



Universitat Oberta
de Catalunya

GRADO DE CIENCIA DE DATOS APLICADA

MARKETING MIX MODELLING

Simulador para la optimización de la inversión en marketing

MEMORIA

18 Junio 2023

TRABAJO FIN DE GRADO

El presente documento representa la memoria del Trabajo Final de Grado



Alberto de Torres Pachón

Dirección académica: Xavier Florit

Responsable académico: Elena Rodríguez

Trabajo final de grado 22536



Esta obra está sujeta a una licencia de Reconocimiento- NoComercial-Compartirlgual
[3.0 España de Creative Commons](#)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>MARKETING MIX MODELLING, Simulador para la optimización de la inversión en marketing</i>
Nombre del autor/a:	<i>Alberto de Torres Pachón</i>
Nombre del consultor:	<i>Xavier Florit</i>
Nombre del PRA:	<i>Elena Rodríguez Gonzàlez</i>
Fecha de entrega:	<i>07/2023</i>
Titulación o programa:	<i>Grado de Ciencia de Datos Aplicadas</i>
Área del Trabajo Final:	<i>Área de Marketing Science</i>
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<i>Marketing Mix Modeling, Time Varying Coefficient Model, Hierarchical Bayesian Model, Bayesian Time Series, ROI Marketing,</i>
Resumen del Trabajo	
<p>El ROI en marketing siempre ha sido uno de los grandes retos a resolver por las empresas. Sin embargo, los modelos actuales, basados en modelos econométricos no son efectivos, ya que la comercialización de los productos ha cambiado drásticamente debido a la llegada de la tecnología de marketing, la publicidad digital y otros disruptores.</p> <p>Para ello, se ha planteado evaluar diferentes modelos analíticos y predictivos, basados en las ventas, precio, distribución de productos, las impresiones, gasto en medios en diferentes canales y factores externos como las fuerzas macroeconómicas, el tiempo, la inflación o la estacionalidad.</p> <p>El desarrollo del proyecto se ha basado en la evaluación de cuatro modelos con diferentes técnicas de aprendizaje automático, la regresión lineal, regresión log-lineal, series temporales y modelo lineal bayesiano. Para ello, se han utilizado técnicas de Minería de Datos, análisis de probabilidad, modelos avanzados de optimización y de predicción.</p> <p>El resultado se presenta a modo de prototipo web de simulador, con la capacidad de evaluar y predecir la asignación presupuestaria óptima. Comprender el rendimiento de los canales de medios simulando cambios en el gasto y los efectos en los KPI objetivo (como las ventas) por canal de comunicación, con la posibilidad de revisar los resultados, hacer pruebas y apoyado en gráficos.</p> <p>Tanto el resultado de los análisis efectuados como la propia aplicación web permiten conocer mejor los modelos de marketing mix de nueva generación, sirviendo de base a futuros proyectos y pudiendo ser utilizado para la optimización de las inversiones de marketing de cualquier empresa.</p>	

FINAL PROJECT SHEET

Title of the work:	<i>MARKETING MIX MODELLING, Simulator for marketing investment optimization</i>
Name of author:	<i>Alberto de Torres Pachón</i>
Name of consultant:	<i>Xavier Florit</i>
Name of Area Teacher:	<i>Elena Rodríguez González</i>
Data of submission (mm/yyyy):	<i>06/2023</i>
Degree of programme:	<i>Degree in Applied Data Science</i>
Area of the Final Paper:	<i>Area of Marketing Science</i>
Language of the paper:	<i>Spanish</i>
Key words	<i>Marketing Mix Modeling, Time Varying Coefficient Model, Hierarchical Bayesian Model, Bayesian Time Series, ROI Marketing,</i>
Abstract	
<p>The ROI of marketing investments has always been one of the great challenges to be solved by companies. However, current models based on econometric models are not effective, since the marketing of products has changed drastically due to the advent of marketing technology, digital advertising, and other disruptions.</p> <p>To this end, it has been proposed to evaluate different analytical and predictive models, based on sales, price, product distribution, impressions, media spending in different channels and external factors such as macroeconomic forces, weather, inflation, or seasonality.</p> <p>The development of the project has been based on the evaluation of four models with different machine learning techniques, linear regression, log-linear regression, time series and Bayesian linear model. For this purpose, data mining techniques, probability analysis, advanced optimization and prediction models have been used.</p> <p>The result is presented as a web prototype simulator, with the ability to evaluate and predict the optimal budget allocation. Understand the performance of media channels by simulating changes in spending and the effects on target KPIs (such as sales) by communication channel, with the ability to review the results, test and supported by graphs.</p> <p>Both the results of the analyses performed and the web application itself provide a better understanding of the new generation marketing mix models, serving as a basis for future projects and can be used to optimize the marketing investments of any company.</p>	

Agradecimientos

A mis estimados compañeros, que han sido una parte integral de este viaje. Hemos compartido desafíos y retos. Su ayuda en el foro ha sido fundamental en mi desarrollo y aprendizaje. Aprecio profundamente su amistad, su apoyo y su valiosa colaboración.

A la Universidad Oberta de Catalunya (UOC), cuyo innovador sistema educativo ha sido el entorno perfecto para mi crecimiento y desarrollo académico. La flexibilidad y accesibilidad que ofrece la UOC me ha permitido equilibrar mis estudios con otras responsabilidades, y la amplia variedad de recursos y herramientas disponibles ha enriquecido enormemente mi experiencia de aprendizaje. Estoy sinceramente agradecido por la oportunidad de formar parte de esta notable institución.

Además, aprovecho para agradecer a los profesores y a mis tutores de por sus enseñanzas, materiales y la ayuda en las asignaturas.

Por último, pero no menos importante, a mi director de TFG, Xavier Florit por su apoyo a lo largo de este desafío académico y acompañamiento durante estos dos semestres, que han sido fundamentales para el éxito en cada etapa de este trabajo de grado. Sus consejos y comentarios me han ayudado a superar obstáculos y a seguir adelante, le estoy muy agradecido.

Índice

Contenido

1.	Introducción.....	1
1.1.	Contexto y justificación del Trabajo.....	1
1.2.	Objetivos.....	3
1.2.1.	Objetivos generales:.....	3
1.2.2.	Objetivos específicos:	3
1.3.	Impacto en sostenibilidad, ético-social y de diversidad	4
1.4.	Enfoque y método.....	5
1.4.1.	Metodología	6
1.5.	Desarrollo de Producto	9
1.6.	Estructura de la Memoria.....	9
2.	Planificación	10
2.1.	Tareas	10
3.	Planificación del trabajo.....	10
3.1.	Tareas y calendario:	10
3.2.	Hitos:	11
3.3.	Análisis de riesgos:.....	11
3.4.	Gestión de Riesgos:	12
4.	Análisis.....	12
4.1.	Conceptos y objetivos:.....	12
4.3.2.	Fuentes de datos externas (Variables de Control).....	18
4.3.3.	Selección de datos y variables del proyecto.....	19
4.4.	Análisis exploratorio de datos	19
4.5.	Efectos de aplicación en el Modelado de un MMM	25
4.5.1.	Efecto “·Ad Stock”	25
4.6.	Efecto “Diminishing Return”	26
4.7.	Modelos Tradicionales de Marketing Mix:	28
4.7.1.	Modelo de Regresión Lineal Multivariante.....	28
4.7.1.1.	Resultados del Modelo de Regresión Multivariante en el proyecto:	28
4.7.2.	Modelo Agregativo o Aditivo.....	31
4.7.3.	Modelo Multiplicativo	32
4.7.3.1.	Resultados del Modelo Multiplicativo en el proyecto:	34
4.8.	Modelos de Nueva Generación de Marketing Mix.....	36

4.8.1.	Modelo Robyn:	37
4.8.1.1.	Resultados del Modelo Robyn en el proyecto:	42
4.8.2.	Modelo Bayesiano con Carryover y Shape Effects.....	49
4.8.3.	Resultados del Modelo Bayesiano en el proyecto:	53
4.8.4.	Modelo Bayesiano con Stan.....	58
4.8.5.	Resultados del Modelo Bayesiano Stan en el proyecto	59
4.8.6.	Comparativa de los Modelos	65
4.8.6.1.	Comparativa de los modelos tradicionales:.....	65
4.8.6.2.	Comparativa de los modelos de nueva generación:.....	66
5.	Desarrollo Aplicación Web: “Nebula Navigator”.....	67
5.1.	Aplicación Web: “Nebula Navigator”	70
6.	Conclusiones.....	78
6.1.	Valoración del proyecto	78
6.2.	Conclusiones de los resultados	79
6.3.	Posibles proyectos derivados	81
7.	Glosario.....	82
8.	Bibliografía	83
9.	Anexos	86

Índice de ilustraciones

Ilustración 1 - Núcleo de la Propuesta Metodológica de enfoque “híbrido”	7
Ilustración 2 - capítulos y fases que se van a desarrollar en este proyecto.....	10
Ilustración 3 - grupos de tareas con su carga de trabajo en horas y porcentaje	10
Ilustración 4 - cronograma tareas	11
Ilustración 5 - análisis DAFO	12
Ilustración 6 - variables del marketing mix	13
Ilustración 7 - interacción variables de marketing	15
Ilustración 8 - impulsores del negocio.....	16
Ilustración 9 - Fuentes de datos	17
Ilustración 10 - gráfico de las ventas como serie temporal	20
Ilustración 11 - Box plots de las variables.....	21
Ilustración 12 - Curvas de las variables y las ventas	21
Ilustración 13 - Curvas de las variables y las ventas	22
Ilustración 14 - Curvas de las variables y las ventas	22
Ilustración 15 - gráfica con distribuciones para las variables numéricas	24
Ilustración 16 - matriz de correlación	25
Ilustración 17 - función Hill	27
Ilustración 18 - gráfica con Roas por canal.....	30
Ilustración 19 - gráfico con resultados del modelo en la inversión por cada medio	31
Ilustración 20 - gráfica estimación ventas sobre predicción	35
Ilustración 21 - gráfica de la inversión por canal predicha	36
Ilustración 22 - Comparación modelos tradicionales MMM vs nuevos modelos..	36
Ilustración 23 - Bases del Modelo de Marketing Mix.....	37
Ilustración 24 - Fases del modelo Robyn.....	37
Ilustración 25 - Fases del Modelo LightweightMMM	49
Ilustración 26 - gráfica con distribuciones posteriores de los efectos de los medios	55
Ilustración 27 - Estimación de la contribución de los medios a las ventas en 6 meses	57
Ilustración 28 - Estimación de la inversión por cada canal.....	58
Ilustración 29 - Estimación de la inversión total	58
Ilustración 30 - Gráficas con los efectos de marketing.....	62
Ilustración 31 - función Hill de cada canal.....	63
Ilustración 32 - Estimación de la contribución para cada canal de los medios en las ventas	63
Ilustración 33 - Contribución Canales de Marketing.....	64
Ilustración 34 - Contribución Canales de Marketing con el ROAS	64
Ilustración 35 - Contribución por canal y ROAS.....	65
Ilustración 36 - Arquitectura aplicación	68
Ilustración 37 - Tecnologías de la aplicación	70
Ilustración 38 - Pantalla de Login	71
Ilustración 39 - Home de la aplicación	71
Ilustración 40 - Pantalla gestión usuarios	71
Ilustración 41 - Gestión de usuarios alta.....	72
Ilustración 42 - Carga datos y variables.....	72
Ilustración 43 - Carga Datos	73
Ilustración 44 - Carga Variables	73
Ilustración 45 - Modelo en ejecución	74
Ilustración 46 - Modelo en ejecución	74
Ilustración 47 - Verificación calidad datos.....	75

Ilustración 48 - Comprobación Variaciones	75
Ilustración 49 - - Distribución impresiones de canales	76
Ilustración 50 - Curvas bayesianas.....	76
Ilustración 51 - - Estimación a 6 meses de la contribución de los canales en las ventas.	77
Ilustración 52 - estimación de la inversión por canales.....	77

1. Introducción

A lo largo de este capítulo se expone el contexto y la justificación del presente Trabajo de Fin de Grado. El tema elegido para este proyecto es la creación de un simulador de Marketing Mix Modelling¹ para la optimización de la inversión en marketing.

La gestión eficiente de los recursos y la asignación de fondos en los diversos canales de marketing son aspectos cruciales para todas las organizaciones en la actualidad. Las decisiones de inversión, en este contexto, deben orientarse no solo por intuiciones o experiencias, sino principalmente por la contribución efectiva de cada canal a las ventas globales. El Marketing Mix Modeling (MMM) es una herramienta esencial en el ámbito del marketing, que nos permite cuantificar la contribución de diversos factores a las ventas. Con esta metodología, podemos orientar de manera precisa la distribución de los recursos financieros a través de los múltiples canales disponibles. El objetivo principal del MMM es maximizar el retorno de la inversión y mejorar la eficiencia y eficacia de nuestras estrategias de marketing.

El MMM es un enfoque analítico que ha ganado amplia aceptación en todos los sectores. Su implementación permite medir y optimizar los presupuestos de marketing, contribuyendo a la toma de decisiones estratégicas más fundamentadas y efectivas. En este contexto, el papel de la ciencia de datos se vuelve particularmente relevante, permitiendo la generación de insights profundos y la mejora de la precisión en la asignación de recursos.

En resumen, el MMM nos brinda una visión clara y precisa de cómo cada elemento del mix de marketing impacta en las ventas, ayudándonos a tomar decisiones más informadas y obtener mejores resultados.

Este Trabajo Final de Grado (TFG) tiene como objetivo explorar en profundidad la aplicación de la ciencia de datos en la Modelización del Mix de Marketing. Pretendemos demostrar cómo la integración de estas dos disciplinas puede proporcionar una mejor comprensión del impacto de las estrategias de marketing y ayudar a las organizaciones a tomar decisiones más informadas y eficaces, destacando su relevancia y aplicabilidad en el escenario empresarial contemporáneo. Y siempre cumpliendo y respetando la normativa vigente y atendiendo a los valores debidos de la ética en el uso de los datos.

1.1. Contexto y justificación del Trabajo

Con la irrupción del marketing digital y el big data, el marketing es cada vez más analítico y metódico que nunca. De hecho, representa una de las mayores áreas de oportunidades para las aplicaciones de ciencia de datos y aprendizaje automático. Además las fuentes de datos en el marketing son cada vez más sofisticadas, que van desde las redes sociales hasta las bases de datos de la web, proporcionando grandes datos a una escala sin precedentes.

Esto provoca una complejidad del entorno del marketing, que entre los métodos analíticos disponibles, la simulación se convierte en la herramienta esencial de análisis y evaluación. Además, la publicidad es cada vez más compleja necesitando nuevas capacidades para evaluar las metodologías de medición. Dada la complejidad del entorno del marketing y la multitud de

¹ (Wikipedia contributors. (2023, May 7). Marketing mix modeling. In Wikipedia, The Free Encyclopedia. Retrieved 10:24, June 4, 2023, from https://en.wikipedia.org/w/index.php?title=Marketing_mix_modeling&oldid=1153635840)



métodos analíticos disponibles, la simulación puede ser una herramienta esencial para evaluar y comparar las opciones de análisis (Zhang & Vaver, 2017).

Actualmente son modelos económicos² que utilizan datos históricos agregados para modelar las ventas a lo largo del tiempo, en función de las variables de marketing o incluso de variables de control como el clima, la estacionalidad y la competencia. Los modelos tradicionales de marketing mix no proporcionan información en tiempo real ni se integran con las herramientas de inteligencia empresarial.

Cabe destacar el documento editado por Deloitte "The future is modeled, A how-to guide for Advanced Marketing Mix Models"³. En el cual se llega a la conclusión que los MMM presentan el máximo nivel de flexibilidad y adaptabilidad a estas dinámicas cambiantes, capaz de absorber nuevos imprevistos externos y recalibrar constantemente su enfoque, en una especie de "evolución darwiniana" de la medición de la eficacia de los medios.

Citando la importancia y ventajas de utilizar MMM:

- (1) Permite a las empresas crecer transformando prácticas de marketing basadas en datos y ciencia;
- (2) ayuda a todas las empresas a acceder a un marketing avanzado, seguro para la privacidad eficacia del marketing en el contexto de evolución de los comportamientos de los consumidores cambiante ecosistema del marketing digital;
- (3) es transparente, ofreciendo modelos personalizables que personalizables que democratizan la econometría y la subjetividad de los analistas.

Además, con la introducción del marco App Tracking Transparency (ATT) de Apple en 2021⁴, ya no se puede acceder a los datos a nivel de usuario en iOS a menos que un usuario opte por el seguimiento (véase la guía iOS 14.5+ Back to basics⁵). Del mismo modo, Google está tratando de limitar el intercambio de datos de usuario en Android a terceros, reduciendo la dependencia de los datos de identificador entre aplicaciones en un esfuerzo por fortalecer la privacidad del usuario en su Google Privacy Sandbox en Android⁶. Este es un punto clave en el que los MMM permiten ser respetuosos con la privacidad.

En esencia, como se cita también en el artículo:" Modelización de la combinación de medios: El regreso del análisis publicitario"⁷, dado que MMM no se alimenta de datos a nivel de usuario, sino que utiliza datos agregados de diversas fuentes y canales, proporciona un sólido análisis de medición de marketing, gracias a los avances en el uso del análisis de datos y el aprendizaje automático.

El presente trabajo fin de grado se basa por mi experiencia en el campo del marketing y la madurez adquirida a lo largo del grado de ciencia de datos aplicadas. Este trabajo está orientado a cubrir la necesidad de usar modelos predictivos, que ayuden a los profesionales del marketing, a maximizar el impacto en la campañas de comunicación, la promoción comercial, así como otras herramientas de marketing para crear planes de eficaces.

² Wikipedia contributors. (2020, December 18). Econometric model. In Wikipedia, The Free Encyclopedia. Retrieved 10:31, June 4, 2023, from https://en.wikipedia.org/w/index.php?title=Econometric_model&oldid=994917572

³ (<https://www2.deloitte.com/content/dam/Deloitte/es/Documents/estrategia/Deloitte-es-estrategia-y-operaciones-guide-advanced-marketing-mix-models.pdf>, n.d.)

⁴ (<https://www.adjust.com/glossary/apptrackingtransparency-att/>, n.d.)

⁵ (<https://www.adjust.com/blog/ios-14-5-back-to-basics-guide/>)

⁶ (https://privacysandbox.com/intl/es_es/android/, n.d.)

⁷ (<https://www.adjust.com/resources/guides/media-mix-modeling/>, n.d.)



Por todo esto mi propuesta es la de desarrollar un simulador de modelos de marketing mix que ayude tanto a medir la inversión en las acciones de comunicación y promoción como optimizar la asignación del presupuesto y distinguir los canales de marketing, tanto con alto, como bajo ROI.

1.2. Objetivos

En este Trabajo Final de Grado, me propongo afrontar una serie de problemas y objetivos concretos relacionados con el MMM utilizando las técnicas y métodos de la Ciencia de Datos. Con esta intención, formularemos preguntas relevantes cuyas respuestas nos permitirán desarrollar una comprensión más profunda de cómo aplicar de manera efectiva la Ciencia de Datos en el ámbito del MMM. Cada objetivo y problema identificado servirá como guía para realizar investigaciones y análisis, con el propósito final de avanzar en el conocimiento y las prácticas de asignación óptima de recursos en el marketing.

1.2.1. Objetivos generales:

Este Trabajo Final de Grado (TFG) se centra en tres objetivos generales principales en el contexto de la Modelización del Mix de Marketing (MMM) utilizando la Ciencia de Datos. Cada uno de estos objetivos se orienta a mejorar nuestra comprensión y uso de la MMM en la práctica del marketing.

- **Evaluación de Metodologías:** Nuestro primer objetivo es realizar una evaluación exhaustiva de las diferentes metodologías existentes para optimizar el presupuesto de marketing. Esta evaluación incluirá un análisis en profundidad de lo que cada metodología aporta a la práctica de la MMM. Buscaremos entender sus fortalezas, limitaciones, y su aplicabilidad en diferentes contextos y situaciones de marketing.
- **Definición de un Modelo de Simulación:** Nuestro segundo objetivo es desarrollar un modelo de simulación que permita estimar el retorno de inversión (ROI) de las diversas actividades de marketing. Este modelo se diseñará para ayudar a determinar la asignación óptima de presupuesto a cada actividad de marketing, en función de su eficacia prevista y demostrada. A través de este modelo, buscamos proporcionar una herramienta que facilite decisiones de inversión más informadas y precisas en marketing.
- **Identificación de Áreas de Mejora:** Finalmente, nuestro tercer objetivo es identificar áreas de mejora en la práctica actual de la MMM que puedan permitir una optimización más efectiva de las variables de marketing. Este objetivo implicará un examen crítico de los enfoques actuales y una búsqueda de oportunidades para mejorar la eficacia de la MMM en la asignación de recursos de marketing.

