COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

COMPARATIVA DE MODELOS DE MACHINE LEARNING APLICADOS A LA PREDICCION DE DATOS DE MARKETING

Autor: Santiago Arenas Martín

Director: Luis Francisco Sánchez Merchante

Madrid

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

COMPARATIVA DE MODELOS DE MACHINE LEARNING APLICADOS A LA PREDICCION DE DATOS DE MARKETING

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico 2024/25 es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo.: Santiago Arenas Martín Fecha: 24/06/2025

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Luis Francisco Sánchez Merchante Fecha: 24/06/2025

Luis Fco Sánchez
2025.06.25
14:39:33 +02'00'

COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

COMPARATIVA DE MODELOS DE MACHINE LEARNING APLICADOS A LA PREDICCION DE DATOS DE MARKETING

Autor: Santiago Arenas Martín

Director: Luis Francisco Sánchez Merchante

Madrid

Agradecimientos

A mis padres, que me han dado todo lo necesario en esta vida para acabar en esta posición. Este trabajo va por vosotros.

COMPARATIVA DE MODELOS DE MACHINE LEARNING APLICADOS A LA PREDICCION DE DATOS DE MARKETING

Autor: Arenas Martín, Santiago.
Director: Sánchez Merchante, Luis Francisco.
Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

RESUMEN DEL PROYECTO

En este trabajo se ha desarrollado una comparativa exhaustiva entre modelos de predicción aplicados al ámbito del marketing digital, utilizando tres conjuntos de datos reales con características heterogéneas. Se han empleado modelos de series temporales como ARIMA, así como algoritmos de Machine Learning supervisado, destacando Random Forest, SVR, LightGBM y redes neuronales recurrentes (RNN). Tras un proceso completo de limpieza, transformación y selección de variables, los resultados han demostrado de forma consistente que Random Forest ha sido el modelo con mejor desempeño, alcanzando valores de R^2 superiores al 0.9 en múltiples tareas y reduciendo notablemente la raíz del error cuadrático medio. En contraste, los modelos temporales han ofrecido un rendimiento deficiente en los datasets utilizados. El trabajo concluye con una propuesta de mejora orientada al ajuste fino de hiperparámetros, el uso de datos de mayor calidad y la exploración de modelos como Prophet en futuras investigaciones.

Palabras clave: Predicción, Random Forest, Machine Learning, Series Temporales, Marketing Digital, ARIMA.

1. Introducción

Este proyecto nace de la necesidad creciente de encontrar modelos predictivos sólidos en un entorno donde los datos de marketing y ventas son cada vez más abundantes, pero también más heterogéneos. A través del análisis comparativo de distintos algoritmos de Machine Learning y modelos de series temporales, se pretende identificar cuál ofrece un mejor rendimiento ante datasets reales con estructuras dispares [1]. El trabajo no solo aborda la preparación y transformación de datos para su uso en modelos [2], sino que también plantea un enfoque metodológico reproducible y adaptable, con el objetivo final de desarrollar soluciones automatizadas que faciliten la toma de decisiones en entornos comerciales complejos.

2. Definición del proyecto

El proyecto se plantea como una investigación aplicada en el ámbito del análisis de datos de marketing, con el objetivo de comparar distintos modelos predictivos frente a datasets que varían tanto en estructura como en contenido. La propuesta parte de una problemática real: en entornos empresariales, no existe un único tipo de dato ni un único patrón de comportamiento [3], lo que obliga a evaluar la versatilidad de los algoritmos utilizados. Así, el núcleo del proyecto consiste en comprobar qué modelo, entre opciones como Random Forest, LightGBM, RNN, SVR o ARIMA (Series Temporales), es capaz de ofrecer el mejor rendimiento generalizado al enfrentarse a bases de datos reales de ventas [4], comportamiento de clientes y campañas publicitarias. Todo ello, bajo un enfoque que combina tanto el rigor técnico del análisis como la visión práctica de su posible aplicación empresarial.

3. Descripción del sistema

El sistema desarrollado en este proyecto se estructura como una pipeline de análisis y predicción que integra múltiples etapas, desde la recopilación y transformación de datos hasta la evaluación y comparación de modelos. Su funcionamiento parte de la entrada de uno o varios conjuntos de datos de marketing, los cuales pueden contener información sobre ventas, campañas publicitarias o perfiles de clientes. Una vez verificados y preprocesados, estos datos son transformados mediante técnicas de ingeniería de variables para extraer nuevas características relevantes que optimicen el rendimiento de los modelos.

El sistema incorpora dos grandes bloques de modelado: uno orientado a series temporales, con algoritmos como ARIMA, SARIMA y SARIMAX; y otro centrado en modelos supervisados de Machine Learning, entre los que destacan Random Forest, LightGBM, RNN y SVR. Para cada bloque, se definen métricas específicas, RMSE y R^2, que permiten evaluar el ajuste de las predicciones a los valores reales.

Además, el sistema incluye mecanismos de selección de variables, ajustes de hiperparámetros mediante Grid Search y generación de visualizaciones comparativas, lo que permite no solo automatizar el proceso, sino también facilitar su interpretación.

Finalmente, los resultados se consolidan en tablas de rendimiento con las métricas descritas antes, que sirven de base para identificar el modelo óptimo en cada escenario, con el objetivo de ofrecer una solución predictiva generalizable a distintos tipos de datos de marketing.

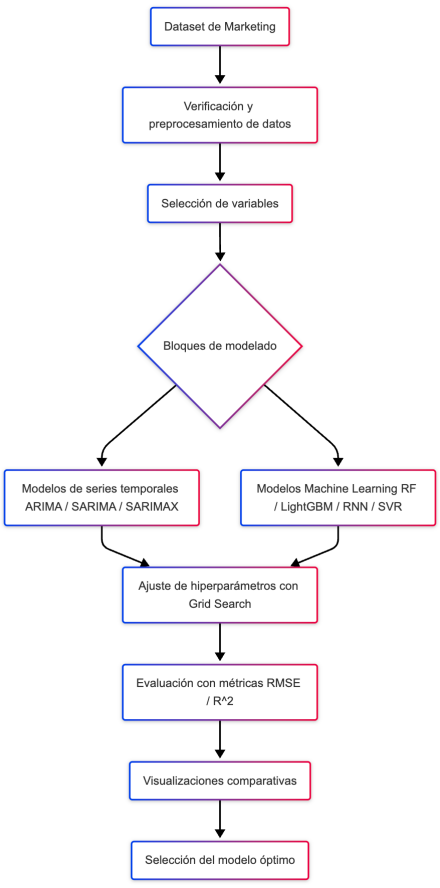

Ilustración 1: Sistema empleado durante el trabajo. Elaboración propia.

4. Resultados

Los resultados obtenidos a lo largo del proyecto confirman que Random Forest ha sido el modelo más eficaz en la mayoría de los escenarios, logrando los mejores valores de R^2 y RMSE en la predicción de las métricas objetivo, seguido de cerca por los modelos de LightGBM. En contraste, los modelos de series temporales, RNN y SVR han ofrecido un rendimiento muy por debajo del esperado, con ajustes pobres y baja capacidad de generalización, indicando que la complejidad de los datos y volumen de estos era superior a lo que podían manejar. La evaluación comparativa entre datasets ha demostrado la importancia de adaptar el modelo al tipo y calidad de los datos, destacando la versatilidad y estabilidad de los algoritmos basados en árboles. Una de las comparativas más claras al respecto del desempeño se ha obtenido en uno de los tres datasets utilizados, donde se puede ver claramente las diferencias mencionadas.

Dataset Clientes	Random Forest	RNN	SVR	LightGBM	Series Temporales
RMSE	2.97	6.61	5.27	3.23	33.767
R^2	0.84	0.23	0.51	0.81	-0.195

Tabla 1: Comparativa de todos los modelos para Clientes.

Cabe mencionar que el R^2 obtenido para este dataset de 0.84 no es el más alto para este trabajo, como puede verse posteriormente.

5. Conclusiones

Este proyecto ha demostrado que, ante la diversidad de datos que caracteriza al entorno del marketing digital, no todos los modelos predictivos responden con la misma eficacia. El uso de múltiples datasets ha permitido poner a prueba la capacidad de generalización de cada enfoque, y los resultados han sido concluyentes: Random Forest se posiciona como el modelo más robusto y preciso, independientemente de la estructura o complejidad de los datos. En cambio, los modelos de series temporales han evidenciado sus limitaciones cuando se aplican a conjuntos sin estacionalidad clara o con ruido estructural. Este trabajo subraya la necesidad de una buena colección y preparación de datos, una validación rigurosa y una selección de modelos que esté alineada con las características reales del problema, más allá de supuestos teóricos o tendencias metodológicas, a la hora de ayudar a los profesionales del sector para tomar las decisiones pertinentes.

6. Referencias

[1] A. Z. H. K. A. &. T. A. M. Bouguettaya, «Machine Learning and Deep Learning as New Tools for Business Analytics. In Advances in business information systems and analytics.,» de 2023 Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE), Las Vegas, NV, USA, 2023.

[2] O. M. V. &. V. A. Tsilingeridis, «Design and development of a forecasting tool for the identification of new target markets by open time-series data and deep learning methods.,» Applied Soft Computing, p. 132, 2022.

[3] N. &. D. K. Vintha, «Comparative Analysis of Deep Learning Approaches for Analysis and Prediction of Multivariate Time Series Data,» IEEE, p. 76–82, 2023.

[4] M. B. A. S. A. H. A. N. A. K. J. M. J. J. &. A. A. S. A. Al Atif, «Data mining with its role in marketing, sales support and customer identification data analysis,» International Journal Artificial Intelligent and Informatics, pp. 104-116, 2022.

COMPARISON OF MACHINE LEARNING MODELS APPLIED TO MARKETING DATA PREDICTION

Author: Arenas Martín, Santiago.
Supervisor: Sánchez Merchante, Luis Francisco.
Collaborating Entity: ICAI – Universidad Pontificia Comillas

ABSTRACT

Throughout this body of work, an exhaustive comparative between predictive models applied to the digital marketing space has been developed, utilizing three datasets and their real-world data, with heterogeneous characteristics. Different models were employed for the findings, from Time Series like ARIMA, as well as supervised learning with Random Forest, SVR, LightGBM and recurrent neural networks (RNN). After a thorough data cleaning process, transformation and selection of variables, the results have shown consistently that Random Forest has been the best model, achieving values of R2 north of 0.9 in multiple instances and notably reducing the root mean squared error. In contrast, Time Series have demonstrated to be clearly deficient in the results obtained for the datasets. The project concludes with a future proposition to improve the results using further fine-tuning of hyperparameters, better quality data and the exploration of Prophet in further investigations.

Keywords: Prediction, Random Forest, Machine Learning, Time Series, Digital Marketing, ARIMA.

1. Introduction

The project in question is born from the growing need of finding strong predictive models in a setting were the Marketing and sales data are ever-so abundant, but also more heterogeneous. Using this comparative analysis of different Machine Learning algorithms and Time Series, it is aimed to identify which one offers better performance when facing real-world data with complex and varied structures [1]. This collection of work not only tackles the preparation and transformation of data to be fed to the models [2], it also provides a methodology of doing so reproductive and adaptively, with the end goal of developing automatized solution that aid in decision making in complex business settings.

2. Project definition

The project is thought of as an applied deep dive into the subject of Marketing data analytics, with the goal of comparing different predictive models that vary in structure as well as content. The proposition starts from a real-world dilemma: in a business setting, there is not one type of data nor one real behavioral pattern [3], thus making it necessary to evaluate the versatility of algorithms used. Therefore, the core of the project involves comparing what model, including Random Forest, LightGBM, RNN, SVR or ARIMA (Time Series), can offer the better functionality overall when facing real databases, whether sales [4], client behavior and marketing campaigns. All of this with an emphasis that combines both technical rigor of the analysis conducted as well as practical vision of the possible commercial application.

3. System Description

The system developed for this project is structured as a pipeline of analysis and prediction that involves multiple steps, from the collection and transformation of data to the evaluation and comparation of the models. Its workings start from the introduction of one or more datasets, which may obtain data regarding sales, client profiles or marketing campaigns. Once verified and pre-processed, they are then transformed using data engineering methods to extract new relevant characteristics that optimize the model's performance.

It incorporates two major modelling blocks: one oriented towards the Time Series analysis, using algorithms like ARIMA, SARIMA and SARIMAX; and another centered around supervised learning algorithms like Random Forest, LightGBM, RNN and SVR. For each of the blocks, there are specific metrics that are sought after, RMSE and R^2, that allow for the evaluation of the fit for the predictions using real values.

More so, the system includes variable selection mechanisms, hyperparameter adjustment using Grid Search and comparative visualization generation, which not only allows for the automation of the process but also supports the correct understanding of the findings.

Finally, the results are consolidated in tables with the aforementioned metrics, which act as a base to identify the optimum model for each scenario, with the goal of offering a generalizable, predictive solution applicable to various Marketing data types.

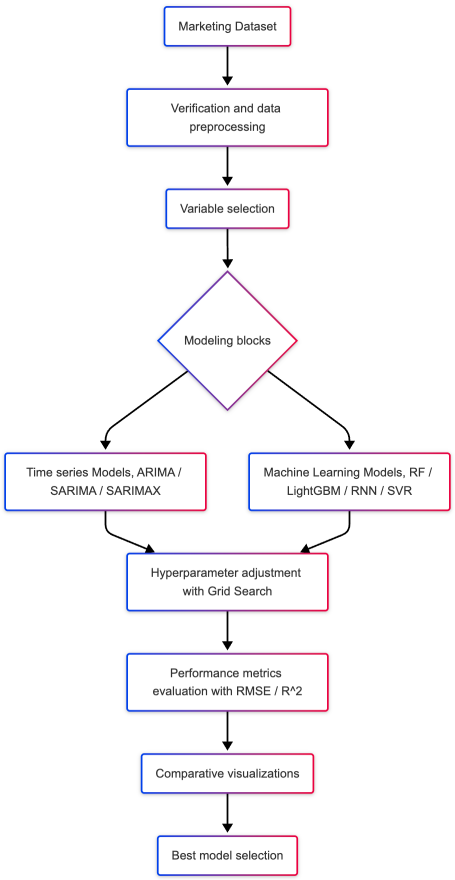

Illustration Ilustración 1- System employed during the project. Own elaboration.

4. Results

The results obtained throughout this project affirm that Random Forest has been the most efficient model in most scenarios, achieving the best R^2 and RMSE in the prediction of metric objectives, with LightGBM being a close second. In contrast, Time Series models, RNN and SVR have displayed a general poor performance, with bad adjustments and low capability of generalization, implying that the data complexity and volume of them were more than they could handle. The comparative evaluation of the datasets has demonstrated the importance of adapting the model to the type and quality of the data, highlighting the versatility and stability of tree-based algorithms. One of the clearest comparison regarding model performance has been obtained in one of the three datasets utilized, where the differences are clearly shown.

Clients Dataset	Random Forest	RNN	SVR	LightGBM	Time Series
RMSE	2.97	6.61	5.27	3.23	33.767
R^2	0.84	0.23	0.51	0.81	-0.195

Tabla 2: All models comparison for Clients.

It is also worth mentioning that the R^2 obtained for this dataset is 0.84, and not the highest obtained throughout the project, as shown in the body.

5. Conclusions

The project has demonstrated that, when facing the diversity that defines the data utilized in digital marketing, not every predictive model is the same efficiency wise. The use of multiple dataset has allowed for testing the generalization capability of the models and the results are clear: Random Forest has positioned itself as the most robust and precise model, regardless of structure or complexity of the dataset. In comparison, Time Series models have shown its limitation when no clear time pattern is present or with structural noise. This collection of work highlights the need for a good collection and preparation fo the data, rigorous validation and selection of models aligned with real problem characteristics, not only theoretically and methodologically, when facing the task of aiding sector professionals in taking the best decisions.

6. References

[1] A. Z. H. K. A. &. T. A. M. Bouguettaya, «Machine Learning and Deep Learning as New Tools for Business Analytics. In Advances in business information systems and analytics.,» de 2023 Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE), Las Vegas, NV, USA, 2023.

[2] O. M. V. &. V. A. Tsilingeridis, «Design and development of a forecasting tool for the identification of new target markets by open time-series data and deep learning methods.,» Applied Soft Computing, p. 132, 2022.

[3] N. &. D. K. Vintha, «Comparative Analysis of Deep Learning Approaches for Analysis and Prediction of Multivariate Time Series Data,» IEEE, p. 76–82, 2023.

[4] M. B. A. S. A. H. A. N. A. K. J. M. J. J. &. A. A. S. A. Al Atif, «Data mining with its role in marketing, sales support and customer identification data analysis,» International Journal Artificial Intelligent and Informatics, pp. 104-116, 2022.

Índice de la memoria

ANEXO II

Índice de figuras

Índice de tablas

Capítulo 1. INTRODUCCIÓN

1.1 MOTIVACIÓN DEL PROYECTO

El campo del análisis de datos ha experimentado un crecimiento significativo en los últimos años, impulsado por la disponibilidad de grandes volúmenes de datos y el avance de técnicas de inteligencia artificial (IA). Este proyecto se centra en la adquisición y análisis de datos de ventas y campañas de marketing, con énfasis en encontrar el mejor modelo ante datos que no siguen un mismo patrón. Con el aumento exponencial de los datos de ventas y marketing, existe una necesidad urgente de técnicas eficientes para extraer información valiosa que pueda mejorar las estrategias comerciales y de marketing.

Los datos de ventas y marketing, como los informes de ventas y las métricas de campañas publicitarias, juegan un papel crucial en la toma de decisiones empresariales, la planificación de estrategias y la evaluación del rendimiento. Sin embargo, los métodos manuales tradicionales para analizar estos datos son lentos y propensos a errores. Este proyecto tiene como objetivo abordar estos desafíos mediante el desarrollo de modelos automatizados para el análisis de datos de ventas y marketing utilizando diferentes modelos de Machine Learning.

Al aprovechar algoritmos y otras técnicas avanzadas de análisis de datos, el proyecto busca proporcionar herramientas precisas y generalizables que minimicen la intervención humana y mejoren la toma de decisiones empresariales en el campo del marketing y las ventas.

1.2 OBJETIVOS PRIMARIOS

- Análisis de los datasets variados de marketing.

Durante el proceso de obtención de las bases de datos que se han utilizado para este trabajo, se hizo notorio la diversidad de éstos en cuánto a los parámetros de los que se componen. Es por esto queun objetivo primario de este proyecto es analizar si hay algún modelo que se comporte particularmente bien frente a toda la tipología de datasets.

- Exploración de modelos de series temporales.

Mediante el uso de técnicas como ARIMA, SARIMA y SARIMAX se busca identificar patrones estacionales para realizar predicciones de las métricas que se analizan.

- Predicción de métricas clave.

Junto con las series temporales, se emplean varios modelos de Machine Learning como Random Forest, RNN, SVR y LightGBM buscando una predicción de lo que puede pasar en un futuro a partir de datos pasados y variables predictoras adicionales.

1.3 OBJETIVOS SECUNDARIOS

- Ingeniería de Datos, construcción de variables adicionales.

Creación de nuevas columnas en los conjuntos de datos que puedan servir para ingestar en los modelos y ayudar a su análisis e interpretación posterior.

- Exploración de datos, validación de calidad.

Asegurarse de que los conjuntos de datos son coherentes y no han sido corrompidos con información artificial. Esto es especialmente importante cuando las fuentes de datos no son conocidas, sino externas a la persona u organización.

- Preparación de datos para modelos predictivos.

Transformación de columnas para poder formar parte de modelos. Especialmente importante en campos donde se introduce texto a la hora de crear nuevas columnas que puedan ser introducidas en las técnicas empleadas.

- Reproducibilidad.

Mantener un registro claro y conciso de las transformaciones aplicadas a los datos, adición de nuevas columnas y pre-procesado de bases, para garantizar que el análisis sea riguroso y reproducible.

1.4 METODOLOGÍA

A lo largo del desarrollo del proyecto se ha trabajado con la metodología Agile, adaptado a las características del entorno académico. Se organizaron sprints semanales con entregables definidos, lo que permitió planificar tareas de forma iterativa y realista. Al final de cada

reunión se fijaban los objetivos del sprint en base al backlog del proyecto, priorizando funcionalidades clave según el estado actual y los recursos disponibles. En lugar de dailys, se optó por reuniones semanales tipo sprint review, donde se revisaban los avances, se discutían bloqueos y se proponían mejoras para el siguiente sprint. Este ritmo constante facilitó mantener una cadencia de trabajo estable, compatible con el desarrollo paralelo de los estudios en el extranjero, promoviendo la entrega continua de valor y permitiendo una adaptación rápida a los cambios surgidos durante el desarrollo.

1.4.1 DEFINICIÓN DEL ALCANCE DEL PROYECTO

En esta etapa se identifican las métricas que buscamos optimizar para los modelos. Este ejercicio consiste en pensar en el problema desde el punto de vista de una empresa, real o ficticia, que desarrollaría este proyecto.

1.4.2 RECOPILACIÓN Y ESTRUCTURACIÓN DE DATOS

Esta parte es que más impacto tiene sobre el resultado final, y como tal se dedicó mucho tiempo a la búsqueda de las bases de datos sobre las que se van a trabajar. Como es de esperar, no hay una gran cantidad de empresas que vayan a publicar sus datos de marketing, ya que estos contienen información confidencial que sólo deben tener acceso determinadas personas internas. Sin unos buenos datos no habría posibilidad de tener unos resultados fiables. Finalmente, y tras una búsqueda realmente intensiva, hubo 3 bases de datos que se seleccionaron, para posteriormente aplicar las transformaciones que fueran necesarias.

1.4.3 VERIFICACIÓN DE LOS DATOS DE ENTRENAMIENTO

Una fase previa al análisis es primero la verificación de los datos y su verosimilitud. Algunas de las bases de datos que se descartaron tenían datos artificiales que se habían constituido de una manera aleatoria, imposibilitando una extracción de conocimiento real.

1.4.4 DESARROLLO Y AJUSTE DE LOS MODELOS

Se definen en cada modelo las variables que se van a predecir y se establece unos parámetros estandarizados para cada uno, que se aplican a todas las bases de datos. En los modelos de series temporales no se hace esto, ya que se busca el mejor ajuste posible.

1.4.5 EVALUACIÓN Y ELECCIÓN DEL MODELO ÓPTIMO

Se recopilan métricas de cada uno de los modelos para todas las bases de datos donde se han utilizado, así como las visualizaciones pertinentes que ayuden en el análisis de los resultados. Se elaboran unas tablas con dichas métricas para poder compararlos equitativamente, y últimamente se elige un modelo como el ganador.

Capítulo 2. ESTADO DE LA CUESTIÓN

2.1 EVOLUCIÓN DEL ANÁLISIS DE DATOS DE VENTAS Y MARKETING

El análisis de datos de ventas y marketing ha experimentado una evolución significativa a lo largo de los años, impulsada por el avance de la tecnología y la creciente disponibilidad de datos. A continuación, se describen las principales etapas de esta evolución:

2.1.1 MÉTODOS TRADICIONALES

En las primeras etapas del análisis de datos, las empresas se apoyaban principalmente en métodos estadísticos básicos y herramientas como hojas de cálculo para interpretar datos históricos. El análisis era mayoritariamente descriptivo, enfocado en resumir datos pasados sin capacidades predictivas o prescriptivas debido a la falta de capacidad de obtención y procesamiento de la época [1]. Estos métodos eran útiles en contextos de datos limitados, pero resultaban ineficaces frente al crecimiento exponencial de la información y la complejidad del mercado [2].

