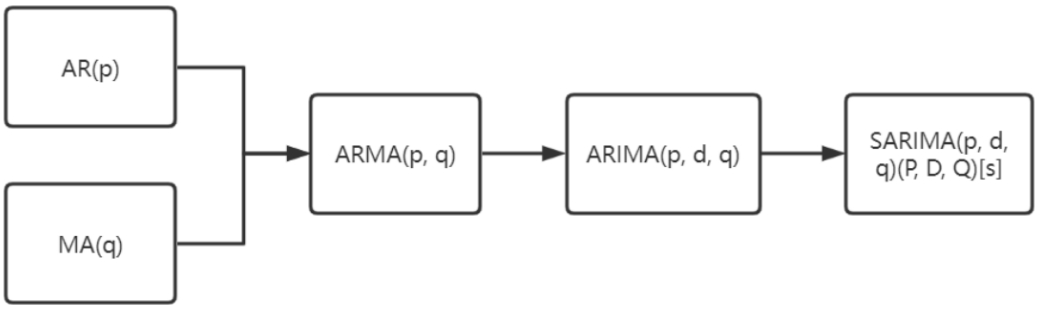
- **Componente estacional (S):** donde S determina el número de períodos de tiempo hasta que el patrón se repite nuevamente. Es decir, considera los comportamientos cíclicos de las series temporales.
- **Componente autorregresivo (AR):** en este caso, el modelo de la serie de tiempo es la representación de un proceso aleatorio, en el que la variable a pronosticar depende de sus observaciones pasadas. En particular, si una serie de tiempo está representada por un modelo AR(4), indica que el pronóstico de un dato depende de las cuatro(4) observaciones anteriores de la serie.
- **Componente de integración:** Para que los datos secuenciales se puedan modelar mediante modelos estadísticos de series de tiempo, se requiere que esta sea estacionaria; es decir; que la media de los datos sea igual a cero (no presente ningún tipo de tendencia) y que la varianza sea constante. En los casos donde los datos no cumplen estas dos condiciones, se puede

realizar un proceso de diferenciación de la serie. Que básicamente consiste en consiste en calcular la diferencia entre cada dato de la serie y el anterior.

- **Componente de media móvil (MA):** Este tipo de modelo se centra en la acumulación de términos de error en el modelo autorregresivo. Por ejemplo, en un modelo de tipo MA(2) indica que el valor actual de lo observado es la acumulación de ruido blanco de las dos últimas observaciones.

A modo de resumen se presenta en la siguiente imagen las tipologías de modelos de series de tiempo tradicionales.

Ilustración 1. Tipologías de modelos SARIMA



Fuente: tomado de [10]

2.2.1. Regresión

La regresión es parte de las técnicas de Machine Learning supervisado, con esta se quiere encontrar una función que recoja la relación que hay entre la variable objetivo y conjunto de características [2]. El ejemplo más sencillo de un algoritmo de regresión es la regresión lineal, en la que se asume una relación lineal entre los predictores y la variable objetivo y se construye una línea media que se utiliza para predecir el valor de la variable objetivo.

Tradicionalmente en Machine Learning, las características (predictores) usadas en la regresión son estáticas y no se tienen en cuenta relaciones con el tiempo, por lo que no se espera que cambien sus valores, por ejemplo, la predicción del valor de una vivienda usando predictores como: el número de cuartos, el número de baños, metros cuadrados, accesibilidad y polución. En estos casos se usan algoritmos de Machine Learning como Random Forest [3], XGBoost [4] incluso regresión lineal. Para los métodos tradicionales de series de tiempo, la regresión es usada para ajustar modelos autoregresivos AR [5] con estos se crean funciones matemáticas con las que se modelan los valores recientes y/o estacionalidad de la serie de tiempo, volviendo al ejemplo de la predicción del precio

de la vivienda ya no se usan variables como el número de baños y el número de habitaciones, sino que se modela usando el precio de venta anterior de la casa, pero esto es viable si se tiene el valor del precio de la vivienda periódicamente.

Existe otra forma de modelar las series de tiempo con regresión, esta requiere que los datos de la serie de tiempo se traten de manera tabular, aquí los valores pasados son las características (predictores) de valores futuros. Se transforma la serie de tiempo de un vector largo de dimensión L x 1 a una matriz L x C, donde D es el largo de la serie de tiempo y C es la cantidad de características seleccionadas de la serie de tiempo. La forma de seleccionar estas características es muy variada, Kang et al. [-], proponen usar los valores pasados de la serie, estadísticas de la serie mensuales, trimestrales, semestrales, anuales, tendencia, estacionalidad, entre otras. Fulcher et al. [6] introduce el algoritmo Highly Comparative Time Series Analysis (HCTSA) con el que se pueden obtener hasta 7000 características de la serie de tiempo y Lubba et al. [7] introduce el algoritmo The Canonical Time Series Characteristics (Catch22) que es una mejora de HCTSA, con el cual reduce el número de características a 22, las más discriminantes.

Una vez seleccionadas las características, se procede a usar cualquier modelo de regresión de la manera usual. Algunas de las más usadas son:

- **Support vector regression (SVR)** [8]: Este algoritmo es usado para predecir valores discretos, usa el mismo principio de Support Vector Machine, se busca encontrar hiperplanos que clasifiquen/separen muy bien los puntos de datos, esto lo hace usando una función kernel que puede ser lineal o sigmoidal o polinomial. La línea que modele el hiperplano que contenga la mayor cantidad de datos es la seleccionada. En el SVR, los hiper parámetros que se deben ajustar son: el hiperplano, el kernel y las líneas límite. Esta técnica de Machine Learning es robusta a datos atípicos [9].
- **Random Forest (RF)** [3]**:** Es un algoritmo muy robusto, este es una agregación bootstrap, también conocida como bagging, la data es separada aleatoriamente con reemplazo, en cada pedazo se entrena un árbol de decisión, luego se agrega el resultado de todos los modelos con la media. Bagging ayuda a reducir la gran varianza de la predicción de cada árbol. En el random forest, los dos hiper parámetros que hay que ajustar son: el número de árboles y el número de características en cada nodo.
- **K-Nearest Neighbor (kNN):** En primer lugar, kNN es un algoritmo de clasificación y regresión no paramétrico, es decir, no hace ningunas presunciones sobre el conjunto de datos elementales. Es conocido por su simplicidad y eficacia y es un algoritmo de

aprendizaje supervisado. La predicción de una entrada se consigue identificando a los k vecinos más cercanos y utilizándolos para determinar, mediante el promedio, el valor predicho de los datos o el vector de entrada. La intuición que subyace a la clasificación por kNN es bastante sencilla: las entradas se predicen según el promedio de sus vecinos más cercanos. A menudo es útil tomar más de un vecino, por lo que la técnica se denomina más comúnmente k-vecinos más cercanos, donde los k- vecinos más cercanos se utilizan para determinar el valor de predicción de las variables independientes [9].

- **Extreme Gradient Boosting XGBoost (GB)** [4]: Similar al Random Forest XGBoost es un ensamble de árboles de decisión, pero el ensamble se realiza con gradient boosting con regularización, esto ayuda que se reduzca el sesgo en la predicción. Booting consiste en construir modelos consecutivos que se entrenan con los errores que quedan después de cada predicción.
- **Regresiones Lasso y Ridge:** Estos son unas técnicas de regularización, consisten en encoger el peso de los coeficientes del modelo, es decir penalizarlos. Cuando se usa la regularización L1 la técnica se llama Lasso (Leat Absolute Shrinkage and Selecto Operator)
  Donde, λ es el término de penalización, determina el tamaño del encogimiento que se va a realizar al modelo, cuando se elige un λ grande, las características menos importantes serán eliminadas. Cuando el valor de λ es 0, tenemos una ecuación equivalente a la de una regresión por mínimos cuadrados.
- **RF-kNN-J.J**: Es una propuesta del estudiante Jose Jorge Muñoz inicialmente para tareas de clasificación, la cual en este trabajo se ha usado con la misma metodología, pero para la tarea de regresión. Es una propuesta de un nuevo modelo de aprendizaje automático que combina los algoritmos Random Forest y k-Nearest Neighbor llamado RF-kNN, en el que Random Forest se poda, reduciendo su sobreajuste y varianza, y los parámetros de k-Nearest Neighbor se optimizan sobre las muestras de cada árbol y se ajustan sobre las muestras de las hojas de cada árbol para reducir el espacio de búsqueda. La propuesta alivia simultáneamente los problemas de RF y kNN, al tiempo que mejorando el rendimiento de estos modelos al componer un método de conjunto. Para cada instancia de entrada, la predicción se realiza como sigue: primero, aplicando la regla del árbol de RF, y luego, encontrar sus k-próximos vecinos que viven en una hoja específica para cada árbol, y por último, se vota para elegir la frecuencia de clase más alta entre todos los árboles. Para tareas de regresión la metodología es igual, cambiando la moda de los vecinos más cercanos, por el promedio de éstos.

2.2.1. LSTM

Las redes neuronales recurrentes son una clase de aprendizaje profundo basada en los trabajos de David Rumelhart en 1986. La característica principal de las redes recurrentes es que la información puede mantenerse en el modelo introduciendo bucles en el diagrama de la red, permitiendo que la serie recuerde los estados previos y utilice esta información para predecir. Esta característica las hace muy adecuadas para manejar series cronológicas con memoria de corto plazo.
Un caso particular de redes neuronales recurrentes son los modelos LSTM (Long Short Term Memory), que tienen la particularidad de poder "recordar" un dato relevante en la secuencia y de preservarlo por varios instantes de tiempo, por lo que el modelo puede tener una memoria tanto de corto plazo como también de largo plazo. Las redes LSTM logran capturar dependencia

Un modelo LSTM (Long Short-Term Memory) es un tipo de red neuronal recurrente (RNN) que puede capturar dependencias a largo plazo en los datos [10]. Las RNN son un tipo de red neuronal que procesa datos secuenciales, como series de tiempo o lenguaje natural, mediante el uso de conexiones de retroalimentación para incorporar información de pasos de tiempo anteriores en la predicción actual [11]. Esto permite que las RNN aprendan patrones temporales complejos en los datos y hagan predicciones precisas. Sin embargo, las RNN tradicionales tienen dificultades para aprender las dependencias a largo plazo porque tienden a olvidar la información antigua a medida que se procesa la información nueva [12]. Esto se conoce como el problema del "gradiente de fuga". Los modelos LSTM superan este problema mediante el uso de un tipo especial de unidad llamada "célula" que puede controlar el flujo de información y evitar que se olvide la información antigua [13].