A través de la consecución de estos objetivos, buscamos aportar un nuevo entendimiento y herramientas útiles para la práctica de la Modelización del Mix de Marketing, mejorando la capacidad de las organizaciones para optimizar sus inversiones en marketing.

1.2.2. Objetivos específicos:

Para la elaboración del objetivo general, es necesario dividirlo y para ello se ha utilizado la técnica SMART, la cual consiste en la definición de estos a través de 5 variables:

S-Especifico M-Medible A-Alcanzable R-Relevante T-Tiempo

Capturar los datos para el modelo de MMM:

- S-** Los datos seleccionados representan la realidad de una empresa, con datos reales.
- M-** El archivo es un fichero CSV que incluye todas las variables de cuatro años de datos semanales.
- A-** Los datos se encuentran en un repositorio Github disponible para su uso..
- R-**Estos datos son muy completos para un proyecto de MMM, por su realidad.
- T-** Son de accesibilidad inmediata, mediante la descarga en la página Github.

Análisis de las variables de marketing y adicionales:

- S-** Analizar los datos de impresiones de media, gastos, variables macroeconómicas y de estacionalidad..
- M-**Cuando se termine se entenderán los datos que incluyen las diferentes variables de marketing. A-Para este proceso se realizará un Análisis Exploratorio de Datos (EDA).
- R-**Es un proceso imprescindible para conocer los datos y crear el modelo.
- T-**Esta fase puede llevar varios días, aproximadamente entre cinco y siete.

Crear y evaluar los modelos predictivos y de optimización:

- S-** Desarrollo de cinco modelos predictivos de marketing mix.
- M-**Este proceso se puede repetir varias veces hasta que el resultado sea óptimo.
- A-**Para este proceso se probarán distintos modelos de Aprendizaje Automático (ML).
- R-**Es un proceso imprescindible para cumplir con el objetivo general de este proyecto.
- T-**Esta fase llevará varias semanas, ya que son modelos complejos y la optimización del modelo será difícil desde el principio o sufrir de sobreajuste.

Crear y evaluar una herramienta de simulación (producto entregable):

- S-** Desarrollo de una aplicación web de simulación.
- M-**Una vez creada la plataforma, el servicio predictivo podrá funcionar para los usuarios.
- A-** Se crea un informe detallado con gráficas de los resultados de las simulaciones..
- R-**Es una herramienta fundamental para cumplir el objetivo general de este proyecto.
- T-**Esta fase puede llevar varias semanas.

Gracias al análisis de los objetivos específicos mediante la técnica SMART, podemos garantizar que con las distintas fases establecidas para la ejecución de este proyecto los objetivos específicos quedan cubiertos y se garantiza el éxito del objetivo general.

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

Durante la asignatura de Contextualización del Trabajo Fin de Grado se realizó un estudio detallado de las dimensiones requeridas en las tres dimensiones de la competencia transversal UOC “Compromiso ético y global”, estos son:

Dimensión sostenibilidad: Como resultado de este TFG se tendrá un impacto positivo de sostenibilidad medioambiental y/o huella ecológica. La creación de un modelo de marketing mix va a permitir reducir el consumo/ahorro energético, ya que al optimizar las campañas de marketing se van a utilizar menos recursos, al ser más eficientes y reducir el número de acciones. Además dentro del mix de marketing se pueden incluir los impactos en huella de carbono que se generan en cada medio, TV, Prensa, Internet,.. pudiendo optimizar no solo la inversión, sino el impacto en el medioambiente y cuantificar el impacto.

El poder incluir estas variables de impacto medioambiental en mi TFG son una aportación innovadora en el uso de las técnicas de marketing para contribuir al ahorro de CO2 y a la sociedad que me han motivado en parte en este trabajo.



Dimensión comportamiento ético y de responsabilidad social (RS): El resultado de este TFG debería impactar positivamente en la parte ético-sociales del marco normativo/legislación al considerar y verificar que los datos que se use deben garantizar la privacidad, en base a las normativas existentes como la GDPR.

Los modelos de marketing mix permiten reducir el sesgo humano, ya que homogeniza la forma de medir la eficacia del marketing, permitiendo que el analista siga un proceso de modelización estandarizado para evaluar cada variable de la forma adecuada. Controlar el sesgo humano es uno de los principales retos de los modelos de Marketing Mix.

Además podría impactar también en el ODS 8 - Decent work and economic growth⁸; ya que estamos creando herramientas para una optimización de los recursos económicos, que van a generar un crecimiento económico y dar más valor a los puestos de trabajo.

Dimensión diversidad, género y derechos humanos: La realización de este proyecto debe tener un impacto positivo en el uso responsable de los datos, al incorporar medidas de control de los sesgos en las variables de los datos a utilizarse, evitando predicciones inexactas y evitando discriminaciones en aspectos de género, diversidad (raza, religión, orientación sexual, funcional, etnia, ideología...).

Además de los datos también se medirá el nivel de sesgo de los algoritmos, que pueden dar lugar a una discriminación entre las personas, pueden existir datos disponibles en que los algoritmos pueden provocar tratos discriminatorios, impactando en los ODS: ODS 5 - Gender equality⁹ y ODS 10 - Reduced inequalities¹⁰.

1.4. Enfoque y método

La metodología a seguir en el desarrollo de este modelo de marketing mix debe ser lo suficientemente flexible como para modelar una amplia variedad de situaciones de marketing que incluyen diferentes combinaciones de gasto en publicidad, niveles de eficacia de los anuncios, tipos de orientación de los anuncios, estacionalidad de las ventas, actividad de la competencia y mucho más.

Tradicionalmente, el método utilizado para el modelado del mix de marketing mediante algún tipo de análisis de regresión, a menudo regresión lineal múltiple. El problema de este enfoque es que hay que estimar al menos una variable para cada punto de contacto de marketing y ventas, donde los datos suelen ser escasos: Los conjuntos de datos típicos se limitan a unos pocos años. Así los datos procesados con frecuencia diaria, como el gasto en medios de comunicación, y el número de impresiones o inversiones que genera este gasto, intentando estimar el valor de miles de observaciones con muchos valores atípicos posibles.

Una de las razones por las que fracasan los modelos de regresión tradicionales es que los datos no están equipados con ningún conocimiento previo: los propios datos no tienen ninguna información sobre las funciones del marketing. Cada parámetro del modelo se determina como si fuera independiente: cada parte del modelo puede girar con total independencia de todos los demás.

⁸ (<https://www.un.org/sustainabledevelopment/wp-content/uploads/2018/09/Goal-8.pdf>, n.d.)

⁹ (<https://www.un.org/sustainabledevelopment/gender-equality/>, n.d.)

¹⁰ (<https://www.un.org/sustainabledevelopment/inequality/>, n.d.)



Lo que necesitamos es una forma de incorporar ese conocimiento del dominio (en este caso, conocimiento específico de la disciplina del marketing) para que pueda guiar el modelo. Para ello, se utilizará la estadística bayesiana¹¹, que es una rama de la teoría de la probabilidad, y que sí incluye el conocimiento del dominio. Esto se expresa a través de lo que se conoce como priors.

La combinación de ambos, la probabilidad a priori y la probabilidad real de lo que podría ocurrir basándose en la creencia que representa la probabilidad a priori, es lo que constituye el modelo final: Lo mejor de ambos mundos. En el contexto de la medición de la eficacia del marketing, sería un Modelado de Marketing Total porque el modelado se basa en una visión más holística. Se intentará adaptar el modelo a cómo sabemos que funciona el marketing a partir de la experiencia y los conocimientos previos, y no al revés.

1.4.1. Metodología

En este Trabajo Final de Grado, que tiene como objetivo el diseño y desarrollo de un producto de ciencia de datos que se aplica al ámbito de la Modelización del MMM me apoyare en dos metodologías clave que hemos estudiado durante este grado: CRISP-DM (Cross-Industry Standard Process for Data Mining)¹² y PMBOK (Project Management Body of Knowledge)¹³.

La metodología CRISP-DM (Cross-Industry Standard Process for Data Mining) es una metodología ampliamente reconocida y utilizada en proyectos de Minería de Datos. Fue desarrollada como un estándar "de facto" en el mundo empresarial y ofrece un enfoque tradicional para llevar a cabo proyectos de Data Mining de manera efectiva.

CRISP-DM propone un conjunto de actividades estructuradas que deben ser realizadas durante el desarrollo del proyecto. Cada actividad se divide en tareas específicas, indicando las salidas generadas por cada tarea y las entradas requeridas. Esta estructura organizada facilita el seguimiento y la gestión del proyecto, asegurando que se cubran todas las etapas necesarias para obtener resultados relevantes.

Una de las características distintivas de CRISP-DM es su naturaleza cíclica. Esto significa que el proceso se realiza en iteraciones, permitiendo ajustes y refinamientos continuos en función de los resultados obtenidos en cada etapa. Esta flexibilidad nos acerca gradualmente a una solución óptima, ya que cada iteración proporciona información valiosa que ayuda a mejorar y optimizar el análisis de datos.

En resumen, la metodología CRISP-DM es ampliamente aceptada como un enfoque efectivo y estructurado para proyectos de Minería de Datos. Su enfoque cíclico y su enfoque en actividades y tareas específicas proporcionan una guía clara y sistemática para el desarrollo de proyectos de Data Mining exitosos.

Por otro lado, la Guía PMBOK es el estándar de gestión de proyectos del PMI (Project Management Institute) [PMBOK Guide 2013], aunque no es en sí misma una metodología, proporciona un conjunto de buenas prácticas esenciales para una gestión efectiva del proyecto. Esto nos asegura tener una visión holística y coherente durante todo el proceso de desarrollo del producto de ciencia de datos.

¹¹ (Estadística bayesiana. (2023, 1 de mayo). Wikipedia, La encyclopédie libre. Fecha de consulta: 20:40, mayo 1, 2023 desde https://es.wikipedia.org/w/index.php?title=Estad%C3%ADstica_bayesiana&oldid=150897601)

¹² (CRISP-DM - Data Science Process Alliance, <https://www.datascience-pm.com/crisp-dm-2/>, n.d.)

¹³ (<https://www.pmi.org/pmbok-guide-standards/foundational/PMBOK>, n.d.)



Es importante destacar que la gestión del proyecto TIC y el ciclo de producción del producto TIC son dos aspectos distintos pero interrelacionados de este Trabajo Final de Grado. Por un lado, la gestión del proyecto se regirá por las buenas prácticas sugeridas por PMBOK, asegurando una coordinación y planificación eficientes. Por otro lado, la creación del producto de ciencia de datos será guiada por la metodología CRISP-DM, que se enmarca en la fase de ejecución del proyecto.

Por lo que adoptare un enfoque metodológico "híbrido" integrando los principios del Project Management Institute (PMI)¹⁴, que divide un proyecto en tres fases: iniciación, intermedia y cierre. Además, la metodología CRISP-DM se alinea con los cinco grupos de procesos definidos en el PMBOK (2013): iniciación, planificación, ejecución, control y cierre. Estos grupos de procesos proporcionan una estructura sólida para llevar a cabo proyectos de Minería de Datos de manera efectiva.

Por otro lado, la metodología también incorpora las ideas propuestas por Charvat (2003), quien establece que toda metodología de proyecto debe incluir fases específicas. Aunque las fases pueden variar según el proyecto o la industria, generalmente se incluyen las siguientes: concepto, desarrollo, implementación y soporte.

La fase de iniciación marca el comienzo del proyecto y tiene como objetivo definir el alcance, los objetivos y los recursos necesarios. La fase de planificación implica la elaboración de un plan detallado que establece las actividades, los plazos, los recursos y los riesgos del proyecto. La fase de ejecución es donde se llevan a cabo las actividades planificadas y se recopilan los datos necesarios para el análisis. La fase de control se centra en supervisar el progreso del proyecto, realizar ajustes según sea necesario y asegurar que se cumplan los objetivos establecidos. Por último, la fase de cierre implica finalizar todas las actividades del proyecto, documentar los resultados y realizar una evaluación final.

Al combinar los cinco grupos de procesos del PMBOK con las fases propuestas por Charvat, la metodología CRISP-DM ofrece una estructura sólida y completa para la gestión de proyectos de Minería de Datos. Esta combinación permite una planificación efectiva, una ejecución adecuada y un cierre exitoso del proyecto, asegurando así la entrega de resultados de alta calidad.

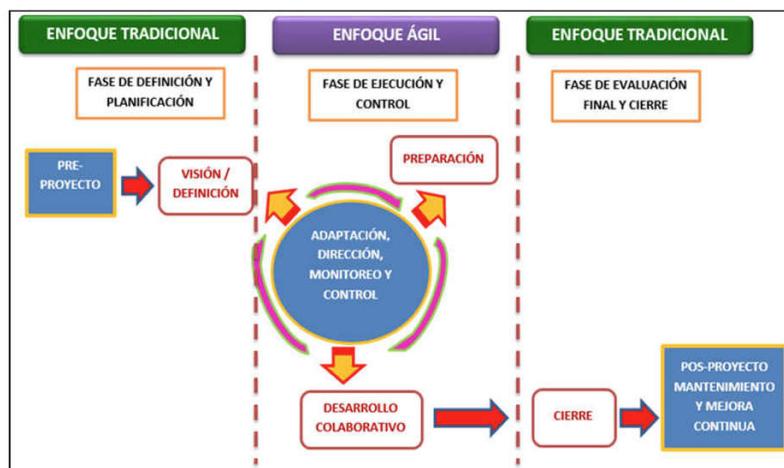


Ilustración 1 - Núcleo de la Propuesta Metodológica de enfoque “híbrido”

Fuente: Patricia Cristaldo, Departamento Ingeniería en Sistemas Universidad Tecnológica Nacional, Argentina

¹⁴ (<https://www.pmi.org/pmbok-guide-standards/foundational/pmbok> ,n.d.)

En consecuencia, nuestro enfoque metodológico híbrido para este proyecto se divide en estas fases clave:

- **Entendimiento del Negocio:** Esta etapa es crucial para entender los objetivos del proyecto desde una perspectiva de negocio. Incluye definir los problemas y oportunidades relacionados con el marketing mix y establecer los objetivos del proyecto de modelado del marketing mix.
- **Entendimiento de los Datos:** En esta fase se recopilan y exploran los datos relacionados con el marketing mix. Esto implica realizar un análisis inicial de los datos, examinar la calidad y la estructura de los datos,
 - Describir los datos e identificar las variables y los tipos de datos que son relevantes para el modelado del marketing mix.
 - Evaluación inicial de la calidad y consistencia de las variables.
 - Hacer un diccionario de los datos e identificar las variables.
 - Realizar Análisis Exploratorio de Datos (Amaratunga, Cabrera,& Morgenthaler, 2018).
 - Para el desarrollo se usaran el lenguaje Python y R. Mediante el IDE DataSpell¹⁵, Rstudio y la plataforma Google Colab¹⁶.
 - Comprobación de la calidad de los datos.(Batini & Scannapieco, 2010)
 - Hay que asegurar que no hay sesgos y discriminaciones (Belenguer, 2022)
- **Preparación de los Datos:** Aquí, los datos se preparan para el modelado.
 - Limpieza de los datos,
 - Corrección de sesgos y creación de nuevas variables o características, si es necesario.
 - Los datos también se procesan de nuevo según las necesidades específicas del modelo de marketing mix.
 - Nuevo procesamiento para verificar que todo está correcto.
 - Igualmente se usara el lenguaje Python, R y los mismos IDEs.
- **Modelado:** En esta fase se seleccionan y aplican las técnicas de modelado apropiadas para los datos del marketing mix.
 - Aplicación de modelos de regresión multivariante¹⁷, logarítmico multiplicativo¹⁸, bayesiano, series temporales¹⁹, rigde regresion²⁰, Numpyro²¹ y Stan²² para la programación probabilística y cualquier otro modelo relevante para los datos del marketing mix.
 - Los datos se dividen en conjuntos de entrenamiento y prueba y se realizan pruebas para evaluar la eficacia de los modelos.
 - Esta fase es muy importante para verificar que los modelos son representativos de la realidad de un MMM.
 - Igualmente se usara el lenguaje Python, R y los mismos frameworks.
- **Evaluación:** En esta fase se evalúa la eficacia del modelo de marketing mix.
 - Se verifica si los resultados cumplen con los objetivos establecidos.

¹⁵ (<https://www.jetbrains.com/es-es/dataspell/>,n.d.)

¹⁶ (<https://research.google.com/colaboratory/faq.html>, n.d.)

¹⁷ (<https://www.mygreatlearning.com/blog/introduction-to-multivariate-regression/>,n.d.)

¹⁸ (Donna, ... Rudolf, (2022) Multiple Regression in Statistical Methods)

¹⁹ (Serie temporal. (2022, 22 de marzo). Wikipedia, La enciclopedia libre. Fecha de consulta: 01:52, marzo 22, 2022, <https://es.wikipedia.org>)

²⁰ (Ridge regression. In Wikipedia, 2023, from https://en.wikipedia.org/w/index.php?title=Ridge_regression&oldid=11288726)

²¹ (Pyro PPL on NumPy, Authors Uber AI Labs, n.d.)

²² (Kentaro (2022), Bayesian Statistical Modeling with Stan, R, and Python, Springer)



- Se lleva a cabo un proceso de revisión que verifica si todos los pasos se han realizado correctamente, corrigiendo cualquier problema que se identifique.
- Esta fase es muy importante para evaluar entre todos los modelo de MMM que aportan y cual optimiza mejor el ROI en la inversión de marketing.
- **Despliegue:** Para que el modelo sea útil, debe ser accesible y utilizable. En esta fase, se despliega el modelo en una aplicación web de simulador de marketing mix. El objetivo es tener un prototipo mínimo viable que pueda ser probado y utilizado por los usuarios.

1.5. Desarrollo de Producto

El alcance de este proyecto limita el resultado a la obtención de un prototipo mínimo viable (MVP), que permita realizar simulaciones de modelos de marketing mix y que sirva de base para un futuro desarrollo de nuevas funcionalidades y modelos de marketing. Para ello se diseñara como interface de usuario un desarrollo web en la nube.

El prototipo incluirá los siguientes aspectos:

- Gestor de usuarios para mantener la privacidad de los datos y modelos de simulación.
- Panel de administración para la gestión de mantenimiento de la aplicación y las APIS.
- Instrucciones para la utilización de la aplicación, carga de datos, selección de las variables.
- Verificación de la calidad de los datos subidos a la plataforma web, con análisis exploratorio y estadístico. Y resultado de las simulaciones con gráficas y medidas del ROI del modelo de marketing mix predictivo por cada canal. Como verificación del resultado de este proyecto.
- Memoria técnica que contempla la totalidad del proyecto y código utilizado.
- Presentación virtual donde se explicarán los aspectos clave de este proyecto y la consecución de los objetivos, además de una visualización del producto.

1.6. Estructura de la Memoria

En el siguiente cuadro se describen los capítulos y fases que se van a desarrollar en este proyecto:

Nº	Nombre del apartado	Descripción del contenido
1	Introducción	En esta sección se expone la extensión y la legitimación del estudio, conduciendo a un conjunto de metas y la propuesta de la técnica que se aplica.
2	Planificación	Se asignan y proponen las diversas actividades, fijando objetivos claves y llevando a cabo una evaluación de los riesgos preliminares.
3	Análisis	Estudio de los datos, las variables con un análisis exploratorio EDA y transformación de los datos.
4	Ejecución de los modelos	Se seleccionan los diferentes modelos de machine learning que se van a evaluar. Preparación de las necesidades de los entornos y ajuste de los hiperparámetros para el ajuste de la optimización y cálculo del ROI.
5	Desarrollo aplicación Web	Se define la arquitectura de la aplicación, la funcionalidad que va a tener, las pantallas de interfaz y los lenguajes de programación. Se desarrollan las APIs para integrar el modelo y que se pueda desarrollar la funcionalidad del simulador.



6	Conclusiones	Evaluación de los resultados obtenidos
7	Glosario	Principales terminología utilizada y su explicación.
8	Bibliografía	Completa documentación de la bibliografía utilizada y leída para el desarrollo del proyecto.
9	Anexos	Se incluyen todos los desarrollos utilizados en el proyecto, así como la documentación usada de apoyo a los resultados del trabajo.

Ilustración 2 - capítulos y fases que se van a desarrollar en este proyecto

2. Planificación

Con relación a la programación del proyecto, se ha confeccionado un cronograma con fechas fundamentales, estableciendo las acciones esenciales para alcanzar las metas definidas. Para lograrlo, se ha considerado la carga de trabajo necesaria para cada tarea individual y se han evaluado los riesgos potenciales que podrían representar un obstáculo para la realización del proyecto.