2.1.2 BUSINESS INTELLIGENCE (BI)

La aparición de sistemas de Business Intelligence (BI) representó un salto cualitativo. Estas herramientas permitieron recopilar, transformar y visualizar grandes volúmenes de datos de forma automatizada [3], facilitando la toma de decisiones estratégicas en tiempo real. Dichos sistemas combinaron capacidades de extracción de datos, generación de informes dinámicos y visualización interactiva, permitiendo a las empresas operar con una visión más completa del negocio y su situación presente [4].

2.1.3 APRENDIZAJE AUTOMÁTICO

El verdadero cambio de paradigma llegó con el aprendizaje automático (Machine Learning, ML). A través de modelos como regresiones lineales y logísticas, árboles de decisión, random forest y support vector machines, los analistas de las empresas comenzaron a anticipar comportamientos, segmentar clientes y optimizar campañas de marketing en base a patrones identificados en los datos [5]. Estos métodos no solo aumentaron la precisión de las predicciones, sino que también facilitaron la personalización de las estrategias comerciales [6].

2.1.4 REDES NEURONALES Y APRENDIZAJE PROFUNDO

Con la introducción del aprendizaje profundo (Deep Learning), el análisis de datos alcanzó nuevos niveles de sofisticación. Las redes neuronales convolucionales (CNN) y recurrentes (RNN), especialmente las LSTM, demostraron buena eficacia en el análisis de datos no estructurados y series temporales complejas [7]. Además, su capacidad para capturar relaciones no lineales ha permitido predecir de mejor manera el valor del cliente, ayuda en la segmentación dinámica y modelado de tendencias emergentes [8].

2.1.5 MODELOS DE SERIES TEMPORALES

El análisis de series temporales ha sido crucial para anticipar fluctuaciones de mercado y planificar estrategias de ventas. Modelos clásicos como ARIMA y sus variantes automáticas, así como métodos de suavizamiento exponencial, siguen siendo ampliamente utilizados. Sin embargo, técnicas híbridas que integran machine learning, como Prophet o LSTM, están ganando popularidad por su capacidad de adaptarse a patrones complejos y no lineales [9], [10].

2.1.6 HERRAMIENTAS Y BIBLIOTECAS MODERNAS

La democratización del análisis de datos ha sido impulsada por herramientas de código abierto como: Pandas, para la manipulación y análisis de datos tabulares; Scikit-learn, para

la implementación de algoritmos de aprendizaje automático; TensorFlow y Keras, para el desarrollo de modelos de redes neuronales; LightGBM, para el entrenamiento de modelos de boosting y pmdarima para el modelado de series temporales utilizando auto ARIMA. Todas estas herramientas, que permiten implementar modelos avanzados con eficiencia computacional y flexibilidad, han acelerado la innovación en marketing predictivo, detección de anomalías y recomendaciones personalizadas [11].

2.1.7 PLATAFORMAS DE ANÁLISIS DE DATOS EN LA NUBE

El análisis de datos en la nube ha transformado la escalabilidad y accesibilidad del procesamiento de datos. Servicios como Amazon Web Services (AWS), Google Analytics [12] y Microsoft Azure ofrecen infraestructura y herramientas para modelar datos a gran escala, realizar análisis en tiempo real y automatizar procesos de marketing digital, sin necesidad de una infraestructura local compleja [13].

2.1.8 INTEGRACIÓN DE APIS

Finalmente, la integración de APIs ha sido esencial para conectar fuentes de datos diversas, como plataformas publicitarias (Facebook Ads, Google Ads, Amazon Advertising API), CRMs y sistemas de e-commerce [14]. Estas conexiones permiten recopilar información en tiempo real, evaluar campañas con mayor precisión y adaptar estrategias de forma ágil y personalizada, optimizando el retorno sobre la inversión y el control de las campañas por parte de los anunciantes [15].

2.2 JUSTIFICACIÓN DEL PROYECTO

En el análisis de datos de marketing, la gran diversidad de datasets, que varían en volumen, estructura y variables, representa un desafío constante para seleccionar el modelo adecuado. Este proyecto surge de la necesidad de identificar, probar y recomendar los modelos más efectivos para diferentes tipos de datos, sin depender de soluciones genéricas poco adaptables. A través de una evaluación sistemática de algoritmos, se busca ofrecer un

enfoque flexible y automatizable que permita seleccionar el modelo óptimo para cada caso, mejorando así la precisión y utilidad de los análisis. Además, se pretende facilitar herramientas accesibles para analistas no expertos, promoviendo el uso de técnicas avanzadas sin necesidad de conocimientos profundos en programación o estadística.

Capítulo 3. FUNDAMENTOS TEÓRICOS

3.1 MÉTRICAS DE MARKETING

Para evaluar y optimizar campañas de marketing resulta imprescindible definir y monitorizar una serie de métricas clave. A continuación, se propone una organización por grandes categorías, con fórmulas y ejemplos prácticos que faciliten su comprensión y permitan incorporar estos indicadores en modelos de machine learning para la predicción de resultados.

3.1.1 MÉTRICAS ECONÓMICAS

Estas métricas cuantifican la dimensión financiera de la campaña y sirven para estudiar la rentabilidad y el control del gasto.

Presupuesto: importe total asignado a la campaña publicitaria. Cuando se fija el presupuesto de una campaña, no es obligado gastar el 100 %: puede configurarse un "techo" máximo y permitir que el sistema optimice el nivel de inversión según el rendimiento. Por ejemplo, si el presupuesto mensual es de 10 000 €, la campaña puede detenerse antes de agotar esa cifra si los resultados caen por debajo de un umbral de eficiencia.

Retorno de la inversión (ROI): mide la eficacia de la campaña como la relación entre los beneficios netos y el coste total:

$$ROI = \frac{Beneficio\ neto\ (ventas - coste)}{Coste\ de\ la\ campaña} \times 100$$

Por ejemplo, una campaña que ha costado 10 000 € ha generado unas ventas de 50 000 €; esto supone un beneficio neto de 40 000 € y un ROI del 400 %. Este tipo de variables suele usarse como variable variable objetivo ("target") para predecir la eficiencia de futuras campañas aplicando técnicas de machine learning al ROI histórico.

ROAS y ACOS: es la misma métrica pero interpretada de forma directa o inversa:

- **ROAS**: se corresponde con el Return On Advertising Spent y se calcula como:

$$ROAS = \frac{Ingresos\ atribuibles\ a\ la\ campaña}{Gasto\ publicitario}$$

  Por ejemplo, un ROAS de 5 significa que por cada euro invertido se obtienen 5 € de ingresos

- **ACOS**: Es el inveros del ROAS, es el Advertising Cost of Sales y se calcula como:

$$ACOS = \frac{Gasto\ publicitario}{Ingresos\ atribuibles\ a\ la\ campaña} \times 100$$

  Por ejemplo, un ACOS del 20% equivale a un ROAS de 5, significando que por cada unidad de ingreso proveniente de la campaña, un 20% se corresponde con el gasto publicitario de dicha campaña

3.1.2 MÉTRICAS DE VISIBILIDAD

Permiten medir el alcance de la campaña y cuantificar cuántas veces y a cuántas personas llegó el mensaje.

Impresiones: número total de veces que un anuncio o producto se muestra en pantalla. Un mismo usuario puede generar múltiples impresiones. Por ejemplo, una campaña para mostrar un *display* publicitario que genera 200 000 impresiones en un día.

Alcance (Reach): número de usuarios únicos que han visto el anuncio al menos una vez. Por ejemplo, un banner publicitario que ha generado 200 000 impresiones puede haber sido visto por 50 000 usuarios diferentes

Visitas o Sesiones: cantidad de accesos (sesiones) a una web o landing page provocados directamente por la campaña. Se mide a través de parámetros de URL (UTM) o códigos de seguimiento. Por ejemplo, se pueden medir el número de visitas que genera un enlace desde un email de marketing.

3.1.3 MÉTRICAS DE INTERACCION

Miden el grado de interés y compromiso de los usuarios con los anuncios.

Clicks: veces que un usuario hace clic sobre el anuncio o enlace. Esta métrica refleja el interés de los diferentes elementos de la campaña.

CTR (Click-Through Rate): mide la relación entre el número de clicks que han realizado los usuarios y las veces que se ha mostrado el anuncio. Por ejemplo, 1200 clicks en 200 000 impresiones supone un CTR de 0.6%. Cuanto más elevado resulta el CTR, mejores se consideran las creatividades y los mensajes de las mismas para la audiencia.

$$CTR = \frac{Clicks}{Impresiones} \times 100$$

Ecuación 1: Click-Through Rate

Coste por Click (CPC) y Coste por Mil (CPM): miden el coste por click y por impresión. Se definen como:

$$CPC = \frac{Gasto\ total}{Número\ de\ clicks} \times 100$$

Ecuación 2: Coste por Click.

$$CPM = \frac{Gasto\ total}{Número\ de\ impresiones} \times 100$$

Ecuación 3: Coste por Mil.

El CPC permite comparar canales de comunicación. Por ejemplo, un CPC de 0.50€ en redes sociales vs. un 0.20% en Google Ads. Mientras, el CPM mide el coste medio por cada mil impresiones, y es más utilizado en campañas de branding.

Tasa de Conversión (Conversion Rate, CVR): mide el porcentaje de usuarios que, tras hacer clic en un anuncio o visita a una página, realizan la acción deseada (por ejemplo, una compra, un registro, una descarga, suscripción, etc.). Se calcula como:

$$CVR = \frac{Número\ de\ conversiones}{Número\ de\ clicks} \times 100$$

Ecuación 4: Conversion Rate, Tasa de Conversión.

Por ejemplo, un anuncio ha recibido 1200 clicks que han generado 60 ventas, esto equivale a un CVR del 5%. De nuevo, información histórica sobre el CVR puede resultar de mucha utilidad como predictor del éxito de futuros lanzamientos.

3.1.4 MÉTRICAS DE COMPORTAMIENTO Y RETENCION

Aunque en este trabajo no se explorarán en profundidad, conviene mencionarlas por su relevancia en análisis de usuarios y modelos de predicción a largo plazo [16].

- Tasa de Rebote (Bounce Rate): % de usuarios que abandonan el sitio tras ver una sola página.
- Páginas por Sesión: media de páginas vistas en cada visita.
- Tiempo Medio en Página: indicador de la calidad de la experiencia de usuario.
- Tasa de Retención: % de usuarios que regresan al sitio tras una primera visita.

Cada métrica puede convertirse en característica (feature) de un modelo predictivo. Por ejemplo, el número de impresiones y el CTR pueden servir como variables de entrada para predecir el ROI o el volumen de ventas. Asimismo, métricas de comportamiento —como páginas por sesión o bounce rate— complementan el perfil del usuario y mejoran la capacidad predictiva de los algoritmos de clasificación o regresión.

3.2 PROCESO DE MODELIZACIÓN

En esta fase se construyen y validan los modelos de machine learning a partir de los datos preprocesados. El proceso de modelización no consiste sólo en aplicar modelos de ML y parametrizarlos hasta conseguir la mejor de las precisiones. Antes de llegar a ese punto hay que realizar una serie de pasos que garanticen que los modelos obtenidos sean los mejores posibles y que sean capaces de generalizar frente a datos desconocidos.

3.2.1 RECOLECCIÓN DE LOS DATOS

La recolección de datos es uno de los pasos más importantes en cualquier proyecto de análisis, ya que de la calidad de los datos depende en gran medida la fiabilidad de los resultados. Para este proyecto, los datos utilizados provienen de fuentes de datos online, contrastadas y que han sido utilizadas con anterioridad. Posteriormente hay que validar la veracidad de dichos datos, con matrices de correlación para ver relaciones que deberían existir, o comprobando que los campos calculados, que se han explicado con anterioridad, tienen los valores que les corresponden.

3.2.2 LIMPIEZA DE DATOS Y PROCESAMIENTO DE MISSING VALUES

Antes de entrenar cualquier modelo, es imprescindible someter el conjunto de datos a una limpieza profunda y al análisis de los valores perdidos (*missing values*), puesto que la presencia de huecos puede introducir sesgos y deteriorar la capacidad predictiva.

Para el tratamiento de *missing values* se identifican primero las columnas y filas con datos faltantes y se evalúa si su ausencia responde a un patrón aleatorio o está relacionado con otras variables o con el propio valor perdido; esta distinción ayuda a decidir el método de tratamiento más adecuado. A continuación, se toman decisiones basadas en el porcentaje de ausencias y en su posible implicación: en casos extremos pueden descartarse registros o atributos con muchos datos faltantes, mientras que cuando se desea preservar la mayor parte de la información se recurre a la imputación mediante estadísticas simples como la media o la moda de la muestra, o bien a técnicas más avanzadas que estiman los valores ausentes a partir de la correlación entre variables.

A la hora de realizar la limpieza, hay que tener en cuenta el motivo de la ausencia original, ya que el hecho de faltar un dato a veces aporta información adicional sobre el comportamiento del usuario.

3.2.3 ELIMINACIÓN DE VARIABLES NO INFORMATIVAS

Una vez recogidos y organizados los datos, se realizó un proceso depurativo para eliminar variables que no aportaban valor de cara a los análisis. En primer lugar se excluyeron identificadores únicos tales como ID del producto, ID usuario o códigos transacción, ya que

si bien son necesarios para la operativa interna, no aportarían información útil para predecir comportamientos.

Además, se suprimieron también las columnas que disponían de un valor constante en todas las observaciones o cuyos valores de varianza fueran considerablemente bajos. Parámetros de esta clase, tal vez un campo que registre siempre lo mismo país o categoría en todos los registros, no aportarían distinción alguna a los datos y, por lo tanto, tampoco contribuyen a una mejora en la capacidad predictiva del modelo. Al conservarlas sólo se estaría aumentando la complejidad del procesamiento sin aportar ventajas.

Este proceso de filtrado ha ayudado a depurar la lista de datos para ofrecer sólo variables que tengan información útil cuyo impacto en el rendimiento del modelo de Machine Learning sea realmente significativo.

3.2.4 TRATAMIENTO DE VARIABLES CATEGÓRICAS

Existen 2 tipos de variables, las categóricas (o cualitativas) y numéricas (o cuantitativas). Los algoritmos de Machine Learning están destinados principalmente al manejo de datos numéricos, teniendo pues que transformar de alguna manera los que no son así.

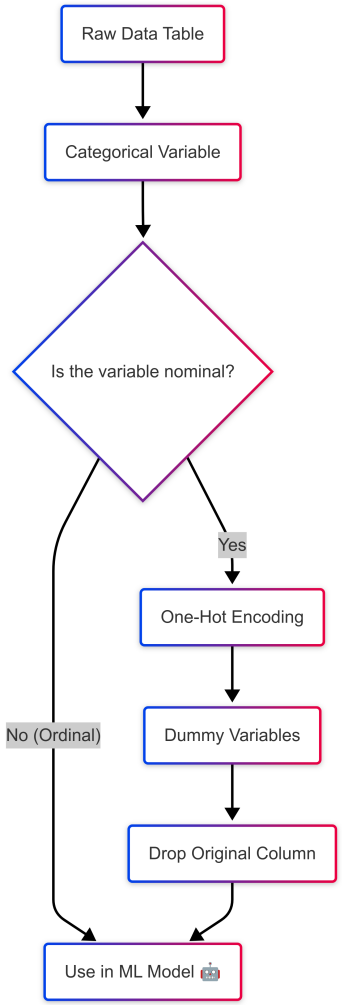

Ilustración 2: Tratamiento de variables categóricas. Elaboración propia.

En la Ilustración 2 se muestra el proceso de conversión de variables categóricas para aplicarlas en modelaciones de machine learning. Al encontrarnos en presencia de una variable categórica, lo primero que debemos establecer es si se trata de una variable nominal o ordinal. Las variables nominales son las que representan categorías diferentes sin una estructura jerárquica entre ellas, a lo que puede corresponder el color de un producto (“Rojo”, “Azul”, “Verde”) o país de origen (“España”, “México”, “Francia”). En dicha situación, no se puede asignar un valor numérico explícito a cada categoría ya que no hay una estructura jerárquica.

Para que se puedan llevar a cabo dichas variables nominales en un modelo, se aplica una técnica llamada codificación one-hot (del inglés *One-hot Encoding*). Esta consiste en agregar una nueva variable para cada una de las posibles categorías. Cada variable nueva, también denominada variable dummy, tiene un valor de 1 si el registro corresponde a dicha categoría, y valor 0 en caso contrario. Por ejemplo, supongamos que tenemos la variable "Color" que tiene tres categorias ("Rojo", "Azul", "Verde"). Se agregarían tres nuevas variables: "Color_Rojo", "Color_Azul" y "Color_Verde". Así que si un producto es azul, el registro contenga "Color_Azul"=1 y "Color_Rojo"=0, "Color_Verde"=0. También se puede eliminar una de las categorías, ya que si todos los valores son 0 quiere decir que pertenece a esta última, reduciendo el número de columnas en la base de datos. En este ejemplo, "Color_Verde" sería prescindible, ya que habría ceros en el rojo y el azul. De esta forma se evita la introducción de un orden ficticio entre categorías, y se puede interpretar correctamente cada opción como independiente de otras. Por último, se elimina la columna donde se encontraba la variable categórica original.

Por otro lado, en caso de que la variable sea ordinal, sí existe un orden lógico entre categorías, y ésta relación de orden se puede y ha de aprovechar. Un caso típico es la clasificación de corredores en una carrera: "Primero", "Segundo", "Tercero". En ese supuesto, es coherente asignarle a cada una de ellas números que respeten la jerarquía natural: "Primero"=1, "Segundo"=2, "Tercero"=3. Esta transcodificación permite a la modelo aprovechar la relación de orden entre las variables.

3.2.5 ESTANDARIZACIÓN

Es crucial que las variables numéricas, es decir, todas las variables después de tratar las variables categóricas se encuentren en escalas similares. Esto se debe a que la mayoría de los algoritmos de Machine Learning son sensibles a diferencias en la escala entre variables, lo que puede impactar en cuanto a rendimiento como a la estabilidad del modelo. Para corregir esta limitación, se utiliza una técnica denominada estandarización cuyo cometido es la supresión de la media de todas las variables y la división entre la desviación estándar. Se obtiene un conjunto de datos donde cada variable tiene una media cero y varianza uno que

aprovecha mejor el aprendizaje de los algoritmos y mejora la convergencia en procesos de optimización.

Este proceso es tan común que hay varias bibliotecas en Python que llevan a cabo automática y eficazmente este proceso. Entre estos se encuentra una de las más populares, que es Scikit-learn que en su módulo de *preprocessing* ofrece la función *StandardScaler* para aplicar esta transformación de forma sencilla en entrenamiento y en prueba [17].

Al utilizar correctamente la estandarización se consigue que todas contribuyan de forma proporcional en el aprendizaje del modelo sin que algunas con rangos más amplios dominen el proceso del entrenamiento.

3.2.6 PARTICIÓN ENTRENAMIENTO Y TEST

Otra acción que llevar a cabo es la separación del conjunto de datos en datos de entrenamiento (train) y datos de prueba (test). Se emplean los datos de entrenamiento para ajustar el modelo, es decir, en fin de aprender los correspondientes patrones que en los datos existan. Los datos de prueba, en su caso, se retienen con la finalidad de evaluar la performance del modelo habiéndolo entrenado, con datos que no ha visto antes. Dichas separación permite predecir en modo más realista cómo comportará en entornos actuales, evitando la posibilidad de sobreajuste o overfitting, con la que el modelo ajusta demasiado a datos de entrenamiento y adquiere poca capacidad de generalización. Para este trabajo, se utiliza una relación estándar del 80% de los datos para el entrenamiento del modelo y el 20% restante para probar la capacidad del mismo.

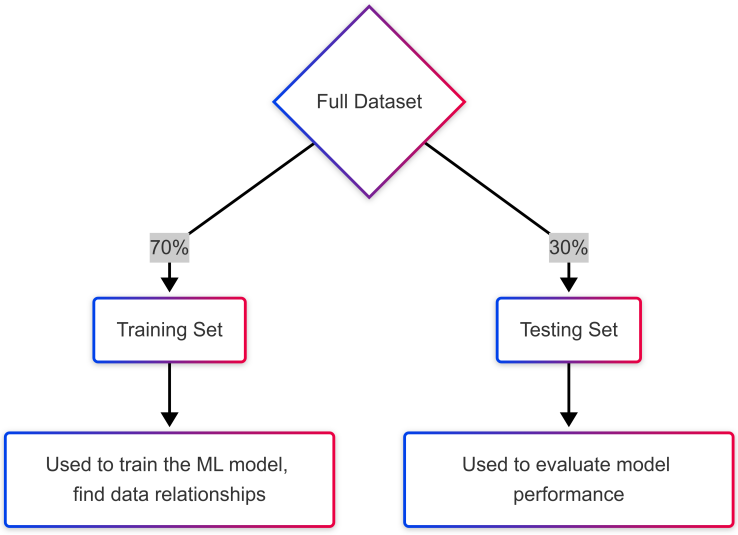

Ilustración 3: separación entre train y test. Elaboración propia.

3.2.7 AJUSTE DEL MODELO

En tareas de regresión, conocer el ajuste del modelo es fundamental para poder apreciar cuál es su capacidad para predecir valores continuos de forma acertada. Las medidas de ajuste más utilizadas en tareas de esta naturaleza son el Coeficiente de Determinación (R^2) y el Error Cuadrático Medio (RMSE). En este trabajo se usan estas dos métricas.

El RMSE (Root Mean Squared Error) es una medida de la magnitud típica de los errores de predicción que castiga más a los grandes errores. Su ecuación es:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Ecuación 5: Root Mean Squared Error, Raíz del Error Cuadrático Medio.

Donde y_i representa los valores reales, $\hat{y}_i$ las estimaciones realizadas por el modelo y n es la cantidad de observaciones presentes en el modelo. Un valor más bajo del RMSE representa un mejor ajuste. Por otra parte, el R^2 se utiliza para medir la fracción de varianza de la variable dependiente que se explica a través de las variables independientes del modelo. Su ecuación es:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y}_i)^2}$$

Ecuación 6: R^2.

Donde $\overline{y}_i$ es la media de los valores reales. Un valor de R^2 cercano a 1 indica que el modelo explica gran parte de la variabilidad de los datos, mientras que valores cercanos a 0 indican un pobre poder predictivo.

Estas métricas permiten comparar modelos de forma objetiva y seleccionar el que mejor equilibre precisión y capacidad de generalización.

3.2.8 GRIDSEARCH

Grid Search es un método de búsqueda de hiperparámetros mediante combinaciones para intentar encontrar la que mejor se ajusta a los valores, obteniendo un modelo final más robusto. Este espacio de pruebas se puede representar como un producto cartesiano entre los valores definidos de cada hiperparámetro.

$$\mathcal{H} = \{h_1, h_2, , h_n\} \Rightarrow GridSearch = h_1 \times h_2 \times \ldots \times h_n$$

Ecuación 7: GridSearch.

Donde:

- $\mathcal{H}$ es el conjunto de hiperparámetros.
- h_i son los posibles valores para el hiperparámetro i.