Una celda LSTM tiene tres puertas: una puerta de entrada, una puerta de salida y una puerta de olvido [10]. Estas puertas se utilizan para controlar el flujo de información que entra y sale de la celda, así como para decidir qué información debe olvidarse. La puerta de entrada y la puerta de olvido son neuronas sigmoideas, que emiten un valor entre 0 y 1. La puerta de salida es una neurona tangente hiperbólica, que escala la salida de la celda entre -1 y 1.

La puerta de entrada controla el flujo de información hacia la celda. Toma como entrada la entrada actual al LSTM y el estado oculto anterior, y genera un valor entre 0 y 1 que representa la cantidad de la entrada actual que debe agregarse al estado de la celda. La puerta de olvido controla el flujo de información fuera de la celda. Toma como entrada la entrada actual al LSTM y el estado oculto

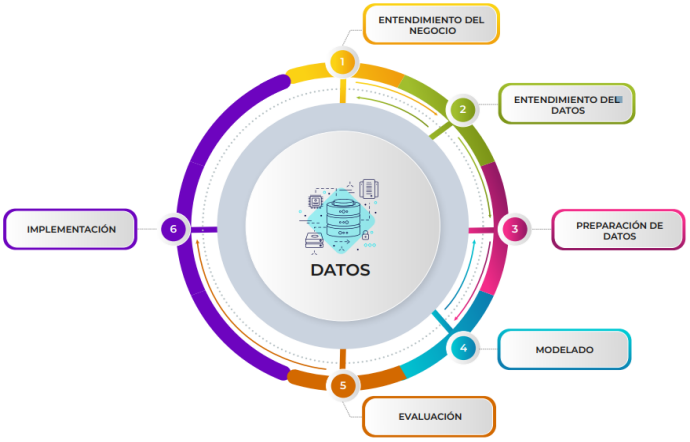
anterior, y genera un valor entre 0 y 1 que representa la cantidad del estado de celda anterior que debe olvidarse. La puerta de salida controla el flujo de información fuera de la celda y hacia la salida del LSTM. Toma como entrada la entrada actual al LSTM, el estado oculto anterior y el estado de celda actual, y genera un valor entre -1 y 1 que representa la salida del LSTM para el paso de tiempo actual.

Al controlar el flujo de información de esta manera, las células LSTM pueden aprender patrones temporales complejos en los datos y hacer predicciones precisas [10]. Los modelos LSTM se usan comúnmente en tareas de procesamiento de lenguaje natural (NLP), como la traducción de idiomas y la clasificación de textos [14], así como en la previsión de series temporales y otras tareas de predicción.

2.4. Metodología CRISP-DM (Cross Industry Standard Process for Data Mining)

CRISP-DM es quizás la guía de referencia más ampliamente utilizadas en proyectos de analítica en el mundo y particularmente en minería de datos. Esta metodología considera las diferentes fases (ver Ilustración 1) de un proyecto de este tipo e incluye la descripción de cada una de las tareas requeridas para el alcance de los objetivos de la fase, todo este proceso es considerado como el ciclo de vida del proyecto (IBM, 2022).

Ilustración 2. Fases empleadas en la metodología CRISP-DM (Adaptado de (IBM, 2022).



Fuente: Adaptado de (IBM, 2022).

A continuación, se describen de manera general cada una de las fases y tareas requeridas en cada una de las fases empleadas en la metodología:

Fase 1. Comprensión del negocio o problema:

Considerada quizás la fase más importante del ciclo de vida, contempla la ejecución de las siguientes tareas:

- **Determinar los objetivos del negocio:** Implica determinar cuál es el problema que pretende resolver, las necesidades propias de los procedimientos a utilizar y definir los criterios de éxito.
- **Evaluación de la situación:** Contempla la valoración de conocimiento específico del problema y la necesidad de datos e información adicional que se requiere.
- **Determinar los objetivos:** definición de las metas
- **Realizar un plan del trabajo para el proyecto:** Pasos a seguir y técnicas a utilizar.

Fase 2. Comprensión de los datos

En esta fase se consideran (entre otras) la ejecución de las siguientes tareas:

- **Recolección de datos iniciales:** Localización de los datos, técnicas para acceder a ellos, inconvenientes y posibles soluciones a estos.
- **Descripción de los datos:** Establece los volúmenes de datos existentes, formatos, descripción de los campos, identificación de los registros, etc.
- **Exploración de los datos**: Busca encontrar la estructura general de los datos a través de estadísticas básicas, tablas de frecuencia, gráficos de distribución entre otros.
- **Verificación de la calidad de los datos:** Tiene como objetivo asegurar la completitud y corrección de los datos, verificando la consistencia de los valores individuales de los campos, la cantidad y distribución de los valores nulos, y para encontrar valores fuera de rango.

Fase 3. Preparación de los datos

Contempla el desarrollo de las siguientes tareas:

- **Seleccionar los datos:** Identificación del conjunto de datos con el que se trabajará de acuerdo con la verificación de calidad de los datos en la fase previa.
- **Limpiar los datos:** Es quizás una de las tareas que más demanda tiempo por la cantidad de técnicas y herramientas de posible uso, tales como: estandarización de los datos, discretización de campos numéricos, tratamiento de valores faltantes, reducción dimensionalidad, entre otras.
- **Estructurar los datos:** Considera la generación de variables o atributos a partir de los ya existentes, transformaciones de variables

- **Integrar los datos:** Generación de nuevos registros, fusión de tablas.
- **Formato de los datos**: Considera a transformación de datos sin modificación de los significados originales, a modo de ejemplo se listan los siguientes: eliminar comas, caracteres especiales entre otros.

Fase 4. Modelado

Contempla el desarrollo de las siguientes tareas:

- **Seleccionar las técnicas de modelado:** Considerando el tipo de problema, los objetivos y las herramientas existentes, tales como clasificación, regresión, visualización, etc.
- **Plan de pruebas:** Probar la calidad y validez de los modelos
- **Construir el modelo:** Ejecución de los modelos seleccionados de forma iterativa.
- **Evaluar el modelo:** Considerando las métricas definidas en el plan de pruebas.

Fase 5. Evaluación

Contempla el desarrollo de las siguientes tareas:

- **Evaluar el resultado:** Evaluación de los resultados en función de las necesidades del negocio
- **Revisión del proceso:** Calificación de todo el proceso desarrollado para identificar oportunidades de mejora.
- **Determinar próximos pasos**

Fase 6. Implementación

Contempla el desarrollo de las siguientes tareas:

- **Plan de implementación**
- **Monitoreo**
- **Retroalimentación del proceso:** En este punto se evalúa qué fue lo correcto y qué lo incorrecto, qué es lo que se hizo bien y qué es lo que se requiere mejorar.

3. DESARROLLO METODOLÓGICO

3.1. Comprensión del negocio

El Centro Abierto de Pronósticos Makridakis (MOFC) de la Universidad de Nicosia lleva a cabo investigaciones de pronósticos comerciales. En este ejercicio específicamente el reto consiste en predecir las ventas diarias en unidades correspondiente a las categorías de alimentos (FOODS), pasatiempos (HOBBIES) en las tiendas de Walmart, ubicadas en tres estados de los Estados Unidos.

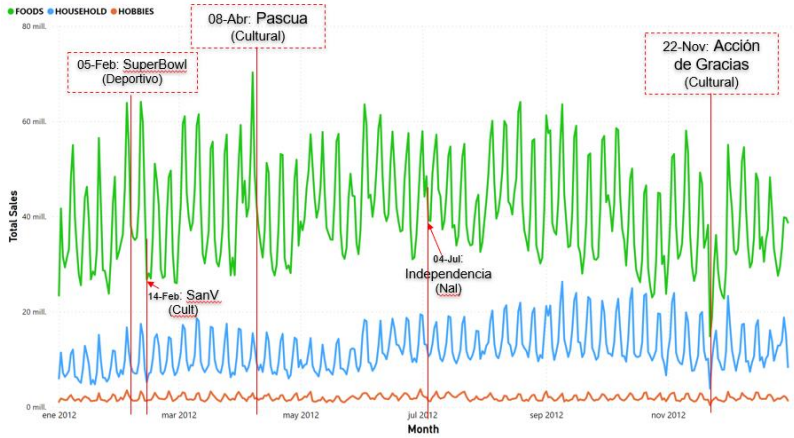
Para el proyecto en particular trabajaremos con los datos correspondientes al estado de California y se estimará el valor total en ventas diarias por cada una de las categorías, con el objetivo de construir una herramienta para la toma de decisiones financieras en el estado, y la planificación de las compras al interior de la compañía.

3.2. Comprensión de los datos

Los datos para utilizar en el análisis incluyen información a nivel de artículo, departamento, categorías de productos y detalles de la tienda. Se cuenta además con variables explicativas como precio, promociones, día de la semana y eventos especiales.
Juntos, este sólido conjunto de datos se puede utilizar para mejorar la precisión de los pronósticos.

A continuación, se puede observar un gráfico ilustrativo del comportamiento de cada una de las series en el año 2012, y el impacto de eventos relevantes en el estado que pueden afectar el comportamiento de las ventas en las tiendas.

Gráfica 1. Comportamiento de las ventas diarias por categorías para el año 2012



Fuente: Elaboración propia

3.3. Preparación de los datos y selección de las características

Con la finalidad de preparar el conjunto de datos para el procesamiento se realizan los siguientes procesos:

1. Eliminar ventas totales nulas,
2. Integración de las tablas mediante join,
3. Filtrado de información
4. Agrupamientos de productos por categorías
5. Creación de la variable respuesta en la que se multiplica el precio de venta de los artículos por la respectiva cantidad vendida (Price*Value).