El volumen de trabajo se distribuye a lo largo de las diferentes fases y etapas durante un periodo estimado de 4 meses, desde marzo hasta junio, calculando un total de 350 horas. Y para la programación, desarrollo y ejecución del proyecto solo se dispone de un solo recurso humano.

2.1. Tareas

Este documento abarca un conjunto de 4 grupos de tareas para segmentar el esfuerzo requerido en la organización del proyecto. En total, se han destinado 60 horas de contingencia, que pueden ser empleadas como acciones correctivas en la gestión del esfuerzo planificado. En la tabla siguiente se pueden apreciar los diferentes grupos de tareas con su carga de trabajo en horas y porcentaje:

Código	Nombre	Esfuerzo (h)	Esfuerzo (%)	Calendario Inicio	Calendario Fin
1	Gestión del proyecto	50	12%	10/03/2023	18/06/2023
2	Análisis y diseño	80	19%	08/10/2022	04/11/2022
3	Desarrollo y ejecución	230	55%	05/11/2022	23/12/2022
4	Gestión de riesgos	60	14%	28/09/2022	23/12/2022
Total		420	100%		

Ilustración 3 - grupos de tareas con su carga de trabajo en horas y porcentaje

3. Planificación del trabajo

3.1. Tareas y calendario:

Se han definido las tareas para cada uno de los objetivos que se han establecido, resultando el siguiente cronograma:



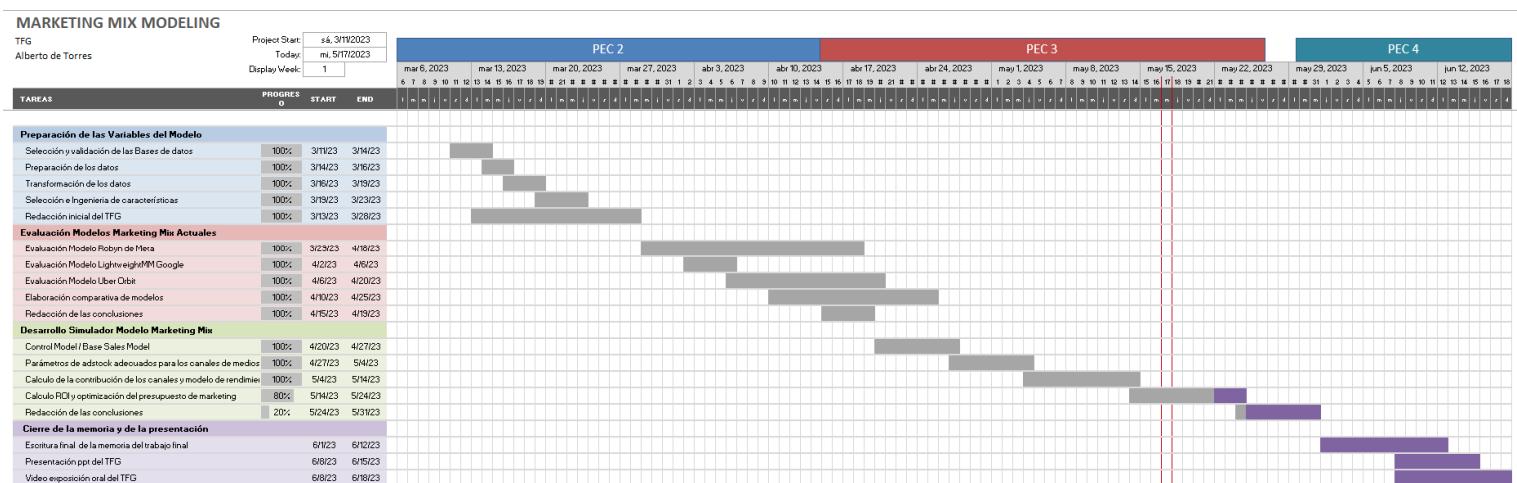


Ilustración 4 - cronograma tareas

3.2. Hitos:

Se han marcado y delimitado unos Hitos, en base a las fechas de las cuatro PECs de la asignatura, en las que se establecen los siguientes hitos que serán claves para avanzar en las sucesivas etapas del proyecto:

- Validación de las bases de datos que se van a utilizar y preparación de las Variables del Modelo. Esto condiciona el poder empezar a desarrollar el modelo para que sea viable.
- Para el desarrollo del modelo de marketing mix, primero se conocerán los modelos actuales más importantes, para utilizar este conocimiento en el desarrollo del modelo definitivo. Este hito se producirá en la mitad del proyecto y permitirá durante la segunda mitad del proyecto desarrollar el modelo.
- Una vez conseguido el modelo óptimo de marketing mix se desarrollara la aplicación web para crear el simulador que pueda validar el proyecto.
- Y el último hito será la redacción final del documento en base a lo conseguido en el modelo y los demás materiales a presentar.

3.3. Análisis de riesgos:

Estos son los factores que pueden repercutir negativamente en el seguimiento del plan de trabajo y en la consecución del proyecto:

- Obtención de fuentes de datos que dispongan de las variables necesarias para poder evaluar y comparar la eficacia del modelo de marketing mix.
- Mitigar los sesgos, bien en la selección de los datos o establecer mecanismos que aseguren la paridad de las características sensibles.
- Entender claramente las necesidades de los profesionales de marketing que necesitarían del simulador de marketing mix para que les permita diseñar estrategias de marketing más eficaces.
- Poder incorporar al modelos las diferentes casuísticas que implica el impacto de la publicidad en las ventas, como el adstock o la relación entre gasto y contribución.
- Poder aplicar al modelo los diferentes algoritmos de regresión múltiple y metodología bayesiana para que den un resultado real y que tenga sentido.
- Dedicar mucho tiempo a la creación del modelo y no profundizar lo suficiente en los resultados y aplicación del modelo.



- No poder acabar este trabajo con la suficiente profundidad por la limitación del tiempo disponible para la realización del TFG.
- Dificultad para crear el modelo al ser muy amplio el campo que abarca el marketing mix, muchas variables y poder determinar cuáles serían realmente las que predicen de forma real mejor.

3.4. Gestión de Riesgos:

- En caso de bloqueos técnicos en el desarrollo del modelo a desarrollar, pediré ayuda a los profesores del Grado o a terceros.
- Balancear el alcance del trabajo propuesto podrá demostrar el uso de del dominio aprendido en el grado y que no sea demasiado ambicioso.
- Con una buena planificación que me ayude a equilibrar los esfuerzos durante el desarrollo del TFG entre las diferentes fases, tanto de investigación, como en la parte de desarrollo técnica del modelo como en la escritura y presentación del TFG.

Como resumen de este estudio realizado, se ha desarrollado un cuadro DAFO que sintetiza los aspectos analizados y la situación de partida de este proyecto.

ANÁLISIS DAFO			
Debilidades		Amenazas	
<ul style="list-style-type: none"> • Insuficientes datos y fuentes. • Validez datos históricos por entorno cambiante. • Dado los pocos datos históricos origina problema de sobreajuste, provocando efecto de multicolinealidad 		<ul style="list-style-type: none"> • Fuentes de datos con verificación de su calidad. • Conseguir corregir sesgos de los datos y algoritmos. • Garantizar el reglamento de protección de datos • Conseguir un modelo con un nivel de precisión adecuada . 	
Fortalezas		Oportunidades	
<ul style="list-style-type: none"> • Modelos Bayesianos que permiten una mayor precisión y fiabilidad. • Conocimiento en las palancas del marketing mix con mucha literatura existente • Necesidad real profesionales del marketing en el simulador. 		<ul style="list-style-type: none"> • Crear herramienta de optimización marketing mix que permita simulaciones fiables. • Desarrollar modelo con aplicación real en el marketing. • Mejorar las investigaciones actuales de los modelos del marketing mix. 	

Ilustración 5 - análisis DAFO

4. Análisis

El apartado de análisis se dedica al estudio de los modelos de marketing, los tipos de enfoque que existen, así como los diferentes modelos y los tipos de resultado. Por otro lado, en el apartado funcional, se definen los requisitos que deben cumplir los modelos para cumplir con los objetivos y también, se define el alcance de los modelos..

4.1. Conceptos y objetivos:

Desde que el término "Marketing Mix Models" (MMM) fue acuñado por primera vez en 1949 por Neil Borden²³, ha experimentado numerosas interpretaciones y, como resultado, ha evolucionado en gran medida a lo largo del tiempo.

²³ (Wikipedia contributors. (2022, May 13). Neil H. Borden, https://en.wikipedia.org/w/index.php?title=Neil_H._Borden&oldid=1087578931)



Los modelos de MMM se refieren a enfoques analíticos que buscan cuantificar y medir el impacto de las diferentes variables de marketing en los resultados empresariales. Estos modelos han experimentado continuas actualizaciones y refinamientos a medida que ha avanzado la tecnología y se han desarrollado nuevas técnicas de análisis.

En sus inicios, los modelos de MMM se centraban en el análisis de las famosas "4P" del marketing: producto, precio, distribución y promoción. Sin embargo, con el tiempo, se han agregado nuevas variables y enfoques, como el impacto de las redes sociales, el marketing digital y otras estrategias de comunicación.

Estos modelos permiten a las organizaciones evaluar el rendimiento de las inversiones en marketing y tomar decisiones más informadas sobre cómo asignar sus recursos. Así podemos definir un modelo de marketing mix como una ecuación que se utiliza para predecir variables objetivo como las ventas, en función de:

- Medios de comunicación, tanto offline como la televisión, la radio, la publicidad exterior, etc.; los medios digitales, como las redes sociales, la medios online de búsqueda pagados, el correo electrónico, el marketing de afiliación, etc.;
- Factores de marketing no relacionados con los medios, como los precios, las promociones, etc.;
- Otras variables no relacionadas con el marketing, como datos macroeconómicos, eventos geopolíticos, etc.

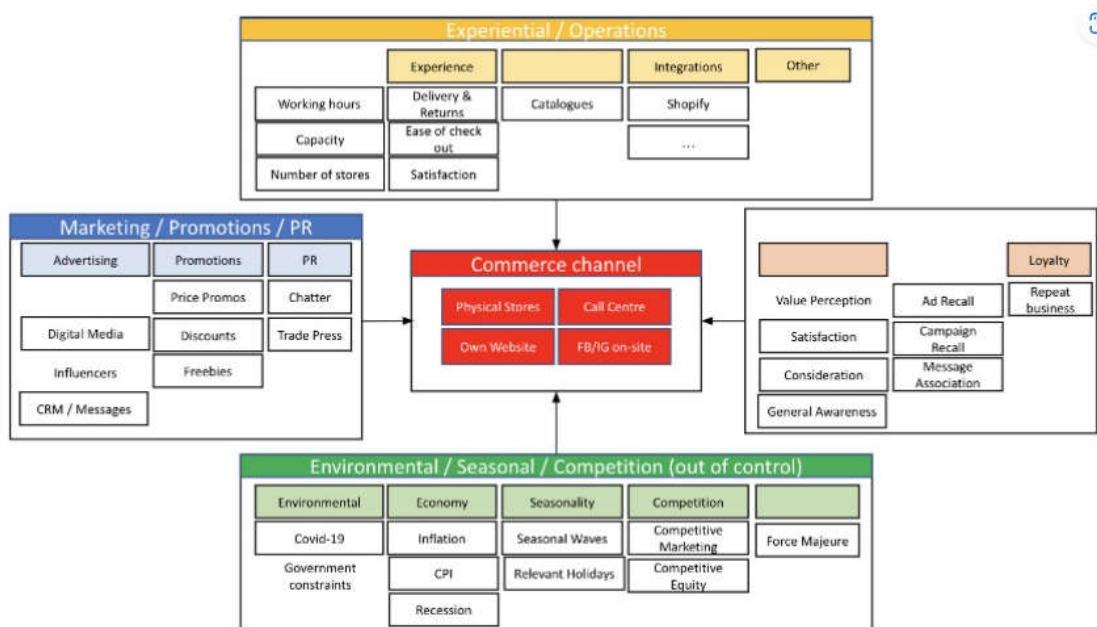


Ilustración 6 - variables del marketing mix

Fuente: <https://josebaruiz.com/marketing-mix-model-desarrollado-con-robyn-paso-a-paso/>

Por lo que un modelo de marketing mix es una combinación compleja de varios modelos estadísticos y funciones de transformación de datos. Pero hasta ahora los MMM basados en modelos económéticos presentan los siguientes retos:

- **Sesgo en el MMM:** Tradicionalmente, construir un MMM requiere de mucho juicio humano, lo que puede dar lugar a "sesgos del analista" al establecer los parámetros del

modelo. El sesgo del analista desempeña un papel en las siguientes actividades críticas en el proceso de construcción del modelo:

- Selección de características: el analista decide qué variables explicativas se deben usar en el modelo.
- Tendencia y estacionalidad: el analista decide cómo se construyen las variables de tendencia/estacionalidad.
- El analista decide la tasa de retardo y la curva de saturación para cada canal de medios.
- El analista decide qué resultado del modelo se debe elegir.
- El analista compara el resultado del modelo y el experimento manualmente.

Como resultado de este sesgo del analista, el MMM tradicional puede ser subjetivo, inflexible y difícil de construir. El MMM tradicional es lento. Puede llevar meses, e incluso todo un año, construir un modelo inicial que generalmente se actualiza una o dos veces al año. Además, la disponibilidad y la granularidad de los datos de entrada también ralentizan el proceso. Debido a esto, los conocimientos obtenidos del MMM tradicional generalmente no son aplicables para obtener información y optimización de campañas en tiempo real.

Debido a este proceso de modelado manual, las extensiones del modelo, como la estructura anidada y la interacción, requieren aún más esfuerzo y tiempo, y pueden introducir sesgos adicionales.(Zhou Leonel Sentana Igor Skokan Antonio Prada, 2021)

Estas ineficiencias puede llevar a una asignación incorrecta de medios y decisiones que perjudican el crecimiento del negocio.

Por lo que el objetivo es desarrollar un MMM que solucione estos retos y conseguir optimizar las inversiones, de tal forma que permitan:

- Conocer la eficacia de los diferentes canales de medios que impacten en las ventas. Pero que permita actualizaciones de forma continua al incorporar nuevos datos, como nuevas campañas o clientes, permitiendo conocer el ROI y optimizar la inversión.
- Entender cómo se relacionan los factores del marketing con la tasa de adquisición de clientes y conocer como impactan las diferentes variables, factores externos, cambios en la estacionalidad, factores económicos, cambios en los precios, etc.
- Conocer el coste de adquisición de clientes por canal o el grado de saturación del canal, esto podría ayudar en la planificación futura del gasto de marketing en todos los canales.
- Optimización en la toma de decisiones de gasto, que permita calcular presupuestos a través de canales de medios que maximizan nuevos clientes para un presupuesto total determinado.
- Realizar simulaciones de marketing, con diferentes pruebas de incremento o disminución para tener una mayor confianza en la asignación de presupuestos.

Como resultado de este trabajo se evaluarán diferentes modelos de MMM, que permitan mejorar la comprensión y que se puedan realizar simulaciones iterativas para ser más precisos en las inversiones y optimizar el ROI.

4.2. Definición del dominio

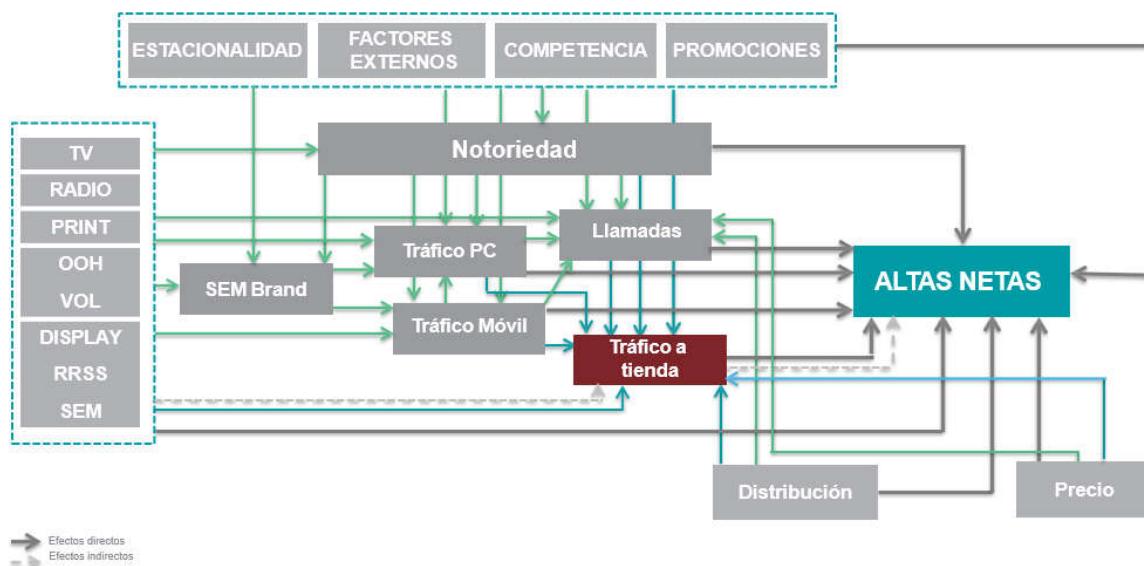
Los modelos de MMM son técnicas analíticas que utilizan datos históricos para medir el rendimiento de las estrategias de marketing y prever el impacto de futuras estrategias. Estos modelos examinan cómo las diversas variables de marketing (como el precio, la publicidad, la distribución y las promociones) afectan las ventas o el volumen de negocio. Los modelos



permiten a los encargados de la toma de decisiones entender cómo cada elemento del mix de marketing contribuye al rendimiento total, aislando los efectos de cada variable del impacto de las demás (Kumar, 2017).

Estos modelos son particularmente útiles para estudiar el volumen de negocios en una o varias regiones, ya que tienen en cuenta la evolución de diferentes variables que influyen en ellas. Los MMM pueden identificar las tácticas de marketing que han tenido un impacto significativo en una región particular y pueden ayudar a los especialistas en marketing a ajustar sus estrategias para maximizar su impacto (Kumar & Leone, 1988).

Por ejemplo, si una empresa lanza una campaña publicitaria en una región y ve un aumento en las ventas, un modelo de marketing mix podría ayudar a determinar si el aumento fue el resultado de la campaña publicitaria o si se debió a otros factores, como un cambio en los precios o las condiciones del mercado.



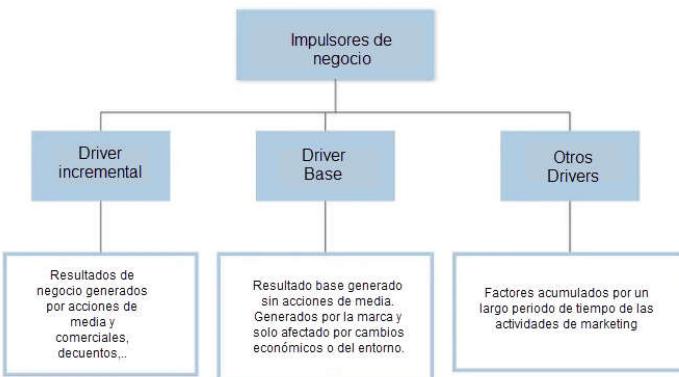
Fuente: a+i

Ilustración 7 - interacción variables de marketing

Como observamos en este cuadro las interacciones de las variables de marketing a tener en cuenta para el desarrollo de un modelo de marketing mix.

4.3. Modelización del caso

El Modelado del Mix de Marketing se emplea para desglosar las métricas de negocio con el objetivo de identificar y cuantificar las contribuciones de diversas fuentes. Estas fuentes se dividen en tres categorías principales: las actividades de marketing y promoción que generan un crecimiento incremental, los factores básicos o inherentes del negocio, y otras variables externas que pueden influir en el rendimiento del negocio. Este proceso de descomposición permite entender mejor cómo cada componente individual impacta en sus resultados globales.



Fuente: elaboración propia

Ilustración 8 - impulsores del negocio

4.3.1. Datos Requeridos

En las fuentes identificadas se deben obtener los datos necesarios para definir las variables del modelo. Tendremos que definir el tipo de granularidad de los datos, que se refiere al nivel de detalle considerado en un modelo, cuanto mayor sea la granularidad, mayor será el nivel de conocimientos detallados que podremos obtener.

En el contexto de las estrategias de marketing, la granularidad de los datos puede resultar extremadamente valiosa. Permite examinar en profundidad los detalles de cada canal de marketing, evaluando su eficacia, eficiencia y retorno de la inversión (ROI) general.

Por ejemplo, en lugar de agrupar todos los esfuerzos de marketing digital bajo un solo paraguas, la granularidad de los datos nos permite analizar el rendimiento de cada plataforma digital individualmente, como Facebook, Instagram, Google Ads, etc. Esto proporciona una visión más precisa de cómo cada canal contribuye a los objetivos y permite tomar decisiones más informadas sobre dónde invertir recursos.