Se busca encontrar una combinación de hiperparámetros que minimice la función de pérdida del RMSE, tal y como se ve en la Ecuación 5. Se prueban todas y después se obtiene la combinación que mejor resultado ha obtenido.

La aplicación para este trabajo es encontrar los siguientes parámetros:

- Random Forest: número de árboles, profundidad máxima y variables por división.
- SVR: tipo de kernel, parámetro de regularización, margen de toleraciancia y parámetro del kernel.
- LightGBM: tasa de aprendizaje, máximo número de hojas, profundidad y regularización.

El uso de GridSearch implica un gran coste computacional, debido a que está creando una gran multitud de modelos y probándolos constantemente, pero asegura que hay una evidencia empírica por la cual se han elegido los parámetros. En este trabajo se utilizaron para el primer modelo, y se mantuvieron para el resto para hacer una comparativa en igualdad de condiciones.

3.2.9 MEDIDAS DE AJUSTE, UNDERFITTING Y OVERFITTING Y VALIDACION CRUZADA

Cuando se entrenan modelos de Machine Learning, uno de los grandes desafíos es encontrar un buen balance entre ajuste, cómo de bien explica el modelo los datos de entrenamiento, y generalización, la predicción de datos fuera del conjunto de entrenamiento. Dos de los problemas más frecuentes que se pueden presentar son underfitting y overfitting [18].

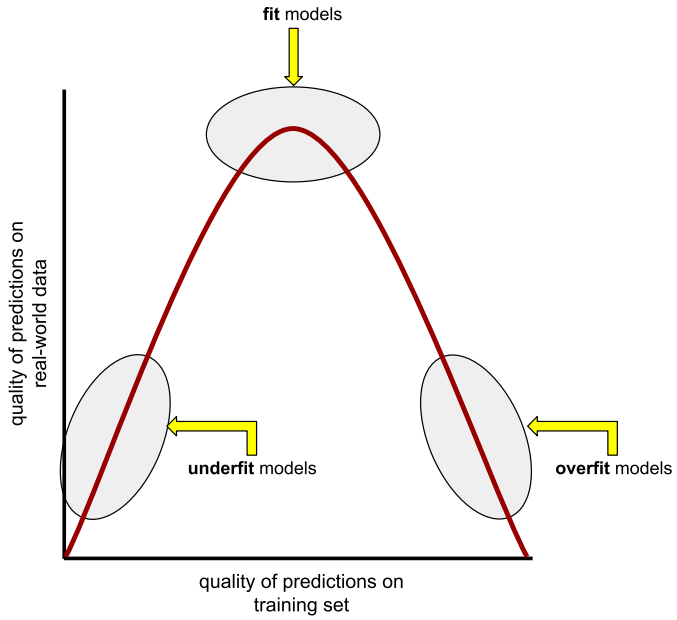

Ilustración 4: overfitting y underfitting. Fuente: Google Developers.

Underfitting ocurre cuando el modelo es demasiado simple para entender la estructura de los datos. Allí tanto el error de validación como el de formación es alto y el modelo no ha podido aprender patrones significantes. Es común para modelos demasiado básicos o por una elección de variables y características inapropiada.

En contrapartida, overfitting sucede cuando el modelo aprende demasiado de los datos de entrenamiento, incluyendo ruido o detalles específicos que no se generalizan muy bien a nuevos datos. Aunque el error de entrenamiento puede ser extremadamente pequeño, el error del conjunto de validación aumenta, lo que sugiere que el modelo no funcionará correctamente fuera de los datos en los que se entrenó el modelo. El punto intermedio es el ideal, como se puede ver en la Ilustración 4, denominado fit models.

Para detectar y prevenir dichos problemas se aplica la validación cruzada [19]. Se trata de una técnica que consiste en dividir los datos en varias particiones (o folds), en una parte de los datos se entrena el modelo y en otra se prueba de forma rotativa. Una de las más habituales en esta técnica es la validación cruzada k-fold, donde se divide el conjunto de datos en k grupos de tamaño aproximadamente igual; se entrena en k ocasiones dejando a uno de los grupos como de prueba y utilizando los demás para enseñar en cada una. La validación cruzada proporciona una estimación más fiable del rendimiento del modelo, así como una determinación rápida de un posible problema de subajuste o sobreajuste. También apoya la selección de los modelos que realmente generalizan bien y no funcionan simplemente en el conjunto de entrenamiento.

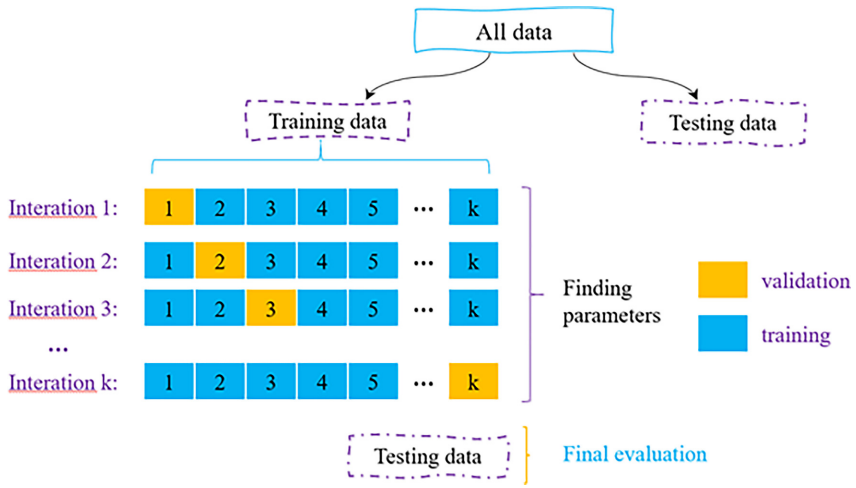

Ilustración 5: k-fold evaluation. Fuente: PLOS ONE.

3.2.10 SELECCIÓN DE MODELO

Una vez que se preprocesan los datos y se establecen qué medidas de evaluación se van a usar, el último paso en el proyecto es la elección del modelo. Se trata de comparar varios algoritmos de Machine Learning y determinar cuál de todos ellos aporta el mejor compromiso entre eficacia, interpretación, complejidad y capacidad de generalización.

A lo largo del proceso se han probado diferentes clases de modelos, desde más sencillos como la regresión lineal temporal hasta más sofisticados como los bosques aleatorios o los métodos de boosting. Cada uno tiene ventajas y limitaciones: los modelos lineales son fáciles de interpretar pero se vuelven erráticos en caso de una relación entre variables que sea no lineal, mientras que los modelos para la regresión basados en árboles aportan una mejor representación de relaciones complejas aunque son menos transparentes.

Esta mejora en precisión venía acompañada de un mayor tiempo de entrenamiento y una menor transparencia en la lógica de predicción, algo que puede ser crítico cuando se requiere justificar las decisiones del modelo ante equipos de negocio o dirección. Aunque existen criterios teóricos como el AIC (Akaike Information Criterion) y el BIC (Bayesian Information Criterion) [20] para comparar modelos desde un enfoque penalizado por complejidad, en este proyecto no se aplicaron sistemáticamente, ya que en este proceso de modelización, se priorizó la precisión predictiva como objetivo principal del estudio, centrándonos en el rendimiento real del modelo sobre datos no vistos. Dado el enfoque práctico de la aplicación, se consideró más relevante obtener el menor error posible que penalizar la complejidad. Aunque no se emplearon criterios como AIC o BIC, se evaluaron múltiples alternativas para garantizar la robustez del modelo final. Esta elección responde a una estrategia orientada a maximizar la utilidad predictiva en el contexto específico del problema.

Finalmente, la elección del modelo no es sólo función de su desempeño numérico, sino que también hay que tener en consideración cuestiones de carácter práctico tales como el tiempo de entrenamiento, la sencillez de ajuste de hiperparámetros, o la interpretabilidad de los

resultados, especialmente en un entorno de marketing donde conocer lo que decide el modelo puede importar lo mismo que su exactitud.

3.3 MODELOS EVALUADOS

3.3.1 SERIES TEMPORALES

Los modelos de series temporales se han diseñado para explorar y predecir datos que se desarrollan a lo largo del tiempo. Mientras los modelos clásicos del Machine Learning suponen que las observaciones son independientes unas de otras, los métodos de series temporales trabajan con datos ordenados cronológicamente y con una alta dependencia entre valores pasados y futuros. Se trata de un tipo de modelado fundamental en áreas donde los acontecimientos pasados están directamente relacionados con los posteriores, por lo que se probaron para este proyecto. De los modelos existentes, se ha optado por utilizar el ARIMA y derivados.

3.3.1.1 ARIMA

El modelo ARIMA (AutoRegressive Integrated Moving Average) es una herramienta estadística utilizada para la predicción de series temporales basada únicamente en los valores previos de la serie y en errores anteriores [21]. Su estructura se define mediante tres parámetros: *p*, *d* y *q*, que corresponden al orden del componente autorregresivo, el número de diferenciaciones aplicadas a la serie y el orden del componente de media móvil, respectivamente. Después de aplicar *d* diferenciaciones a la serie y_t, el modelo ARIMA(p, d, q) se expresa como:

$$y'_t = \phi_1 y'_t - 1 + \phi_2 y'_t - 2 + \cdots + \phi_p y'_t - p + \epsilon_t + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q}$$

Ecuación 8: Modelo ARIMA.

Donde:

- y'_t es la serie diferenciada de orden *d*.
- ϕ_i son los coeficientes autorregresivos.
- θ_j son los coeficientes de media móvil.
- ϵ_t es el error aleatorio en el instante *t*.

3.3.1.2 SARIMA

El modelo Seasonal ARIMA (SARIMA) extiende el ARIMA al incluir componentes estacionales, comunes en series con patrones periódicos (e.g., ventas mensuales). SARIMA añade cuatro parámetros: *P*, *D*, *Q,* y *s*, indicando el orden autorregresivo, la cantidad de diferenciaciones, y el orden de la media móvil estacional, así como el período estacional s (por ejemplo, 12 para series mensuales). Su formulación de SARIMA(p, d, q)(P, D, Q)$_s$ es la siguiente:

$$\Phi_P(B^S)\phi_p(B)\nabla^d\nabla_s^D y_t = \Theta_Q(B^S)\theta_q(B)\epsilon_t$$

Ecuación 9: Modelo SARIMA.

Donde:

- B es el operador de retardos: $By_t = y_{t-1}$.
- $\nabla^d = (1-B)^d$ representa la diferenciación regular.
- $\nabla_s^D = (1-B^S)^D$ representa la diferenciación estacional.
- $\phi_p(B)$ y $\Phi_P(B^S)$ son los polinomios de los componentes de AR (AutoRegresión), regulares y estacionales.
- $\theta_q(B)$ y $\Theta_Q(B^S)$ son los polinomios de los componentes de MA (Moving Average) regulares y estacionales.

3.3.1.3 SARIMAX

SARIMAX (Seasonal ARIMA with eXogenous variables) ofrece la ventaja adicional de poder agregar variables externas X_t que pueden afectar la serie temporal. Esto es útil, por ejemplo, si se quiere predecir ventas con influencia de campañas publicitarias o eventos especiales. La fórmula de SARIMAX(p, d, q)(P, D, Q)$_s$(X_t) es la siguiente:

$$y_t = SARIMA(y_t) + \beta X_t + \epsilon_t$$

Ecuación 10: Modelo SARIMAX.

β representa los coeficientes que vinculan las variables externas X_t con la serie objetivo y_t. SARIMAX es una herramienta útil a la hora de que las series temporales no sólo se basen en datos anteriores, y que tengan en cuenta las variables explicativas externas.

3.3.1.4 Estacionalidad, ACF y PACF

En el contexto de una serie temporal, la estacionalidad se refiere a patrones cíclicos que se repiten en intervalos de tiempo fijos, como los picos de ventas durante la temporada navideña. Este tipo de comportamiento no es compatible con la estacionariedad, ya que implica que las propiedades estadísticas de la serie, como la media o la varianza, cambian a lo largo del tiempo.

Una serie con estacionalidad no es estacionaria porque su patrón repetitivo altera el promedio de la serie con el paso del tiempo. Detectar esta estacionalidad es un paso clave antes de aplicar modelos como ARIMA, que requieren una estructura más estable, pasando por lo tanto a un modelo SARIMA más complejo. Para identificar la estacionalidad, se utilizan comúnmente dos herramientas gráficas: la Función de Autocorrelación (ACF) y la Función de Autocorrelación Parcial (PACF). Dichas herramientas se encuentran dentro de la librería *statsmodels* [22].

ACF mide la correlación entre los valores actuales y sus valores pasados en distintos retardos (lags). Si una serie presenta estacionalidad clara, mostrará picos significativos en los retardos que coincidan con los intervalos del ciclo estacional (por ejemplo, cada 12 meses si el ciclo es anual). Esta herramienta es útil para detectar la presencia general de patrones repetitivos.

PACF, por su parte, mide la correlación entre una observación y un retardo específico, eliminando la influencia de los valores intermedios. Esto permite identificar cuántos retardos directos afectan realmente al valor actual. En los modelos ARIMA, la PACF resulta clave para determinar el orden p del componente autorregresivo. Un ejemplo de gráficas que se usará más tarde, y su interpretación, se encuentra debajo.

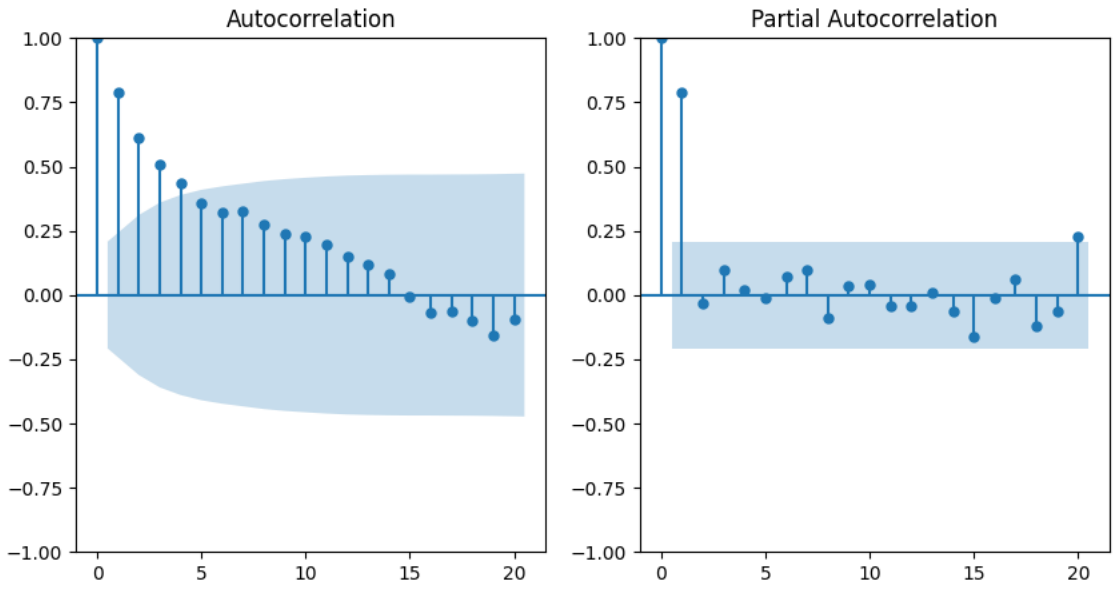

Ilustración 6: Gráfico ACF y PACF. Elaboración propia.

En el primer gráfico, ACF, el primer valor es 1 indicando que la serie está totalmente correlada con ella misma, como es de esperar. Después, se ve un descenso gradual a medida que aumentan los lags, indicando la presencia de una estructura autorregresiva ya que los valores pasados influyen en los futuros. Muchos de los lags están por encima del área sombreada, el intervalo de confianza, indicando que las correlaciones son significativas. La serie no es aleatoria y tiene memoria, una componente autorregresiva importante. Si el descenso fuese más abrupto, sería indicación de que no hay componente autorregresiva.

En el segundo gráfico, el PACF, se comprueba el componente autorregresivo que se ha descubierto en el paso anterior. Se puede observar cómo el primer valor no está dentro del intervalo de confianza, con lo que se podría decir que la serie puede ser modelada con una componente AR(1), el valor actual depende directamente sólo del valor anterior, sin necesidad de más retardos. Por lo tanto, la variable p valdrá 1.

Ambas herramientas se utilizan conjuntamente para decidir si una serie requiere una diferenciación estacional y para definir los parámetros correctos del modelo SARIMA. Cuando se observan valores de autocorrelación significativamente distintos de cero en

retardos estacionales (como el retardo 12 en una serie mensual), es una señal clara de estacionalidad que debe abordarse antes del modelado.

3.3.1.5 Diferenciación, Test de Dickey-Fuller Aumentado (ADF)

Casi todos los modelos de series temporales, como ARIMA, asumen la estacionariedad de la serie, o sea, la constancia en el tiempo de sus propiedades estadísticas (tanto la media como la varianza). Pero, en el contexto real, la mayor parte de las series no son estacionarias, debido a las tendencias o estacionalidades. La técnica más común empleada para estabilizar una serie no estacionaria es la diferenciación, eliminando tendencias lineales y estabilizando la media. Esto nos dará el parámetro *d* de ARIMA, o cuántas veces habrá de ser derivada.

Se empieza con la hipótesis nula de que la serie tiene una raíz unitaria, por lo que tendría no estacionariedad, de la siguiente forma:

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \delta_1 \Delta y_{t-1} + \cdots + \delta_p \Delta \mathrm{y}_{t-p} + \epsilon_t$$

Ecuación 11: Test de Dickey-Fuller Aumentado (ADF).

Donde:

- Δy_t es la diferencia del valor actual con el anterior.
- α es el intercepto, permite capturar una media no nula en la serie.
- βt es el término que permite que el modelo sea lineal en la serie original.
- $\delta_p \Delta \mathrm{y}_{t-p}$ son los retardos de las primeras diferencias, para capturar la dinámica de corto plazo, haciendo que los errores no estén correlacionados, el parámetro *p*.
- γ es el coeficiente clave que se prueba.

Si $\gamma<0$ con significancia estadística $p<0.05$, se rechaza la hipótesis de la raíz unitaria y se acepta la estacionariedad de la serie, por lo que $d=0$. Si $p\geq0.05$, se debería aplicar alguna diferenciación adicional, siendo $d\neq0$.

3.3.2 RANDOM FOREST

Random Forest es un método supervisado de aprendizaje inductivo fundado sobre el uso combinado de varios árboles de decisión. Fue desarrollado por Leo Breiman el año 2001 [23] como optimización del uso individual de los árboles, integrando varias modelos débiles (árbol) para producir un modelo fuerte, exacto y menos sujeto al sobreajuste.

La base son los árboles de decisión, dividiendo el espacio de características en regiones homogéneas para después utilizar medidas de evaluación de modelos para seleccionar las variables óptimas. Para los problemas de regresión, se utilizan métricas de errores comunes, como el RMSE presente en la Ecuación 5, o el MSE:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Ecuación 12: Median Squared Error, Error cuadrático medio.

Donde, como en el RMSE, y_i es el valor real que se intenta predecir y $\hat{y}_i$ es el valor que predijo el modelo, con n siendo el número de muestras de las que se dispone para entrenar el modelo. El árbol divide el espacio de entrada en subconjuntos que minimizan esta métrica en cada división.

3.3.2.1 Bagging, bootsrap aggregation.

Random Forest se basa en hacer bagging (Bootstrap Aggregation), consistente en hacer divisiones aleatorias del conjunto de entrenamiento mediante el muestreo sin remplazo, y entrenar un regresor en cada una de esas divisiones, reducienzo la varianza del modelo combinado sin aumentar mucho el sesgo, debido a la diversidad entre árboles. Dado un conjunto de datos original:

$$D = \{(x_i, y_i)\}_{i=1}^{n}$$

Ecuación 13: conjunto de datos original.

Se generan B muestras de Bootstrap:

$$D^1,\ D^2, \ldots, D^B$$

Ecuación 14: Divisiones del conjunto de datos por bootstrapping.

Y sobre cada una se entrena un árbol de regresión $h_b(x)$. La predicción final será un promedio de estas salidas.

$$\hat{f}RF(x) = \frac{1}{B}\sum b = 1^B h_b(x)$$

Ecuación 15: Obtención de la salida final por agregación.

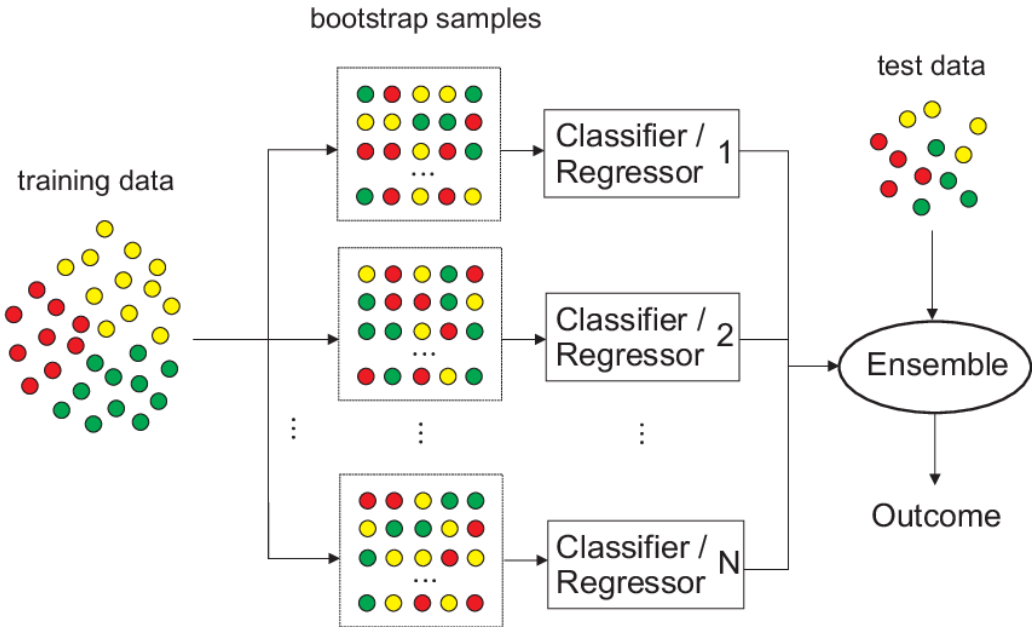

Ilustración 7: Bagging. Fuente: ResearchGate.

Random Forest también aleatoriza la selección de variables. Esto quiere decir que, en lugar de usar todas las variables disponibles para cada división de Bootstrap, elige un subconjunto menor que la totalidad. Esto promueve independencia entre árboles, su diversidad, y por lo tanto su capacidad de generalización en el modelo.

3.3.2.2 Ventajas y limitaciones.

Random Forest tiene un bajo error de generalización por los métodos que utiliza, que se han explicado antes. A diferencia de un solo árbol de regresión, que tiene un alto riesgo de overfitting, el promediado de los resultados de la totalidad de los árboles hace que el sobreajuste no sea un problema generalizado. También se puede estimar la importancia de las variables mediante medidas de disminución del error o reducción de impureza, dando

más información a un equipo de marketing que tenga que tomar decisiones a partir de un modelo.

Por otra parte, hay un elevado coste computacional asociado con este modelo, especialmente en comparación con otros más simples. Tampoco es sencillo obtener una interpretabilidad directa, ya que el modelo final es una combinación de muchos árboles con sus decisiones internas.