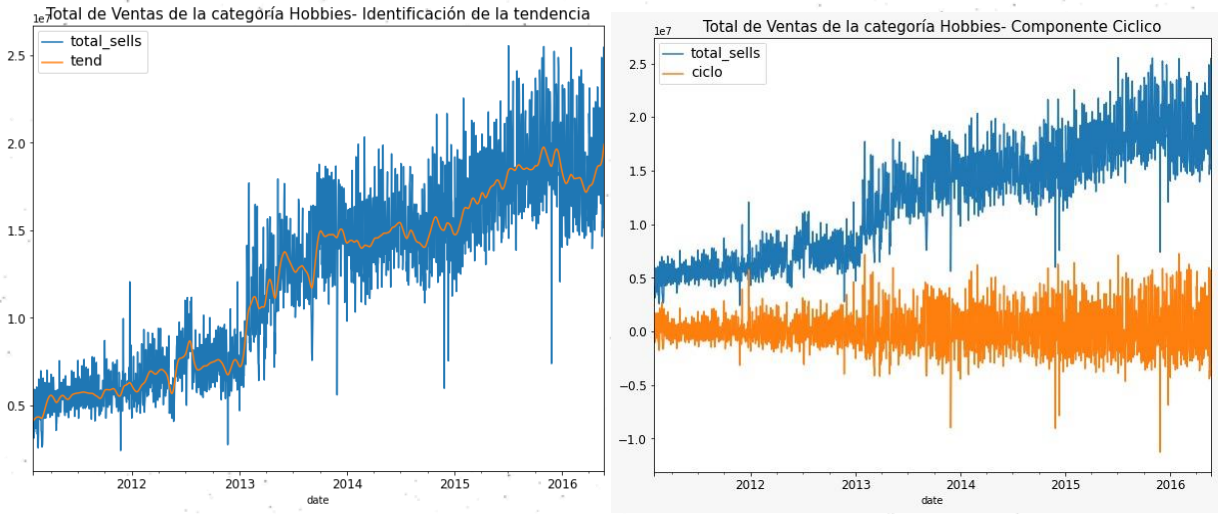
Finalmente, el DataFrame quedará con 5.808 registros correspondiente a la cantidad en dólares vendida.
Las categorías Hobbies y Foods tendrán 1.941 cada una.

3.3.1. Preparación de los datos Modelo SARIMA

A continuación, se presenta; a modo ejemplo; para la categoría de Hobbies cada uno de los pasos realizados en la preparación de los datos previo a la modelación de la serie a través de modelos estadísticos tradicionales. Estos mismos pasos fueron implementados para las categorías Foods y Houshold y pueden ser consultados en el enlace del repositorio.

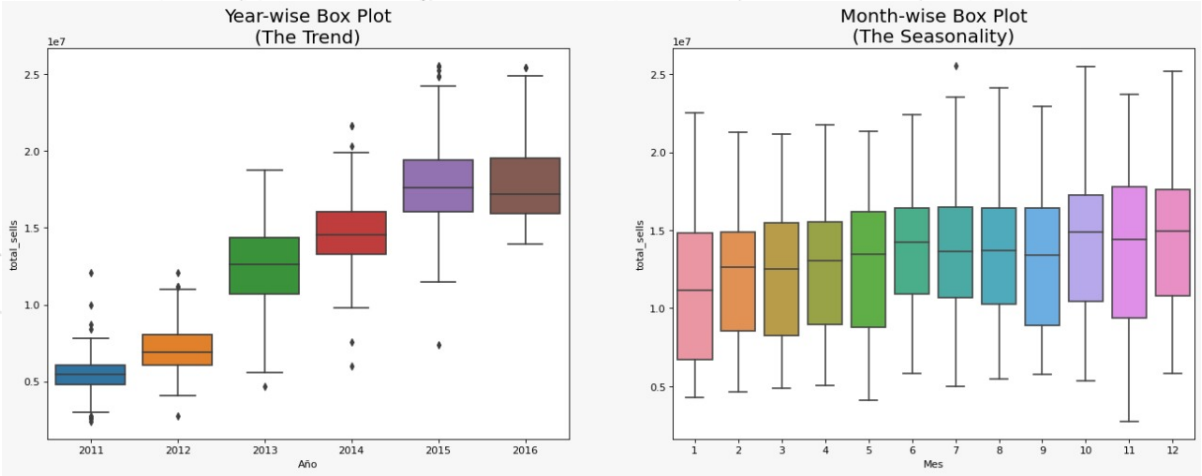
1. **Análisis gráfico:** Los modelos estadísticos clásicos para el modelamiento de series de tiempo requieren que estas cumplan las propiedades de estacionaridad, es decir, que tanto la media como la varianza sean constantes en el tiempo. Con el objetivo de validar dichas propiedades se construye el gráfico de las ventas de la categoría Hobbies utilizando los datos originales, como se puede observar en la Gráfica 1.

Gráfica 2. Comportamiento de las ventas diarias de la categoría Hobbies



Fuente: Elaboración propia

Gráfica 3. Comportamiento de las ventas semanales y mensuales correspondiente a la categoría Hobbies



Fuente: Elaboración propia

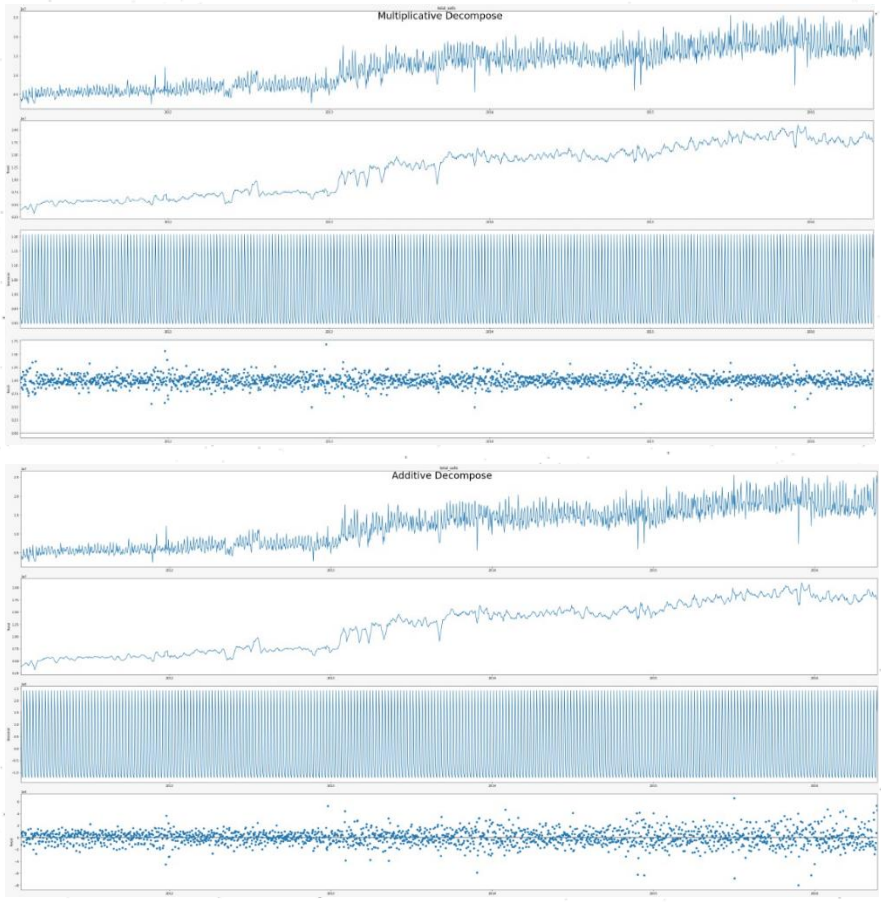
Gráficamente se puede observar comportamiento de tendencia creciente no lineal a lo largo del tiempo, varianza no constante, patrones cíclicos o fluctuaciones recurrentes en periodos anuales y la presencia de outliers que indica necesidad de realizar transformaciones que garanticen el correcto modelamiento de los datos.

2. **Descomposición de la serie en Aditiva y Multiplicativa:** Una serie de tiempo puede presentar diferentes patrones tanto en el comportamiento de la tendencia como de la

estacionalidad y sus residuales con ello puede variar el modelamiento de esta bien sea aditiva (sumatoria de sus componentes) o multiplicativa (expresada como el producto).

A continuación, evaluaremos los componentes de manera individual para determinar la tipología en el caso de las ventas de artículos de la categoría Hobbies en las tiendas Walmart de California.

Gráfica 4. Descomposición de las ventas diarias de la categoría Hobbies en sus componentes



Fuente: Elaboración propia

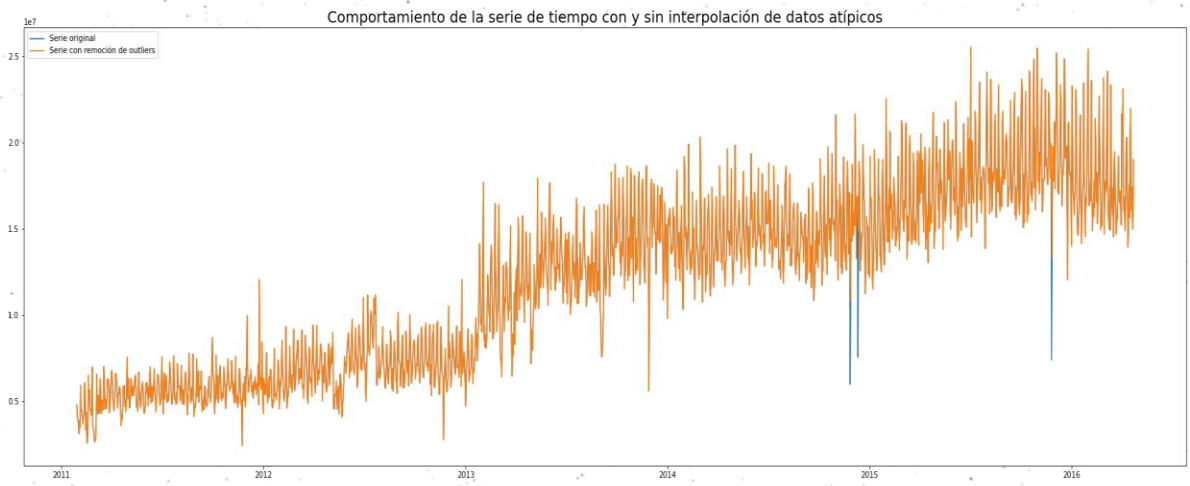
Al observar los residuos de ambas descomposiciones se puede notar una mayor aleatoriedad en los datos utilizando la descomposición multiplicativa, mientras que en el caso de la descomposición aditiva se nota una mayor dispersión y patrones no modelados. Por lo tanto, en el análisis de la categoría Hobbies se recomienda la descomposición multiplicativa.