Además, una granularidad de datos mayor permite identificar patrones y tendencias específicos que pueden pasar desapercibidos en un análisis menos detallado. Esto puede incluir todo, desde el rendimiento de anuncios individuales hasta las variaciones en la actividad del consumidor durante diferentes períodos o eventos estacionales.

Los factores que afectan en el modelado del MMM son los siguientes:

Categoría	Factores
Factores Base	Precio Distribución
Trade Marketing	Reducción temporal precios Display Skus Features
Promociones	Promo Packs Lanzamientos Combos Sampling
Media	Tv Digital Media Radio Revistas Periódicos Print
Estacionalidad	Temperatura Precipitaciones Vacaciones
Factores Macro económicos	PIB Renta Per capita Inflación Precio Gasolina

Fuente: Elaboración propia

Ilustración 9 - Fuentes de datos

Para construir un modelo de MMM sólido y preciso, necesitamos distintos tipos de datos. A continuación, se detallan los mismos.

- **Datos del producto** se refieren a la información del producto y del subproducto, incluyendo precio y cantidad de unidades vendidas / no vendidas. Estos son necesarios para comprender diversos aspectos relacionados con el producto - ¿Es un producto nuevo? ¿En qué categoría de productos se clasifica? ¿Es el producto más vendido? ¿Cuál es la tasa de crecimiento de las ventas de este producto? ¿Cuál es el precio de cada producto en la categoría?
- **Datos Promocionales:** incluyen detalles sobre los días en que las promociones u ofertas estuvieron activas y el tipo de oferta, como envío gratuito, reembolsos, etc.
- **Datos de Media:** se utilizan para medir la eficacia de una publicidad. En la publicidad televisiva, los GRPs, que es un método que mide la exposición total de la audiencia a la publicidad. También se pueden incluir otros métodos tradicionales como periódicos, revistas y radio. Y la publicidad digital como buscadores, display en banners, entre otros que se miden en impresiones.
- **Estacionalidad:** El incremento en las ventas debido a la estacionalidad puede sesgar el cálculo para identificar los principales impulsores de las ventas. No podríamos evaluar el impacto de la promoción en las ventas. También necesitamos incorporar festivos y eventos importantes programados en un mes específico. Por ejemplo, las ventas durante la temporada navideña suelen ser superiores a las ventas promedio.
- **Demografía:** factores como la edad, la raza y el sexo, se refieren a información socioeconómica expresada estadísticamente, que también incluye empleo, educación, ingresos, tasas de matrimonio, tasas de natalidad y mortalidad, y más factores.
- **Datos macroeconómicos:** Las ventas también pueden verse afectadas por factores macroeconómicos como la inflación, la tasa de desempleo, el PIB, etc. Las empresas generalmente reportan una tasa de crecimiento negativa en ventas durante la recesión.



Necesitamos incorporar estos factores en nuestro modelo para que entienda los efectos de recesión y cíclicos.

- **Ventas:** No es posible construir un modelo de Mezcla de Mercadotecnia sin la variable de ventas. Las ventas pueden ser en volumen en unidades, así como en ingresos

4.3.2. Fuentes de datos externas (Variables de Control)

Las variables de control son elementos esenciales en cualquier modelo de análisis, ya que permiten aislar el efecto de las variables de interés al ajustar por factores que podrían influir en los resultados pero que no son el foco principal del estudio. En el contexto de un modelo de marketing mix, las variables de control podrían incluir factores como las condiciones económicas generales, las tendencias de la industria y las actividades de los competidores.

Los datos externos pueden proporcionar una valiosa información para estas variables de control. Por ejemplo, los datos económicos del gobierno pueden usarse para controlar el impacto de las condiciones económicas generales en las ventas. Las estadísticas de la industria pueden proporcionar información sobre las tendencias generales que podrían estar influyendo en el comportamiento de los consumidores. Y los datos de los competidores, como su gasto en publicidad o sus precios, pueden ayudar a aislar el impacto de las propias actividades de marketing de una empresa.

Estos serían unos ejemplos de datos externos:

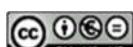
Tráfico web: Utilizando SEMrush, plataforma de análisis de marketing digital que proporciona una amplia gama de herramientas y datos para ayudar a las empresas a mejorar su presencia en línea. Con SEMrush, los usuarios pueden realizar análisis de la competencia, investigar palabras clave, realizar auditorías de sitios web, realizar seguimiento de posiciones en los motores de búsqueda y mucho más

Búsquedas web: Los datos indexados de búsquedas en Google ofrecen información sobre las tendencias y los intereses de los usuarios. Estos datos pueden ser desglosados por palabras clave, región geográfica y rango de fechas seleccionado, lo que permite identificar patrones y oportunidades de marketing.

Notoriedad publicitaria: Admo.tv es una herramienta de análisis de publicidad televisiva y digital que permite a las empresas medir y optimizar el rendimiento de sus campañas publicitarias. Proporciona datos precisos sobre la visibilidad y el impacto de los anuncios en la televisión y en plataformas digitales.

Agregadores financieros: Aplicaciones como Personal Capital, Yolt, Mint, proporcionan información valiosa sobre productos financieros, condiciones y tarifas de diferentes empresas del sector. Estos datos son una fuente invaluable para comprender las estrategias de precios y productos de la competencia.

Indicadores macroeconómicos: Fuentes como el Instituto Nacional de Estadística (INE) y Eurostat proporcionan datos macroeconómicos clave, como la tasa de desempleo, el índice de precios al consumidor (IPC), el Producto Interno Bruto (PIB) y otros indicadores de sentimiento económico. Estos datos pueden ayudar a contextualizar y comprender el entorno económico en el que operan las empresas.



Datos de movilidad: Google ofrece datos de movilidad que muestran los patrones de movilidad de las personas, como los viajes hacia el trabajo, centros comerciales, parques, entre otros. Estos datos pueden ser útiles para comprender los comportamientos de desplazamiento de los consumidores y adaptar las estrategias de marketing en consecuencia.

La incorporación de estas fuentes externas de datos en el MMM permitirá una evaluación más precisa y completa de las estrategias de marketing, brindando una visión más holística y actualizada del mercado y sus tendencias. Esto ayudará a las empresas a tomar decisiones más informadas y maximizar el impacto de sus inversiones en publicidad.

4.3.3. Selección de datos y variables del proyecto

Para la construcción de los modelos de MMM se ha seleccionado un data set que represente lo más parecido a un caso real, con un histórico de cuatro años (209 semanas) de ventas, impresiones de medios y gastos de medios a nivel semanal.

1. Variables de medios

- Impresión de medios (prefijo = 'mdip_'): impresiones de 13 canales de medios: prensa, audio digital, radio, televisión, video digital, redes sociales, visualización online, correo electrónico, SMS, afiliados, SEM.
- Gasto en medios (prefix='mdsp_'): gasto de los canales de medios.

2. Variables de Control

- Macroeconomía (prefijo='me_'): IPC, precio del gas.
- Rebaja (prefijo='mrkdn_'): rebaja/descuento.
- Recuento de tiendas ('st_ct')
- Días festivos minoristas (prefijo = 'hldy_'): codificación one-hot.
- Estacionalidad (prefix='seas_'): mes, con noviembre y diciembre divididos en semanas.

3. Variable de Ventas

Será nuestra variable objetivo del modelo, en el dataset se denomina 'sales'.

Se adjunta un diccionario de las variables, con la descripción y tipología de cada una de ellas.(Ver en anexo GitHub)

4.4. Análisis exploratorio de datos

El análisis exploratorio de datos (EDA), mediante técnicas permite comprender mejor el conjunto de datos que se utiliza, siendo un factor fundamental para una alta calidad en los datos.

4. Características de las variables y los datos:

- **Dimensionalidad de los datos:** 209 filas por 80 columnas
- **Análisis de datos duplicados:** No hay datos duplicados

- **Análisis de Valores nulos o perdidos:** La no disponibilidad de datos para una determinada observación o cálculo en una variable. Suele deberse a la no ocurrencia de eventos, a la no disponibilidad durante todo el tiempo de análisis, a que la gente no menciona la entrada de datos o a que faltan de forma aleatoria. En nuestro data set no se han detectado valores nulos.
- **Selección y análisis variable objetivo:** En nuestro caso serán las ventas y verificaremos que se trata de una serie temporal:

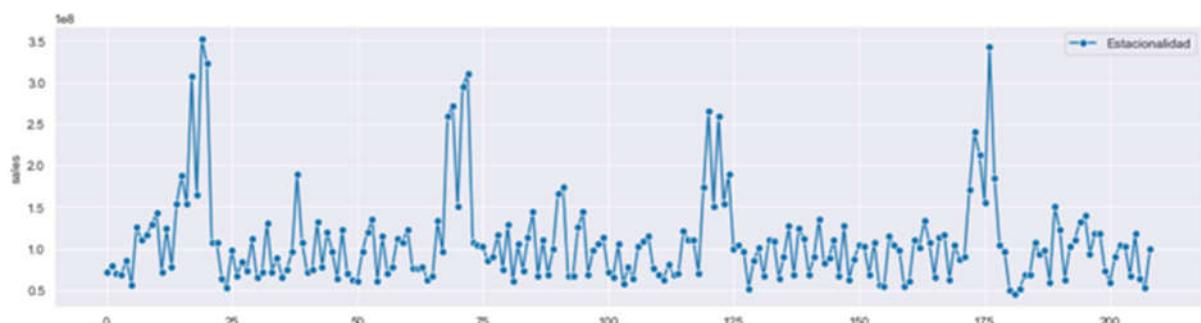


Ilustración 10 - gráfico de las ventas como serie temporal

- **Análisis de valores extremos:**

Vemos la existencia de valores extremos para entender si son atípicos o se deben mantener. Hay que considerar que un valor atípico puede deberse a la estacionalidad, una campaña, una promoción, descuentos, etc.

Para su análisis se ha desarrollado una función (ver Notebook en anexos GITHUB), obteniendo:

- mdip_dm: 2 outliers (0.96%)
- mdip_inst: 6 outliers (2.87%)
- mdip_nsp: 10 outliers (4.78%)
- mdip_auddig: 9 outliers (4.31%)
- mdip_audtr: 8 outliers (3.83%)
- mdip_vidtr: 17 outliers (8.13%)
- mdip_viddig: 15 outliers (7.18%)
- mdip_so: 2 outliers (0.96%)
- mdip_on: 9 outliers (4.31%)
- mdip_em: 7 outliers (3.35%)
- mdip_sms: 5 outliers (2.39%)
- mdip_aff: 10 outliers (4.78%)
- mdip_sem: 9 outliers (4.31%)
- mdsp_dm: 9 outliers (4.31%)
- mdsp_inst: 9 outliers (4.31%)
- mdsp_nsp: 8 outliers (3.83%)
- mdsp_auddig: 13 outliers (6.22%)
- mdsp_audtr: 5 outliers (2.39%)
- mdsp_vidtr: 14 outliers (6.7%)
- mdsp_viddig: 10 outliers (4.78%)
- mdsp_so: 8 outliers (3.83%)
- mdsp_on: 5 outliers (2.39%)
- mdsp_sem: 16 outliers (7.66%)

Y se complementa con la visualización a través de gráficos Box Plots:

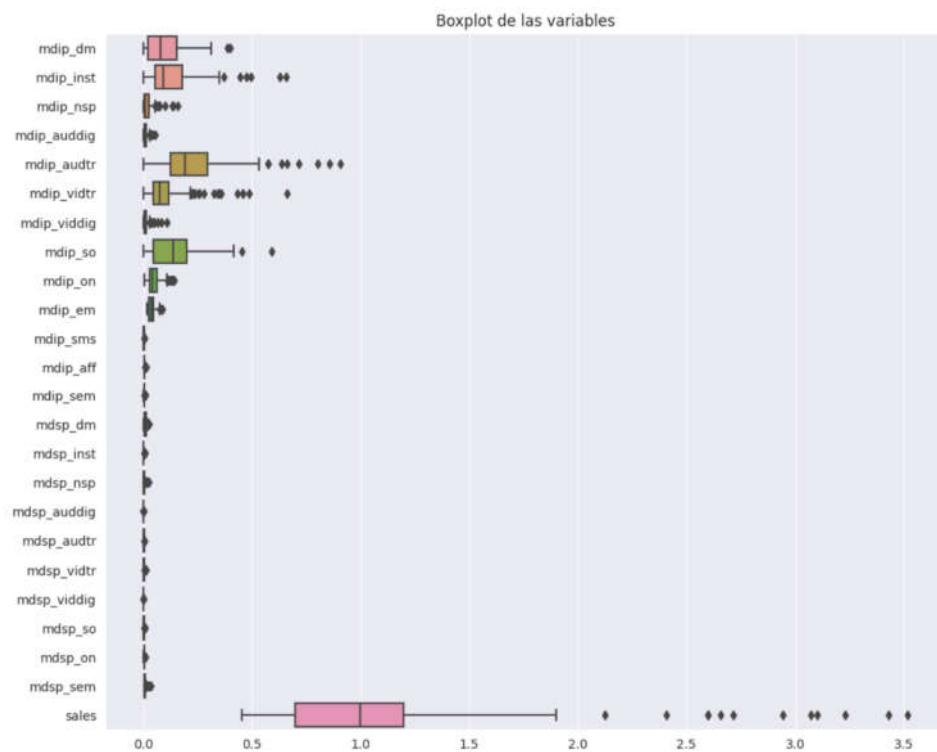


Ilustración 11 - Box plots de las variables

Según la variabilidad en los datos, nos indica que muchas de las variables presentan outliers, observaciones atípicas o extremas en esos casos. Por ejemplo, las variables mdip_vidtr, mdip_viddig, mdsp_sem, mdsp_vidtr tienen una cantidad considerable de outliers, representando entre el 6% y el 8% de las observaciones.

Pero como la variable objetivo es una serie temporal, es posible que estos valores sean por este motivo, lo compruebo con un análisis gráfico (anexo en notebook EDA están todas las gráficas), así en el caso de las tres variables con más outliers:

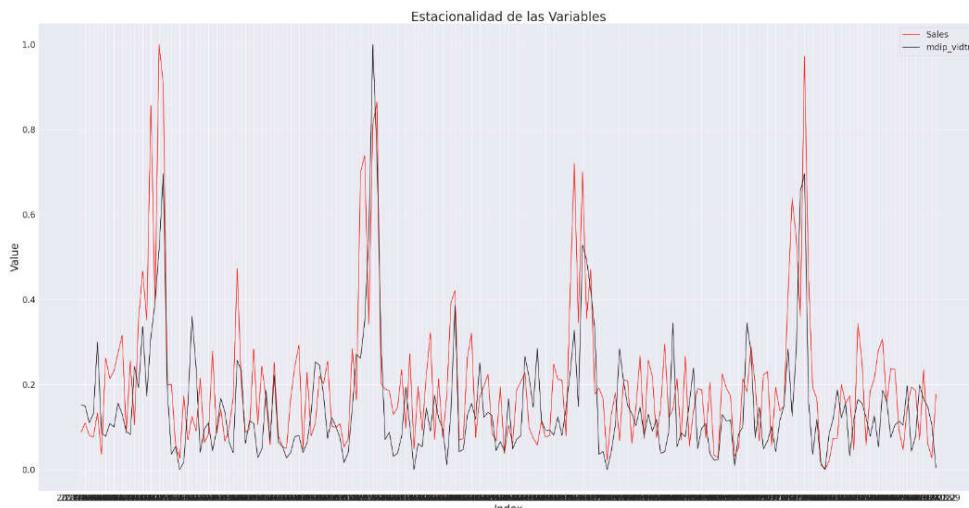


Ilustración 12 - Curvas de las variables y las ventas

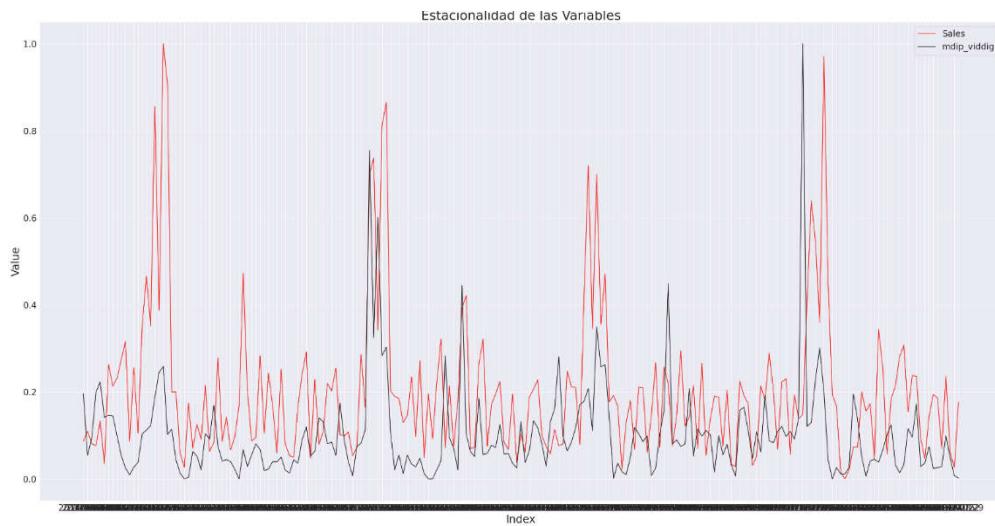


Ilustración 13 - Curvas de las variables y las ventas

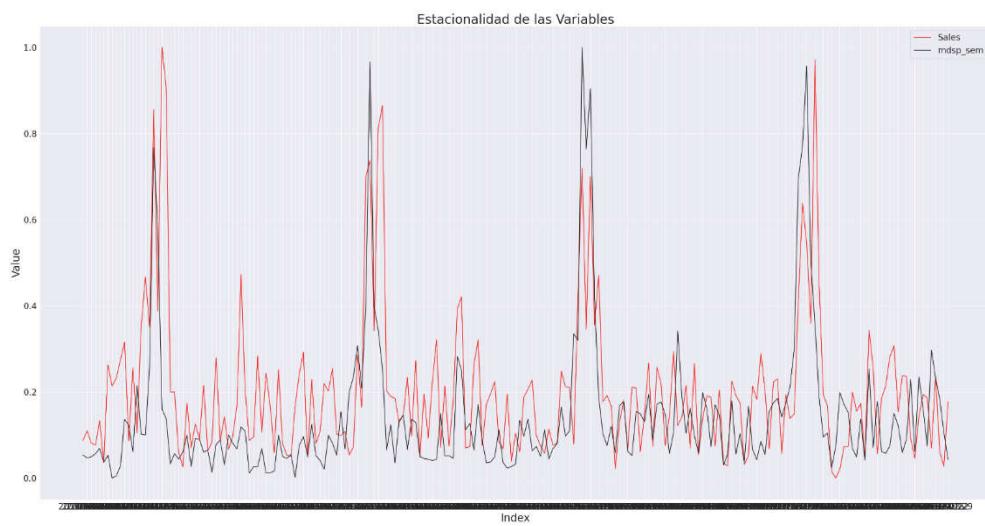


Ilustración 14 - Curvas de las variables y las ventas

Nos indican que en todos los casos los valores atípicos son debidos a la estacionalidad de las variables, por los que los mantenemos en el data set.