3.3.3 RNN, REDES NEURONALES RECURRENTES

Las Redes Neuronales Recurrentes (RNN) son un tipo de redes neuronales pensadas específicamente para operar sobre datos secuenciales. Mientras que las redes neuronales convencionales tratan una entrada de manera independiente, las RNN poseen una estructura interna que les permite almacenar información de entradas previas, haciendo que sean perfectas para aplicaciones que puedan tener alguna estructura temporal, tengan procesamiento de lenguaje natural o análisis de un comportamiento secuencial de usuarios.

3.3.3.1 Arquitectura básica

A diferencia de sólo recibir una entrada actual x_t, también se recibe la salida del paso anterior h_{t-1}, generando dependencias temporales entre elementos de la secuencia. Esta es la razón por la cual se seleccionó este tipo de redes neuronales frente a otras opciones. La activación en un paso t es la siguiente:

$$h_t = \sigma(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$$

Ecuación 16: Activación de las redes neuronales recurrentes.

Donde:

- h_t es el estado oculto en el tiempo *t*.
- x_t es la entrada en el tiempo es la entrada en el tiempo *t.*
- W_{xh} y W_{hh} representan las matrices de peso que conectan entrada y salida del estado previo al siguiente.
- b_h es el sesgo.
- σ es la función de activación.

La salida de la red, por lo tanto, se obtiene con la siguiente fórmula.

$$y_t = W_{hy} h_t + b_y$$

Ecuación 17: salida de una red neuronal recurrente.

3.3.3.2 Ventajas y limitaciones.

Como se ha ido mencionando, las redes neuronales recurrentes comparten los mismos pesos a lo largo del tiempo, adaptando las predicciones que realiza en función del contexto secuencial, algo que podría parecer útil teniendo en cuenta que estos datos contienen un campo temporal.

De todas maneras, esta estructura también introduce el problema del desvanecimiento o explosión del gradiente, sobre todo con dependencias temporales de largo plazo. El desvanecimiento ocurre cuando las derivadas parciales que llevan al cálculo de gradientes se vuelven extremadamente pequeñas, tendiendo a cero a medida que se propaga hacia el pasado:

$$\frac{\partial \mathcal{L}}{\partial \theta} \propto \prod_{t=1}^{T} \frac{\partial h_t}{\partial h_{t-1}}$$

Ecuación 18: Desvanecimiento en RNNs.

También está el problema de la explosion del gradiente, cuando las variables involucradas en el cálculo del gradiente son mayores que uno, haciendo que las multiplicaciones sucesivas lleven a un número que va creciendo exponencialmente, haciendo que vaya tendiendo al infinito:

$$\prod_{t=1}^{T} \left| \frac{\partial h_t}{\partial h_{t-1}} \right| \rightarrow \infty$$

Ecuación 19: explosión de gradiente en RNNs.

Para solucionar estos problemas se han ideado una serie de soluciones, como las variantes LSTM (Long Shor-Term Memory) y las GRU (Gated Recurrent Unit) [24], pero no han sido incluidas en este trabajo, aunque sí formarían parte de una buena exploración futura.

3.3.4 SVR

Support Vector Regression (SVR) es una variante del algoritmo de Support Vector Machine (SVM), inicialmente creado para la clasificación y después adaptado para su uso en problemas de regresión. A diferencia de otros modelos cuyo, objetivo es minimizar estrictamente la diferencia entre la prevista y la real, SVR intenta crear una función $f(x)$ capaz de aproximar los datos de entrada x con la mayor precisión posible, sin preocuparse por pequeñas diferencias dentro de un margen aceptable ε:

$$f(x) = \langle w, x \rangle + b$$

Ecuación 20: Función del SVR.

Donde:

- w es el vector de pesos.
- x es la variable de entrada.
- b es el término independiente.
- $\langle w, x \rangle$ es el producto escalar.

3.3.4.1 Función de pérdida ε-sensible

Como se apuntaba antes, SVR no penaliza los errores dentro de un margen ε, pero sí lo hace de manera lineal aquellos que lo superan:

$$L_{\mathrm{e}}\big(y, f(x)\big) = \begin{cases} 0, & |y - f(x)| \leq \mathrm{e} \\ |y - f(x)| - \mathrm{e}, & otherwise \end{cases}$$

Ecuación 21: Penalización ε-sensible de SVR.

Este tratamiento de márgenes tolera desviaciones creadas por ruido, generando robustez frente a él y evitando el sobreajuste.

3.3.4.2 Problema de optimización

SVR intenta optimizar la siguiente función:

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*)$$

Siendo:

- ξ_i , ξ_i^* las variables de holgura que permiten errores mayores que ε.
- C el parámetro de regularización que monitoriza la complejidad del modelo y tolerancia al error.

Sujeto a 3 restricciones:

$$y_i - \langle w, x_i \rangle - b \leq \varepsilon + \xi_i$$

Ecuación 22: Restricción del margen inferior de SVR.

Asegurando que la predicción que esté por debajo de y_i, el valor que hay que intentar replicar, la diferencia no sea mayor que ε teniendo en cuenta la holgura ξ_i.

$$\langle w, x_i \rangle + b - y_i \leq \varepsilon + \xi_i^*$$

Ecuación 23: Restricción del margen superior de SVR.

Asegurando que si la predicción está por encima de y_i, la diferencia no se pase del margen ε teniendo en cuenta la holgura ξ_i^*.

$$\xi_i + \xi_i^* \geq 0$$

Ecuación 24: Restricción de positividad de holguras de SVR.

Asegurando que las variables de holgura son positivas, nunca compensando negativamente.

3.3.4.3 Uso de kernels

SVR puede hacer uso de funciones kernel para transformar datos a espacios de mayor dimensión donde las relaciones sean lineales entre variables. Estos son 3 de los kernels comunes:

$$K(x_i, x_j) = x_i \, x_j$$

Ecuación 25: Kernel lineal SVR.

Sólo calcula el product escalar entre variables originales, ideal para datos que se sospeche que tienen una relación lineal, y el más rápido de entrenar.

$$K(x_i, x_j) = (\langle x_i, x_j \rangle + 1)^d$$

Ecuación 26: Kernel Polinomial SVR.

Este kernel eleva el producto escalar a una potencia (grado del polinomio). Con él, el modelo puede aprender relaciones no lineales con curvaturas suaves o complejas, dependiendo del grado del polinomio se elija.

$$K(x_i, x_j) = exp(-\gamma \|x_i - x_j\|^2)$$

Ecuación 27: Kernel RBF o gaussiano.

Este es el más flexible y por lo tanto el más popular. Funciona muy bien cuando no hay una idea clara de la forma de la relación entre los datos. El RBF mide la distancia entre puntos y le da más peso a los que están más cerca. Es ideal para capturar patrones no lineales complejos.

3.3.4.4 Ventajas y limitaciones

Una de las principales ventajas del modelo SVR es que se adapta muy bien cuando queremos hacer predicciones sin que los datos más extremos o ruidosos afecten demasiado. Gracias a ese margen de tolerancia que tiene, puede centrarse en lo importante y no en cada pequeño desvío. También funciona muy bien cuando las relaciones entre las variables no son lineales, especialmente si usamos kernels adecuados. Ahora bien, no todo son puntos a favor: entrenarlo puede llevar tiempo si el conjunto de datos es grande, y afinar los parámetros para que funcione realmente bien puede requerir bastantes pruebas. Además, no es de los modelos más intuitivos de entender si lo que buscamos es explicar fácilmente por qué toma ciertas decisiones.

3.3.5 LIGHTGBM

LightGBM (Light Gradient Boosting Machine) es un algoritmo muy rápido y eficiente para resolver problemas de predicción, ya sea de regresión o clasificación. Se basa en una técnica llamada boosting que lo que hace es construir varios árboles de decisión de forma secuencial, de tal manera que cada nuevo árbol intenta corregir los errores que cometieron los anteriores, como puede verse en la Ilustración 8 [25]. A diferencia de otros métodos que tratan de encontrar una única solución buena de una vez, LightGBM va ajustando poco a poco hasta encontrar un modelo que, en conjunto, funcione muy bien [26].

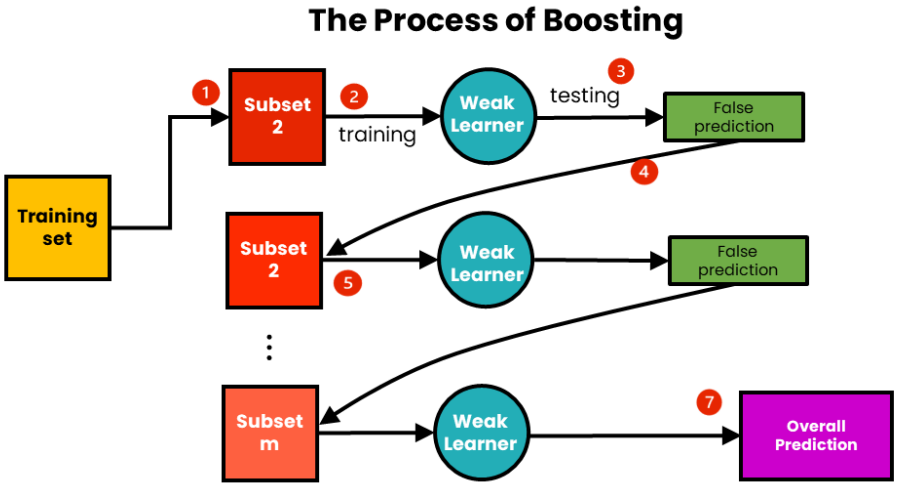

Ilustración 8: Algoritmo de Boosting. Fuente: Medium.

Una de las principales razones por las que LightGBM es tan popular es su velocidad. A diferencia de otros algoritmos similares (como XGBoost), LightGBM no construye los árboles hoja por hoja, sino que los expande hoja a hoja (leaf-wise). Esto permite encontrar divisiones más precisas con menos recursos. Además, utiliza técnicas como el uso de histogramas para acelerar aún más el proceso y reducir el uso de memoria, lo cual se agradece mucho cuando se trabaja con conjuntos de datos grandes o con muchas variables.

Otra ventaja clara es que soporta datos categóricos directamente, sin necesidad de convertirlos previamente en dummies o hacer one-hot encoding, lo que simplifica mucho el preprocesamiento. En este trabajo sí se convierten previamente porque el resto de los modelos sí precisan este pre-procesado. También maneja muy bien los valores faltantes y permite ajustar muchos parámetros para afinar el rendimiento del modelo.

También tiene limitaciones. El hecho de que crezca los árboles de forma leaf-wise puede hacer que se sobreajuste si no se controla bien. Por eso, es importante ajustar parámetros como el número máximo de hojas o la profundidad máxima del árbol para que no aprenda demasiado de los datos de entrenamiento y falle con datos nuevos.

Capítulo 4. TRABAJO DESARROLLADO

4.1 TECNOLOGÍA EMPLEADA

En este apartado se describe el lenguaje de programación y las librerías que hemos utilizado para para llevar a cabo el proceso de análisis y modelización que nos ocupa.

4.1.1 PYTHON

Python es un lenguaje de programación de alto nivel ampliamente utilizado en el análisis de datos y el desarrollo de modelos de Machine Learning e Inteligencia Artificial [27]. Su sintaxis clara y su extensa biblioteca de paquetes lo convierten en una herramienta ideal para el procesamiento de datos y la implementación de algoritmos de aprendizaje automático.

4.1.2 PANDAS

Pandas es una biblioteca de Python que proporciona estructuras de datos y herramientas de análisis de datos de alto rendimiento [28]. Es especialmente útil para la manipulación y el análisis de datos tabulares, como los datos de ventas y marketing.

4.1.3 SCIKIT-LEARN

Scikit-learn es una biblioteca de Python que incluye una amplia gama de algoritmos de aprendizaje automático [29]. Es utilizada para tareas como la clasificación, regresión y agrupamiento de datos, y es fundamental para el desarrollo de modelos predictivos en este proyecto. Como se ha mencionado con anterioridad, esta librería también es usada para la regularización [17].

4.1.4 TENSORFLOW Y KERAS

TensorFlow es una biblioteca de código abierto para el aprendizaje automático desarrollada por Google [30]. Keras es una API de alto nivel para TensorFlow que facilita la construcción y el entrenamiento de modelos de redes neuronales profundas [31].

4.1.5 PMDARIMA

La librería pmdarima es un recurso de Python que proporciona herramientas para el modelado de series temporales, incluyendo la implementación de auto ARIMA [32]. Auto ARIMA es un algoritmo que automatiza el proceso de ajuste de modelos ARIMA, SARIMA y SARIMAX.

4.1.6 LIGHTGBM

LightGBM [33] es una biblioteca de aprendizaje automático que se especializa en algoritmos de boosting [33]. Es utilizada para entrenar modelos de regresión y clasificación de manera eficiente y rápida, y es particularmente útil para manejar grandes volúmenes de datos.

4.2 CONJUNTOS DE DATOS

Una de las grandes dificultades a la hora de desarrollar un proyecto de predicción sobre comportamiento de clientes o rendimiento de campañas es el acceso a datos reales. Esta dificultad es más notable por tratarse de datos de marketing relacionado con campañas publicitarias que podrían revelar a la competencia las estrategias y los recursos que dedican estas empresas. Por estos motivos, no es sencillo que las empresas compartan públicamente esta información Por ello, se ha optado por utilizar tres conjuntos de datos distintos, que se han utilizado anteriormente en trabajos publicados [34]. Cada uno de estos datasets tiene características únicas que aportan riqueza y variedad al análisis. Esta decisión responde a la necesidad de trabajar con datos que reflejen distintos escenarios reales, lo cual permite validar la versatilidad de los modelos aplicados y evaluar su capacidad para adaptarse a diferentes estructuras y objetivos de negocio. De esta manera, estos son los tres datasets con los que se trabajó, y en paréntesis el nombre que se les dará de ahora en adelante:

- Datos de ventas en uno de los mayores portales de e-commerce mundial, Amazon, en India (**Amazon India**) puesto que era el único dataset de este tipo que hay publicado en Internet a día de hoy [35].
- Perfiles y comportamientos de consumidores (**Clientes**), donde se analizan sus perfiles y las compras que hicieron [36].

- Métricas publicitarias multicanal (**Anuncios**), abarcando desde la predicción de compras hasta el análisis de campañas de marketing digital [37].

Esta diversidad no solo refuerza la aplicabilidad práctica del trabajo, sino que también simula un entorno realista donde los datos son heterogéneos y presentan desafíos distintos. Estos tres datasets fueron elegidos entre muchos otros por la aparente naturalidad de sus datos, frente a otros que se encontraron que eran de naturaleza artificial, aparte de representar información de la que pudiese disponer un departamento de marketing de una empresa.

A continuación, se proporciona una breve explicación del contenido de cada dataset. En el Anexo II se proporciona un análisis más detallado de todas las variables de cada dataset.

4.2.1.1 Amazon India

Como se detallaba antes, esta base de datos contiene información de las ventas en Amazon de una empresa textil india, basado en los productos de los que se habla. Presenta una estructura, que incluye tanto variables numéricas (fechas, importes y cantidades) como variables categóricas que, para mejorar su procesamiento en un proceso de modelización, se han adaptado en forma de variables dummies. En la Ilustración 9 se presenta la matriz de correlación de las variables que se han incluido en los modelos.
En este caso, se intenta predecir la variable ***Amount***, que contiene información sobre cuánto gasta el cliente.

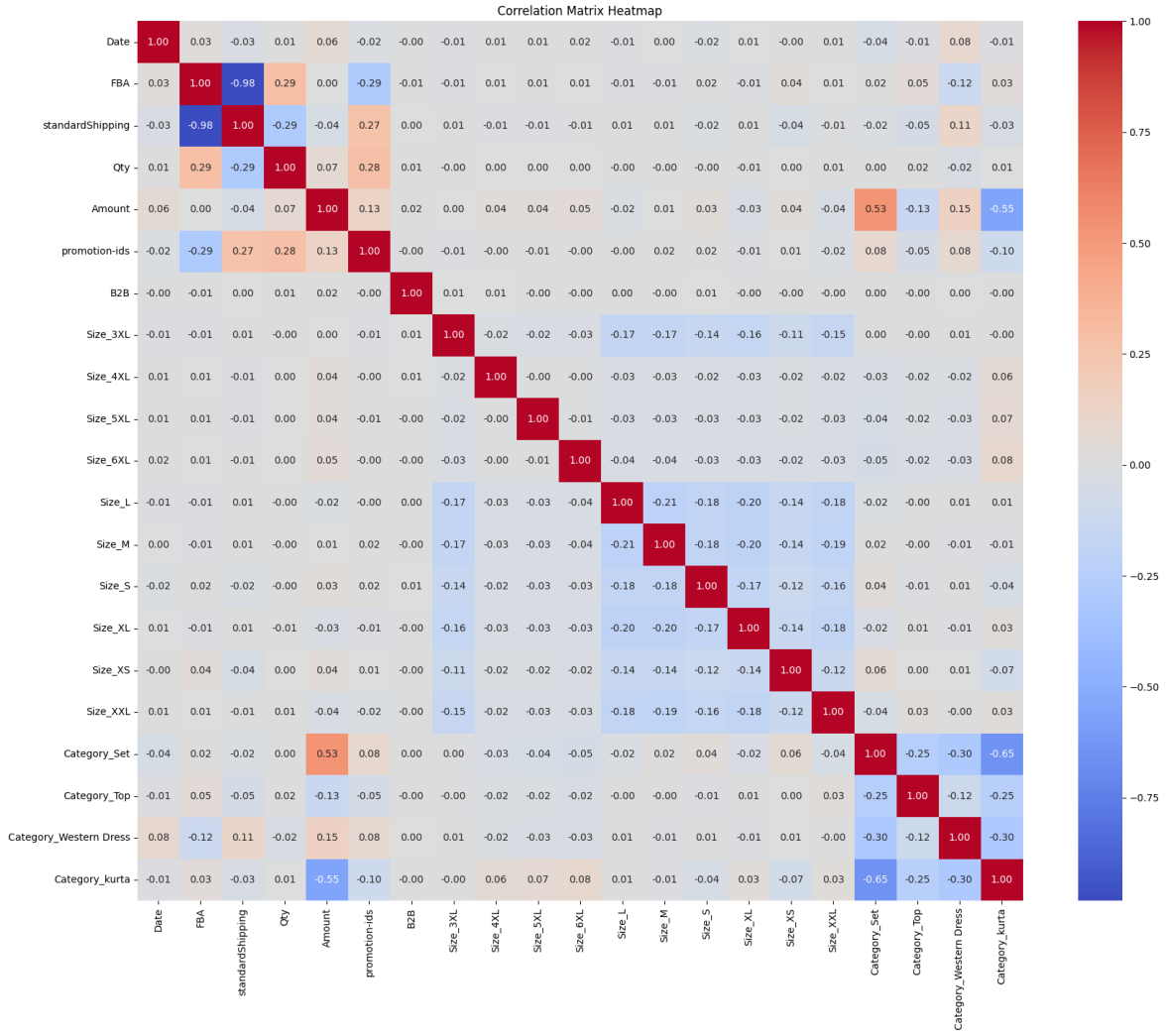

Ilustración 9: Matriz de correlación Amazon India.

En líneas generales, se puede observar cómo no hay una colinealidad muy alta en las variables, lo cual es algo deseable y positivo de cara a la modelización. Estar muy correlacionado podría implicar que hubiese dos variables que explican lo mismo.

De entre las correlaciones fuertes, se pueden destacar las siguientes:

- Hay una correlación positiva esperable entre las variables Qty (quantity) y Amount, el precio total de la venta. A mayor cantidad vendida, más alto será el precio total de venta.
- Promotion-ids, una variable que ha sido transformada a un 1 si se aplica alguna promoción y 0 si no, está:

  - Correlada negativamente con FBA (Fullfilled By Amazon), indicando que si Amazon gestiona el pedido habrá menos promociones aplicadas. Por el contrario, si es standardShipping (gestionado por la empresa) sí que suelen aplicar más promociones.
  - Ligeramente correlada positivamente con Amount, lo cuál indica que la existencia de promociones hace que haya un mayor gasto total.
- Correlaciones grandes para categorías que son mutualmente excluyentes, las que son de dummies, normal debido a que sólo suelen pertenecer a una categoría, por ejemplo, la talla. Si hubiese alguna correlación más baja podría indicar algún problema, pero no es el caso.

4.2.1.2 Clientes

Para esta base de datos hay una buena variedad de variables continuas, categóricas transformadas mediante one-hot encoding y binarias, dando mucha información sobre los clientes que permiten entenderlos de manera completa. La matriz de correlación de la Ilustración 10 muestra las conexiones entre las que se han introducido en los modelos, algo que se explica brevemente más adelante. El listado completo de variables antes del preprocesado se puede encontrar en el Anexo II, Anuncios.

En este caso, se intenta predecir una variable que acumula la ***cantidad de compras totales***.

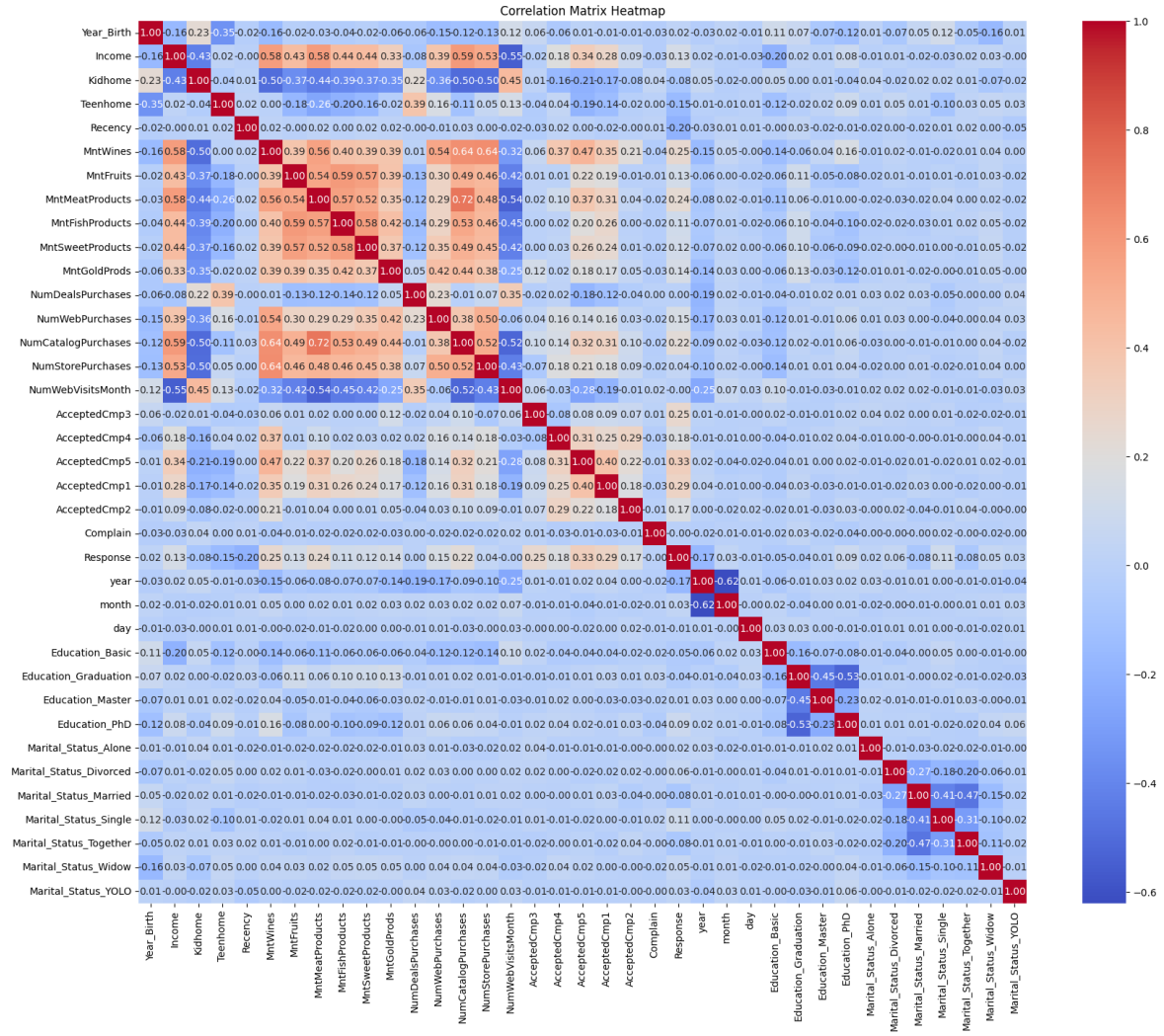

Ilustración 10: Matriz de correlación Clientes.