Considerando el análisis anterior, la serie de tiempo correspondiente a las ventas de la categoría Hobbies en las tiendas de Walmart puede ser expresada el producto de sus componentes, como se observa a continuación:

$$Y_t = T_t * C_t * E_t$$

3. **Tratamiento de Outliers**: Como se mencionó previamente, la serie de tiempo presenta datos atípicos que deben ser interpolados de manera que la predicción de la serie de tiempo disminuya el error. Para solucionar este inconveniente se propone reemplazar lo outliers utilizando el promedio entre las fechas vecinas más cercanos con ayuda de la librería kats de Python

Gráfica 5. Detección de datos atípicos de las ventas diarias de la categoría Hobbies



Fuente: Elaboración propia

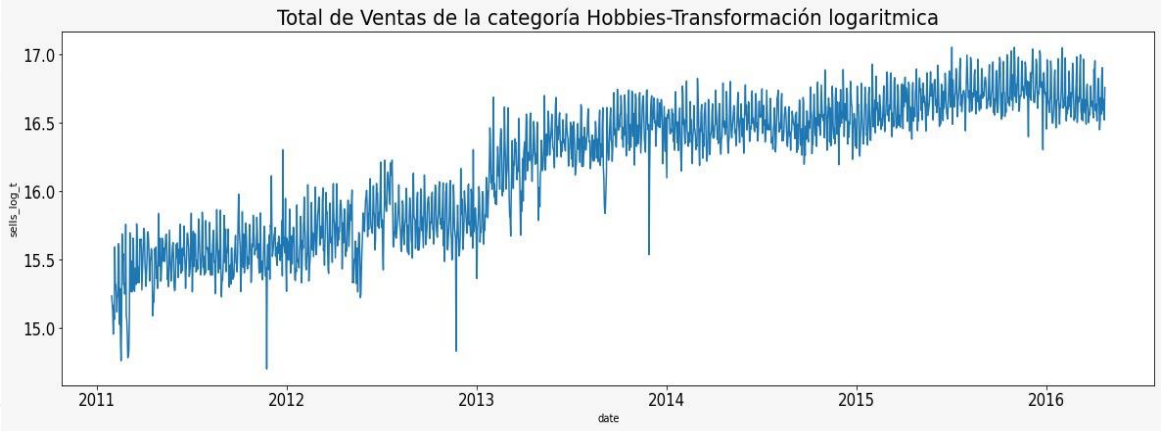
Gráficamente se puede observar que al realizar la imputación de los datos mantiene la misma estructura, pero disminuyendo el efecto de los datos atípicos.

4. **Transformaciones a la serie:** Desde la gráfica inicial de la serie habíamos identificado que presenta tendencia creciente y que su varianza tiene comportamientos diferenciales en ciertos momentos. Por lo tanto, la serie no es estacionaria, luego debemos realizar transformaciones

que la lleven a cumplir las propiedades y validar ese comportamiento mediante la prueba analítica de Dickey-Fuller cómo se puede observar a continuación:

- **Para la estabilización de la varianza haremos uso de la transformación logarítmica**

Gráfica 6. Transformación logarítmica de las ventas diarias de la categoría Hobbies

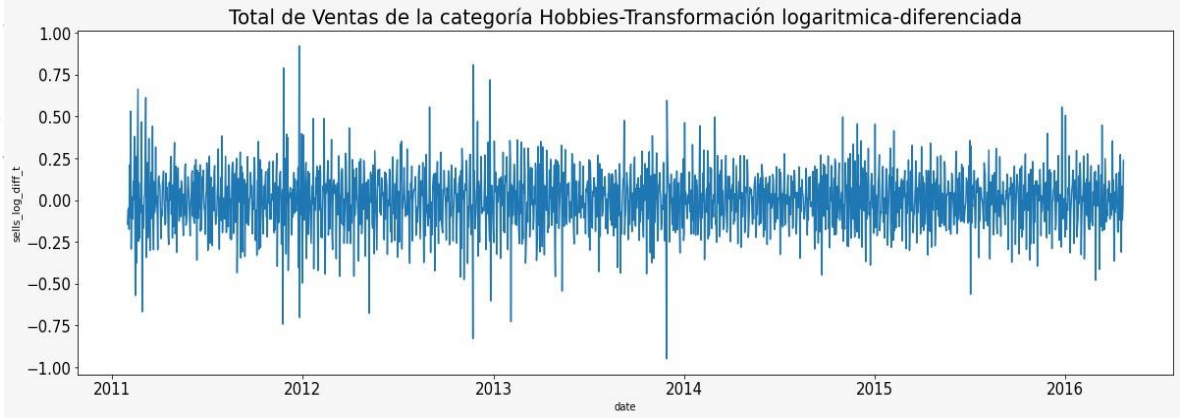


Fuente: Elaboración propia

Si bien se observan algunos picos en la serie logarítmica se observa una varianza más estable.

- **Para la estabilización en media aplicaremos diferenciación:** La transformación que elimina la tendencia de modo que pueda garantizar la estacionaridad en media es la diferenciación, que se define como la resta del valor actual con el valor anterior

Gráfica 7. Serie logarítmica diferenciada de las ventas diarias de la categoría Hobbies



Fuente: Elaboración propia

De la gráfica anterior se puede observar una oscilación de la serie alrededor del cero por lo que se considera que la primera diferencia logra eliminar la tendencia de la serie de ventas para la categoría Hobbies.

5. **Test de Dicky Fuller:** es una prueba de raíz única que detecta estadísticamente la presencia de conducta tendencial estocástica en las series temporales de las variables mediante un contraste de hipótesis que se presenta a continuación. Busca determinar la existencia o no estacionariedad.

Prueba de Hipótesis

- **Hipótesis nula:** H0: La media de los datos es no estacionaria, p-value =>0.05
- **Hipótesis alternativa:** H1: La media de los datos es estacionaria, p-value <0.05

Gráfica 8. Salida test ADF para la serie logarítmica diferenciada de las ventas diarias de la categoría Hobbies

Test ADF para prueba de estacionariedad

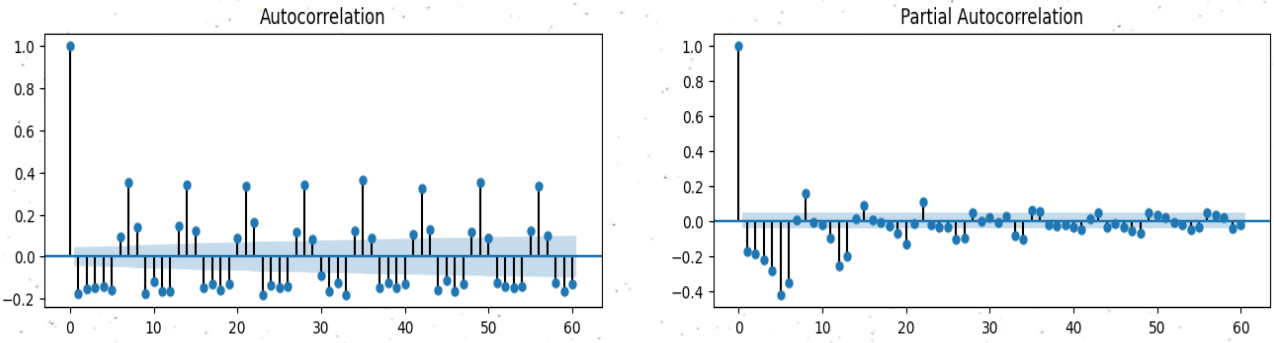
```
[92] test_results = adfuller(train["sells_log_diff_t"])

     print(f"ADF test statistic: {test_results[0]}")
     print(f"p-value: {test_results[1]}")
     print("Critical thresholds:")

     for key, value in test_results[4].items():
        print(f"\t{key}: {value}")

     ADF test statistic: -13.911464186853042
     p-value: 5.500614844971446e-26
     Critical thresholds:
             1%: -3.433827555443887
             5%: -2.8630761443612065
             10%: -2.5675877863879206
```

Fuente: Elaboración propia

6. **Correlogramas:** Mediante las gráficas de autocorrelación y autocorrelación parcial se puede analizar el orden de la serie; es decir; los componentes autorregresivos y/o de media móvil de los datos analizados. Específicamente, la función de autocorrelación (ACF), muestra la correlación entre las ventas actuales versus las versiones retrasadas de sí misma. Si una serie está significativamente auto correlacionada, significa que los valores anteriores de la serie pueden ser útiles para predecir el valor actual; mientras que la función de autocorrelación parcial (PACF) muestra la correlación entre las ventas actuales versus las versiones retrasadas excluyendo los efectos de los retrasos anteriores

A continuación, se presentan los correlogramas de la serie de ventas de la categoría Hobbies

Gráfica 9. Correlogramas de la serie logarítmica diferenciada de las ventas diarias de la categoría Hobbies



Fuente: Elaboración propia

De los gráficos anteriores se puede observar desde el componente autorregresivo un comportamiento secuencial con seis rezagos significativos y un componente cíclico marcado en el comportamiento semanal.