- **Resumen estadístico de las variables numéricas:**

```

      mdip_dm      mdip_inst      mdip_nsp      mdip_auddig      mdip_audtr
count  2.090000e+02  2.090000e+02  2.090000e+02  2.090000e+02  2.090000e+02
mean   9.544510e+06  1.247717e+07  1.616957e+06  1.002816e+06  2.295103e+07
std    8.293082e+06  1.024959e+07  2.203341e+06  8.122848e+05  1.567124e+07
min    0.000000e+00  4.853300e+04  0.000000e+00  1.561800e+04  0.000000e+00
25%   2.087021e+06  5.304240e+06  2.542340e+05  4.577220e+05  1.236705e+07
50%   7.664954e+06  8.911466e+06  8.870720e+05  8.061170e+05  1.910160e+07
75%   1.533852e+07  1.786920e+07  2.248483e+06  1.344765e+06  2.956004e+07
max   3.979871e+07  6.545146e+07  1.553181e+07  5.418819e+06  9.066538e+07

      mdip_vidtr      mdip_viddig      mdip_so      mdip_on      mdip_em
count  2.090000e+02  2.090000e+02  2.090000e+02  2.090000e+02  2.090000e+02
mean   9.846007e+06  1.096360e+06  1.382028e+07  4.742089e+06  3.585528e+06
std    9.254274e+06  1.247631e+06  1.091767e+07  2.669165e+06  1.467660e+06
min    0.000000e+00  0.000000e+00  0.000000e+00  5.028230e+05  1.446695e+06
25%   4.378348e+06  3.931700e+05  4.305911e+06  2.893497e+06  2.416255e+06
50%   7.562079e+06  8.357480e+05  1.381069e+07  4.100627e+06  3.452070e+06
75%   1.139192e+07  1.298726e+06  1.974306e+07  6.108048e+06  4.431058e+06
max   6.582956e+07  1.078819e+07  5.882166e+07  1.394427e+07  8.614296e+06

      ...      mdsp_inst      mdsp_nsp      mdsp_auddig      mdsp_audtr \
count ...  209.000000  2.090000e+02  209.000000  209.000000
mean ...  79474.858947  2.545628e+05  3844.330287  122606.298373
std ...  69706.379634  3.415438e+05  2574.853370  66082.688970
min ...  1138.730000  0.000000e+00  1.620000  0.000000
25% ...  36827.560000  5.102103e+04  2014.370000  77073.280000
50% ...  61208.050000  1.286947e+05  3237.400000  116023.470000
75% ...  99921.330000  3.619149e+05  4891.120000  158288.110000
max ...  590148.110000  2.198467e+06  13064.830000  435614.540000

      mdsp_vidtr      mdsp_viddig      mdsp_so      mdsp_on \
count  2.090000e+02  209.000000  209.000000  209.000000
mean   1.681586e+05  18495.923349  102010.544498  215863.998038
std    1.685799e+05  17093.178098  95765.738144  124662.993635
min    0.000000e+00  0.000000e+00  0.000000e+00  40324.230000
25%   6.029077e+04  6840.090000  35081.440000  125161.210000
50%   1.123532e+05  12991.090000  70799.440000  186763.140000
75%   2.185535e+05  24837.910000  143463.910000  281687.620000
max   1.100083e+06  104352.440000  573355.550000  695750.180000

      mdsp_sem      sales
count  2.090000e+02  2.090000e+02
mean   6.261338e+05  1.080543e+08
std    4.981457e+05  5.461240e+07
min    1.997579e+05  4.531651e+07
25%   3.559545e+05  6.970993e+07
50%   4.855040e+05  9.955665e+07
75%   6.920217e+05  1.194684e+08
max   3.134565e+06  3.515980e+08

```

Las principales conclusiones que nos indican este análisis es:

- Las variables mdip_inst, mdip_audtr, y mdip_on tienen valores máximos significativamente altos en comparación con otros variables. Esto indica que hay observaciones con valores extremadamente altos en esas variables.
- La variable mdip_dm tiene un valor mínimo de 0, lo que sugiere que existen observaciones con cero en esa variable. Es importante considerar la interpretación de esos valores cero y su impacto en el análisis.
- Las variables mdip_auddig, mdip_vidtr, y mdip_viddig tienen una amplia variabilidad, como se refleja en sus desviaciones estándar relativamente altas.



d) La variable sales tiene un promedio de aproximadamente 108 millones y un rango que va desde aproximadamente 45 millones hasta 351.6 millones. Esto indica que las ventas varían significativamente en el conjunto de datos.

e) Las variables mdsp_nsp, mdsp_audtr, y mdsp_on tienen valores máximos más altos que otras variables en el conjunto de datos.

- **Análisis de las distribuciones para las variables numéricas:** realizamos histogramas para visualizarlo:

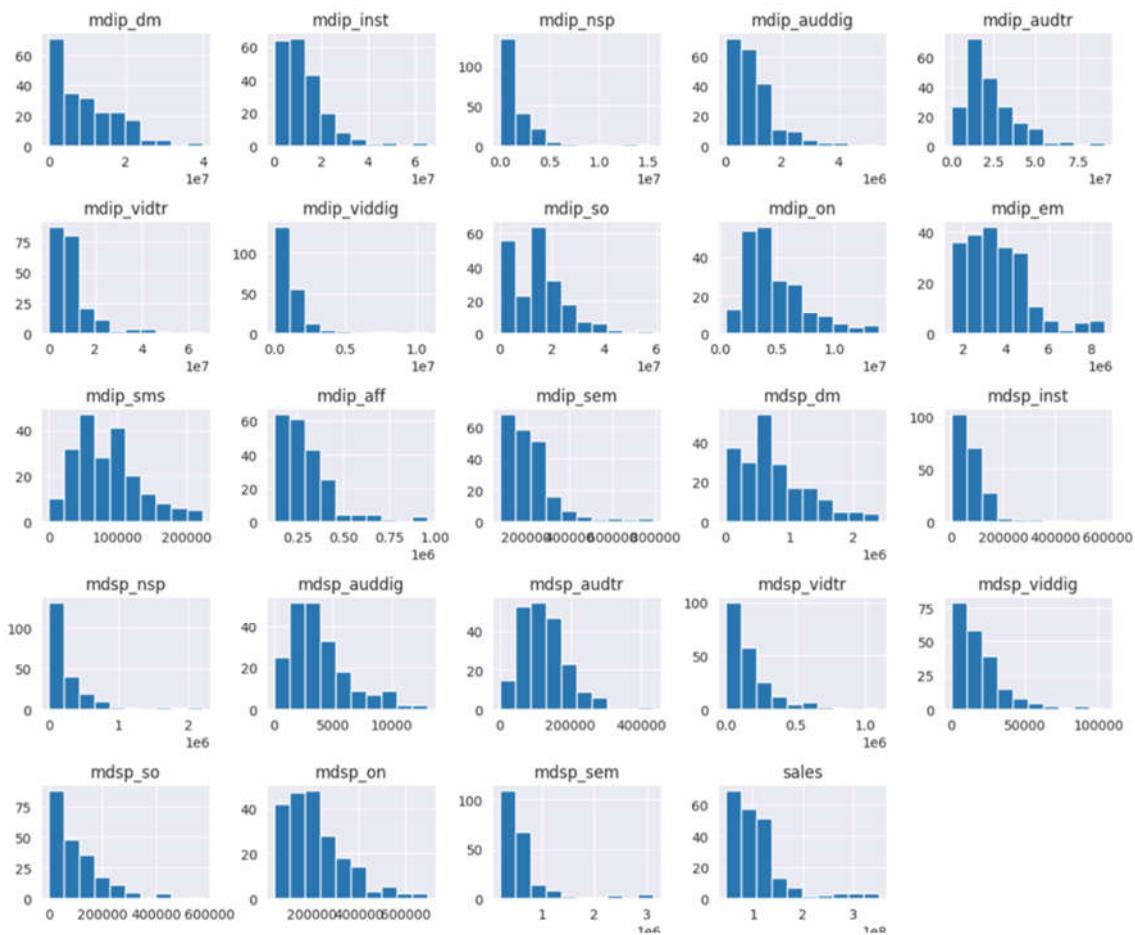


Ilustración 15 - gráfica con distribuciones para las variables numéricas

- **Análisis de la correlación de las variables:**

Para este análisis usaremos la Matriz de correlación que nos permite comprender las relaciones existentes entre las variables en nuestro conjunto de datos. La matriz de correlación nos permite cuantificar la fuerza y la dirección de la asociación lineal entre pares de variables.

El objetivo principal al obtener la matriz de correlación es identificar posibles patrones o dependencias entre las variables. Al examinar los coeficientes de correlación, podemos obtener una idea de cómo se mueven las variables juntas. Si dos variables están altamente correlacionadas, significa que hay una relación lineal fuerte entre ellas. Por otro lado, una correlación débil o cercana a cero indica una relación más débil o prácticamente inexistente.

Matriz de correlación																								
mdip_dm	1	0.22	0.29	-0.11	0.13	0.25	0.14	-0.041	0.25	0.1	-0.04	0.15	0.096	0.55	0.17	0.3	0.042	0.065	0.38	0.11	-0.023	0.14	0.12	0.24
mdip_inst	0.22	1	0.5	0.12	0.47	0.56	0.31	-0.091	0.49	0.074	-0.066	0.26	0.21	0.17	0.84	0.48	0.32	0.41	0.64	0.29	0.089	0.27	0.27	0.47
mdip_nsp	0.29	0.5	1	-0.12	0.24	0.4	0.17	-0.41	0.31	-0.2	-0.19	0.15	0.026	0.27	0.49	0.89	0.07	0.33	0.51	0.0046	-0.3	0.031	0.11	0.38
mdip_auddig	-0.11	0.12	-0.12	1	0.45	0.38	0.24	0.3	0.22	0.36	0.12	0.22	0.23	-0.14	0.13	-0.1	0.77	0.36	0.3	0.37	0.45	0.45	0.26	0.27
mdip_audtr	0.13	0.47	0.24	0.45	1	0.62	0.34	0.22	0.39	0.35	0.1	0.32	0.35	0.088	0.42	0.31	0.64	0.6	0.58	0.42	0.37	0.43	0.42	0.47
mdip_vidtr	0.25	0.56	0.4	0.38	0.62	1	0.46	0.15	0.52	0.34	0.023	0.35	0.38	0.19	0.55	0.53	0.54	0.52	0.86	0.52	0.27	0.51	0.44	0.68
mdip_viddig	0.14	0.31	0.17	0.24	0.34	0.46	1	0.28	0.4	0.23	0.053	0.39	0.39	0.11	0.31	0.23	0.3	0.28	0.52	0.71	0.36	0.38	0.41	0.39
mdip_so	-0.041	-0.091	-0.41	0.3	0.22	0.15	0.28	1	0.16	0.45	0.21	0.26	0.42	-0.11	-0.082	-0.33	0.22	-0.038	0.09	0.42	0.67	0.38	0.36	0.12
mdip_on	0.25	0.49	0.31	0.22	0.39	0.52	0.4	0.16	1	0.28	0.0083	0.39	0.45	0.15	0.5	0.32	0.32	0.32	0.62	0.45	0.39	0.63	0.53	0.56
mdip_em	0.1	0.074	-0.2	0.36	0.35	0.34	0.23	0.45	0.28	1	0.23	0.33	0.43	0.0061	0.071	-0.13	0.38	0.19	0.31	0.41	0.47	0.44	0.46	0.29
mdip_sms	-0.04	-0.066	-0.19	0.12	0.1	0.023	0.053	0.21	0.0083	0.23	1	0.1	0.18	-0.11	-0.083	-0.14	0.13	-0.031	-0.0078	0.0096	0.21	-0.0092	0.077	-0.0035
mdip_aff	0.15	0.26	0.15	0.22	0.32	0.35	0.39	0.26	0.39	0.33	0.1	1	0.67	0.12	0.28	0.16	0.3	0.12	0.43	0.42	0.46	0.41	0.64	0.46
mdip_sem	0.096	0.21	0.026	0.23	0.35	0.38	0.39	0.42	0.45	0.43	0.18	1	0.67	0.19	0.26	0.065	0.24	0.11	0.41	0.51	0.55	0.49	0.79	0.59
mdsp_dm	0.55	0.17	0.27	-0.14	0.088	0.19	0.11	-0.11	0.15	-0.0061	-0.11	0.12	0.019	1	0.18	0.29	0.015	0.11	0.29	0.081	-0.063	0.13	0.049	0.15
mdsp_inst	0.17	0.84	0.49	0.13	0.42	0.55	0.31	-0.082	0.5	0.071	-0.083	0.28	0.26	0.28	1	0.53	0.32	0.41	0.58	0.25	0.14	0.27	0.27	0.52
mdsp_nsp	0.3	0.48	0.89	-0.1	0.31	0.53	0.23	-0.33	0.32	-0.13	-0.14	0.16	0.065	0.29	0.53	1	0.15	0.37	0.56	0.046	-0.19	0.049	0.14	0.42
mdsp_auddig	0.042	0.32	0.07	0.77	0.64	0.54	0.3	0.22	0.32	0.38	0.13	0.3	0.24	0.015	0.32	0.15	1	0.49	0.5	0.35	0.42	0.47	0.3	0.37
mdsp_audtr	0.065	0.41	0.33	0.36	0.6	0.52	0.28	-0.038	0.32	0.19	-0.031	0.12	0.11	0.11	0.41	0.37	0.49	1	0.51	0.25	0.11	0.26	0.14	0.28
mdsp_vidtr	0.38	0.64	0.51	0.3	0.58	0.86	0.52	0.09	0.62	0.31	-0.0078	0.43	0.41	0.29	0.58	0.56	0.5	0.51	1	0.55	0.24	0.55	0.48	0.66
mdsp_viddig	0.11	0.29	0.0046	0.37	0.42	0.52	0.71	0.42	0.45	0.41	0.0096	0.42	0.51	0.081	0.25	0.046	0.35	0.25	0.55	1	0.47	0.56	0.56	0.46
mdsp_so	-0.023	0.089	-0.3	0.45	0.37	0.27	0.36	0.67	0.39	0.47	0.21	0.46	0.55	-0.063	0.14	-0.19	0.42	0.11	0.24	0.47	1	0.58	0.55	0.33
mdsp_on	0.14	0.27	0.031	0.45	0.43	0.51	0.38	0.38	0.63	0.44	-0.0092	0.41	0.49	0.13	0.27	0.049	0.47	0.26	0.55	0.56	0.58	1	0.56	0.48
mdsp_sem	0.12	0.27	0.11	0.26	0.42	0.44	0.41	0.36	0.53	0.46	0.077	0.64	0.79	0.049	0.27	0.14	0.3	0.14	0.48	0.56	0.55	0.56	1	0.59
sales	0.24	0.47	0.38	0.27	0.47	0.68	0.39	0.12	0.56	0.29	-0.0035	0.46	0.59	0.15	0.52	0.42	0.37	0.28	0.66	0.46	0.33	0.48	0.59	1
mdip_dm																								
mdip_inst																								
mdip_nsp																								
mdip_auddig																								
mdip_audtr																								
mdip_vidtr																								
mdip_viddig																								
mdip_so																								
mdip_on																								
mdip_em																								
mdip_sms																								
mdip_aff																								
mdip_sem																								
mdsp_dm																								
mdsp_inst																								
mdsp_nsp																								
mdsp_auddig																								
mdsp_audtr																								
mdsp_vidtr																								
mdsp_viddig																								
mdsp_so																								
mdsp_on																								
mdsp_sem																								
sales																								

Ilustración 16 - matriz de correlación

Además, la matriz de correlación nos ayuda a identificar variables que están altamente correlacionadas entre sí, lo que puede indicar la presencia de multicolinealidad.

La multicolinealidad puede ser problemática en algunos modelos de análisis, ya que puede afectar la interpretación de los coeficientes y la estabilidad de los modelos.

- ## 4.5. Efectos de aplicación en el Modelado de un MMM
- ### 4.5.1. Efecto “·Ad Stock”
- Este concepto es especialmente relevante en la medición de la efectividad de la publicidad, se refiere a los efectos que la publicidad en las ventas tienen debido a que las campañas de comunicación tienen un efecto de recuerdo de la marca que persiste más allá del período en el que se realizó la publicidad. En otras palabras, la publicidad tiene un efecto acumulativo o "en stock" que se desvanece con el tiempo.
- La teoría del ad stock sugiere que el impacto de la publicidad en el comportamiento del consumidor no se da de inmediato, sino que se acumula y se desvanece con el tiempo. Esto se debe a varios factores, como la retención de la información por parte del consumidor y la repetición de los mensajes publicitarios.
- Matemáticamente, el ad stock a menudo se modela utilizando una función de decaimiento, que captura cómo el impacto de la publicidad se reduce con el tiempo. Una función de decaimiento comúnmente utilizada es la siguiente:



$$w_{t-l} = D^{(l-p)^2} \text{ for each } l \text{ in } [0, L),$$

$$x_t^* = Adstock(x_t, \dots, x_{t-(L-1)}; L, P, D) = \frac{\sum_{l=0}^{L-1} w_{t-l} \cdot x_{t-l}}{\sum_{l=0}^{L-1} w_{t-l}}$$

Donde:

L: duración del efecto de los medios de comunicación

P: pico/retardo del efecto de los medios, cuántas semanas se retrasa con respecto a la primera exposición

D: tasa de decaimiento/retención del canal mediático, concentración del efecto

El efecto mediático de las semanas actuales es una media ponderada de la semana actual y las semanas anteriores (L- 1).

La elección de L puede depender de varios factores, como el tipo de publicidad y la frecuencia con que los consumidores están expuestos a ella. En general, cuanto más tiempo se espera que dure el impacto de la publicidad, más cercano a 1 será L.

Este efecto ad Stock se calculará a continuación, en alguno de los modelos que se han seleccionado para evaluar los modelos de MMM de este proyecto.

4.6. Efecto “Diminishing Return”

El concepto de "Diminishing Returns" o de rendimientos decrecientes es un principio económico clave, que también se aplica en el marketing. En el contexto del marketing, los rendimientos decrecientes se refieren a la idea de que, después de cierto punto, cada unidad adicional de inversión en una actividad de marketing (como la publicidad, las promociones, etc.) produce un rendimiento menor que la unidad anterior.

Este fenómeno ocurre porque, inicialmente, al aumentar la inversión en marketing, se pueden alcanzar nuevos segmentos de mercado o mejorar la percepción de la marca entre los consumidores existentes. Sin embargo, después de cierto punto, la mayoría de los consumidores ya están alcanzados o informados, y las inversiones adicionales en marketing pueden tener menos impacto.

Por ejemplo, si una empresa lanza una nueva campaña publicitaria, es probable que vea un aumento significativo en las ventas al principio. Sin embargo, a medida que la campaña continúa, cada dinero adicional gastado en publicidad puede generar menos ventas adicionales que el dinero anterior.

La Ley de Rendimientos Decrecientes puede modelarse matemáticamente en diversas formas, aunque a menudo se utiliza una función de producción Cobb-Douglas o una función de respuesta logarítmica. En términos de modelado de marketing mix, esta ley puede incorporarse asumiendo que la elasticidad de la publicidad (el porcentaje de cambio en las ventas debido a un porcentaje de cambio en la publicidad) disminuye a medida que aumenta la inversión en publicidad.

Para su medición se puede utilizar la función de Hill que permite describir cómo la respuesta a la publicidad o a otra actividad de marketing cambia con el nivel de esa actividad (Yao, S., & Mela, C. F. (2011)).

La forma básica de la función de Hill es:

$$Hill(x; K, S) = \frac{1}{1 + (x/K)^{-S}}$$

K: punto de semi saturación

S: pendiente

Esta función genera una curva en forma de S, que comienza en cero, aumenta rápidamente con pequeños incrementos en X y luego se aplana a medida que X se hace grande, lo que representa los rendimientos decrecientes.

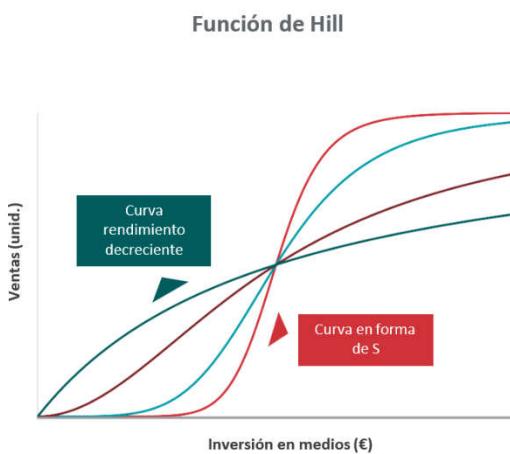


Ilustración 17 - función Hill

Este efecto se puede obtener a través del modelo bayesiano con efecto carry - over y no lineal:

$$y_t = \tau + \sigma_{m=1}^M \beta_m Hill(x_{t,m}^* ; K_m, S_m) + \sigma_{c=1}^C \gamma_c Z_{t,c} + \varepsilon_t,$$

Carry - over

$x_{t,m}^*$ es el efecto carryover sobre variables de medios

τ es el nivel base

γ_c son el resto de variables del modelo

ε_t es el error del modelo

Efecto no lineal

$$Hill(x_{t,m} ; K_m, S_m) = \frac{1}{1 + \left(\frac{x_{t,m}}{K_m}\right)^{-S_m}}, x_{t,m} \geq 0$$

$S_m > 0$ es la pendiente de la curva

$K_m > 0$ es la mitad del punto de saturación

Es importante tener en cuenta que, aunque la función de Hill puede ser útil para modelar los rendimientos decrecientes, puede que no sea apropiada para todas las situaciones. (Sahni, N. S. (2016)). Por ejemplo, puede no capturar bien las respuestas que no siguen una forma de S, o puede ser demasiado compleja para los datos disponibles.

Este efecto de rendimientos decrecientes o “Diminishing Return”, también se calculará a continuación, en alguno de los modelos que se han seleccionado para evaluar los modelos de MMM de este proyecto.

4.7. Modelos Tradicionales de Marketing Mix:

4.7.1. Modelo de Regresión Lineal Multivariante

La regresión es una técnica estadística utilizada para examinar la relación entre dos o más variables. Los modelos de regresión permiten hacer predicciones a partir de datos y, lo que es más importante, inferir relaciones causales.