A continuación, se analizan algunas de las correlaciones destacadas:

- La variable Income muestra correlaciones moderadas y positivas con casi todos los tipos de gasto (MntWines, MntMeatProducts, MntGoldProds…), lo que tiene sentido: los clientes con mayores ingresos tienden a gastar más.
- Las variables relacionadas con el número de compras (NumWebPurchases, NumCatalogPurchases, NumStorePurchases) están bastante correlacionadas entre sí, indicando que los clientes activos lo son en varios canales a la vez.

- Entre las campañas (AcceptedCmp1–5, Response), se observan correlaciones positivas entre sí, lo que sugiere que quienes aceptan una campaña tienden a participar también en otras, reflejando un perfil más receptivo al marketing.
- La fecha (a través de year, month, day) no parece tener impacto fuerte sobre las demás, lo cual sugiere que podrían ser menos relevantes para ciertos modelos predictivos. Algo que se verá más adelante con los modelos de series temporales.

Este dataset presenta una buena base para entrenar modelos de predicción sobre comportamiento de compra o respuesta a campañas. Las correlaciones reflejan patrones coherentes y no se observan señales graves de multicolinealidad, aunque sí hay redundancia entre algunas variables relacionadas con canales de compra o tipos de gasto, que podrían combinarse o analizarse por separado según el objetivo del análisis. Estas combinaciones se explicarán más adelante.

4.2.1.3 Anuncios

Por último, el dataset de Anuncios, centrado en campañas de publicidad digital y métricas asociadas a la inversión, creatividad y resultados de anuncios. A través de la matriz de correlación de la Ilustración 11: Matriz de correlación Anuncios. se pueden observar relaciones claras entre variables presupuestarias, rendimiento en medios y configuración de campañas.

Al haber varias variables que se podrían predecir en este dataset, se decidió probar a predecir ***approved_budget, impressions, clicks y media_cost_usd***, sacando cada una de esas del entrenamiento y manteniendo las demás para cada caso.

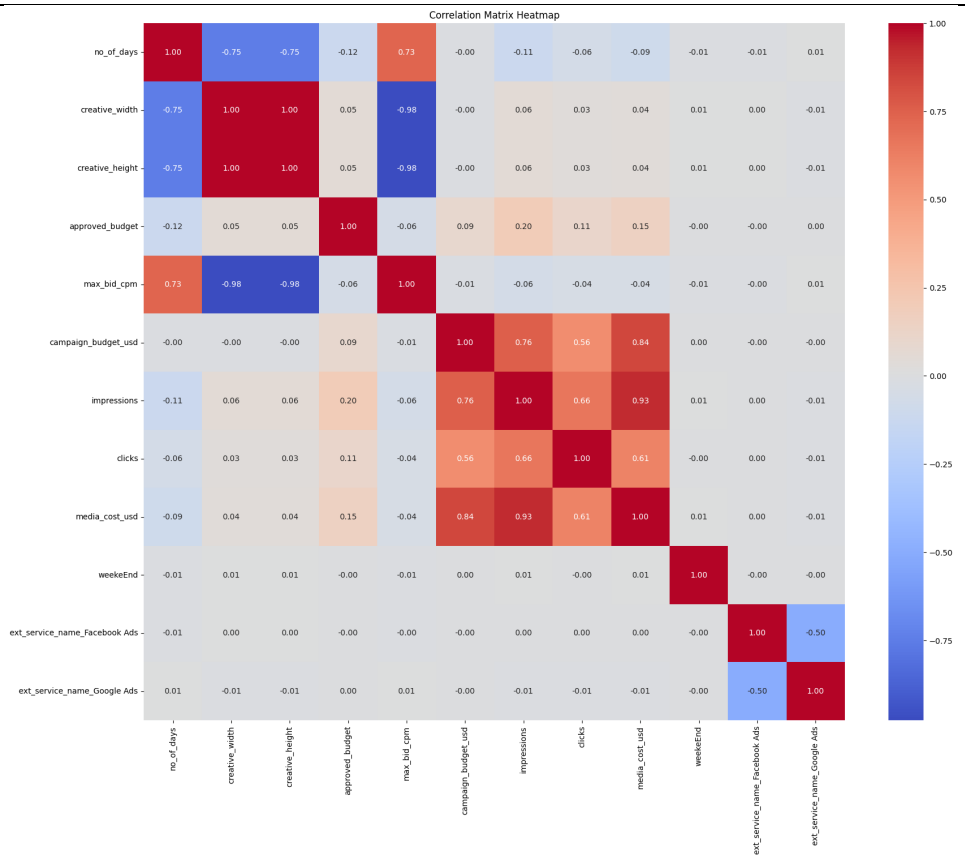

Ilustración 11: Matriz de correlación Anuncios.

Las correlaciones más destacadas en este caso serían las siguientes:

- Hay una fuerte correlación entre campaign_budget_usd y variables como impressions (0.76), clicks (0.66) y media_cost_usd (0.84). Esto sugiere que a mayor presupuesto asignado, mayor es el alcance y la interacción obtenida, como es esperable en campañas bien estructuradas.
- clicks y impressions también presentan una relación fuerte (0.86), lo que refleja un comportamiento lógico de que mayor visibilidad suele traducirse en más interacciones.
- Se presenta una fuerte correlación negativa entre el tamaño de los anuncios (creative_width y creative_height) y la max_bid_cpm (-0.96 y -0.99), lo que podría estar indicando que los anuncios de mayor tamaño se asocian a pujas más bajas, posiblemente por limitaciones de formato o estrategia de colocación.

- Las variables categóricas codificadas como dummies (ext_service_name_Facebook Ads, ext_service_name_Google Ads) muestran una relación negativa entre sí (-0.50), como es lógico por tratarse de plataformas excluyentes.
- La variable weekEnd no presenta correlaciones fuertes con el resto, lo que sugiere que el día de la semana no influye de forma clara sobre el resto de métricas en este subconjunto.

El dataset muestra patrones esperados que confirman relaciones clave entre inversión y rendimiento, con algunos matices interesantes en el comportamiento según el tipo de plataforma o características creativas del anuncio. Esto lo convierte en un buen candidato para modelar el impacto de decisiones presupuestarias sobre resultados de campañas.

4.3 ESTIMACIÓN BASADA EN LOS MODELOS

En este apartado se recogen los resultados obtenidos tras aplicar diferentes modelos de predicción sobre los tres conjuntos de datos presentados anteriormente. El análisis se ha dividido en dos bloques principales: por un lado, se evalúan modelos específicamente diseñados para series temporales, y por otro, se aplican los algoritmos de Machine Learning más frecuentemente empleados en procesos de modelización sobre los mismos datos, una vez adaptados. En ambos casos, la comparación entre modelos se ha realizado utilizando las métricas RMSE (Root Mean Squared Error) representado en la Ecuación 5, y R^2 representado en la Ecuación 6. Ambas métricas permiten valorar tanto la precisión de las predicciones como el grado en que los modelos explican la variabilidad de los datos. Además, se ha dedicado un apartado específico al análisis de la importancia de las variables, especialmente relevante en modelos como Random Forest, donde es posible identificar qué características del dataset tienen mayor peso a la hora de realizar las predicciones. Esta combinación de evaluación cuantitativa y análisis interpretativo proporciona una visión más completa del rendimiento y utilidad práctica de cada enfoque aplicado.

4.3.1 SERIES TEMPORALES

4.3.1.1 Amazon India

En este dataset en particular, debido al escaso número de observaciones una vez se agregan los datos para poder hacer un análisis de serie temporal, se eligió una partición entre entrenamiento y test de 80%-20%, en atención a que la métrica r^2 mejoraba de manera sustancial, indicando que aprendía mejor.

Para mantener una comparativa similar, los modelos de Machine Learning posteriores que tratan con éste también utilizan esa partición.

Después se aplica el algoritmo de auto ARIMA, con los siguientes resultados:

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                      y   No. Observations:                   62
Model:               SARIMAX(0, 1, 0)   Log Likelihood                -777.490
Date:                Sun, 11 May 2025   AIC                           1556.981
Time:                        16:06:16   BIC                           1559.092
Sample:                  04-01-2022   HQIC                          1557.808
                       - 06-01-2022
Covariance Type:                 opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
sigma2       6.78e+09   8.94e+08      7.586      0.000    5.03e+09    8.53e+09
===================================================================================
Ljung-Box (L1) (Q):                   0.26   Jarque-Bera (JB):                 13.41
Prob(Q):                              0.61   Prob(JB):                          0.00
Heteroskedasticity (H):               0.43   Skew:                             -0.77
Prob(H) (two-sided):                  0.06   Kurtosis:                          4.70
===================================================================================
```

Código 1: SARIMAX Amazon India.

Como puede comprobarse en el código, el modelo determina que la mejor manera de ajustarse a los datos es mediante una variable constante. Este es un ajuste realmente pobre, que podrá verse cuando se hable de los resultados en la Tabla 3.

La partición de entrenamiento y test está presente en la Ilustración 12. Es aquí donde puede comprobarse visualmente el resultado del modelo, y su completa incapacidad de capturar cualquier patrón.

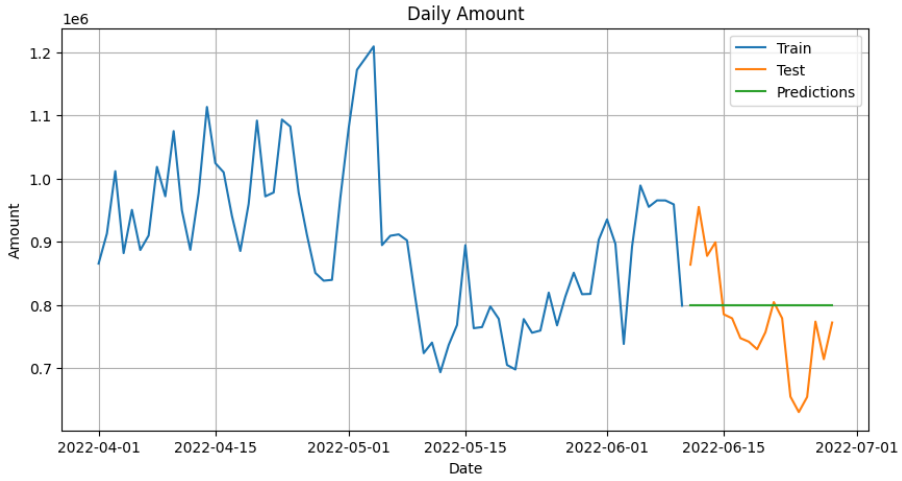

Ilustración 12: Train, Test y Predicciones Time Series Amazon India.

Los valores de RMSE y R^2 son francamente malos. Haciendo énfasis en el valor de R^2, y como se podrá ver en varios modelos más adelante, es posible que éste sea un valor negativo, indicando que el modelo es peor que hacer la media de los datos de test e incluir una variable constante, lo cual daría R^2 = 0.

Tabla 3: Resultados Time Series Amazon India.

	RMSE	R^2
Amazon India	156979.85	-1.28

4.3.1.2 Clientes

Para el segundo dataset, la función auto ARIMA sí que es capaz de interpretar una relación entre los datos. El modelo es uno relativamente complejo, al resultar con un SARIMAX(5, 1, 0)x(2, 0, [1], 7), radicalmente distinto que el anterior Código 1 que hace un ARIMA(0,1,0).

```
                                SARIMAX Results
==========================================================================================
Dep. Variable:                                   y   No. Observations:                  560
Model:             SARIMAX(5, 1, 0)x(2, 0, [1], 7)   Log Likelihood               -2793.723
Date:                             Thu, 15 May 2025   AIC                           5605.445
Time:                                     15:08:18   BIC                           5644.381
Sample:                                 07-30-2012   HQIC                          5620.650
                                      - 02-09-2014
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.8710      0.040    -21.596      0.000      -0.950      -0.792
```

```
ar.L2          -0.7185      0.054    -13.383       0.000      -0.824      -0.613
ar.L3          -0.5372      0.058     -9.245       0.000      -0.651      -0.423
ar.L4          -0.3110      0.053     -5.825       0.000      -0.416      -0.206
ar.L5          -0.1683      0.046     -3.627       0.000      -0.259      -0.077
ar.S.L7         0.8210      0.078     10.581       0.000       0.669       0.973
ar.S.L14        0.1208      0.041      2.922       0.003       0.040       0.202
ma.S.L7        -0.9077      0.069    -13.166       0.000      -1.043      -0.773
sigma2       1277.9414     69.559     18.372       0.000    1141.608    1414.275

Ljung-Box (L1) (Q):                   0.30   Jarque-Bera (JB):                38.46
Prob(Q):                              0.58   Prob(JB):                         0.00
Heteroskedasticity (H):               0.60   Skew:                             0.56
...
```

Código 2: SARIMAX Clientes.

Cuando se hace un gráfico para comparar los resultados del test versus los valores actuales, presente en la Ilustración 13, se puede observar la mayor complejidad de las predicciones, aunque al ser los valores de los coeficientes tan bajos, le resulta imposible capturar de una manera significativa cualquier pico.

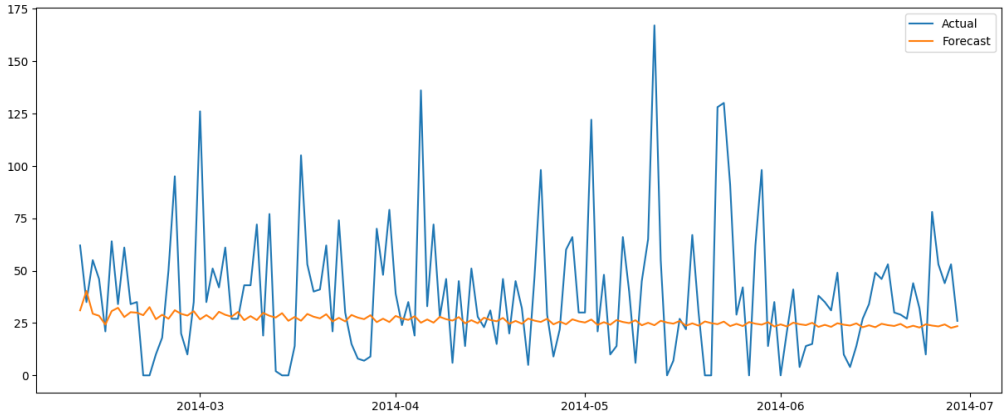

Ilustración 13: Test vs Predicted Time Series Clientes.

Cuando se sacan los resultados, se puede observar también como el R^2 es un valor negativo, indicando lo mismo que en el apartado anterior, el modelo es peor que hacer la media de los valores del test y utilizarlo como intercepto.

Tabla 4: Resultados Time Series Clientes.

	RMSE	R^2
Clientes	33.767	-0.195

4.3.1.3 Anuncios

Los siguientes códigos fueron los resultados de la aplicación de auto ARIMA para la predicción de las 4 variables objetivo de este dataset. Aquí se puede ver, como en códigos anteriores, como los modelos finales del SARIMAX Budget Anuncios. y SARIMAX Impressions Anuncios. son un simple ARIMA(0,1,0), mientas que SARIMAX Clicks Anuncios. y SARIMAX Media Cost USD Anuncios. sí que son modelos más complejos, indicando que en los últimos sí que ha sido capaz de interpretar alguna relación entre las variables, aunque después se comprobará cómo los resultados de estos modelos son los peores a nivel interpretativo de todo el trabajo.

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                          y   No. Observations:                  179
Model:                 SARIMAX(0, 1, 0)   Log Likelihood               -2056.691
Date:                  Thu, 15 May 2025   AIC                           4115.382
Time:                          14:37:32   BIC                           4118.563
Sample:                      05-01-2022   HQIC                          4116.672
                           - 10-26-2022
Covariance Type:                    opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
sigma2      6.326e+08   2.45e+07     25.772      0.000    5.85e+08    6.81e+08
===================================================================================
Ljung-Box (L1) (Q):                   2.03   Jarque-Bera (JB):              1209.38
Prob(Q):                              0.15   Prob(JB):                         0.00
Heteroskedasticity (H):               7.44   Skew:                             1.10
Prob(H) (two-sided):                  0.00   Kurtosis:                        15.58
===================================================================================
```

Código 3: SARIMAX Budget Anuncios.