3.2.2. Preparación de los datos modelación Machine Learning

1. Separación en dos sets de datos uno con cada categoría: Foods y Hobbies.
2. La variable por predecir es las ventas en dólares en cada día
3. Se crean las variables año y mes a partir de la fecha, como variables ordinales
4. Se crean variables rezago desde un día antes hasta 30 días antes, correspondientes a las ventas del día rezago respecto al día actual.
5. Se crean variables rezago correspondientes al máximo, mínimo, media, desviación estándar y curtosis de 30 días antes, 60 días antes y 180 días antes de las ventas del día actual
6. Se eliminan los datos que no tienen la información anterior completa después de la creación de variables del paso 4 y 5
7. Se escalan los valores de las variables anteriores con el escalado:

$$X_{escalada} = X_{std} * (max - min) + min$$

$$con \quad X_{std} = \frac{X - X_{min}}{X_{max} - X_{min}},$$ min y max el rango de la variable

8. Se seleccionan las variables que entran al modelo de tres maneras:

- **Coeficiente de correlación con la variable a predecir:**

El coeficiente de correlación de Pearson es una variable de dependencia lineal entre dos variables aleatorias continuas. Esta medida es independiente de la escala de las variables
Las variables elegidas según correlación de Pearson (mayor a 0.7) son:

Tabla 1. Variables set de datos para la categoría Hobbies

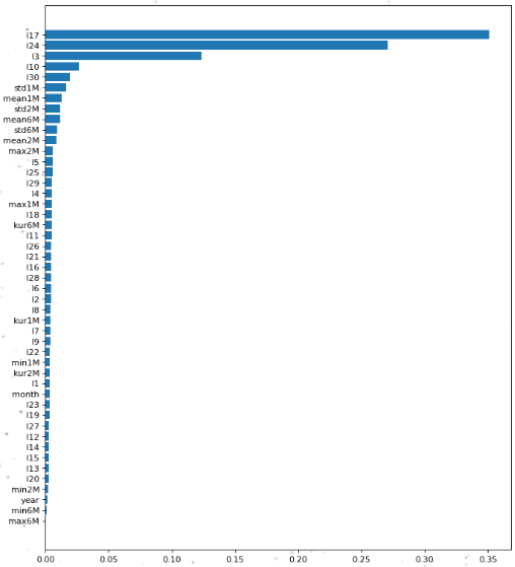
Variables del set hobbies	Correlación Pearson con ventas_Totales
l24 (rezago 7)	0.861
l17(rezago 14)	0.853
l3(rezago 28)	0.843

l10(rezago 21)	0.839
mean1M	0.765
mean2M	0.755
l30(rezago 1)	0.748
max6M	0.744
max1M	0.742
mean6M	0.740
max2M	0.733
std6M	0.725
l18(rezago 13)	0.723
l4(rezago 17)	0.719
l16(rezago 15)	0.719
l11(rezago 20)	0.719
l23(rezago 8)	0.718
l25(rezago 6)	0.717
l2(rezago 29)	0.710
year	0.710
l9(rezago 22)	0.709

Fuente: Construcción propia

- **Importancia en modelo Random Forest:**

Gráfica 10. Importancia en el modelo Random Forest (hobbies)



Fuente: Elaboración propia

Se eligen 9 variables: L17(rezago 14), L24(rezago 24), L3(rezago 28), L10 (rezago 21), L30 (rezago 1), std1M (desviación estándar últimos 30 días), mean1M (media últimos 30 días), std2M (desviación estándar últimos 60 días), mean6M (media últimos 6 meses). Estas 9 variables son las utilizadas en el proceso interno de divisiones del Random Forest. Estas son las variables más relevantes, el 84.4% de los árboles con los que se hizo la selección de variables tenían al menos una de estas 9 variables.

- **Selección de variables recursiva**

Dado un estimador externo que asigna pesos a las variables, el objetivo de la eliminación recursiva de variables es seleccionar variables considerando recursivamente conjuntos de características más pequeños. Primero, el estimador se entrena con el conjunto de características inicial y la importancia de cada característica se obtiene a través de una medida de ajuste/ importancia. Luego las características menos importantes se eliminan del conjunto actual de características [15]. Este proceso se repite recursivamente en el conjunto podado hasta que finalmente se alcanza el número deseado de características por seleccionar. Para el ejercicio planteado en este trabajo, el número de variables a seleccionar es 7. Con el algoritmo Random Forest Regressor.

Las variables seleccionadas para set de datos 'hobbies', por este método son:
L3(rezago 28), L10(rezago 21), l17(rezago 14), l24(rezago 7), l30(rezago 1), mean1M (media últimos 30 días), mea6M (media últimos 6 meses)

- **Selección hacia atrás (Backward selection)**

Este selector secuencial de características elimina (selección hacia atrás) características para formar un conjunto de variables más parsimonioso y que ayude a la generalización del modelo. En cada etapa, este estimador elige la mejor característica para agregar en función de la puntuación de validación cruzada de un estimador, para un caso como el de este estudio, el supervisado. Este selector de características secuenciales observa solo las características, no las salidas deseadas de la variable a predecir [16]. El estimador usado para la selección hacia atrás fue Regresión Lasso con puntuación basada en coeficiente de determinación.

Para el caso de la serie hobbies tenemos como características seleccionadas: L3(rezago 28), L4(rezago 27), L8(rezago 23), L9(rezago 20), L10(rezago 21), L11(rezago 18), L12(rezago 19), L14(rezago 17), L28(rezago 3), L30(rezago 1), L17(rezago 14), L18(rezago 13), L20(rezago 11),

L21(rezago 10), L22(rezago 9), L24(rezago 7), max1M (máximo valor del último meses), max2M (máximo valor de los últimos 2 meses), max6M (máximo valor de los últimos 6 meses), min1M (mínimo valor del último meses), min6M (mínimo valor de los últimos 6 meses), mean1M (media último meses)), year (variable ordinal que representa el año)

9. Variables seleccionadas

Se eligen 9 variables: L17(rezago 14), L24(rezago 24), L3(rezago 28), L10 (rezago 21), L30 (rezago 1), std1M (desviación estándar últimos 30 días), mean1M (media últimos 30 días), std2M (desviación estándar últimos 60 días), mean6M (media últimos 6 meses). Estas 9 variables son que concuerdan en los métodos que se probaron en el apartado anterior, con ellas se alcanza un set de características parsimonioso y con gran explicabilidad de la variable a predecir.

3.2.3. Preparación de los datos modelación Deep Learning

En el contexto de Deep Learning se realiza una separación similar al tratamiento en los modelos de machine Learning, utilizando 3 conjuntos: entrenamiento, validación y pruebas. Esto con el fin de asegurar la optimización de los parámetros de los modelos mientras se controla el sobreajuste a los datos. Además, se realizó la estandarización de los datos de manera similar al tratamiento de los datos en machine Learning y se transformaron los datos de series de tiempo a manera que el esquema se asemeje a una tarea de aprendizaje supervisado, siendo las etiquetas la predicción y las características los rezagos determinados. Finalmente, se utilizó la diferenciación de los datos con el fin de facilitar el aprendizaje de la red neuronal.

3.4. Modelado

3.4.1. Modelos SARIMA

Para la implementación de los modelos SARIMA se sigue el siguiente proceso:

1. **Partición de los datos:** Para el entrenamiento del modelo se propone particionar los datos considerando 1.911 registros para entrenar el modelo; fechas comprendidas entre el 29 de enero del 2011 y el 07 de abril de 2016 y se dejan los últimos 30 días (hasta el 22 de mayo de 2016) para el testeo.

2. **Modelo propuesto:** Considerando los resultados de las fases de exploración, entendimiento y preparación de los datos se propone el evaluar el modelo SARIMA (6,1,0)(1,0,0,52)

haciendo uso de la librería Statmodels de Python. Los resultados se pueden observar a continuación:

Gráfica 11. Resultados modelo SARIMA (6,1,0)(1,0,0,52)

SARIMAX Results

Dep. Variable:	sells_log_t	No. Observations:	1910
Model:	SARIMAX(6, 1, 0)x(1, 0, 0, 52)	Log Likelihood	1063.233
Date:	Fri, 02 Dec 2022	AIC	-2110.465
Time:	02:45:38	BIC	-2066.031
Sample:	01-30-2011 - 04-22-2016	HQIC	-2094.111
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.5886	0.018	-33.012	0.000	-0.624	-0.554
ar.L2	-0.5783	0.020	-29.452	0.000	-0.617	-0.540
ar.L3	-0.5748	0.020	-28.237	0.000	-0.615	-0.535
ar.L4	-0.5688	0.020	-28.848	0.000	-0.607	-0.530
ar.L5	-0.5749	0.018	-31.442	0.000	-0.611	-0.539
ar.L6	-0.3512	0.016	-22.428	0.000	-0.382	-0.320
ar.S.L52	-0.0168	0.024	-0.713	0.476	-0.063	0.029
sigma2	0.0192	0.000	48.828	0.000	0.018	0.020

Ljung-Box (L1) (Q):	0.00	Jarque-Bera (JB):	891.88
Prob(Q):	0.97	Prob(JB):	0.00
Heteroskedasticity (H):	0.54	Skew:	-0.20
Prob(H) (two-sided):	0.00	Kurtosis:	6.32

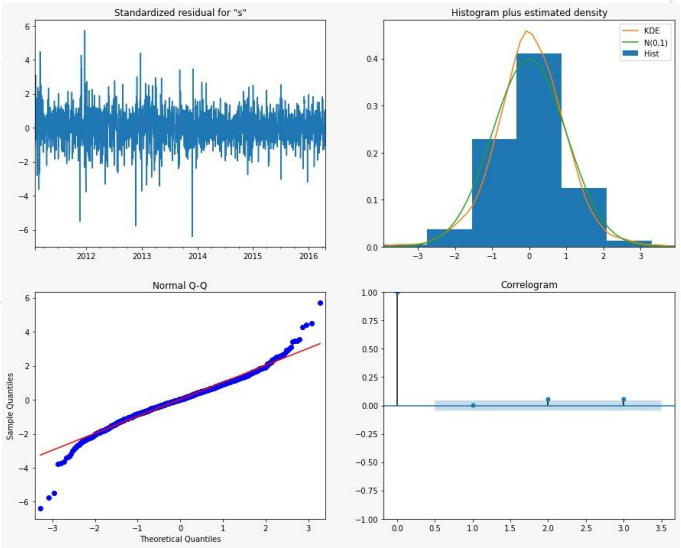
Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

Fuente: Elaboración propia

Los resultados de este modelo con S= 52 el AIC = -2110.465 y un valor de los parámetros no es significativo ar.S.L52 porque su p-value es mayor que 0.05.