En estos modelos los datos se dividen en dos categorías: variables dependientes (VD) y variables independientes (VDI). El análisis de cómo las variables independientes pueden influir en el resultado de las variables dependientes es el objetivo de la regresión. (Leeflang, P. S. H., Wittink, D. R., Wedel, M., & Naert, P. A. (2000)).

Un modelo de regresión simple puede tener la forma:

$$Y = a + bX + e$$

donde:

Y es la variable dependiente (por ejemplo, las ventas).

X es la variable independiente (por ejemplo, el gasto en publicidad).

a es la intersección, que representa el valor esperado de Y cuando X es 0.

b es la pendiente, que representa el cambio esperado en Y por cada cambio unitario en X.

e es el error aleatorio. (Gujarati, D. N. (2003)).

En un modelo de regresión múltiple, hay varias variables independientes. Por ejemplo:

$$Y = a + b_1X_1 + b_2X_2 + \dots + b_nX_n + e$$

Aunque el análisis de regresión se utiliza principalmente para el análisis causal, la previsión del impacto de un cambio, la previsión de tendencias, etc, por lo que este tipo de modelo podría usarse para analizar el efecto de varias partes del mix de marketing en la variable ventas.

La regresión puede ser una herramienta poderosa para el análisis del mix de marketing. Pero tiene sus limitaciones. Por ejemplo, puede ser difícil captar relaciones no lineales o interacciones entre variables. Además, la regresión puede inferir correlación, pero no necesariamente causalidad. (Wooldridge, J. M. (2012)). Además, este método no funciona bien con grandes cantidades de datos, ya que es sensible a los valores atípicos, la multicolinealidad y la correlación cruzada.

4.7.1.1. Resultados del Modelo de Regresión Multivariante en el proyecto:

Se ha desarrollado un modelo de regresión multivariante, utilizando el algoritmo LR_model.fit(xtrain, ytrain), técnica de regresión lineal que permite predecir una variable de



respuesta continua a partir de múltiples variables predictoras, en nuestro caso se han tomado todas las variables del marketing mix, tanto las impresiones como las inversiones y las variables de control.

El algoritmo de regresión lineal busca estimar los coeficientes que ponderan cada variable predictora de manera que se minimice el error cuadrático medio (MSE) entre las predicciones del modelo y los valores reales de la variable de respuesta. Esto implica ajustar una línea o un hiperplano en un espacio de mayor dimensión, dependiendo del número de variables predictoras involucradas.

Para evaluar el MMM con la regresión lineal se ha calculado el MAPE (Mean Absolute Percentage Error), que es una medida comúnmente utilizada para evaluar la precisión de los modelos de pronóstico. El MAPE se calcula tomando el valor absoluto del error porcentual para cada pronóstico, y luego calculando la media de estos errores.

El resultado obtenido por nuestro modelo es del 29.28%, significa que, en promedio, los pronósticos del modelo están desviados en aproximadamente un 29.28% del valor real. Esto puede interpretarse como que el modelo, en promedio, está sobreestimando o subestimando los valores reales en un 29.28395%.

En el caso del modelo MMM (Multiplicative Mixed Model), esto puede ser especialmente útil para entender cómo el modelo está realizando los pronósticos en términos porcentuales, en lugar de valores absolutos, lo que puede proporcionar una evaluación más justa del rendimiento del modelo, especialmente cuando estás tratando con series temporales que tienen valores con diferentes magnitudes.

Aunque un MAPE del 29.2% puede parecer alto dependiendo del contexto (en muchos casos, un MAPE inferior al 10% se considera bueno), pero dado que la variable objetivo son las ventas, que dan poca estabilidad motivada por la tendencia y estacionalidad, este valor podría ser aceptable, veremos más adelante cómo se comportan los modelos más complejos

```
[ ] 1 from sklearn.metrics import mean_absolute_percentage_error  
[ ] 1 mean_absolute_percentage_error(ytest, y_pred)  
0.2935266996559805
```

Calculamos los ratios siguientes para ver la calidad del modelo:

MSE (Mean Squared Error): El valor del MSE es extremadamente alto, aproximadamente 1.38e+15. El MSE es una medida de la calidad general del modelo y representa el promedio de los errores al cuadrado entre las predicciones y los valores reales. Un MSE alto indica que las predicciones tienen un error cuadrático promedio significativo en comparación con los valores reales. En este caso, el MSE indica que las predicciones pueden tener un error cuadrático promedio muy alto.

RMSE (Root Mean Squared Error): El valor del RMSE es aproximadamente 3.72e+07. El RMSE es la raíz cuadrada del MSE y tiene las mismas unidades que la variable objetivo. Un RMSE más bajo indica un mejor ajuste del modelo a los datos. En este caso, el valor del RMSE también es alto, lo que sugiere que las predicciones pueden tener un error considerable en comparación con los valores reales.

MAE (Mean Absolute Error): El valor del MAE es aproximadamente 2.97e+07. El MAE es una medida del error absoluto promedio entre las predicciones y los valores reales. Cuanto menor sea el valor del MAE, mejor será el modelo en términos de precisión. En este caso, el valor del MAE indica que las predicciones pueden tener un error absoluto promedio de alrededor de 2.97e+07 en comparación con los valores reales.

R^2 (Coeficiente de determinación): El valor de R^2 es aproximadamente 0.214, lo que indica que alrededor del 21.35% de la variabilidad en los datos puede ser explicada por las variables incluidas en el modelo. El R^2 se utiliza para evaluar la capacidad del modelo para capturar la variabilidad de los datos y proporciona una medida de qué tan bien se ajusta el modelo a los datos. En este caso, el valor de R^2 es bajo, lo que sugiere que el modelo tiene un ajuste deficiente a los datos y no puede explicar una gran parte de la variabilidad en los datos.

En resumen, las métricas indican que el modelo de regresión lineal no se ajusta bien a los datos y tiene un rendimiento insatisfactorio. El MSE, RMSE y MAE son altos, lo que indica que las predicciones tienen un error significativo en comparación con los valores reales. El R^2 es bajo, lo que sugiere que el modelo no puede explicar gran parte de la variabilidad en los datos. Por lo que seguiremos evaluando otras técnicas de modelado para obtener un mejor ajuste y rendimiento.

Otro indicador que se ha obtenido en este modelo es el ROAS es un acrónimo que significa "Return On Advertising Spend" (Retorno Sobre el Gasto Publicitario). Es una métrica que se utiliza en el marketing online para medir la eficacia de la inversión publicitaria. ROAS se calcula dividiendo los ingresos generados por los canales publicitarios entre el coste de los medios.

Los valores obtenidos de los ROAS han sido:

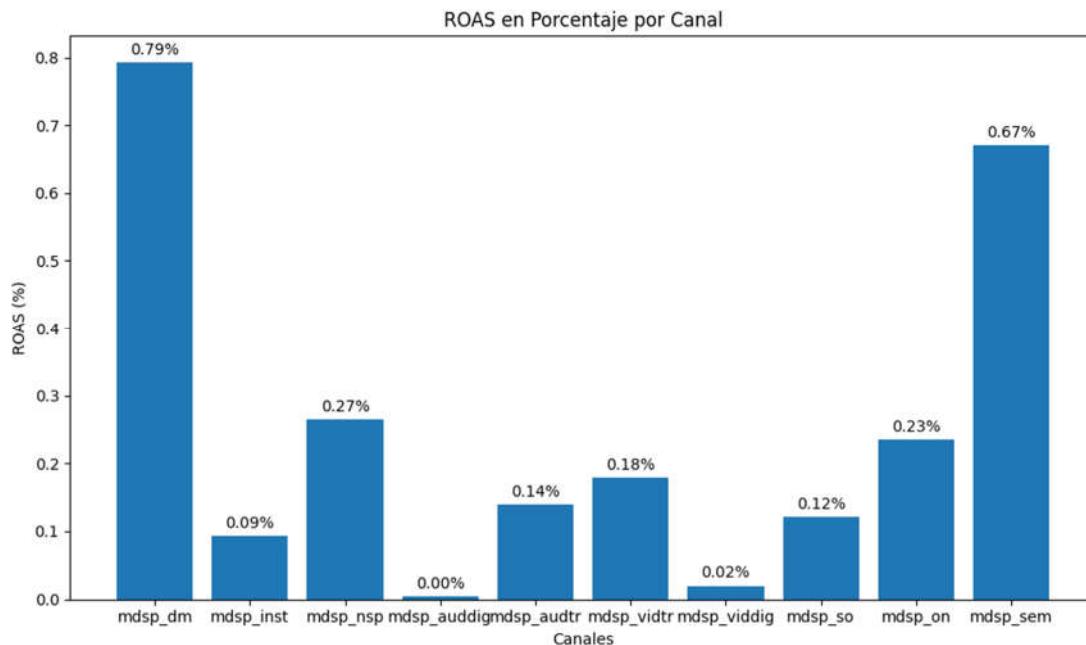


Ilustración 18 - gráfica con Roas por canal

Estos valores representan el rendimiento de la inversión en marketing. En general, los valores de ROAS superiores a 1 indican que la campaña generó más ingresos que la inversión, y son por lo tanto rentables. Los valores por debajo de 1 indican que la campaña generó menos ingresos que el costo, y por lo tanto no fueron rentables. En nuestro caso todos los canales presentan un nivel por debajo de uno, con lo que con este modelo de regresión lineal, se indica que el ROI de las inversiones no es muy adecuado, pero ya veremos que nos indican los otros modelos.

Los resultados del modelo en la inversión por cada medio, comparado con la inversión realizada, ha sido el siguiente:

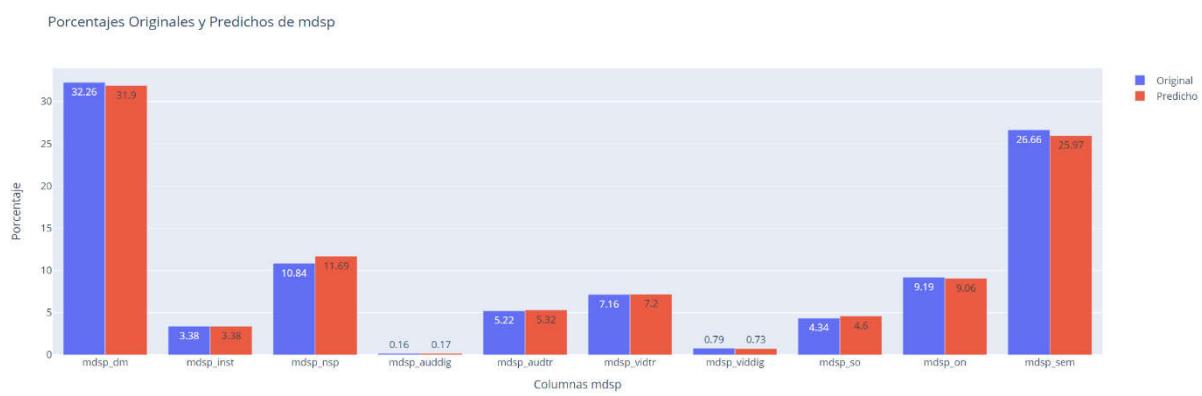


Ilustración 19 - gráfico con resultados del modelo en la inversión por cada medio

Como se puede observar este modelo de regresión lineal presenta un ajuste muy bajo entre lo real y lo predicho, esto indica que las inversiones reales son adecuadas y no nos aporta mucho este tipo de modelos que no consideran otras variables más complejas, como la estacionalidad, los efectos del marketing y variables externas.

Aunque los métodos de regresión son potentes herramientas de modelización, vemos claramente sus limitaciones para un MMM, siendo quizás las causales y las de extrapolación las más importantes. En primer lugar, aun cumpliéndose las hipótesis, una relación de regresión no implica una relación causal. Ya que, la relación de regresión y la variable de respuesta no implica que sea la causa o viceversa. Y además, un modelo ajustado con valores observados no debe utilizarse para hacer inferencias sobre valores fuera del estudio, lo que también se conoce como extrapolación (Freund, Wilson & Regression, 2006). Estos modelos son buenos para mostrar un comportamiento cercano a la linealidad en un subespacio mientras que se alejan de un comportamiento lineal en todo el espacio.

4.7.2. Modelo Agregativo o Aditivo

Los modelos aditivos en el mix de marketing son métodos estadísticos útiles para identificar y cuantificar las diversas formas en que los diferentes componentes del mix de marketing influyen en los resultados de marketing, como las ventas o la participación de mercado.

Los modelos aditivos se llaman así porque presuponen que los efectos de las diferentes variables de entrada (por ejemplo, la publicidad, el precio, etc.) se pueden sumar para obtener una medida del resultado total. En términos matemáticos, si Y es el resultado (por ejemplo, las ventas) y X_1 ,

X₂, ..., X_n son las variables de entrada (los componentes del mix de marketing), un modelo aditivo simple puede tener la forma Y = X₁ + X₂ + ... + X_n. Esto significa que cada variable de entrada contribuye de forma independiente y lineal al resultado.

En el contexto del mix de marketing, esto podría significar que se supone que cada dinero gastado en publicidad añade una cierta cantidad a las ventas, independientemente de lo que se gaste en, por ejemplo, desarrollo de productos o descuentos de precio. De manera similar, cada cambio en el precio se supone que tiene un efecto directo en las ventas, independientemente de las demás variables.

Esto tiene sus ventajas y sus limitaciones. Los modelos aditivos son relativamente sencillos y fáciles de entender, lo que puede facilitar la interpretación de los resultados y la toma de decisiones. Sin embargo, al suponer que las variables de entrada contribuyen de forma independiente al resultado, los modelos aditivos pueden no capturar interacciones complejas entre diferentes componentes del mix de marketing. Por ejemplo, es posible que la publicidad sea más efectiva cuando se combina con una estrategia de precios adecuada, una interacción que un modelo aditivo simple no podría captar.

Si comparamos los modelos aditivos y los modelos de regresión multivariable ambas técnicas estadísticas se pueden utilizar para analizar la relación entre múltiples variables independientes y una variable dependiente. Ambos enfoques permiten examinar cómo varias variables de entrada afectan a un resultado de interés y pueden ayudar a identificar las variables que tienen el mayor impacto en el resultado.

La principal diferencia entre los dos enfoques tiene que ver con la forma en que modelan las relaciones entre las variables.

Un modelo de regresión multivariable lineal, como su nombre indica, asume que las relaciones entre las variables independientes y la variable dependiente son lineales. Esto significa que un cambio en una variable independiente se asocia con un cambio constante en la variable dependiente, sin importar el valor de la variable independiente. Además, se asume que el efecto de cada variable independiente en la variable dependiente es constante, sin importar el valor de las otras variables independientes. (James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013)).

Este modelo no está en el alcance de este proyecto, ya que los modelos aditivos implican un efecto absoluto constante de cada unidad adicional de variables explicativas. Sólo son adecuados si los negocios se desarrollan en entornos más estables y no se ven afectados por la interacción entre variables explicativas.

4.7.3. Modelo Multiplicativo

Los modelos de regresión multiplicativos son una excelente herramienta en el análisis de marketing mix. A diferencia de los modelos de regresión aditiva, que asumen que los efectos de las variables independientes sobre la variable dependiente son aditivos, los modelos de regresión multiplicativa asumen que estos efectos son multiplicativos (Tellis, G. J. (1988)).

Esto significa que, en lugar de sumar los efectos de las variables independientes, como en un modelo aditivo, los efectos se multiplican entre sí. Hanssens, (D. M., Parsons, L. J., & Schultz, R. L. (2001)). Matemáticamente, un modelo de regresión multiplicativa podría verse de esta manera:

$$Y = a * X_1^{b_1} * X_2^{b_2} * \dots * X_n^{b_n} * e$$



$$y = \beta_0 \cdot x_{TV}^{\beta_{TV}} \cdot x_{SEM}^{\beta_{SEM}} \cdot \dots \cdot x_{ctrl}^{\beta_{ctrl}}$$

↓ ↓ ↓

Sales Media Control

Donde:

Y es la variable dependiente (por ejemplo, las ventas).

X1, X2, ..., Xn son las variables independientes (por ejemplo, los componentes del mix de marketing).

a es la constante de proporcionalidad.

b1, b2, ..., bn son los exponentes que representan el impacto de las respectivas variables independientes en la variable dependiente.

e es el error aleatorio.

El uso de un modelo de regresión multiplicativo puede ser apropiado cuando existe una interacción entre las variables independientes, es decir, cuando el efecto de una variable independiente sobre la variable dependiente depende del nivel de otra variable independiente. Este es a menudo el caso en el marketing, donde los componentes del mix de marketing pueden interactuar entre sí de maneras complejas.(Leefflang, P. S. H., Wittink, D. R., Wedel, M., & Naert, P. A. (2000)).

Para superar las limitaciones inherentes a los modelos lineales, a menudo se prefieren los modelos multiplicativos. Estos modelos ofrecen una representación más realista de la realidad que los modelos lineales aditivos.

Hay dos tipos de modelos multiplicativos:

- **Modelos semilogarítmicos:** Los modelos semilogarítmicos son aquellos en los que la variable dependiente se transforma logarítmicamente, mientras que las variables independientes se mantienen en su forma original. Esta transformación logarítmica permite linealizar la relación entre las variables y facilita la estimación del modelo como una función aditiva.
- **Modelos logarítmicos:** Los modelos logarítmicos, por otro lado, implican la transformación logarítmica tanto de la variable dependiente como de las variables independientes. Esta transformación logarítmica se aplica a todas las variables involucradas en el modelo, lo que permite una representación log-lineal de la relación entre las variables.

$$\log y = \beta_0 + \beta_{TV} \log x_{TV} + \beta_{SEM} \log x_{SEM} + \dots + \beta_{ctrl} \log x_{ctrl}$$

↑ ↓ ↓ ↓

In(Sales) Intercept Media Control

La principal diferencia entre los modelos Log-Lineal y Log-Log se encuentra en la forma en que se interpretan los coeficientes de respuesta. En los modelos Log-Log, los coeficientes se interpretan como el porcentaje de cambio en el resultado empresarial (ventas) en respuesta a un cambio del 1% en la variable independiente. Esto significa que cada coeficiente representa la elasticidad porcentual de la variable dependiente con respecto a la variable independiente.

En contraste, en los modelos Log-Lineal, los coeficientes no se interpretan como porcentajes de cambio directo. En su lugar, representan el cambio en el logaritmo natural de la variable dependiente en respuesta a un cambio unitario en la variable independiente. Estos coeficientes tienen una interpretación más directa en términos de cambios relativos en la variable dependiente.

4.7.3.1. Resultados del Modelo Multiplicativo en el proyecto:

El modelo multiplicativo que se ha utilizado ha sido el algoritmo multiplicative_func(result.x, *X.T.values):

```
y_pred = multiplicative_func(result.x, *X_test.T.values)
X_pred = df[variables]
```

Donde result.x representa los coeficientes estimados del modelo y X.T.values es una matriz de valores de variables predictoras.

Este modelo ha generado un MAPE (Mean Absolute Percentage Error) de aproximadamente 28.19% indicando el porcentaje promedio de error absoluto entre las predicciones y los valores reales de las ventas.

Este resultado del MAPE indica qué tan cerca están las predicciones del modelo en comparación con los valores reales. Un MAPE más bajo indica una mejor precisión del modelo, ya que significa que las predicciones están más cerca de los valores reales en promedio.

En este caso, un MAPE del 28.19% indica que, en promedio, las predicciones del modelo tienen un error absoluto del 28.19% en comparación con los valores reales de las ventas. Esto significa que las predicciones pueden variar en un 28.19% en promedio con respecto a los valores reales.

Para ajustar el modelo se ha utilizado la función least_squares() a los datos de entrenamiento. Esta función optimiza los parámetros del modelo de manera que minimiza la suma de los cuadrados de los residuos, con los siguientes resultados:

Metric	Value
0 MSE	1.111821e+15
1 RMSE	3.334398e+07
2 MAE	2.709231e+07
3 R^2	6.254281e-01

MSE (Mean Squared Error): El valor del MSE es aproximadamente 1.11e+15. El MSE es una medida de la calidad general del modelo y representa el promedio de los errores al cuadrado entre las predicciones y los valores reales. Cuanto menor sea el valor del MSE, mejor será el modelo en términos de ajuste a los datos. En este caso, el valor del MSE es muy alto, lo que sugiere que el modelo puede tener un ajuste deficiente a los datos y que las predicciones pueden tener un error considerable.

RMSE (Root Mean Squared Error): El valor del RMSE es aproximadamente 3.33e+07. El RMSE es una métrica que representa la raíz cuadrada del MSE y tiene las mismas unidades que la variable objetivo. Un RMSE más bajo indica un mejor ajuste del modelo a los datos. En este caso, el valor del RMSE es alto, lo que sugiere que las predicciones del modelo pueden tener un error considerable en comparación con los valores reales de las ventas.