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                          y   No. Observations:                  179
Model:                 SARIMAX(0, 1, 0)   Log Likelihood               -2273.841
Date:                  Thu, 15 May 2025   AIC                           4549.682
Time:                          14:37:35   BIC                           4552.863
Sample:                      05-01-2022   HQIC                          4550.972
                           - 10-26-2022
Covariance Type:                    opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
sigma2      7.257e+09    3.8e+08     19.107      0.000    6.51e+09       8e+09
===================================================================================
Ljung-Box (L1) (Q):                   2.61   Jarque-Bera (JB):               298.54
Prob(Q):                              0.11   Prob(JB):                         0.00
```

Heteroskedasticity (H):	7.31	Skew:	-0.64
Prob(H) (two-sided):	0.00	Kurtosis:	9.22

Código 4: SARIMAX Impressions Anuncios.

SARIMAX Results

Dep. Variable:	y	No. Observations:	179
Model:	SARIMAX(1, 1, 5)	Log Likelihood	-1895.446
Date:	Thu, 15 May 2025	AIC	3804.892
Time:	14:37:47	BIC	3827.164
Sample:	05-01-2022	HQIC	3813.924
	- 10-26-2022		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.8002	0.170	4.695	0.000	0.466	1.134
ma.L1	-0.6292	0.191	-3.290	0.001	-1.004	-0.254
ma.L2	-0.3725	0.146	-2.556	0.011	-0.658	-0.087
ma.L3	0.3214	0.144	2.238	0.025	0.040	0.603
ma.L4	-0.1140	0.207	-0.550	0.582	-0.520	0.292
ma.L5	-0.1608	0.113	-1.422	0.155	-0.382	0.061
sigma2	1.19e+08	1.05e-08	1.13e+16	0.000	1.19e+08	1.19e+08

Ljung-Box (L1) (Q):	0.00	Jarque-Bera (JB):	33895.56
Prob(Q):	0.96	Prob(JB):	0.00
Heteroskedasticity (H):	97.21	Skew:	6.17
Prob(H) (two-sided):	0.00	Kurtosis:	69.47

Código 5: SARIMAX Clicks Anuncios.

SARIMAX Results

Dep. Variable:	y	No. Observations:	179
Model:	SARIMAX(0, 1, 0)x(2, 0, 0, 7)	Log Likelihood	-1467.165
Date:	Thu, 15 May 2025	AIC	2940.329
Time:	14:37:51	BIC	2949.874
Sample:	05-01-2022	HQIC	2944.200
	- 10-26-2022		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.S.L7	-0.0829	0.095	-0.872	0.383	-0.269	0.103
ar.S.L14	-0.2367	0.062	-3.800	0.000	-0.359	-0.115
sigma2	8.652e+05	5.14e+04	16.819	0.000	7.64e+05	9.66e+05

Ljung-Box (L1) (Q):	0.30	Jarque-Bera (JB):	298.85
Prob(Q):	0.58	Prob(JB):	0.00
Heteroskedasticity (H):	10.94	Skew:	-0.84
Prob(H) (two-sided):	0.00	Kurtosis:	9.12

Código 6: SARIMAX Media Cost USD Anuncios.

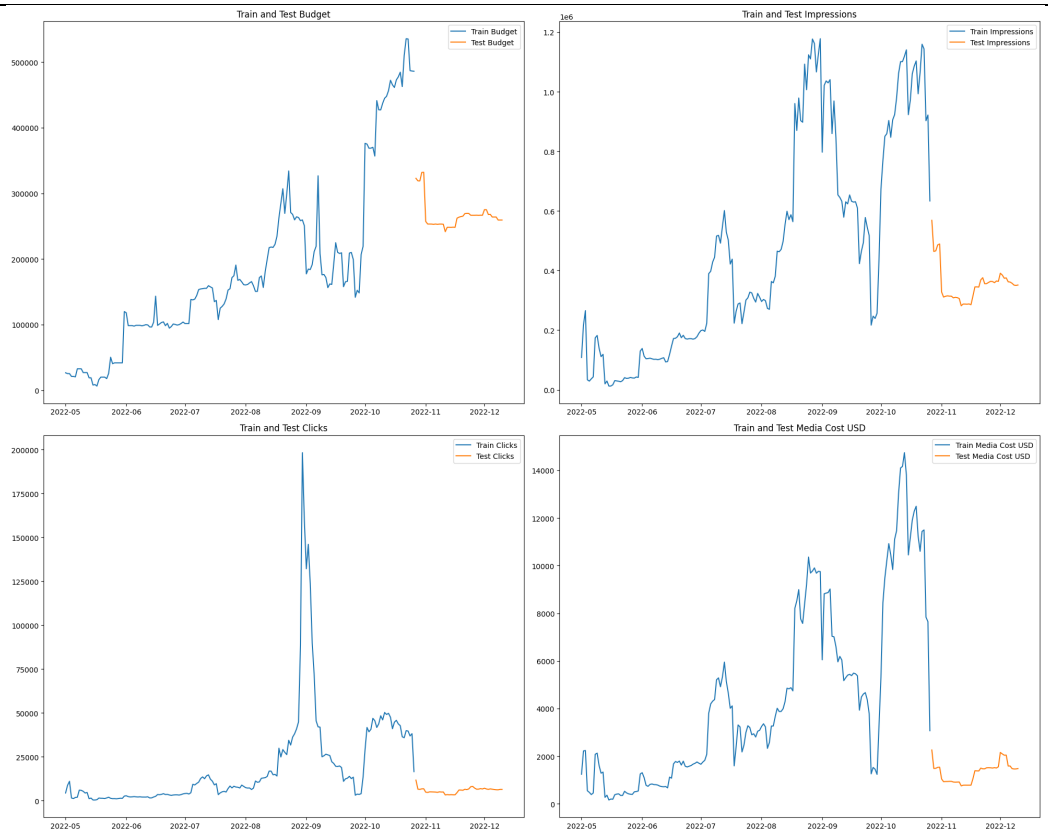

Ilustración 14: Train, Test de todos los parámetros, Time Series Anuncios.

Para establecer un contexto importante a la hora de interpretar visualmente los resultados de la Ilustración 15, se hace una gráfica con las particiones de entrenamiento y test de los 4 modelos. Es útil acordarse de la forma que tienen las gráficas de test, delineadas en naranja en la Ilustración 14, porque también están presentes en la Ilustración 15, en azul esta vez.

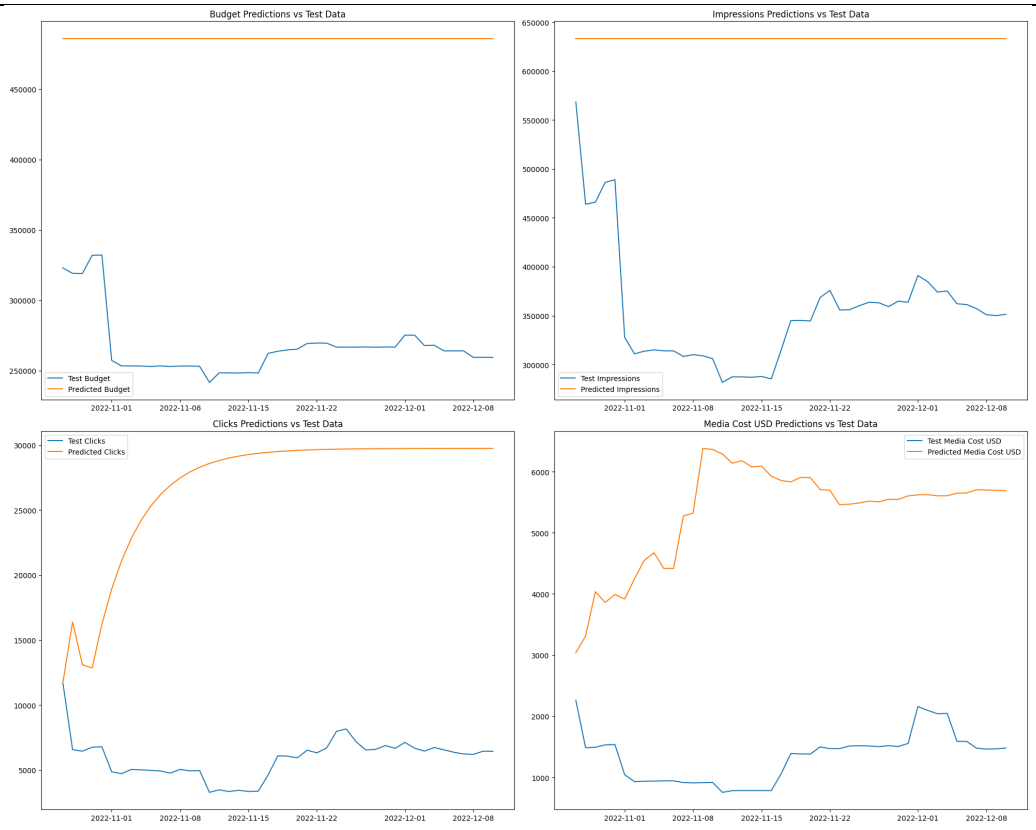

Ilustración 15: Test vs Predecidos Time Series en Anuncios.

A simple vista se puede comprobar cómo hay un desajuste considerable entre las predicciones y la realidad. Estos resultados presentes numéricamente en la

Anuncios	RMSE	R^2
Budget	219839.92	-99.67
Impressions	284999.48	-22.21
Clicks	21681.01	-199.90
Media Cost USD	4117.63	-104.00

Tabla 5 son los peores de todo el proyecto, indicando que el modelo no es capaz de aprender ni la forma ni la variabilidad de las series temporales, tanto la forma como la escala presentan un desajuste muy grande.

Anuncios	RMSE	R^2
Budget	219839.92	-99.67
Impressions	284999.48	-22.21

Clicks	21681.01	-199.90
Media Cost USD	4117.63	-104.00

Tabla 5: Resultados Time Series Anuncios.

Ninguna de las series temporales, para todos los datasets que se han probado, se han ajustado bien. La consistencia en resultados pobres obtenida indica inequívocamente que las series temporales ARIMA no son buenos regresores para cualquier dato de marketing, y no deberían considerarse para una aplicación en el mundo real.

4.3.2 MACHINE LEARNING

Para garantizar que cada modelo operase en condiciones óptimas, se aplicó una búsqueda exhaustiva mediante Grid Search sobre los principales hiperparámetros de cada uno. En el caso de Random Forest, se seleccionó un número de árboles (*n_estimators*) de 100, una profundidad máxima (*max_depth*) de 10 y se utilizó el criterio de división mse, junto con *max_features='sqrt'*, lo que favoreció un equilibrio entre precisión y velocidad.

Para el modelo Support Vector Regression (SVR), los mejores resultados se obtuvieron con un *kernel radial (rbf),* un parámetro de regularización $C = 100$, un $\varepsilon = 0.1$ como margen de tolerancia y un $\gamma = 0.01$ para el control de la curvatura del kernel.

En el caso de LightGBM, la combinación más efectiva fue un *learning_rate* $= 0.05$, *num_leaves* $= 31$, *max_depth* $= 7$, y regularización $\lambda_{L_1} = 0.1$ y $\lambda_{L_2} = 0.2$, lo que permitió reducir el riesgo de sobreajuste sin comprometer el rendimiento.

Finalmente, para la RNN, se probaron configuraciones de tipo *SimpleRNN* con una sola capa oculta de *100 unidades*, función de activación *tanh*, un *batch_size* de 50, *dropout* del 20%, y el *optimizador adam con learning_rate* $= 0.001$. Estos valores fueron seleccionados tras evaluar el rendimiento con validación cruzada, utilizando como métricas principales el RMSE y el R^2.

Las particiones de train y test se realizaron aleatorizando los pertenecientes a cada uno mediante el uso del *random_state=42*, algo que no se podía hacer para las series temporales, debido a que en ellas se debe mantener una integridad temporal, forzando a la primera parte a ser el entrenamiento y la última a ser el test.

4.3.2.1 Amazon India

se

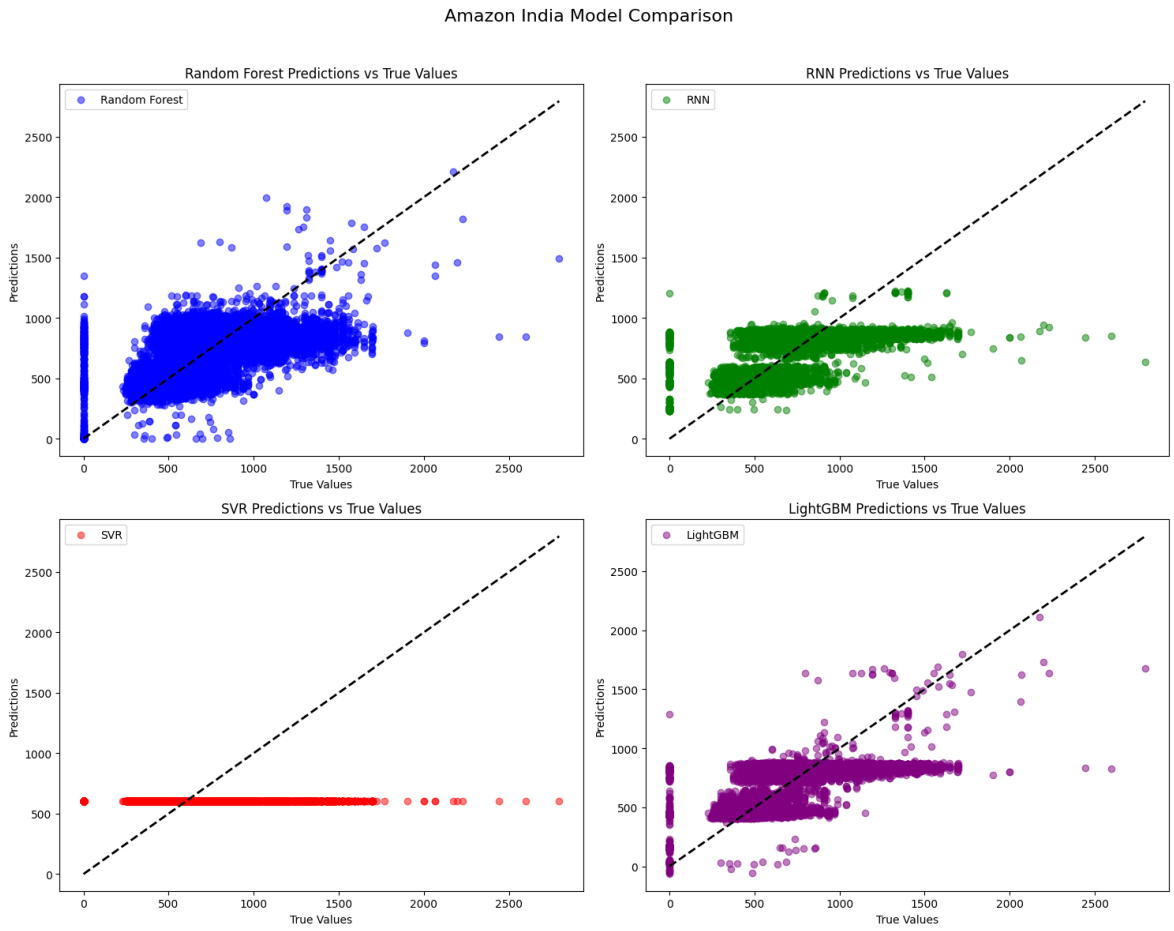

Ilustración 16: Comparación de modelos de Machine Learning Amazon India.

Amazon India	Random Forest	RNN	SVR	LightGBM
RMSE	207.06	208.91	283.21	200.37
R^2	0.45	0.44	-0.02	0.47

Tabla 6: Resultados de Machine Learning Amazon India.

LightGBM es el modelo con mejor rendimiento general: consigue el RMSE más bajo (200.37) y el mayor R^2 (0.47), lo que indica que no solo predice con mayor precisión, sino que además explica mejor la variabilidad de los datos. En el gráfico correspondiente se aprecia una buena densidad de puntos en torno a la diagonal (línea de predicción perfecta), aunque con cierta dispersión en valores altos.

Random Forest y RNN ofrecen rendimientos muy similares, con RMSEs casi idénticos (207.06 y 208.91 respectivamente) y valores de R^2 también parecidos (0.45 y 0.44). Aunque ambos modelos predicen de forma razonable, muestran más dispersión respecto a la diagonal, especialmente la RNN, donde se aprecia una mayor concentración de puntos por debajo de la línea, lo que sugiere cierta subestimación sistemática.

SVR, por el contrario, tiene un desempeño muy pobre: su RMSE es el más alto (283.21) y su R^2 es negativo (-0.02), lo que significa que el modelo predice incluso peor que simplemente usar la media como estimación constante, también es el que más tiempo tarda en crearse, demostrando que no es el modelo a seguir. Gráficamente, esto se confirma con una nube de puntos claramente alejada de la diagonal, con predicciones agrupadas artificialmente en un mismo valor o con una falta clara de ajuste.

A pesar de que en el caso de este dataset Random Forest no obtiene los mejores resultados, éste tiene un desempeño muy parecido a LightGBM. Más tarde se verá como Random Forest sí que es el modelo más apropiado para todos los demás datasets, y es por esto que se estudiará en mayor detalle más adelante.

4.3.2.2 Clientes

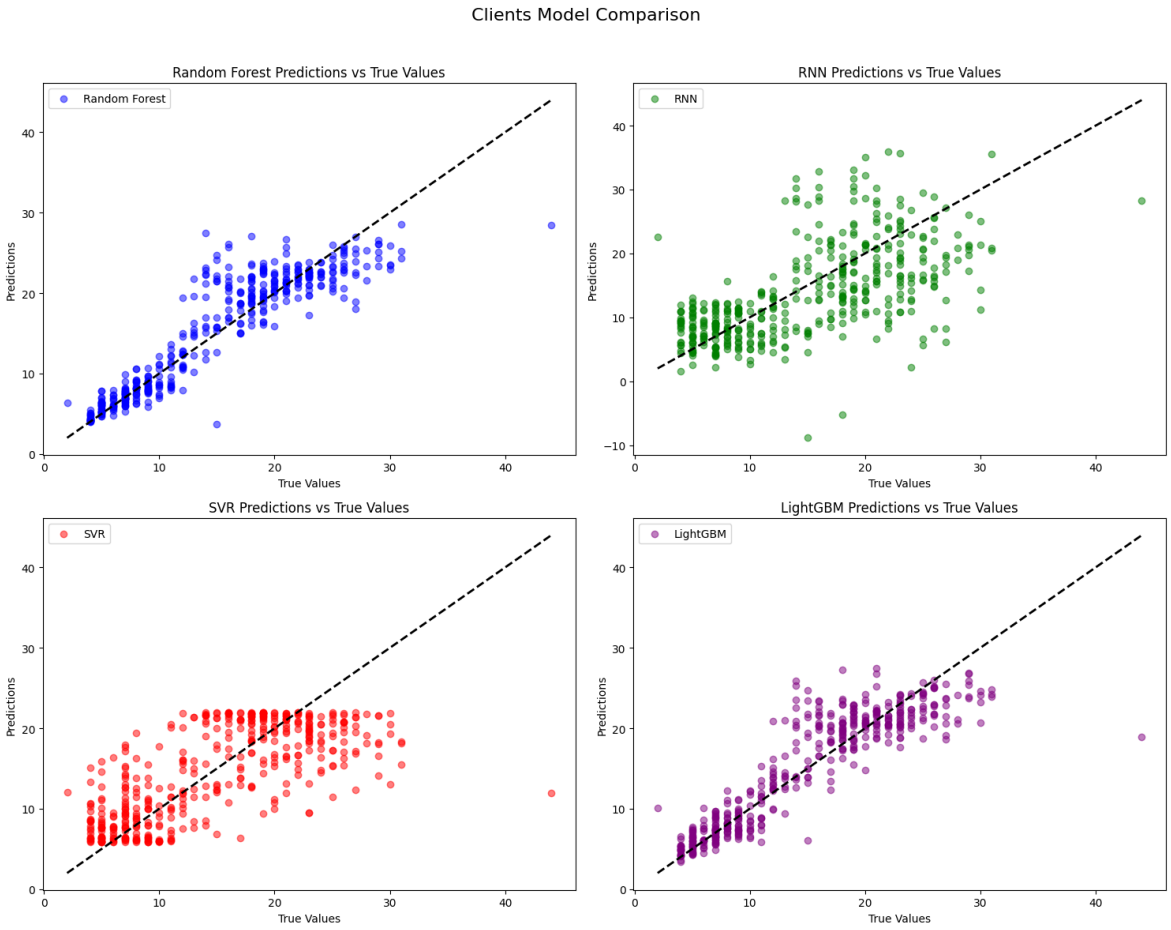

Ilustración 17: Comparativa de modelos de Machine Learning Clientes.

Clientes	Random Forest	RNN	SVR	LightGBM
RMSE	2.97	6.61	5.27	3.23
R^2	0.84	0.23	0.51	0.81

Tabla 7: Resultados de Machine Learning Clientes.

Random Forest destaca como el modelo más efectivo, con el mejor R^2 (0.84) y el menor RMSE (2.97). Su gráfico muestra una alta alineación de puntos en torno a la diagonal, indicando que el modelo consigue una gran precisión a lo largo de todo el rango de valores. La nube de puntos es compacta, sin grandes desviaciones sistemáticas.

El modelo LightGBM también obtiene resultados excelentes, con un R^2 de 0.81 y un RMSE de 3.23, valores muy cercanos a los de Random Forest. Visualmente, sus predicciones siguen bien la diagonal, aunque con algo más de dispersión. Aun así, se confirma como una alternativa muy sólida.

Por otro lado, el modelo SVR, aunque con una mejora notable respecto al apartado anterior, muestra un rendimiento más moderado: su R^2 de 0.51 y RMSE de 5.27 indican que capta parte de la estructura de los datos, pero con errores más amplios. En su gráfico, los puntos tienden a agruparse por debajo de la diagonal, sugiriendo cierta subestimación sistemática.

Por el contrario, el modelo neuronal recurrente, RNN, obtiene el rendimiento más bajo: RMSE de 6.61 y un pobre R^2 de 0.23. Aunque visualmente se aprecia cierta tendencia positiva, el patrón de dispersión es mucho más amplio, con predicciones menos ajustadas y más ruido. Esto puede deberse a que la variable objetivo no presenta suficiente secuencia temporal para que la RNN sea efectiva, o a que el modelo no ha sido suficientemente afinado para este caso.

Para este dataset, tanto Random Forest (en el que se explorarán sus variables más tarde) como LightGBM ofrecen una capacidad predictiva excelente, destacando por su precisión y estabilidad. Son modelos recomendables cuando se busca interpretar el comportamiento de los clientes a partir de múltiples variables. SVR puede considerarse aceptable, aunque menos preciso, y RNN no resulta eficaz en este contexto concreto.

4.3.2.3 Anuncios

Al haber 4 variables objetivo para este dataset, se van a proceder a comparar cada uno en su apartado.

4.3.2.3.1 Budget

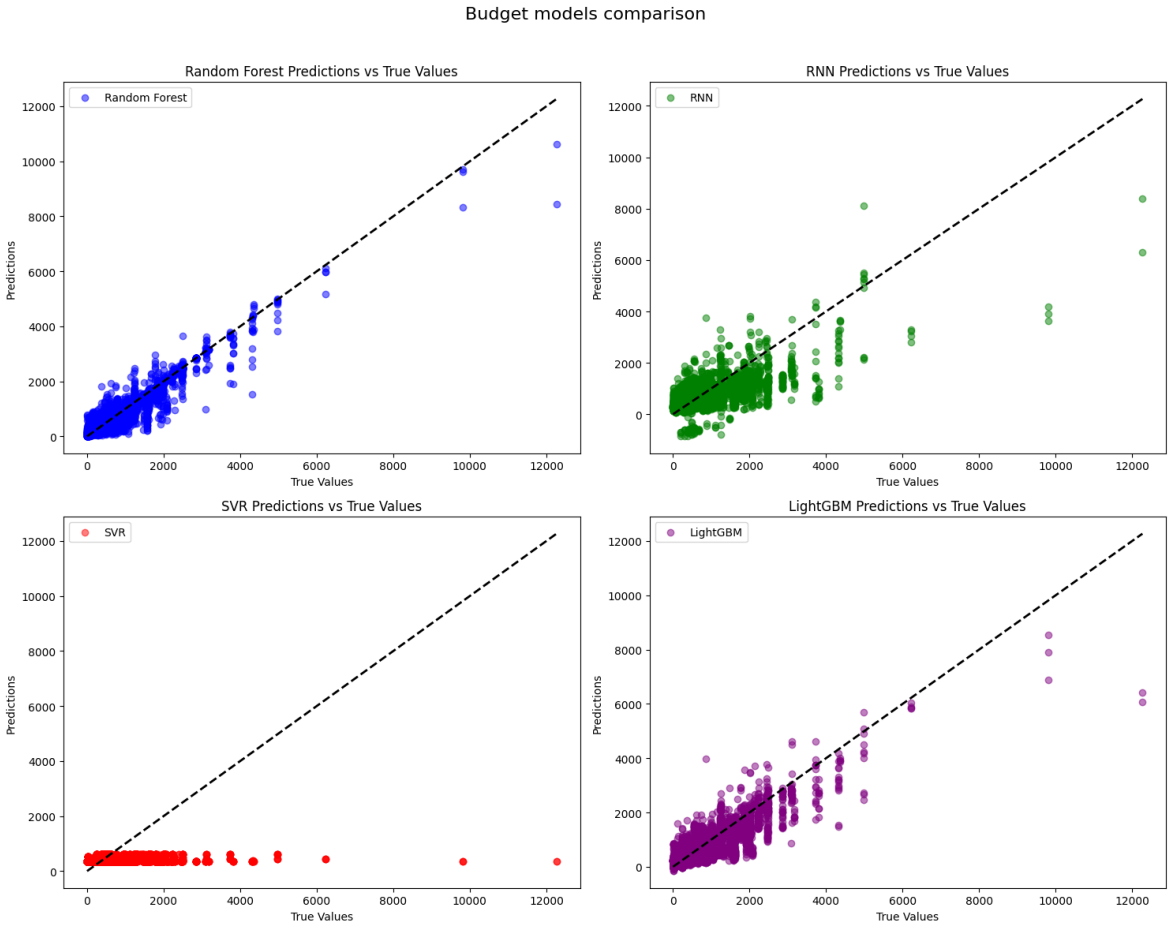

Ilustración 18: Comparativa de modelos Budget Anuncios.

Budget	Random Forest	RNN	SVR	LightGBM
RMSE	185.95	742.42	605.81	311.26
R^2	0.90	-0.58	-0.05	0.72

Tabla 8: Resultados de Machine Learning Budget Anuncios.

Random Forest es claramente el modelo con mejor rendimiento en esta tarea: alcanza un RMSE muy bajo (185.95) y un excelente $R^2 = 0.90$, lo que indica que explica el 90 % de la

variabilidad del presupuesto. Su gráfico muestra una alineación precisa a lo largo de la diagonal, lo que refleja predicciones cercanas a los valores reales en casi todo el rango.

LightGBM también se comporta de forma notable, con un RMSE de 311.26 y un R^2 de 0.72, lo que lo convierte en la segunda mejor opción. Aunque presenta algo más de dispersión y errores en valores extremos, sigue ofreciendo un ajuste sólido y generalizable.

SVR, en cambio, tiene un comportamiento pobre, con un RMSE de 605.81 y un R^2 negativo (-0.05). Esto significa que el modelo predice peor que simplemente usar la media de los datos, lo cual queda reflejado gráficamente en la nube de puntos agrupados en una franja horizontal, muy alejada de la diagonal.

RNN es el modelo con peor rendimiento, con un RMSE alto (742.42) y un R^2 negativo significativo (-0.58). Su gráfico muestra un patrón muy disperso, con predicciones sistemáticamente sesgadas hacia abajo o estancadas, lo que sugiere que la red no ha logrado aprender una relación útil a partir de los datos temporales disponibles.

El modelo Random Forest demuestra ser el más eficaz para predecir presupuestos publicitarios, seguido de LightGBM, que también ofrece resultados sólidos. En cambio, tanto SVR como RNN fallan en capturar la estructura de los datos, con errores elevados y ajustes deficientes.

4.3.2.3.2 Impressions

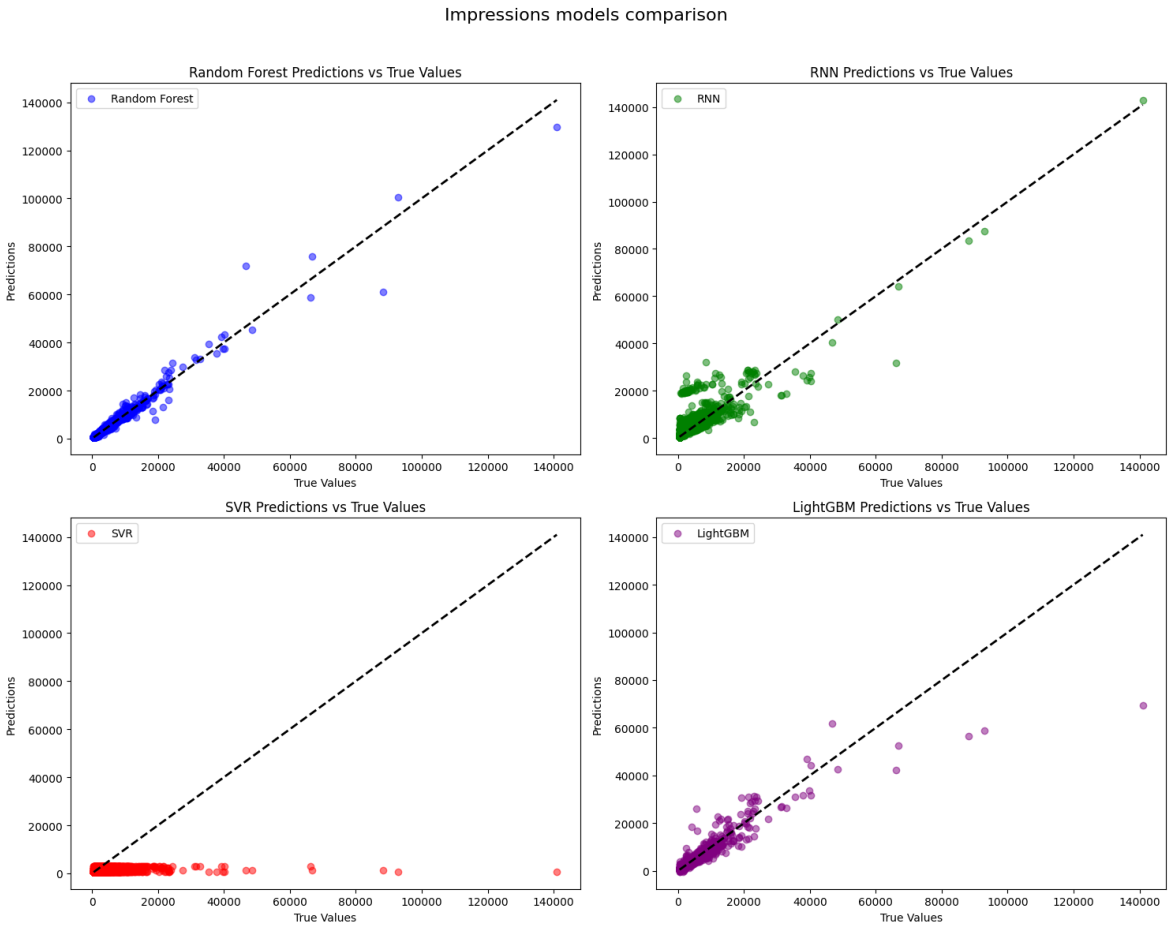

Ilustración 19: Comparativa de modelos de Maachine Learning Impressions Anuncios.

Impressions	Random Forest	RNN	SVR	LightGBM
RMSE	451.52	884.81	2621.26	953.03
R^2	0.97	0.89	0.08	0.88

Tabla 9: Resultados de Machine Learning Impressions Anuncios.

De nuevo, el Random Forest destaca como el mejor modelo, logrando un RMSE de 451.52 y un altísimo $R^2 = 0.97$. Su gráfico muestra una gran concentración de puntos sobre la diagonal, lo que indica una capacidad sobresaliente para predecir los valores reales en prácticamente todo el rango. Es, sin duda, el modelo más fiable para esta métrica.

Las redes recurrentes RNN también ofrecen un rendimiento notable, con un RMSE de 884.81 y un excelente $R^2 = 0.89$. Aunque es algo menos preciso que Random Forest, sigue siendo muy competente. En el gráfico se observan algunas predicciones dispersas en valores altos, pero la tendencia general sigue la diagonal de forma consistente.

El modelo LightGBM mantiene un buen nivel de rendimiento con un RMSE de 953.03 y un $R^2 = 0.88$. Aunque se sitúa ligeramente por debajo de los dos anteriores, sigue ofreciendo una predicción robusta, con puntos bastante alineados y pocos errores graves.

Por el contrario, la regresión basada en máquinas de vectores soporte o SVR, muestra un desempeño claramente deficiente: el RMSE es muy elevado (2621.26) y el R^2 apenas llega al 0.08, lo que indica que apenas explica la variabilidad de los datos. Visualmente, la mayoría de sus predicciones se agrupan en un rango estrecho muy por debajo de la diagonal, lo que sugiere una grave subestimación sistemática.

4.3.2.3.3 Clicks

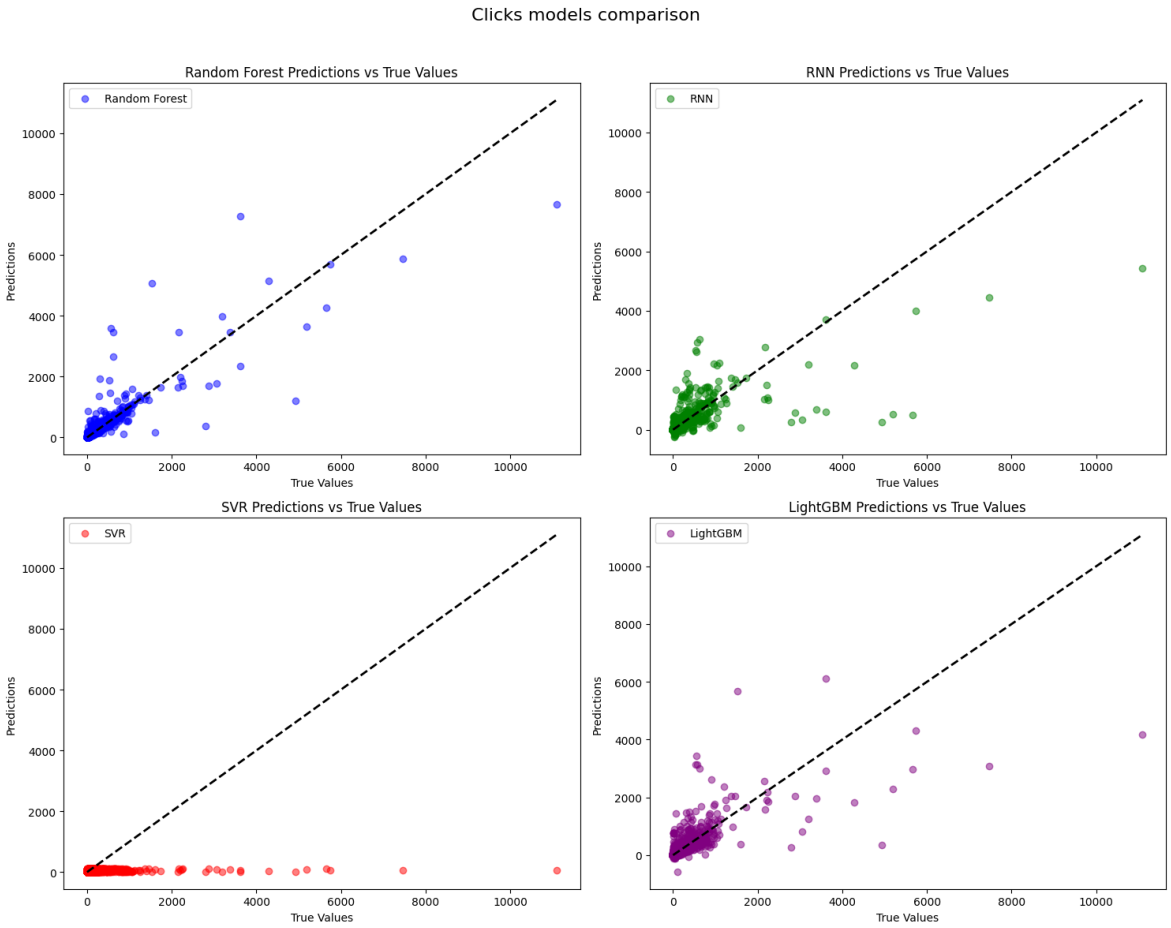

Ilustración 20: Comparativa de modelos de Machine Learning Clicks Anuncios.

Clicks	Random Forest	RNN	SVR	LightGBM
RMSE	90.06	119.80	187.93	128.73
R^2	0.78	0.61	0.04	0.55

Tabla 10: Resultados de Clicks Anuncios.