Veamos a continuación, el gráfico de los residuales correspondiente al modelo propuesto:

Gráfica 12. Residuales modelo SARIMA (6,1;0)(1,0,0,52)



Fuente: Elaboración propia

Interpretando los gráficos podemos observar lo siguiente:

- **Arriba a la izquierda:** los residuos del modelo parece que siguen un proceso de Ruido Blanco (White Noise) y no son predecibles. Esto implica que nuestro modelo ha extraído toda la información de los datos.
- **Arriba a la derecha:** vemos que la distribución de los residuos sigue una distribución próxima a la Normal (0, 1).
- **Abajo a la derecha:** vemos que la autocorrelación parcial entre los residuos y residuos - k, dan lugar a valores significativos. Esto implica que "nos queda" información a extraer de los residuos, es decir el modelo ha sido capaz de reproducir el patrón de comportamiento sistemático de la serie y estaría bien formulado.
- **Abajo a la izquierda:** la distribución ordenada de los residuos sigue una Normal.

Considerando que el componente estacional propuesto (52) no resulta ser significativo en el modelo y de acuerdo con los análisis previos de los datos, se propone un modelo adicional con estacionalidad mensual como sigue:

Gráfica 13. Resultados modelo SARIMA (6,1,0)(1,0,0,12)

SARIMAX Results

Dep. Variable:	sells_log_t	No. Observations:	1910
Model:	SARIMAX(6, 1, 0)x(1, 0, 0, 12)	Log Likelihood	1068.218
Date:	Fri, 02 Dec 2022	AIC	-2120.437
Time:	03:36:23	BIC	-2076.002
Sample:	01-30-2011 - 04-22-2016	HQIC	-2104.083
Covariance Type:	opg		

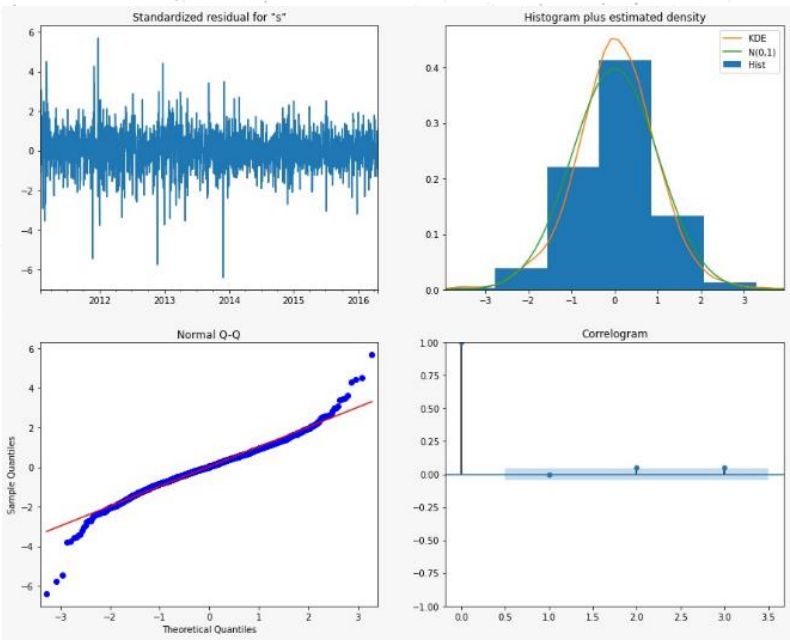
	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.5850	0.018	-32.744	0.000	-0.620	-0.550
ar.L2	-0.5641	0.020	-28.473	0.000	-0.603	-0.525
ar.L3	-0.5729	0.020	-28.078	0.000	-0.613	-0.533
ar.L4	-0.5694	0.020	-28.814	0.000	-0.608	-0.531
ar.L5	-0.5634	0.019	-30.182	0.000	-0.600	-0.527
ar.L6	-0.3427	0.016	-21.579	0.000	-0.374	-0.312
ar.S.L12	-0.0763	0.024	-3.176	0.001	-0.123	-0.029
sigma2	0.0191	0.000	48.408	0.000	0.018	0.020

Ljung-Box (L1) (Q):	0.00	Jarque-Bera (JB):	872.96
Prob(Q):	0.97	Prob(JB):	0.00
Heteroskedasticity (H):	0.53	Skew:	-0.22
Prob(H) (two-sided):	0.00	Kurtosis:	6.28

Fuente: Elaboración propia

Para este nuevo modelo todos los parámetros resultan ser significativo dado que su valor-p es inferior a 0.05. Observemos el comportamiento de los residuales:

Gráfica 14. Residuales del modelo SARIMA (6,1,0)(1,0,0,12)



Fuente: Elaboración propia

En general, el comportamiento de los residuales de ambos modelos es muy similar.

3. **Mejor Modelo Autoarima:** Adicionalmente, se decide hacer uso de la librería Autoarima de Python para apoyarnos en la identificación de un modelo alterno, tomando como métrica de selección el menor valor de AIC. El modelo seleccionado por la función se muestra a continuación:

Gráfica 15. Resultados modelo Autoarima (0,1,3)(2,0,2,12)

SARIMAX Results

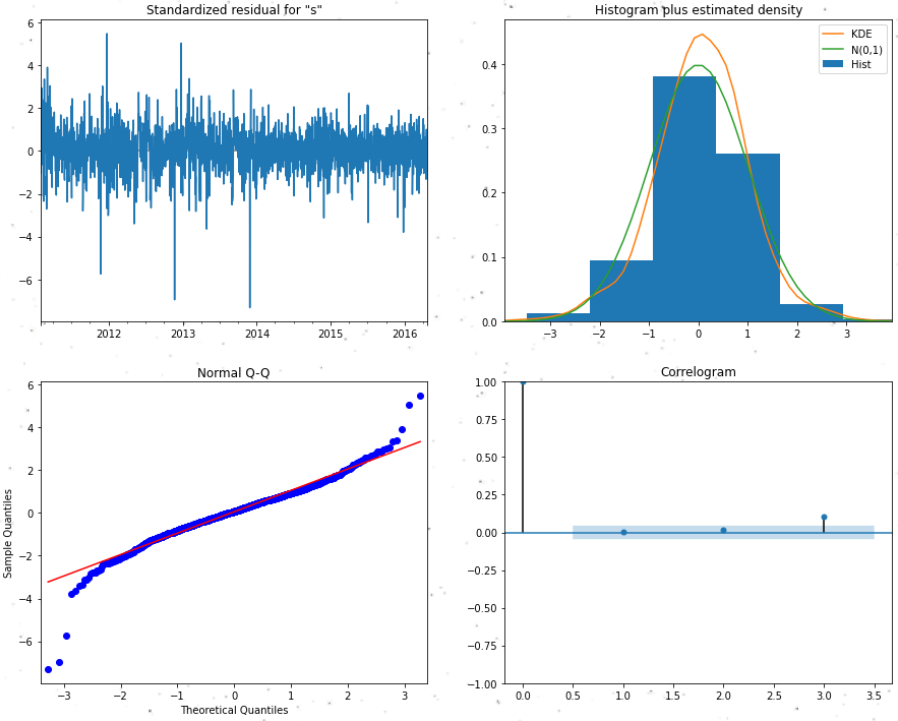
Dep. Variable:	sells_log_t	No. Observations:	1909
Model:	SARIMAX(0, 1, 3)x(2, 0, [1, 2], 12)	Log Likelihood	1165.972
Date:	Sat, 03 Dec 2022	AIC	-2315.945
Time:	01:11:32	BIC	-2271.514
Sample:	01-31-2011	HQIC	-2299.592
	- 04-22-2016		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ma.L1	-0.7246	0.020	-37.145	0.000	-0.763	-0.686
ma.L2	-0.1484	0.027	-5.563	0.000	-0.201	-0.096
ma.L3	-0.0038	0.021	-0.176	0.860	-0.046	0.038
ar.S.L12	-0.4463	0.002	-270.590	0.000	-0.450	-0.443
ar.S.L24	-0.9997	0.000	-2033.996	0.000	-1.001	-0.999
ma.S.L12	0.4403	0.008	53.044	0.000	0.424	0.457
ma.S.L24	0.9795	0.015	65.195	0.000	0.950	1.009
sigma2	0.0166	0.000	48.299	0.000	0.016	0.017

Ljung-Box (L1) (Q):	0.03	Jarque-Bera (JB):	1594.57
Prob(Q):	0.85	Prob(JB):	0.00
Heteroskedasticity (H):	0.57	Skew:	-0.45
Prob(H) (two-sided):	0.00	Kurtosis:	7.39

Fuente: Elaboración propia

Gráfica 16. Gráfico de residuales del modelo Autoarima (0,1,3)(2,0,2,12)