MAE (Mean Absolute Error): El valor del MAE es aproximadamente $2.71e+07$. El MAE es una medida del error absoluto promedio entre las predicciones y los valores reales. Cuanto menor sea el valor del MAE, mejor será el modelo en términos de precisión. En este caso, el valor del MAE indica que las predicciones del modelo pueden tener un error absoluto promedio de alrededor de $2.71e+07$ en comparación con los valores reales de las ventas.

R² (Coeficiente de determinación): El valor de R² es aproximadamente 0.625, lo que indica que alrededor del 62.54% de la variabilidad de las ventas puede ser explicada por las variables incluidas en el modelo. El R² se utiliza para evaluar la capacidad del modelo para capturar la variabilidad de los datos y proporciona una medida de qué tan bien se ajusta el modelo a los datos. Un valor de R² más cercano a 1 indica un mejor ajuste del modelo, mientras que un valor cercano a 0 indica un ajuste deficiente. En este caso, el valor de R² indica que el modelo puede explicar aproximadamente el 62.54% de la variabilidad en las ventas.

Y finalmente para realizar las predicciones se ha utilizado los parámetros optimizados para hacer predicciones de ventas en el conjunto de prueba utilizando la función `multiplicative_func()`, con los siguientes resultados que podemos ver en esta gráfica:

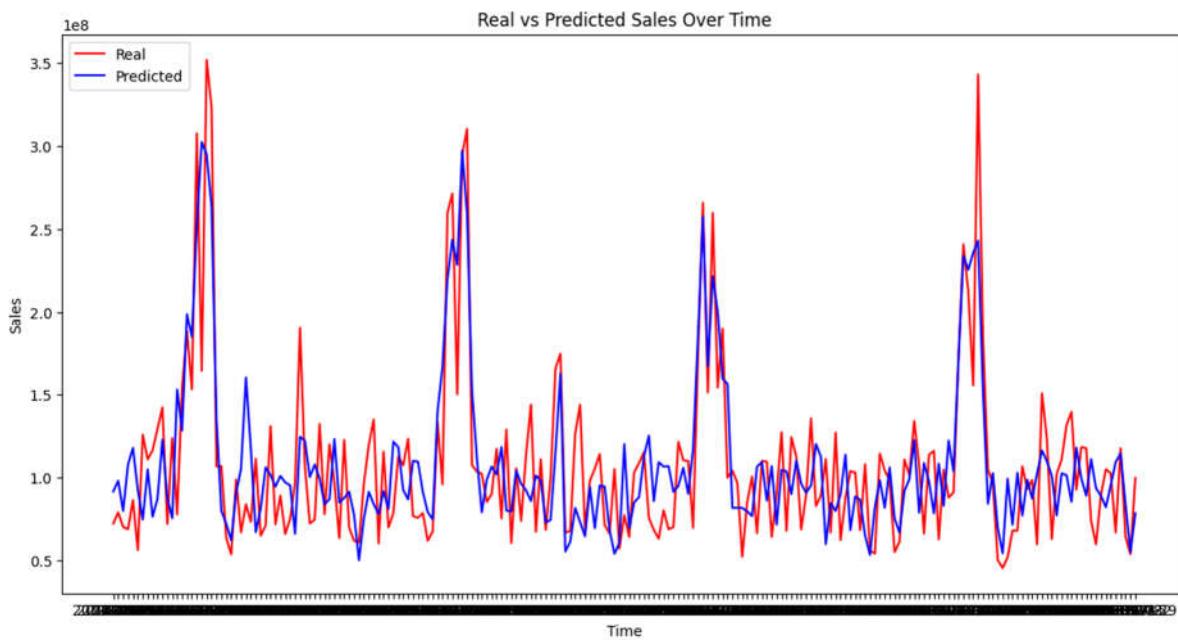


Ilustración 20 - gráfica estimación ventas sobre predicción

Y finalmente se hace la predicción de la inversión en marketing para cada canal comparado con la inversión real por medio:

Comparativa de porcentajes: Predicho vs Original

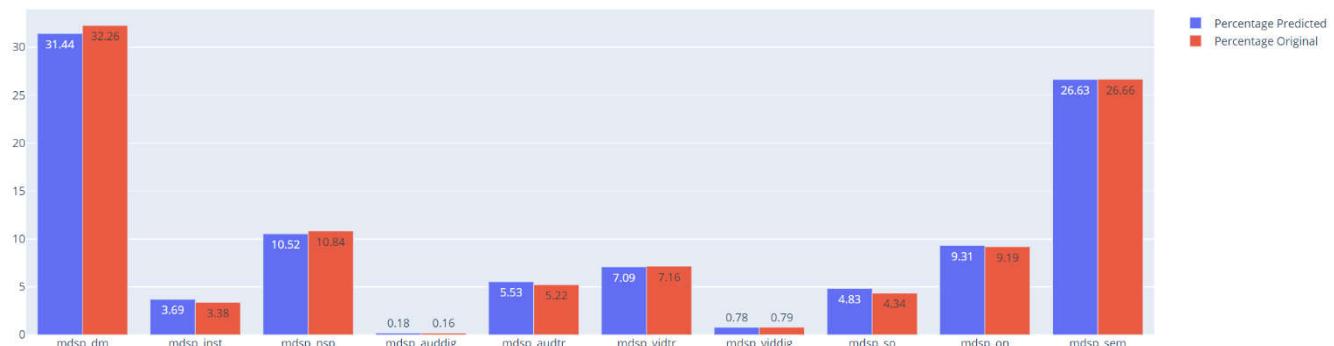


Ilustración 21 - gráfica de la inversión por canal predicha

En general, los resultados indican que el modelo multiplicativo no ajusta adecuadamente los datos de respuesta, ya que las métricas de evaluación (MSE, RMSE y MAE) tienen valores altos. Por lo que se puede decir que el modelo puede tener ciertas limitaciones y que hay margen de mejora en términos de precisión y ajuste a los datos.

4.8. Modelos de Nueva Generación de Marketing Mix

Los modelos analizados anteriormente no tienen en cuenta muchas de los retos actuales que se tiene en las actuales variables de marketing, como son:

- Los medios publicitarios evolucionan a gran velocidad, lo que obliga a los modelos para tener en cuenta constantemente los nuevos mercados.
- Los modelos actuales tienen un nivel alto de granularidad de los datos, que puede dar lugar a observaciones dispersas y valores atípicos.
- La naturaleza secuencial de los datos los hace más susceptibles a errores correlacionados, lo que condiciona los modelos lineales.

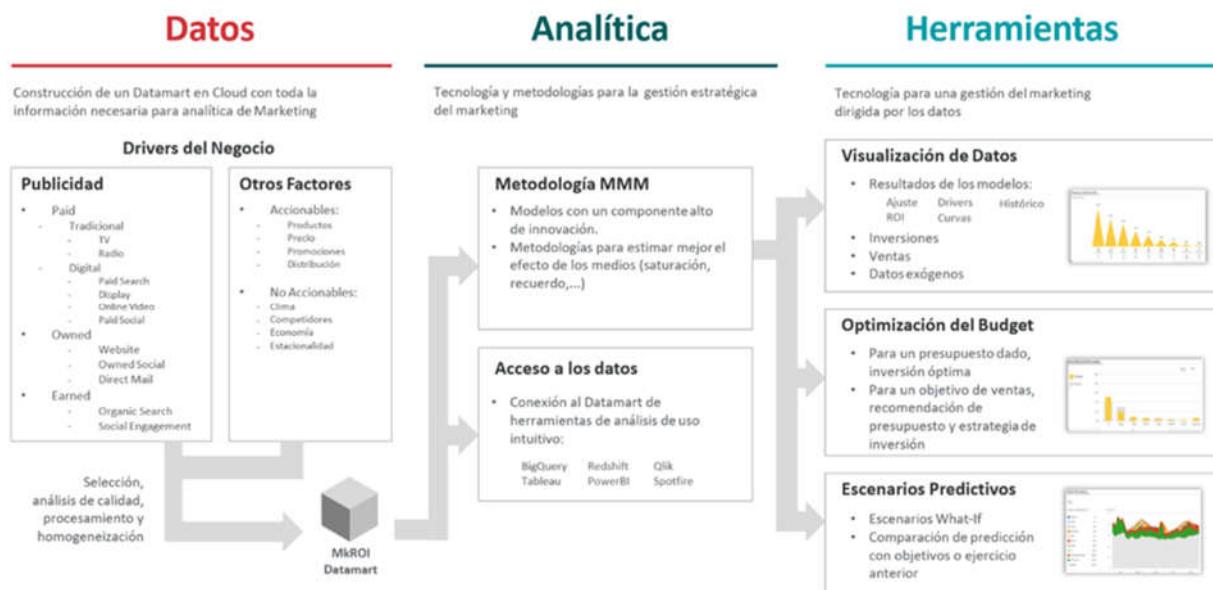
Y además, existen graves problemas de endogeneidad y multicolinealidad debido a la planificación de marketing y a la dinámica de los medios de comunicación.

ANTIGUO PARADIGMA	NUEVO PARADIGMA
Modelización manual	Selección automatizada de hiperparámetros
Modelos estáticos multi-year	Regresión de Ridge y Regularización
Problemas de Multicolinealidad y sobreajuste	Estacionalidad y tendencia modelada
Estacionalidad manual y variables dummy	Validación y calibrado con experimentación
Solo validación estadística	Modelos actualizados dinámicamente y frecuentemente

Ilustración 22 - Comparación modelos tradicionales MMM vs nuevos modelos

La mayor granularidad y variedad de la información y la evolución de los algoritmos han provocado cambios significativos en la manera de medir el ROI de la publicidad en los últimos años. A continuación se han planteado y evaluado varios modelos de nueva generación que incluyen estos conceptos que son fundamentales para que un modelo pueda representar la realidad del marketing actual.

Los elementos que deberán tener los modelos de marketing mix de nueva generación:



Fuente: i+a española

Ilustración 23 - Bases del Modelo de Marketing Mix

4.8.1. Modelo Robyn:

Robyn, de Facebook Experimental, es un código de modelado de marketing mix (MMM) que se encuentra actualmente en versión beta (modelo de Robin experimental en R en el repositorio <https://github.com/facebookexperimental/Robyn>).

Estas son las etapas que conforman el modelo:

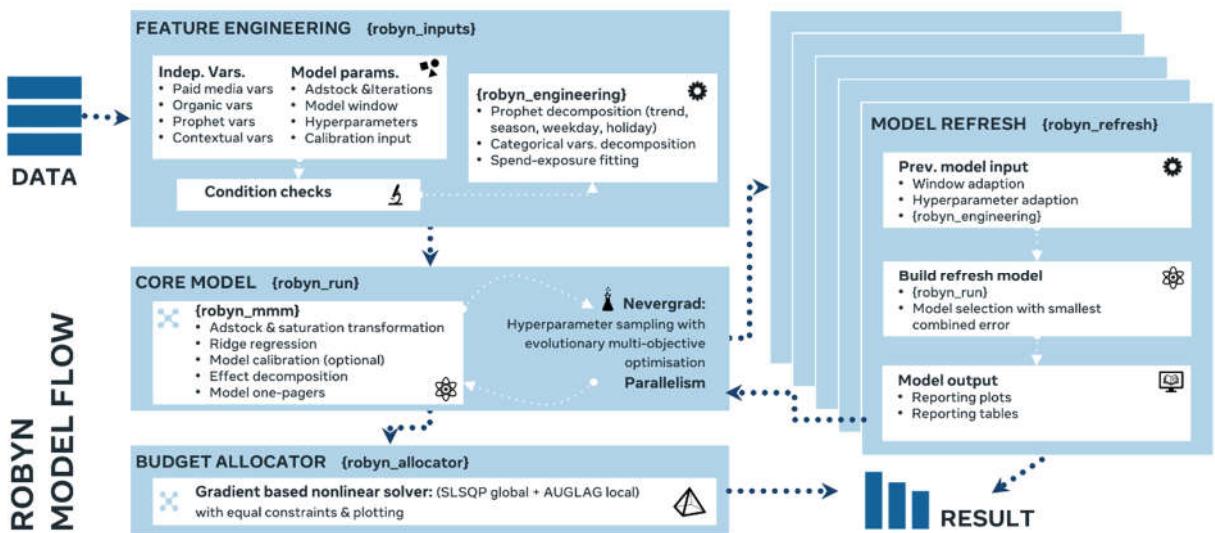


Ilustración 24 - Fases del modelo Robyn

Fuente:Facebook

Recopilación y Preparación de Datos: Esta etapa implica la recopilación de datos sobre las ventas (u otra métrica de interés) y los gastos de marketing en diferentes canales, así como de

posibles variables de control como días festivos, tendencias económicas, etc. Los datos luego son preparados y preprocesados para el análisis.

Descomposición de Series Temporales con Prophet: La librería Prophet , una herramienta de predicción de series temporales desarrollada por Facebook, para la previsión de datos de series temporales, que permite descomponer los datos en tendencia, estacionalidad, días festivos y días de la semana, con el fin de mejorar el ajuste del modelo y la capacidad de previsión. Tradicionalmente, sería necesario recopilar y modelizar los datos de estacionalidad y días festivos como variables ficticias independientes en el modelo.

Para recoger las series temporales la estacionalidad se basa en las series de Fourier para proporcionar un modelo flexible de efectos periódicos (Harvey & Shephard 1993). Sea P el periodo periódico que esperamos que tenga la serie temporal (por ejemplo $P = 365.25$ para datos anuales o $P = 7$ para datos semanales), aproximando los efectos estacionales suaves arbitrarios con:

$$s(t) = \sum_{n=1}^N \left(a_n \cos\left(\frac{2\pi nt}{P}\right) + b_n \sin\left(\frac{2\pi nt}{P}\right) \right)$$

una serie estándar de Fourier. El componente estacional es entonces:

$$s(t) = X(t)\beta.$$

Es un modelo generativo con $\beta \sim \text{Normal}(0; \sigma^2)$ para imponer un suavizado a priori sobre la estacionalidad.

Los días festivos y otros eventos importantes causan interrupciones notables e impredecibles, las cuales no siguen una pauta regular. Debido a esto, sus efectos no pueden ser correctamente modelados utilizando ciclos regulares. Esto se hace de forma similar a la estacionalidad, generando una matriz de regresores:

$$Z(t) = [\mathbf{1}(t \in D_1), \dots, \mathbf{1}(t \in D_L)]$$

tomando

$$h(t) = Z(t)\kappa.$$

Al igual que con la estacionalidad, utilizamos una prior $\kappa \sim \text{Normal}(0; v^2)$.

Al combinar las características de estacionalidad y vacaciones de cada observación en una matriz X y los indicadores de punto de cambio en una matriz A , podemos expresar fácilmente el modelo completo a través del código Stan (Carpenter et al. 2017), que se muestra a continuación:

```

model {
    // Priors
    k ~ normal(0, 5);
    m ~ normal(0, 5);
    epsilon ~ normal(0, 0.5);
    delta ~ double_exponential(0, tau);
    beta ~ normal(0, sigma);

    // Logistic likelihood
    y ~ normal(C ./ (1 + exp(-(k + A * delta) .* (t - (m + A * gamma)))) +
               X * beta, epsilon);
    // Linear likelihood
    y ~ normal((k + A * delta) .* t + (m + A * gamma) + X * beta, sigma);
}

```

Listing 1: Example Stan code for our complete model.

Transformación de Medios y Variables de Control: Los datos recopilados se transforman utilizando una serie de funciones para reflejar suposiciones clave sobre cómo los gastos de marketing afectan las ventas. Esto puede incluir funciones que capturan la duración de los efectos del gasto en marketing ad stock (carry-over) y el impacto de la saturación (efecto Hill).

Para el cálculo del ad stock el modelo utiliza estos métodos:

- **Geometric:**

Se utiliza la función de decaimiento exponencial uniparamétrica con theta como parámetro de decaimiento de tasa fija. Por ejemplo, un valor de theta = 0,75 significa que el 75% de los anuncios del Periodo 1 se trasladan al Periodo 2. La implementación de Robyn de la transformación geométrica se puede encontrar aquí y se muestra conceptualmente como sigue:

$$x_{adstocked_{i,j}} = x_{raw_{i,j}} + \theta_j * x_{raw_{i-1,j}}$$

A partir de la creación histórica de modelos, se han determinado algunos parámetros como por ejemplo para la TV entre 0,3 y 0,8, OOH/Print/Radio entre 0,1-0,4, y Digital entre 0,0 - 0,3.

- **Weibull PDF & CDF:**

El modelo ofrece la función Weibull bi paramétrica en los formatos PDF (función de densidad de probabilidad) y CDF (Función de Distribución Acumulativa). Comparada con la función Geométrica uniparamétrica donde theta es igual a la tasa de decaimiento fija, Weibull produce tasas de decaimiento variables en el tiempo a través de una mayor flexibilidad en la transformación con los parámetros forma y escala.

$$x_{adstocked_{i,j}} = x_{raw_{i,j}} + \theta_{i-1,j} * x_{raw_{i-1,j}}$$

- **Weibull CDF adstock:** La Función de Distribución Acumulativa de Weibull tiene dos parámetros, forma y escala, y tiene una tasa de decaimiento flexible, comparada con la Geometric adstock con tasa de decaimiento fija. El parámetro de forma controla la forma de la curva de caída. El límite recomendado es c(0.0001, 2). Cuanto mayor sea la forma, más forma de S tendrá. Cuanto más pequeña, más forma de L. La escala controla el punto de inflexión de la curva de La recomendación es un rebote muy conservador de c(0, 0,1), porque la escala aumenta mucho la vida media del adstock.

- **Weibull PDF adstock:** La función de densidad de probabilidad de Weibull también tiene dos parámetros, forma y escala, y también tiene una tasa de decaimiento flexible como Weibull CDF. La diferencia es que la PDF de Weibull ofrece un efecto retardado. Debido a la gran flexibilidad de la PDF de Weibull, lo que significa más libertad en los espacios de hiper parámetros que puede explorar Nevergrad, también requiere más iteraciones para converger.

Para el cálculo de la saturación de cada canal el modelo utiliza la función Hill, que es una función bi paramétrica con alfa y gamma:

- Alpha controla la forma de la curva entre exponencial y en forma de s, con valor estándar con un límite de $c(0,5, 3)$.
- Gamma controla el punto de inflexión, con valor estándar con un límite de $c(0,3, 1)$.

Para la previsión del crecimiento, y saturación el componente central del proceso de generación de datos es un modelo de cómo ha crecido la población y cómo se espera que siga creciendo. Este tipo de crecimiento se hace con el modelo de crecimiento logístico:

$$g(t) = \frac{C}{1 + \exp(-k(t - m))},$$

siendo C la capacidad de carga, k la tasa de crecimiento y m un parámetro de compensación.

Para los problemas de previsión que no muestran un crecimiento saturado, una tasa de crecimiento constante a intervalos proporciona un modelo útil, con el modelo de tendencia:

$$g(t) = (k + a(t)^T \delta)t + (m + a(t)^T \gamma),$$

donde como antes k es la tasa de crecimiento, δ tiene los ajustes de la tasa, m es el parámetro de compensación, y γ se fija en $-s_j\delta_j$ para que la función sea continua.

Especificación del Modelo: Se especifica un modelo estadístico que relaciona las ventas con los gastos de marketing y las variables de control. El MMM utiliza el modelo de regresión, cuyo objetivo es obtener una ecuación que describa la variable dependiente. El modelo tiene como objetivo asignar un coeficiente a cada variable independiente, donde sólo las variables que son estadísticamente significativas permanecen en el modelo.

Estimación del Modelo con Regresión Ridge: El modelo se ajusta a los datos utilizando el método de Stan, un tipo de algoritmo de Monte Carlo Hamiltoniano, y la regresión Ridge, un método de regularización para prevenir el sobreajuste. Estos se usan para estimar los parámetros del modelo.

El modelo se basa en la descomposición de la variable dependiente mediante la regresión Ridge, que es una variante de la regresión lineal que incluye la norma de regularización L2. Se diferencia del modelo lineal clásico en que impone una “penalización” a variables insignificantes, porque las variables en los modelos MMM a veces están altamente correlacionadas entre sí, lo que significa que no se pueden sacar conclusiones estadísticamente significativas del modelo.