El modelo Random Forest vuelve a ser el modelo más preciso, con el RMSE más bajo (90.06) y un alto $R^2 = 0.78$. Su gráfico muestra una buena alineación de las predicciones con la diagonal, especialmente en el rango medio de valores. Aunque se observan ciertos errores para clics muy altos, el comportamiento general es robusto y fiable.

El modelo RNN ofrece un rendimiento correcto, con un RMSE de 119.80 y un $R^2 = 0.61$. El gráfico refleja cierta subestimación en valores elevados, pero el patrón general sigue la diagonal. Esto sugiere que, si bien la red neuronal capta una parte importante de la estructura de los datos, sufre cierta pérdida de precisión en extremos.

En este caso, LightGBM logra resultados intermedios entre RNN y Random Forest, con un RMSE de 128.73 y $R^2 = 0.55$. Aunque su capacidad predictiva no es la mejor en esta tarea concreta, mantiene un rendimiento aceptable. El gráfico muestra un comportamiento correcto pero algo más disperso, con errores más notorios en valores altos.

De nuevo, el modelo SVR es el menos eficaz, con un RMSE de 187.93 y un R^2 muy bajo (0.04). En la gráfica, la mayoría de las predicciones se agrupan artificialmente en torno a un mismo rango, sin adaptarse a la escala real de los datos. Esto refleja una clara limitación del modelo para manejar variabilidad o relaciones no lineales complejas.

4.3.2.3.4 Media Cost

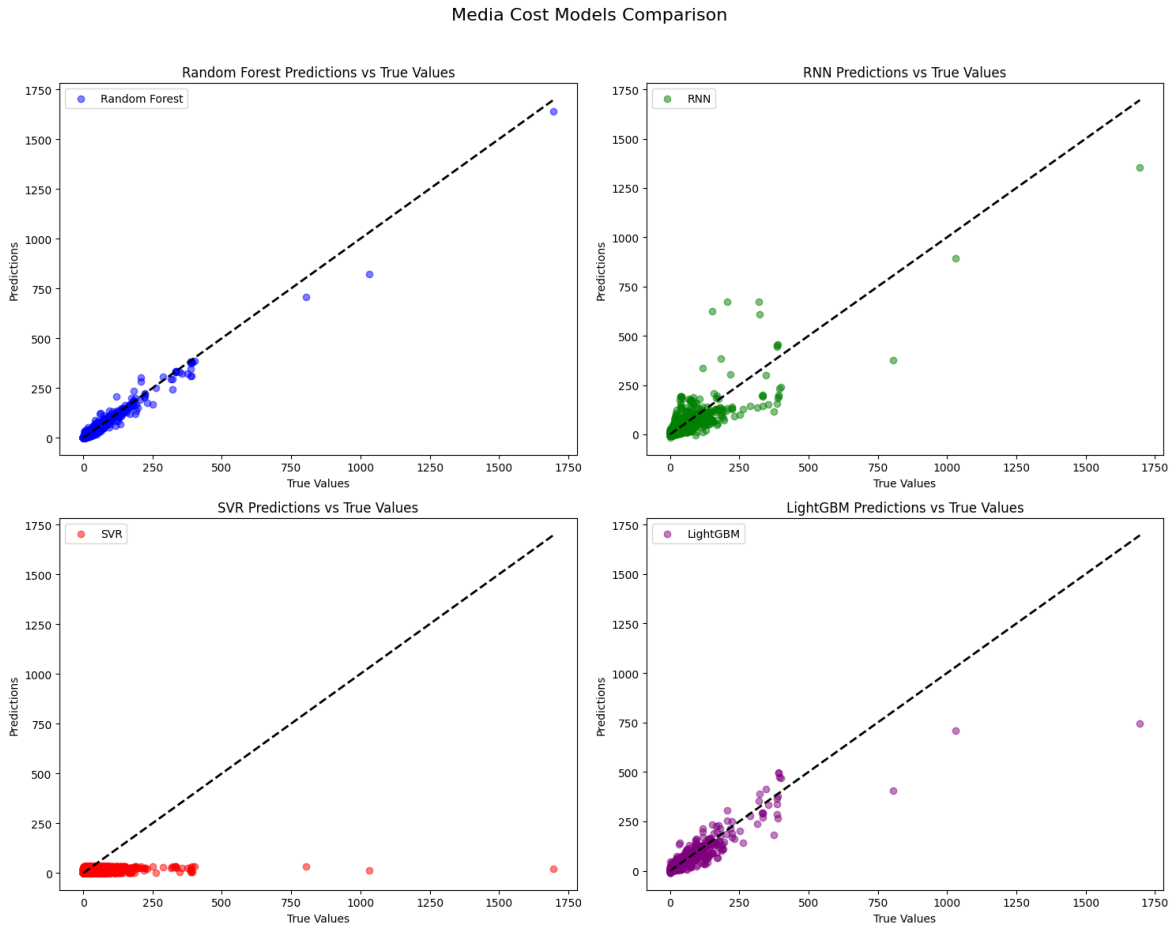

Ilustración 21: Comparativa de modelos de Media Cost Anuncios.

Media Cost	Random Forest	RNN	SVR	LightGBM
RMSE	4.19	14.85	27.75	11.74
R^2	0.98	0.74	0.10	0.84

Tabla 11: Resultados de Media Cost Anuncios.

Una vez más, el modelo basado en Random Forest más destacado. Con un RMSE mínimo de 4.19 y un R^2 casi perfecto de 0.98, ofrece una predicción extremadamente precisa del coste en medios. En su gráfico, las predicciones se alinean de forma clara con la diagonal, mostrando mínima dispersión incluso en valores altos.

El modelo LightGBM también muestra un rendimiento excelente, con un RMSE de 11.74 y un $R^2 = 0.84$. Aunque es ligeramente menos preciso que Random Forest, su comportamiento general es muy bueno, como se refleja en una nube de puntos bien alineada en torno a la diagonal.

También se obtienen resultados aceptables con RNN pero con menor precisión: RMSE de 14.85 y $R^2 = 0.74$. Aunque capta parte de la tendencia, en su gráfico se observan más errores en valores extremos, lo que indica una menor capacidad para generalizar correctamente el coste.

La regresión SVR vuelve a presentar un bajo rendimiento, con un RMSE elevado (27.75) y un $R^2 = 0.10$. El modelo subestima sistemáticamente los valores reales, y sus predicciones se agrupan en torno a valores muy bajos, lejos de la diagonal. Esto refleja una limitación estructural del modelo para esta tarea.

Como conclusión de esta sección, el modelo Random Forest se impone claramente como la opción más precisa y fiable, seguido por LightGBM, que también ofrece resultados sólidos. RNN se mantiene como alternativa intermedia, mientras que SVR queda descartado por su bajo poder explicativo y error elevado.

4.3.3 IMPORTANCIA DE VARIABLES

Como se ha podido comprobar, el mejor modelo en cada uno de los casos, salvo para que se probaron es el de Random Forest. Al ser este el caso, se profundizará en la importancia de las variables, una característica que se puede comprobar de este modelo a partir del atributo *feature_importances*. Mediante este análisis se intentará llegar a una conclusión que podría llegar un departamento de marketing de una empresa.

4.3.3.1 Amazon India

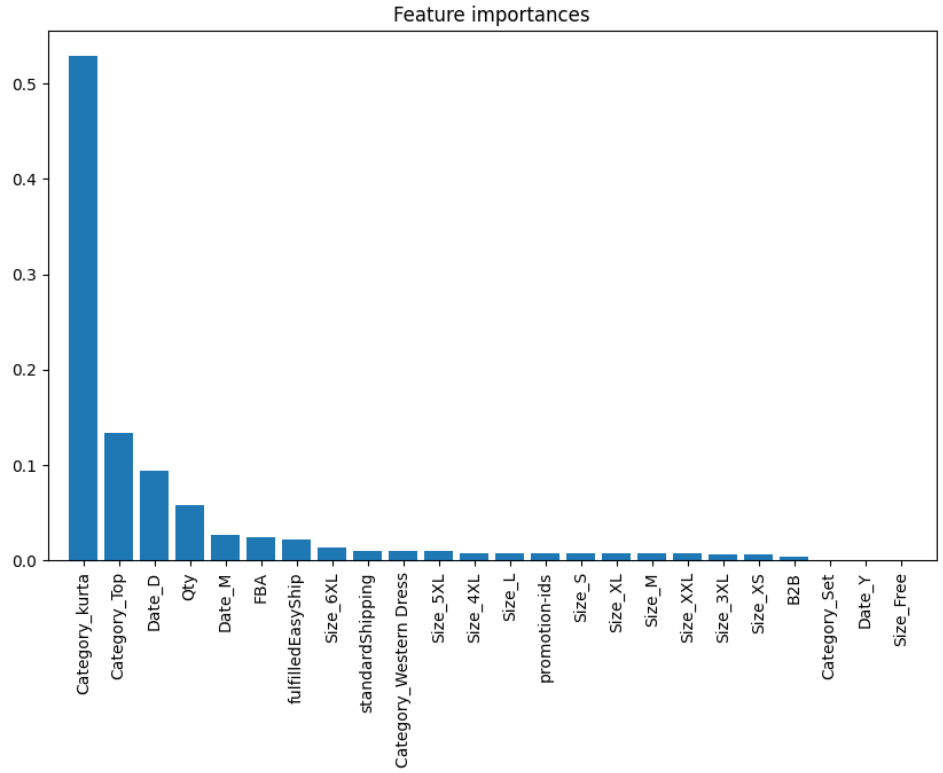

Ilustración 22: Importancia de variables de Amazon India.

En el primer dataset, donde tratamos de predecir la variable del precio de la venta, la característica más importante es la categoría del producto (kurta, top) seguido de variables relacionadas con fechas, día y mes en este caso, probablemente por estar relacionadas con la demanda y el tipo del cliente. Siguiendo en la lista, la cantidad y el tipo de fulfillment tienen cierto peso, aunque en menor medida. Las tallas no influyen de manera significativa, por lo que se puede entender que no influyen en el precio final de las prendas.

4.3.3.2 Clientes

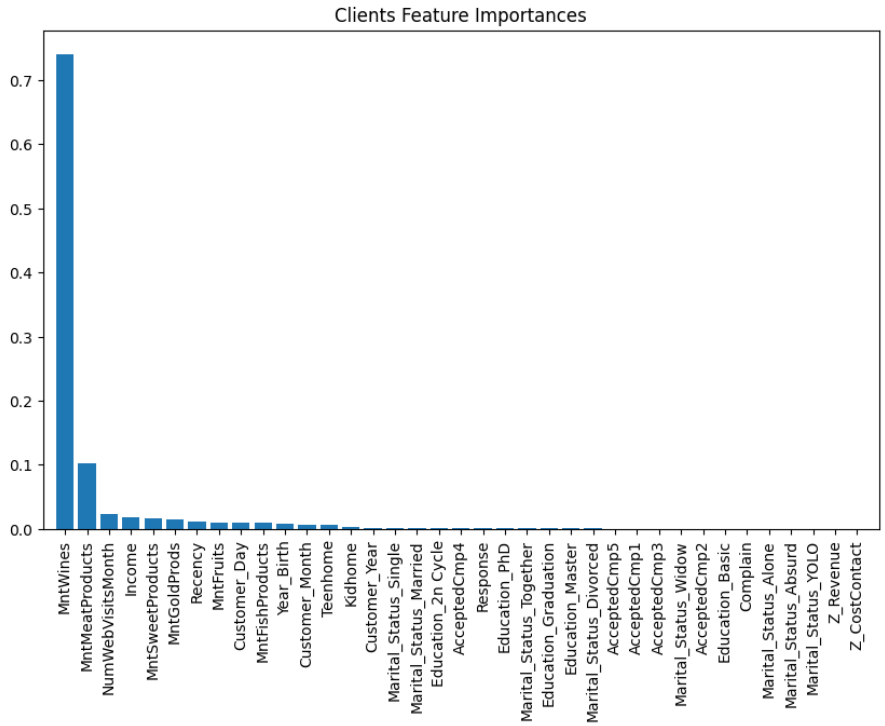

Ilustración 23: Importancia de variables de Clientes.

A la hora de querer predecir la cantidad de compras totales, las variables importantes en este dataset son la cantidad de vino que se compra, MntWines, con un peso superior, seguido a mucha distancia de MtnMeatProducts y NumWebVisitsMonth, lo que significa que el volumen de consumo y la actividad obline reciente son los mejores indicadores útiles del comportamiento del cliente. Otras variables como ingresos (Income), el número de promociones aceptadas (AcceptedCmpX), e indicadores de la estructura familiar o edad (Kidhome, Teenhome, Year_Birth) presentan importancias marginales. Esto indica que, en este caso, las características socio-demográficas del cliente aportan muy poca información al modelo en comparación con su historial de gasto.

Este patrón es especialmente útil para fines de segmentación y retargeting, ya que demuestra que los datos transaccionales (lo que el cliente hace) tienen mucho más valor predictivo que los datos declarativos (quién es el cliente).

4.3.3.3 Anuncios

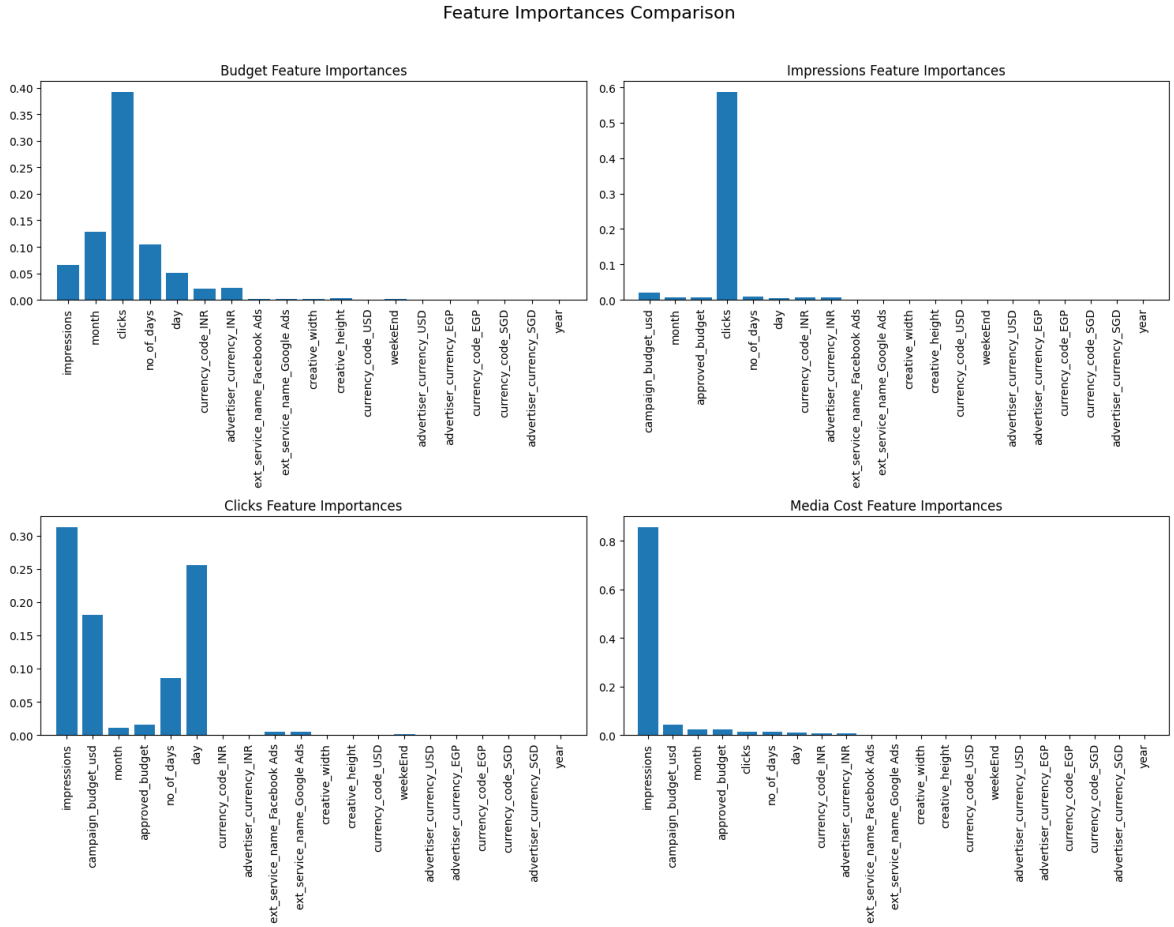

Ilustración 24: Importancia de variables de Clientes.

A través del análisis de la database de anuncios se puede observar que, aunque hay ciertos factores que se repiten como relevantes, la influencia de cada variable varía significativamente según la métrica objetivo.

- **Budget Feature Importances**

En este caso, la variable más relevante es clicks, seguida de month, no_of_days e impressions. Esto sugiere que el presupuesto asignado varía estacionalmente, y está altamente correlacionado tanto con el presupuesto previamente aprobado como con el rendimiento histórico de la campaña.

- **Impressions Feature Importances**

Aquí, la importancia se concentra de forma abrumadora en clicks, que supera con creces al resto. Es una señal clara de que la cantidad de clicks es el principal (y casi único) determinante del volumen de impresiones, lo cual tiene sentido al estar las dos variables altamente correladas. El resto de variables tienen un peso prácticamente nulo.

- **Clicks Feature Importances**

En la predicción de clics, el modelo otorga más equilibrio: impressions encabeza con fuerza la lista, seguida de campaign_budget_usd y day. Esto refleja la estructura típica de una campaña digital: cuantas más impresiones se obtienen, mayor es la probabilidad de generar clics. La variable no_of_days también aparece con un impacto moderado, lo que podría asociarse a ciclos de consumo o campañas estacionales.

- **Media Cost Feature Importances**

El coste en medios está casi exclusivamente explicado por impressions, lo que es coherente con modelos de compra basados en CPM (coste por mil impresiones). Esto confirma que el coste no depende directamente de aspectos como el tiempo, la moneda o el anunciante, sino del volumen de visibilidad adquirida. El resto de variables aportan información insignificante.

Capítulo 5. CONCLUSIONES Y DISCUSIÓN

A lo largo de este trabajo se han explorado distintos enfoques para la predicción y análisis de variables clave en el contexto del marketing digital y el comportamiento de clientes, utilizando tres conjuntos de datos representativos. Tras aplicar técnicas de preprocesamiento, transformación, selección de variables y modelado, se ha podido contrastar el rendimiento de modelos basados en series temporales y algoritmos de Machine Learning clásicos y avanzados. Los resultados han sido concluyentes: Random Forest se ha consolidado como el modelo más eficaz, superando al resto tanto en precisión como en capacidad de generalización, con valores de R^2 consistentemente altos y errores reducidos en prácticamente todos los escenarios.

Paralelamente a la comparación de modelos, también se ha buscado un análisis de las variables más significativas, ya que Random Forest permite esto y resultó ser el ganador. Resulta algo evidente que a la hora de predecir la cantidad que se va a ingresar en una tienda, la categoría de producto sería la variable más importante, frente a otras características como el tipo de envío. El hecho de que la importancia sea mayor que 0 indica algo de relación, y ahí entra el análisis a ser realizado por las personas correspondientes. En otros datasets utilizados, para predecir las compras totales el mejor indicador resulta ser las cantidades del producto primario que vende la tienda, estando esta por encima de compras adyacentes que pueda realizar el cliente. Por otro lado, cuando se intentan predecir métricas, los buenos predictores suelen ser las que están correlacionadas unas con otras, al seguir formas similares, lo cual puede ser útil si se tienen claras algunas métricas previamente para realizar predicciones.

En contraste, los modelos de series temporales han ofrecido un rendimiento claramente insuficiente, con predicciones alejadas de los valores reales y métricas negativas en muchos casos, lo que evidencia su limitada utilidad en los conjuntos de datos utilizados, posiblemente por la falta de estacionalidad fuerte o por la escasa capacidad de capturar

dinámicas complejas en ventanas de tiempo cortas. Modelos como SVR o RNN, aunque en ciertos contextos ofrecieron resultados aceptables, tampoco lograron superar a los modelos basados en árboles, que demostraron ser más estables, interpretables y versátiles ante variables heterogéneas. Hay que identificar que el comportamiento de los compradores es algo que tiene un elevado grado de complicación, con gente que dedica su vida a esta tarea, e incluso en conversaciones con ellos me han asegurado que nunca es una ciencia cierta. La alta dificultad puede explicar cómo, los modelos más complejos que son capaces de establecer relaciones, tienen mejores resultados. LightGBM, pero especialmente Random Forest, obtienen sus resultados frutos del *bagging*, un proceso incluso más complicado que los otros modelos, y es por ello que son capaces de entender mejor lo que sucede.

En definitiva, este trabajo no solo ha permitido evaluar qué modelos funcionan mejor en escenarios reales de marketing, sino también destacar la importancia de una buena selección de variables y una metodología rigurosa. De cara a aplicaciones futuras en entornos empresariales, Random Forest emerge como una herramienta potente, fiable y adaptable a diferentes tipos de datos y objetivos analíticos, lo que la convierte en una elección estratégica para tareas predictivas en entornos donde la precisión es clave.

A lo largo de todo este estudio se ha comprobado que la capacidad de generalización, como ya ocurre en otras áreas, es muy complicada. No existe un modelo que sea perfecto, aunque claramente hay algunos mejores que otros, y siempre hay que tener en cuenta la complejidad del mismo, ya que se busca que su aplicación sea correcta para cualquiera que sea el conjunto de datos. Igualmente, aunque los expertos en mercados pueden tener cierta intuición sobre la relevancia de ciertas variables a la hora de predecir la evolución de ventas, cantidades o métricas de campañas, estos análisis han revelado que existen otras variables que deberían ser vigiladas de cara a una mejor planificación de las campañas, además de la cuantificación numérica de su importancia, favoreciendo la identificación de métricas clave o periodos óptimos para el desarrollo de la actividad comercial o campaña.

Capítulo 6. TRABAJOS FUTUROS

A partir de los resultados obtenidos, surgen diversas líneas de mejora y exploración para trabajos futuros. En primer lugar, una de las principales áreas de interés sería profundizar en el ajuste de hiperparámetros del modelo Random Forest, ya que, aunque se ha aplicado Grid Search, existe margen para una optimización más fina mediante técnicas como la búsqueda bayesiana o validación cruzada más específica por segmentos. Mejorar este ajuste permitiría no solo afinar la precisión, sino también reducir tiempos de procesamiento y mejorar la estabilidad del modelo.

Otro aspecto clave es la calidad de los datos. Muchos de los errores de predicción observados podrían estar relacionados con variables incompletas, ruido o una falta de granularidad en las series temporales. Trabajar con datasets más limpios, ricos y representativos del comportamiento real del consumidor permitiría entrenar modelos más robustos y con mayor capacidad de generalización. Esto abre otro experimento que sería ideal para comprobar este análisis frente a un nivel base, que sería la comparación de dos campañas reales, una con las conclusiones que se puedan sacar de este análisis y otro sin ello. Esta prueba ciega podría determinar de una manera empírica los resultados, y ver si merece la pena o se tiene que cambiar ciertos parámetros para que se ajuste mejor al funcionamiento real.

Además, en el ámbito del análisis temporal, sería interesante explorar alternativas más modernas y adaptadas como Amazon Prophet, que incorpora de forma nativa estacionalidad, festividades y cambios de tendencia. Este tipo de modelos podrían resultar más adecuados que los ARIMA clásicos para ciertos tipos de series reales del mundo del marketing digital, donde los patrones de comportamiento suelen estar influenciados por eventos externos y ciclos de campaña, desde luego algo a tener en cuenta a la hora de determinar trabajos futuros.

Capítulo 7. BIBLIOGRAFÍA

[1] H. K. W. &. H. I. Lee, «Developing a business performance evaluation system: An analytic hierarchical model.,» *The Engineering Economist,* pp. 343-357, 1995.

[2] R. Webber, «The evolution of direct, data and digital marketing.,» 2013. [En línea]. Available: https://link.springer.com/article/10.1057/dddmp.2013.20.