Fuente: Elaboración propia

4. Comparación de Modelos

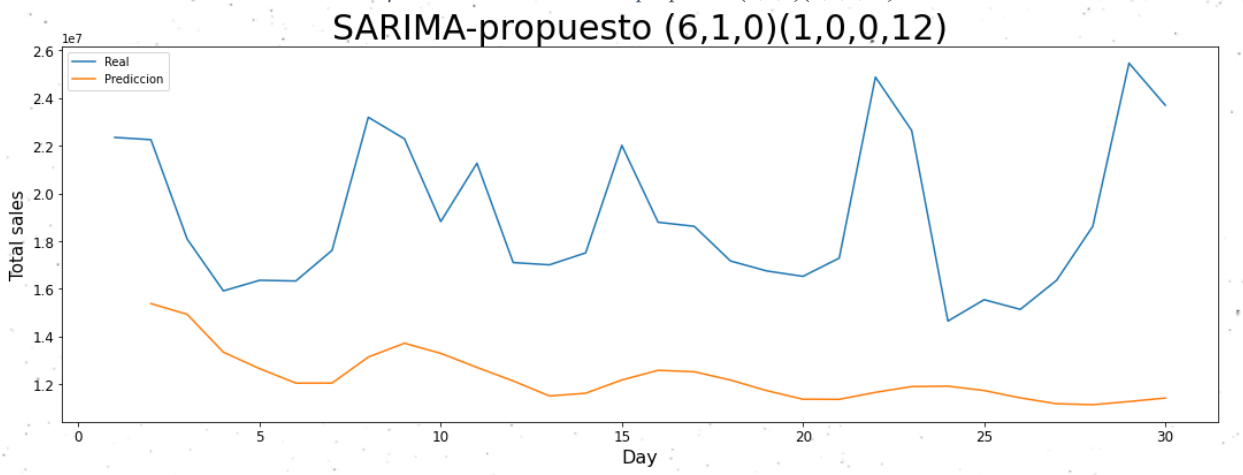
Los resultados muestran que el modelo propuesto por autorima es ligeramente mejor que el propuesto en el analisis al comparar el error absoluto medio,MAE. Este valor se interpreta como el promedio de los errores entre el valor real y el predicho del valor de ventas para 30 días. Entre más se acerca está el valor del MAE a 0, más cerca está el valor pronosticado del valor real. Los resultados muestran que el modelo se puede mejorar, quizá revisando la incorporación de variables exógenas.

Tabla 2. Resultados de los modelos SARIMA categoría Hobbies: propuesto vs best model autoarima

Modelo	AIC	MAE Test
SARIMA-Modelo propuesto autoarima (0,1,3)(0,0,2,12)	-2315.945	5,739,715.44
SARIMA-propuesto (6,1,0)(1,0,0,12)	-2118.6	6607194.96

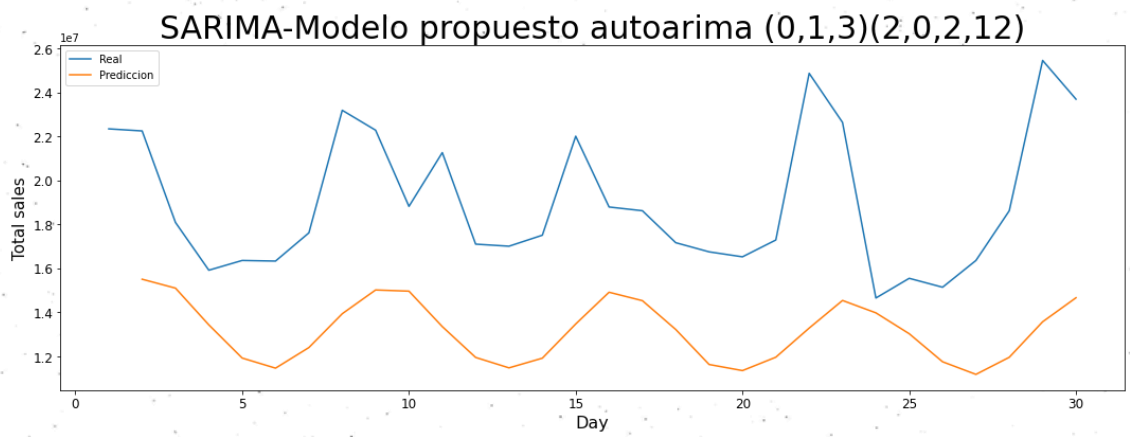
Fuente: Elaboración propia

Gráfica 17. SARIMA-Modelo propuesto (6,1,0)(1,0,0,12)



Fuente: Elaboración propia

Gráfica 18. SARIMA-Modelo propuesto Autoarima (0,1,3)(2,0,2,12)



Fuente: Elaboración propia

3.4.2. Modelos de Machine Learning

Los modelos de Machine Learning serán optimizados mediante validación cruzada con el set de entrenamiento y set de validación. Los modelos de Machine Learning con sus hiper parámetros optimizados entraran a competir con SARIMA y LSTM en el set de prueba tomando sus propias predicciones como variables descriptoras.

3.4.2.1 Modelación Machine Learning para HOBBIES

Para la implementación de los modelos ML se sigue el siguiente proceso:

1. **Partición de datos:**
   - Entrenamiento = 1.816 registros consecutivos en días
   - Validación = 15 registros consecutivos en días
   - Prueba = 30 registros consecutivos en días

2. **Modelos usados**

Se cuenta entonces con una estructura de datos que permite hacer aprendizaje supervisado, por lo que se pretende encontrar una función matemática, por medio de un modelo de machine Learning, que permita expresar las ventas diarias en dólares para la categoría hobbies, usando las 9 variables predictoras encontradas en el apartado 3.2.2 [L17(rezago 14), L24(rezago 24), L3(rezago 28), L10 (rezago 21), L30 (rezago 1), std1M (desviación estándar últimos 30 días), mean1M (media últimos 30 días), std2M (desviación estándar últimos 60 días), mean6M (media últimos 6 meses)]

Los modelos seleccionados para encontrar la interacción de las variables predictoras con las ventas en dólares para la categoría hobbies son:

- kNN
- Random Forest
- Gradient Boosting
- Suppor Vector Regression
- RF-kNN

3. Optimización de hiper parámetros

La optimización de parámetros se realiza usando validación cruzada:

- **kNN:** número de vecinos cercanos entre 1 y 50
- **Random Forest:**
  Número de estimadores 30, 50, 100, 150
  Máximo de características por árbol: 6, 7, 8, 9
  Máximo número de hojas: 15, 20, 22, 25
- **Gradient Boosting**
  Número de estimadores 30, 50, 100, 150
  Máximo de características por árbol: 6, 7, 8, 9
  Máximo número de hojas: 15, 20, 23, 26
- **Support Vector Regression**
  Kernel
  C: 1, 10, 100, 1000
  Degree: 2, 3, 4, 5, 6

Después de optimizar los hiper parámetros se obtienen los siguientes errores de entrenamiento y validación para la mejor versión de cada modelo.

Tabla 3. Modelos con variable a predecir sin escalamiento

Modelo	MAE train	MAE Validation
kNN	458,995.7	498,964.29
RF	428,888.49	563,891.17
Gradient Boosting	413,914.5	584,023.99
SVR	763,292.32	634,416.23
RF-kNN	459,432.09	493,392.91

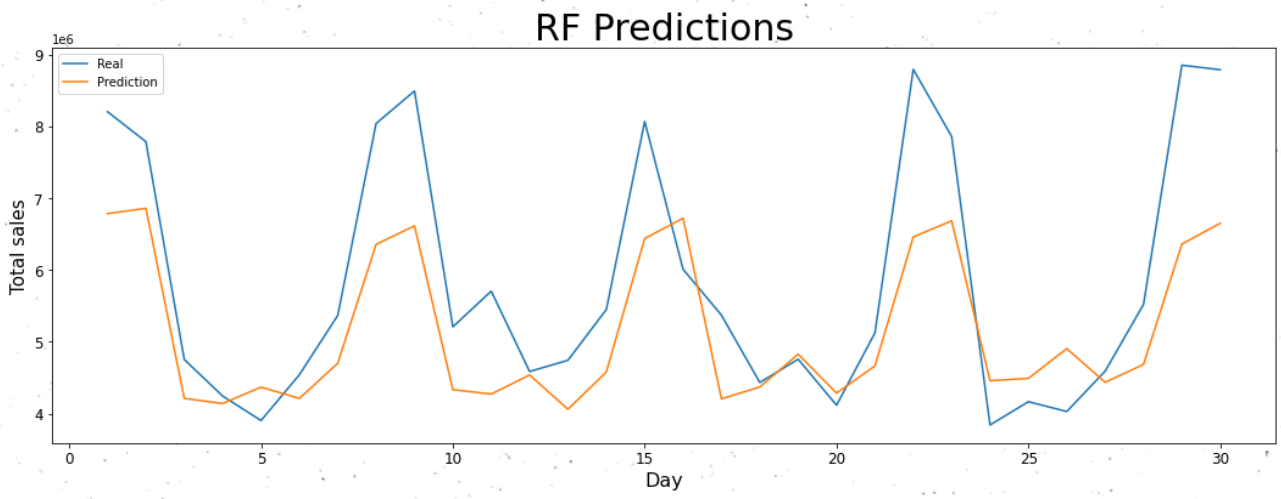
Fuente: Elaboración propia

Tabla 4. Modelos con variables escalada a predecir

Modelo	MAE Train	MAE Validation
kNN	455,056.43	489,077.06
RF	423,522.97	527,205.10
Gradient Boosting	3,427,712.86	5,007,366.38
SVR	763,292.32	634,416.23
RF-kNN	456,893.1	479,575.37

Fuente: Elaboración propia

Gráfica 19. Predicción RF vs serie real (mejor modelo en entrenamiento)



Fuente: Elaboración propia

El modelo de la gráfica anterior es un Random Frorest, con 100 estimadores, máximo número de características por árbol de 6 y máximo número de nodos hojas de 25. Tienen un MAPE de 13,94%,

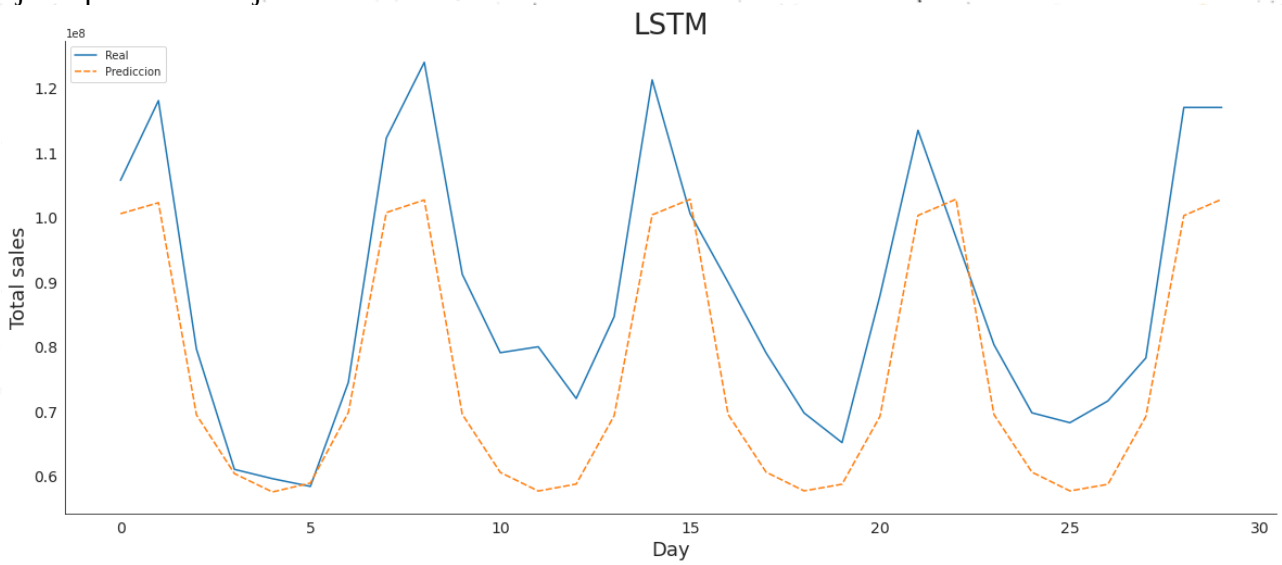
3.4.3. Modelos de Deep Learning

Se separaron los datos en validación, entrenamiento y pruebas. Posteriormente, se transforman los datos empleando los rezagos de 180 días para realizar las predicciones correspondientes. Ya que se busca la predicción de una ventana de tiempo amplia (30 días) se emplea el método recursivo para la predicción de múltiples pasos. Además, se realizan experimentos con diferente número de épocas para identificar el valor óptimo, siendo 20. Posteriormente, se seleccionó el mejor modelo y se reentrenó agregando los datos de validación para obtener los resultados finales.

Gráfica 20. Ajuste del modelo por épocas

Fuente: Elaboración propia

El modelo entrenado logra identificar el comportamiento de la serie de tiempo, no obstante, el ajuste puede ser mejorado.

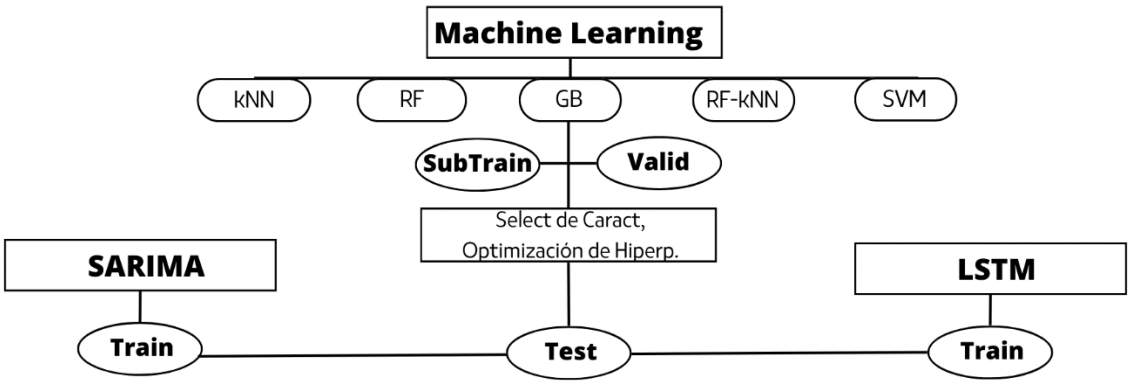


3.5. Evaluación de los modelos

En esta sección compiten los Frameworks SARIMA, Machine Learning y LSTM mediante el set de Testing con 30 días como se especificó en la sección de Modelado. Los modelos de Machine Learning fueron optimizados mediante un subset de entrenamiento y un set de validación, es decir, de los datos sobrantes (set de Train) después de realizar la partición del set de Test, se realizó una nueva partición donde resulta un subset del Train y un set de Validación.

La métrica de comparación es el Error Absoluto Medio tanto para la categoría HOBBIES como FOODS. La siguiente figura muestra un diagrama para la metodología de evaluación que se usó.

Figura 21. Diagrama de evaluación para los tres Frameworks



Fuente: Elaboración propia

A continuación, en las dos siguientes subsecciones se presentan tablas de los resultados de los modelos para los 3 Frameworks.

3.5.1 HOBBIES

Tabla 5. Resultados de los Frameworks en el set de Test

Modelo	MAE TEST
kNN con variables escaladas	999,664.91
RF con variables escaladas	**903,844.14**
Gradient Boosting con variables escaladas	926,338.89
SVR con variables escaladas	1,073,776.60

RF-kNN con variables escaladas	971,490.96
SARIMA propuesto	6,607,194.96
SARIMA Autoarima	5,739,715.44
LSTM	6,255,130.81

Fuente: Elaboración propia

1.5.1 FOODS

Tabla 6. Resultados de los Frameworks en el set de Test

Modelo	**MAE TEST**
kNN con variables escaladas	12,186,285.10
RF con variables escaladas	11,727,245.21
Gradient Boosting con variables escaladas	**9,913,601.50**
SVR con variables escaladas	11,781,817.34
RF-kNN con variables escaladas	11,727,245.21
SARIMA propuesto	108,799,642.95
SARIMA Autoarima	122,990,454.14
LSTM	18,875,740.5

Fuente: Elaboración propia

4. CONCLUSIONES Y RECOMENDACIONES

Se implementaron 3 frameworks diferentes para la predicción de series de tiempo, logrando evidenciar las características y requisitos principales para que cada uno tenga un funcionamiento correcto. Las principales conclusiones son:

- Los modelos de Machine Learning lograron el mayor desempeño, siendo el más destacado el modelo Random Forests, no obstante, este modelo presenta sobreajuste. Por lo tanto, para evitarlo se recomienda el uso del modelo RF-KNN
- Tanto los modelos SARIMA como la LSTM pueden mejorarse ya que presentan un error MAE considerablemente mayor en comparación a los modelos de machine Learning.
- La iteración de nuevas pruebas para los tres frameworks es importante y recomendada ya que con distintos conjuntos de parámetros y combinaciones de estas se podría alcanzar un mejor desempeño.

El repositorio con toda la información referente al proyecto se encuentra en:
https://github.com/JoseJorgeMM/Proyecto-Integrador-2022-2

UNIVERSIDAD EAFIT

REFERENCIAS BIBLIOGRÁFICAS

[1] J. Durbin and S. J. Koopman, "Time Series Analysis by State Space Methods," *Oxford University Press,* 2001.

[2] W. G. Sammut C, "Computationally efficient heart rate estimation during physical," *Encyclopedia of machine learning. Springer, Berlin,* p. 2478–2481, 2017.

[3] Breiman, "Random Forest. Mach Learn," 2001, pp. 5-32.

[4] G. C. Chen T, "XGBoost: a scalable tree boosting system.," *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining,* p. 785–794, 2016.

[5] S. E. A. V. Makridakis S, "The M4 competition: results, findings, conclusion and," *Int J Forecast,* vol. 34, no. 4, p. 802–808, 2018.

[6] L. M. J. N. Fulcher BD, "Highly comparative time-series analysis: the empirical structure of," *J R Soc Interface ,* vol. 10, no. 83, 2013.

[7] C. S. S. K. P. e. a. Lubba, "catch22: CAnonical Time-series CHaracteristics.," *Data Min Knowl Disc,* vol. 33, p. 1821–1852, 2019.

[8] B. C. K. L. S. A. V. V. Drucker H, "Support vector regression machines," in *Advances in neural information processing systems*, 1997, p. 155–161.

[9] A. Raj, "Unlocking the True Power of Support Vector Regression," Using Support Vector Machine for Regression Problems, 03 10 2020. [Online]. Available: https://towardsdatascience.com/unlocking-the-true-power-of-support-vector-regression-847fd123a4a0. [Accessed 22 11 2022].

[10] F. A. Gers, J. Schmidhuber and F. Cummins, "Learning to Forget: Continual Prediction with LSTM," *Neural Comput,* vol. 12, p. 2451–2471, 2000.

[11] J. L. Elman, "Finding structure in time," *Cogn Sci,* vol. 14, no. 2, p. 179–211, 1990.

[12] Y. Bengio, P. Simard and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans Neural Netw,* vol. 5, no. 2, p. 157–166, 1994.

[13] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput,* vol. 9, no. 8, p. 1735–1780, 1997.

[14] I. Sutskever, O. Vinyals and Q. v Le, "Sequence to Sequence Learning with Neural Networks," *Advances in Neural Information Processing Systems,* vol. 27, 2014.

[15] I. W. J. B. S. a. Guyon, "Gene Selection for Cancer Classification using Support Vector Machines.," *Machine Learning,* vol. 46, no. https://doi.org/10.1023/A:1012487302797 , p. 389–422, 2002.

[16] F. &. P. P. &. H. M. Ferri, "Comparative Study of Techniques for Large-Scale Feature Selection.," *Pattern Recognition in Practice,* no. 10.1016/B978-0-444-81892-8.50040-7. , 2001.

[17] IBM, "IBM," 05 06 2022. [Online]. Available: https://www.ibm.com/docs/es/SS3RA7_18.3.0/pdf/ModelerCRISPDM.pdf.