[3] C. A. O. J. H. K. O. I. &. R. P. A. Tavera Romero, «Business intelligence: business evolution after industry 4.0.,» *Sustainability,* pp. 13-18, 2021.

[4] A. Z. H. K. A. &. T. A. M. Bouguettaya, «Machine Learning and Deep Learning as New Tools for Business Analytics. In Advances in business information systems and analytics.,» de *2023 Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE)*, Las Vegas, NV, USA, 2023.

[5] M. B. A. S. A. H. A. N. A. K. J. M. J. J. &. A. A. S. A. Al Atif, «Data mining with its role in marketing, sales support and customer identification data analysis,» *International Journal Artificial Intelligent and Informatics,* pp. 104-116, 2022.

[6] K. S. E. &. S. A. K. Mehta, «Time Series Analysis: A Machine Learning Approach,» de *Springer*, Singapore, 2023, p. 193–204.

[7] O. M. V. &. V. A. Tsilingeridis, «Design and development of a forecasting tool for the identification of new target markets by open time-series data and deep learning methods.,» *Applied Soft Computing,* p. 132, 2022.

[8] B. S. D. K. B. S. N. A. P. S. &. B. D. Jamalpur, «Applications of Deep Learning in Marketing Analytics: Predictive Modeling and Segmenting Customers.,» de *2024 International Conference on Communication, Computer Sciences and Engineering (IC3SE)*, Gautam Buddha Nagar, India, 2024.

[9] E. Spiliotis, «Time Series Forecasting with Statistical, Machine Learning, and Deep Learning Methods: Past, Present, and Future,» *Springer International Publishing,* pp. 49-75, 2023.

[10] N. &. D. K. Vintha, «Comparative Analysis of Deep Learning Approaches for Analysis and Prediction of Multivariate Time Series Data,» *IEEE,* p. 76–82, 2023.

[11] A. Z. H. K. A. &. T. A. M. Bouguettaya, «Machine Learning and Deep Learning as New Tools for Business Analytics,» de *Handbook of Research on Foundations and Applications of Intelligent Business Analytics*, Souk Ahras, Algeria, IGI Global, 2022, p. 166–188.

[12] M. McGuirk, «Performing web analytics with Google Analytics 4: a platform review,» *Journal of Marketing Analytics,* pp. 854-868, 2023.

[13] N. R. A. D. M. R. Y. &. P. A. Kewate, «A review on AWS-cloud computing technology.,» *nternational Journal for Research in Applied Science and Engineering Technology,* pp. 258-263, 2022.

[14] R. W. O. P. A. V. W. E. A. &. T. C. Bello, «Architecture for detecting advertisement types,» Department of Computer Systems Engineering, Faculty of Information and Communication Technology, Tshwane University of Technology, Tshwane, South Africa, 2025.

[15] O. S. M. L. D. Z. U. &. P. C. Zimmermann, «Patterns for API design: simplifying integration with loosely coupled message exchanges,» Addison-Wesley Professional, 2022.

[16] E. A. a. W. L. S. Roychowdhury, «OPAM: Online Purchasing-behavior Analysis using Machine learning,» de *International Joint Conference on Neural Networks (IJCNN)*, Shenzhen, China, 2021.

[17] Scickit-Learn, «StandardScaler,» [En línea]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html.

[18] Google Developers, «Ajuste, sobreajuste y subajuste.,» Google, [En línea]. Available: https://developers.google.com/machine-learning/crash-course/overfitting/overfitting?hl=es-419#fitting.

[19] L. Q. N. V. Q. T. A. Q. Ngo, «K-fold cross validation diagram,» PLOS ONE, 8 Agust 2023. [En línea]. Available: https://plos.figshare.com/articles/figure/K-fold_cross_validation_diagram_/23405926/1.

[20] J. D. Diaries, «Choosing the Best Model: A Friendly Guide to AIC and BIC,» Medium, 6 November 2024. [En línea]. Available: https://medium.com/@jshaik2452/choosing-the-best-model-a-friendly-guide-to-aic-and-bic-af220b33255f.

[21] A. Nielsen, Practical Time Series Analysis. Prediction with Statistics and Machine Learning, O'Reilly Media, 2019.

[22] Statsmodels, «statsmodels.tsa.stattools.acf,» 3 October 2024. [En línea]. Available: https://www.statsmodels.org/stable/generated/statsmodels.tsa.stattools.acf.html#statsmodels-tsa-stattools-acf.

[23] J. &. T. M. Bernardino, «Business Intelligence Tools,» de *Computational Intelligence and Decision Making*, Springer, 2013, pp. 267-276.

[24] J. &. L. S. Struye, «Hierarchical temporal memory and recurrent neural networks for time series prediction: An empirical validation and reduction to multilayer perceptrons,» Neurocomputing, 2020.

[25] B. Soni, «Understanding Boosting in Machine Learning: A Comprehensive Guide,» Medium, 27 April 2023. [En línea]. Available: https://medium.com/@brijesh_soni/understanding-boosting-in-machine-learning-a-comprehensive-guide-bdeaa1167a6.

[26] A. D. K. Y. N. &. P. Y. Hartanto, «Stock Price Time Series Data Forecasting Using the Light Gradient Boosting Machine (LightGBM) Model.,» *JOIV: International Journal on Informatics Visualization,* p. 57, 2023.

[27] Python Software Foundation, «Python: Powerful Programming Language,» 2024. [En línea]. Available: https://www.python.org/.

[28] The Pandas Development Team, «Pandas: Data Analysis and Manipulation Library,» 2024. [En línea]. Available: https://pandas.pydata.org/.

[29] F. e. a. Pedregosa, «Scikit-learn: Machine Learning in Python,» 2024. [En línea]. Available: https://scikit-learn.org/.

[30] M. e. a. Abadi, «TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems,» 2024. [En línea]. Available: https://www.tensorflow.org/.

[31] F. Chollet, «Keras: Deep Learning for Humans,» 2024. [En línea]. Available: https://keras.io/.

[32] J. e. a. Smith, «pmdarima: Time Series Forecasting with Auto-ARIMA,» 2024. [En línea]. Available: https://alkaline-ml.com/pmdarima.

[33] G. e. a. Ke, «LightGBM: A Fast, Distributed, High-Performance Gradient Boosting Framework,» 2024. [En línea]. Available: https://lightgbm.readthedocs.io/.

[34] I. Desale, «E-commerce Sales Forecasting Using Machine Learning Algorithm,» *Dublin Business School,* 2024.

[35] A. Sharma, 2022. [En línea]. Available: https://data.world/anilsharma87/sales/workspace/file?filename=Amazon+Sale+Report.xlsx.

[36] Unknown, «Kaggle,» 2021. [En línea]. Available: https://www.kaggle.com/datasets/oviyaa/marketing-data-csv.

[37] R. Chavan, «Kaggle,» 2022. [En línea]. Available: https://www.kaggle.com/datasets/rahulchavan99/marketing-campaign-dataset.

[38] R. G. E. M. S. D. T. &. N. M. S. Rangasamy, Machine Learning-Based Functionalities for Business Intelligence and Data Analytics Tools, Auerbach Publications, 2024, p. 100–116.

[39] C. S. S. R. S. &. R. B. S. Reddy, «A Survey on Business Intelligence Tools for Marketing, Financial, and Transportation Services,» Singapore, Springer, 2019, pp. 495-504.

[40] D. &. Y. H. Dzyabura, «Machine learning and marketing.,» de *Handbook of Marketing Analytics*, EE Elger, 2018, p. 255–279.

[41] A. R. Pathe, «Machine Learning-Based Outlier Detection for Business Intelligence: A Scalable Time Series Analysis Framework,» *Indian Scientific Journal Of Research In Engineering And Management,* pp. 1-6, 2024.

ANEXO I: ALINEACIÓN DEL PROYECTO CON LOS ODS

ODS 8: Trabajo decente y crecimiento económico: El análisis de datos de campañas de marketing y comercio electrónico mejora la eficiencia de las estrategias empresariales, fomentando un crecimiento económico sostenible. Este trabajo permite identificar patrones y tendencias para optimizar los recursos invertidos en publicidad, promoviendo un uso más eficiente del presupuesto.

ODS 9: Industria, innovación e infraestructura: El uso de técnicas avanzadas de análisis de datos, como estos modelos de predicción, fomenta la innovación en el sector del comercio. A través de este proyecto, se impulsa la adopción de tecnologías modernas que fortalecen la infraestructura digital y promueven la transformación tecnológica.

ODS 12: Producción y consumo responsables: El análisis de datos de ventas y campañas publicitarias puede ayudar a las empresas a comprender mejor las necesidades de los consumidores, reduciendo el desperdicio de recursos y promoviendo patrones de consumo más responsables. Este enfoque contribuye a una gestión más sostenible de los recursos en el ámbito del comercio electrónico.

ANEXO II

7.1 LISTA DE VARIABLES DE LOS DATASETS

7.1.1 AMAZON INDIA

- **Order ID**: código único que identifica cada pedido realizado.
- **Date**: fecha en la que se generó el pedido.
- **Status**: estado actual del pedido.
- **Fulfilment**: indica si el pedido fue gestionado por Amazon o por el vendedor directamente.
- **Sales Channel**: plataforma por la que se realizó la venta (como Amazon.in u otros canales).
- **ship-service-level**: tipo de envío seleccionado (estándar, exprés, prioritario, etc.).
- **Style**: hace referencia al estilo o modelo del producto, especialmente útil en moda.
- **SKU**: código interno del producto para identificarlo en el inventario.
- **Category**: categoría general del producto (ropa, tecnología, hogar, etc.).
- **Size**: talla o dimensión del producto, si aplica.
- **ASIN**: identificador único de Amazon para cada producto.
- **Courier Status**: estado del envío por parte del servicio de mensajería.
- **Qty**: cantidad de unidades vendidas en ese pedido.
- **currency**: moneda en la que se realizó la transacción (probablemente INR).
- **Amount**: valor monetario total del pedido (precio * cantidad).
- **ship-city**: ciudad de destino del envío.
- **ship-state**: estado o región del destinatario.
- **ship-postal-code**: código postal del cliente.
- **ship-country**: país al que se envía el producto (en este caso, India u otros si aplica).
- **promotion-ids**: identificadores de promociones aplicadas al pedido (si las hay).
- **B2B**: indica si la venta fue hecha a una empresa (Business to Business).
- **fulfilled-by**: quién se encargó de preparar y enviar el pedido (Amazon o vendedor).

- **Unnamed**: 22: columna vacía o sin nombre, probablemente un error o residuo del archivo original.

7.1.2 CLIENTES

- **ID**: identificador único de cada cliente.
- **Year_Birth**: año de nacimiento del cliente.
- **Education**: nivel educativo alcanzado.
- **Marital_Status**: estado civil del cliente.
- **Income**: ingresos anuales declarados por el cliente.
- **Kidhome**: número de niños pequeños que viven en el hogar.
- **Teenhome**: número de adolescentes que viven en el hogar.
- **Dt_Customer**: fecha en la que el cliente se registró por primera vez.
- **Recency**: número de días desde la última compra realizada.
- **MntWines**: gasto total en vinos durante los últimos dos años.
- **MntFruits**: gasto total en frutas.
- **MntMeatProducts**: gasto total en productos cárnicos.
- **MntFishProducts**: gasto total en pescado.
- **MntSweetProducts**: gasto total en dulces y productos azucarados.
- **MntGoldProds**: gasto total en productos de lujo o de valor elevado (como oro).
- **NumDealsPurchases**: número de compras realizadas con descuentos o promociones.
- **NumWebPurchases**: número de compras hechas a través de la tienda online.
- **NumCatalogPurchases**: número de compras a través del catálogo impreso o digital.
- **NumStorePurchases**: número de compras realizadas en tienda física.
- **NumWebVisitsMonth**: número de visitas al sitio web en el último mes.
- **AcceptedCmp3–AcceptedCmp2**: variables que indican si el cliente aceptó cada una de las campañas de marketing numeradas.
- **Complain**: indica si el cliente ha presentado alguna queja.
- **Z_CostContact**: variable de control sin información real (valor constante).
- **Z_Revenue**: otra variable de control sin valor informativo real.
- **Response**: indica si el cliente respondió positivamente a la última campaña.

7.1.3 ANUNCIOS

- **campaign_item_id**: identificador único del ítem dentro de la campaña.
- **no_of_days**: duración de la campaña o del anuncio en días.
- **time**: fecha en la que se ejecutó la acción o medición.
- **ext_service_id**: código del servicio publicitario externo utilizado.
- **ext_service_name**: nombre de la plataforma de anuncios.
- **creative_id**: identificador del diseño publicitario utilizado.
- **creative_width / creative_height**: dimensiones del anuncio en píxeles.
- **search_tags**: etiquetas temáticas asociadas al anuncio para su categorización.
- **template_id**: código del diseño o plantilla publicitaria empleada.
- **landing_page**: URL de destino a la que se redirige el usuario tras hacer clic.
- **advertiser_id**: identificador único del anunciante.
- **advertiser_name**: nombre del anunciante que gestiona la campaña.
- **network_id**: código interno de la red publicitaria usada.
- **approved_budget**: presupuesto aprobado para la campaña, en moneda local.
- **advertiser_currency**: moneda en la que se fijó el presupuesto inicial.
- **channel_id / channel_name**: identificador y nombre del canal donde se mostró el anuncio.
- **max_bid_cpm**: puja máxima dispuesta a pagar por cada mil impresiones (CPM).
- **network_margin**: margen que retiene la red publicitaria.
- **campaign_budget_usd**: presupuesto total de la campaña convertido a dólares.
- **impressions**: número de veces que el anuncio fue mostrado.
- **clicks**: número de clics recibidos por el anuncio.
- **stats_currency**: moneda en la que se registran las métricas.
- **currency_code / exchange_rate**: código de moneda y tipo de cambio respecto al dólar.
- **media_cost_usd**: coste total invertido en medios, expresado en dólares.
- **position_in_content**: ubicación del anuncio dentro del contenido (si se conoce).
- **unique_reach / total_reach**: número de usuarios únicos y total de veces que se vio el anuncio.

- **search_tag_cat**: categoría asociada a las etiquetas de búsqueda.
- **cmi_currency_code**: moneda de referencia usada en el sistema de gestión.
- **timezone**: zona horaria desde la que se gestionó la campaña.
- **weekday_cat**: categoría del día (por ejemplo, día de semana o fin de semana).
- **keywords**: palabras clave asociadas al contenido del anuncio o al público objetivo.