La motivación de Meta para utilizar un método de regularización fue abordar la multicolinealidad entre muchos regresores y evitar que el modelo se ajustara en exceso. La ecuación del modelo con los principales componentes de la función es la siguiente:



The diagram shows the S-Curve regression model equation:

$$y_t = \text{Intercept} + \beta_j \times \frac{x_{decay_{t,j}}^\alpha}{x_{decay_{t,j}}^\alpha + \gamma^\alpha} + \beta_{hol} \cdot hol_t + \beta_{sea} \cdot sea_t + \beta_{trend} \cdot trend_t + \dots + \beta_{ETC} \cdot ETC_t + \varepsilon$$

Annotations explain the components:

- Main components of the function:** Intercept, $\beta_j \times \frac{x_{decay_{t,j}}^\alpha}{x_{decay_{t,j}}^\alpha + \gamma^\alpha}$, $\beta_{hol} \cdot hol_t$, $\beta_{sea} \cdot sea_t$, $\beta_{trend} \cdot trend_t$, ..., $\beta_{ETC} \cdot ETC_t$, ε .
- S-Curve component for each media (j):** $\frac{x_{decay_{t,j}}^\alpha}{x_{decay_{t,j}}^\alpha + \gamma^\alpha}$
- Holiday, Seasonality and Trend effect***: $\beta_{hol} \cdot hol_t$, $\beta_{sea} \cdot sea_t$, $\beta_{trend} \cdot trend_t$, ..., $\beta_{ETC} \cdot ETC_t$
- Independent Variables**: $x_{decay_{t,j}}$, γ , hol_t , sea_t , $trend_t$, ETC_t , ε .

Below the main equation, specific components are defined:

1. Adstock transformation: $X_{decay_{t,j}} = X_{t,j} + \theta_j \cdot X_{decay_{t,j-1}}$
2. S Curve transformation: $S\text{ Curve}(x, j) = \beta_j \times \frac{x_{decay_{t,j}}^\alpha}{x_{decay_{t,j}}^\alpha + \gamma^\alpha}$

Where:

- y_t = revenue at time t
- t = time index of dependent and independent variable (week)
- j = media index (e.g. FB, TV, OOH) and $\beta, \alpha, \gamma, \theta$ = regressor specific to each media j
- y implemented on the S-Curve is a transformed y where $y_{tran} = \text{quantile}(X_{decay_{j+1}}, Y)$
- β_{ETC}, ETC_t = further independent variables to be added to the model (e.g. competitor, promotions)
- ε = Error term (accounting for all the other factors not addressed in the model)

Fuente: Meta

Donde y_t es nuestra variable dependiente ingresos en el momento t . Las variables independientes se definen por el intercepto, seguido del componente transformado ad-stock y s-curve para cada medio j . Los efectos de vacaciones, estacionalidad y tendencia se representan por hol, sea y trend. Las variables independientes adicionales se definen por ETC seguido del término de error ε .

El uso de la regresión ridge es una buena solución, porque permite mejorar la automatización en el proceso de modelado, pero puede conducir a una situación en la que el algoritmo impone una penalización en un medio que el usuario decide usar mucho.

Optimización de Hiperparámetros con Nevergrad: Nevergrad, un framework de optimización de código abierto basado en la derivada libre desarrollado por Facebook, para realizar una optimización multiobjetivo que equilibra la relación entre la cuota de gasto y la cuota de descomposición del coeficiente de los canales, proporcionando un conjunto de soluciones modelo óptimas de Pareto.

Nevergrad permite optimizar de manera eficiente los hiper parámetros y las métricas para evaluar la calidad del modelo, que incluyen:

- NRMSE: cuanto menor sea, menor será la varianza de los residuos de los modelos.
- DECOMP.RSSD: cuanto menor sea, mejor emparejará el modelo la descomposición con la distribución del gasto en medios.

En base a esto, Nevergrad elige un conjunto de modelos óptimos de Pareto. Para ello Nevergrad realiza una cantidad de iteraciones seleccionadas por el usuario (se recomienda no menos de 2000) que crean múltiples modelos. Luego minimiza las métricas seleccionadas y selecciona modelos óptimos de Pareto de entre ellos.

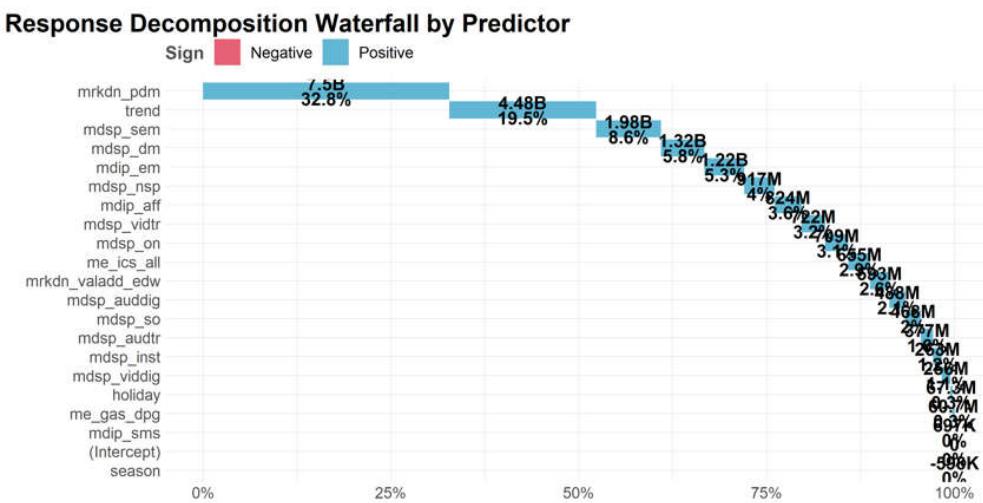
Validación y Prueba del Modelo: El modelo se valida y se prueba su rendimiento mediante métricas como el Error Cuadrático Medio (RMSE) y el coeficiente de determinación ajustado (R-cuadrado).

Interpretación y Uso de los Resultados: Los coeficientes del modelo son interpretados como los efectos marginales de los gastos de marketing en diferentes canales sobre las ventas. Estos resultados se pueden usar para optimizar la asignación del presupuesto de marketing.

4.8.1.1. Resultados del Modelo Robyn en el proyecto:

Los resultados del MMM representados en estos gráficos generan las soluciones óptimas del modelo como resultado del proceso óptimo de Pareto de optimización multiobjetivo.

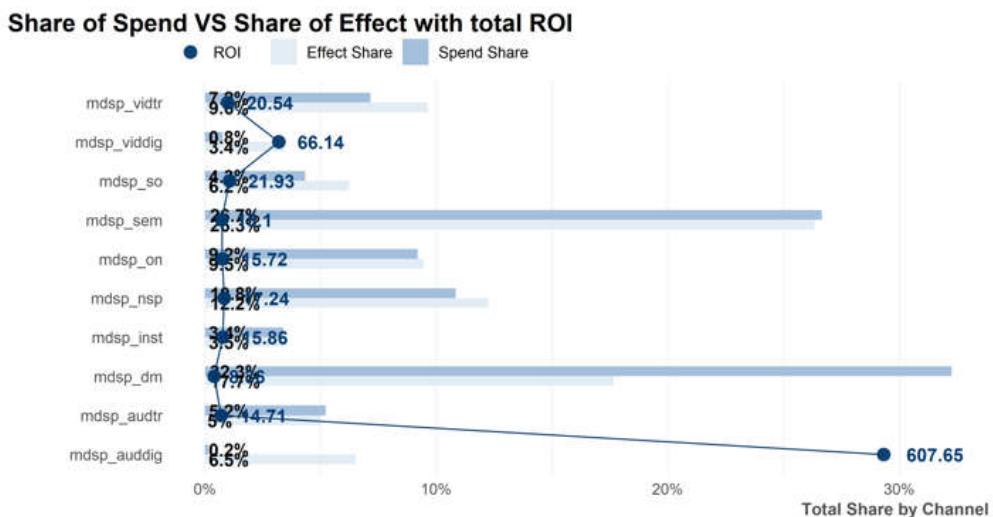
1. Descomposición de la respuesta por el predictor:



Este gráfico refleja el porcentaje del efecto de cada una de las variables (variables Base y Media + intercepto) sobre la variable respuesta. Donde vemos que el efecto de la reducción precios permanente es del 32,8%, significa que el 32,8% de las ventas totales pueden atribuirse a las reducciones de precios. El intercepto y las variables de series temporales, en este caso la tendencia constituye una gran parte de su descomposición, con el 19,5% . Esto tiene sentido ya que el dataset toma datos de una marca está establecida, esto significa que se tiene una gran base de ventas sin gasto en medios pagados.

2. Porcentaje de gasto frente a efecto con ROI total:

El gráfico siguiente muestra el impacto detallado de las contribuciones de los medios comparando y contrastando varias métricas diferentes:



La cuota de gasto refleja el gasto relativo de cada canal; La cuota de efecto es lo mismo que la contribución en volumen, es decir, la cantidad de ventas incrementales impulsadas por cada canal; El ROI (retorno de la inversión) representa la eficacia de cada canal y se calcula dividiendo los ingresos incrementales generados por un canal por la cantidad gastada en ese canal.

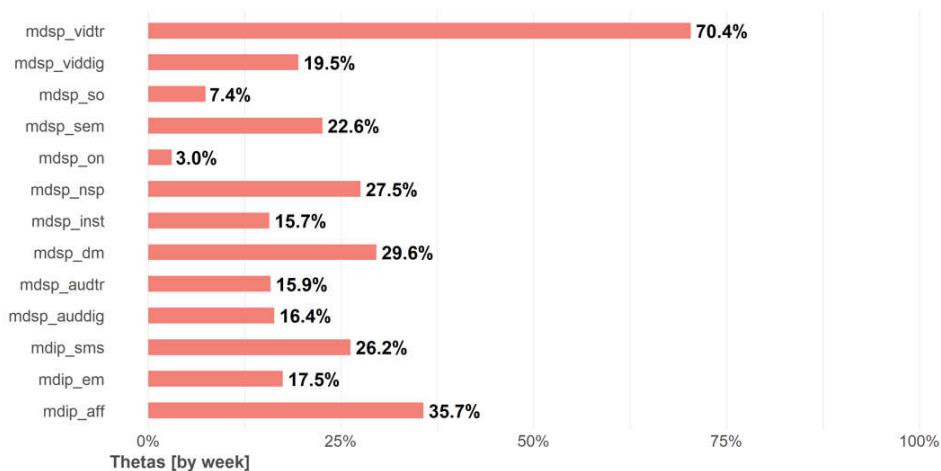
El canal de audio digital, presenta un ROI alto pero una contribución y un gasto bajos, esto puede indicar un posible aumento del gasto, dado que está ofreciendo buenos rendimientos y es probable que no esté saturado debido al bajo gasto.

El canal de SEM, TV y mailing presentan un ROI bajo pero una contribución y un gasto elevados, esto puede indicar que su rendimiento es bajo y que, por lo tanto, habría que reducir el gasto en él; sin embargo, es un gran impulsor del rendimiento y, por lo tanto, podría no ser acertado. Aquí se debería considerar ver la forma de cómo optimizar estos canales, ya que son canales importantes.

3. Tasa media de caída de adstock:

En este gráfico se representa, en promedio, cuál es el porcentaje de decaimiento que tuvo cada canal. Cuanto mayor sea la tasa de decaimiento, mayor será el efecto en el tiempo de la exposición a los medios de ese canal específico. Se trata de un cálculo sencillo para los adstocks geométricos, con una tasa de decaimiento media del adstock = theta. Para los adstocks Weibull este cálculo es un poco más complicado.

Geometric Adstock: Fixed Rate Over Time

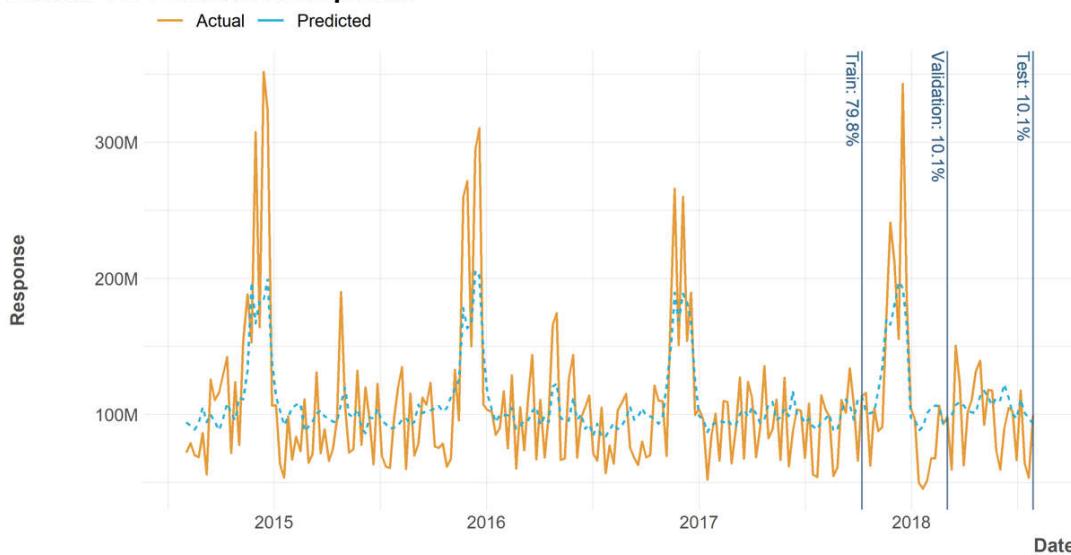


Comparando las tasas de adstock entre las diferentes fases del embudo de ventas, observamos que los medios del canal del superior o de captación de leads son más bajas en el recuerdo de los consumidores que los adstocks el canal del embudo inferior. Se pueden sacar algunas conclusiones de la efectividad de estas acciones en estos canales, que se debería conseguir subir su adstock o cambiar la estrategia de medios.

4. Respuesta real frente a respuesta prevista:

Este gráfico nos muestra los datos reales de la variable de respuesta, en nuestro caso, las ventas, y cómo los datos previstos del modelo para esa variable de respuesta reflejan la curva real. El modelo ha buscado modelos que puedan capturar la mayor parte de la varianza de los datos reales y, por lo tanto, el R-cuadrado está más cerca de 1, mientras que el NRMSE es bajo.

Actual vs. Predicted Response



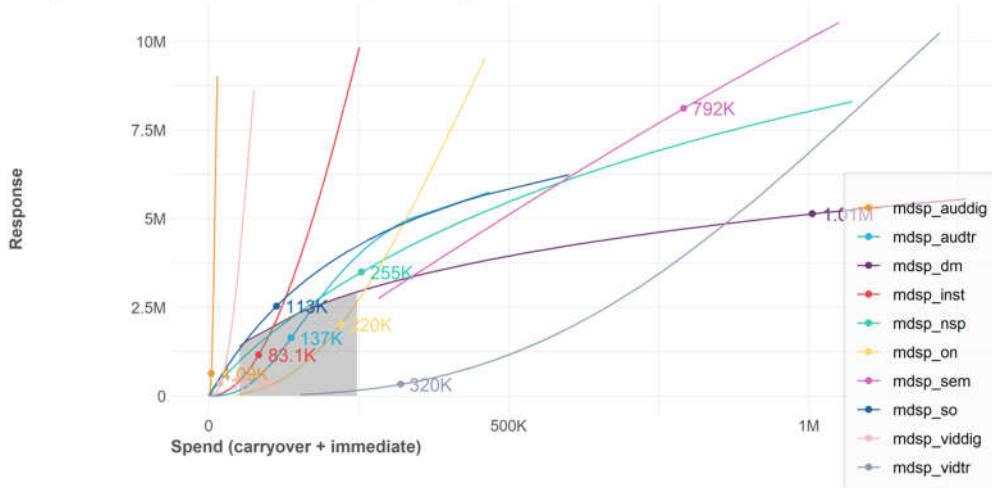
En general el modelo predice de forma similar en todos los períodos. Si bien es cierto, que en los períodos de mayor ventas o picos promocionales la curva predicha es más suave que la real,

esto nos puede indicar que pueda existir alguna variable contextual que debería incluirse en el modelo.

5. Curvas de respuesta y gasto medio por canal:

Son las curvas de respuesta de rendimiento decreciente de la función de los impactos de subida. Representan el grado de saturación de un canal y, por tanto, pueden sugerir posibles estrategias de reasignación presupuestaria. Cuanto más rápido lleguen las curvas a un punto de inflexión y a una pendiente horizontal/plana, más rápido se saturarán con cada gasto adicional (\$).

Response Curves and Mean Spends by Channel

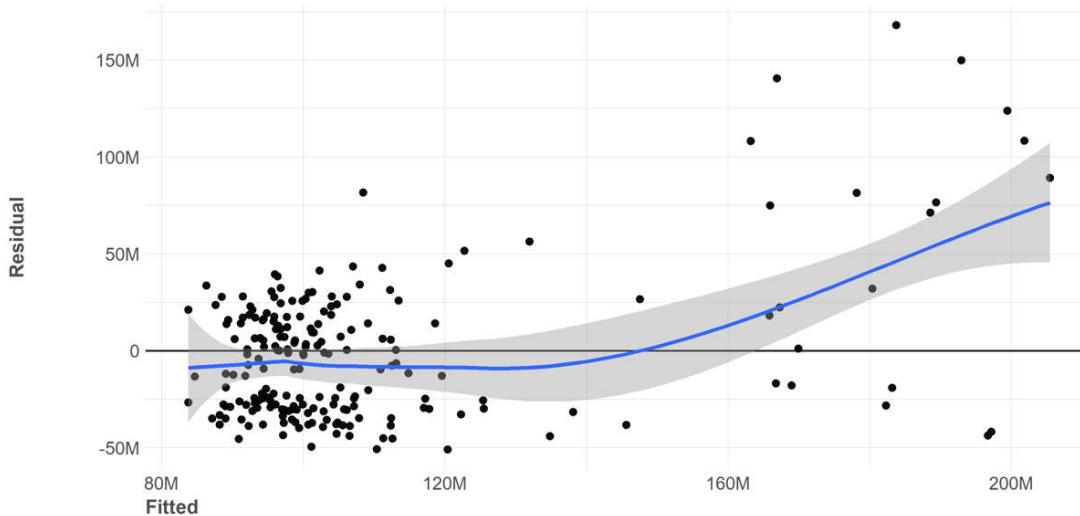


Comprobamos que las campañas SEM tiene un excelente comportamiento, con una alta subida y alto valor de 729k, seguidas de las campañas online, promociones y video digital que aunque tienen un curva más pronunciada, su valor y crecimiento con el gasto es inferior.

6. Ajustado frente a residual:

Este gráfico muestra la relación entre valores ajustados y residuales. Un valor residual es una medida de cuánto se aleja verticalmente una línea de regresión de un punto de datos. Un gráfico residual se utiliza normalmente para encontrar problemas con una regresión. Algunos conjuntos de datos no son buenos candidatos para la regresión, como los puntos a distancias muy variables de la línea. Si los puntos de un gráfico residual están dispersos aleatoriamente alrededor del eje horizontal, un modelo de regresión lineal es apropiado para los datos; de lo contrario, un modelo no lineal es más apropiado.

Fitted vs. Residual



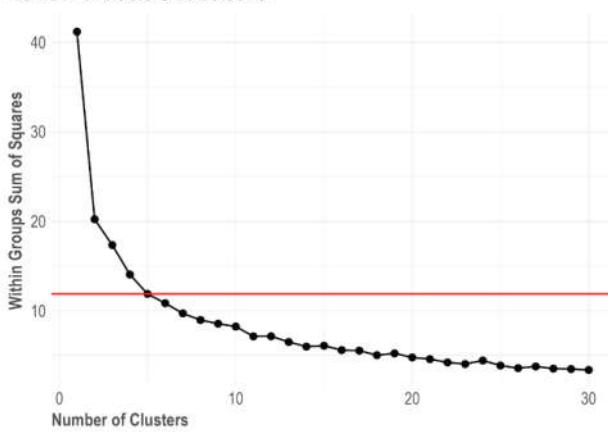
Como comprobamos que hay una alta dispersión de los puntos de error sobre la regresión lineal, con lo que podemos confirmar que un modelo de regresión lineal no es el más adecuado.

7. Agrupación de soluciones modelo

En lugar de explorar todas las soluciones Pareto-óptimas (de las que podría haber docenas), buscamos un numero de clústers para encontrar soluciones similares que ofrecer a los usuarios y encontrar las mejores.

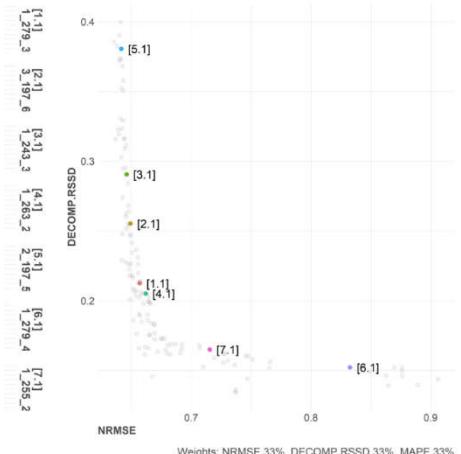
Total Number of Clusters

Number of clusters selected: 5



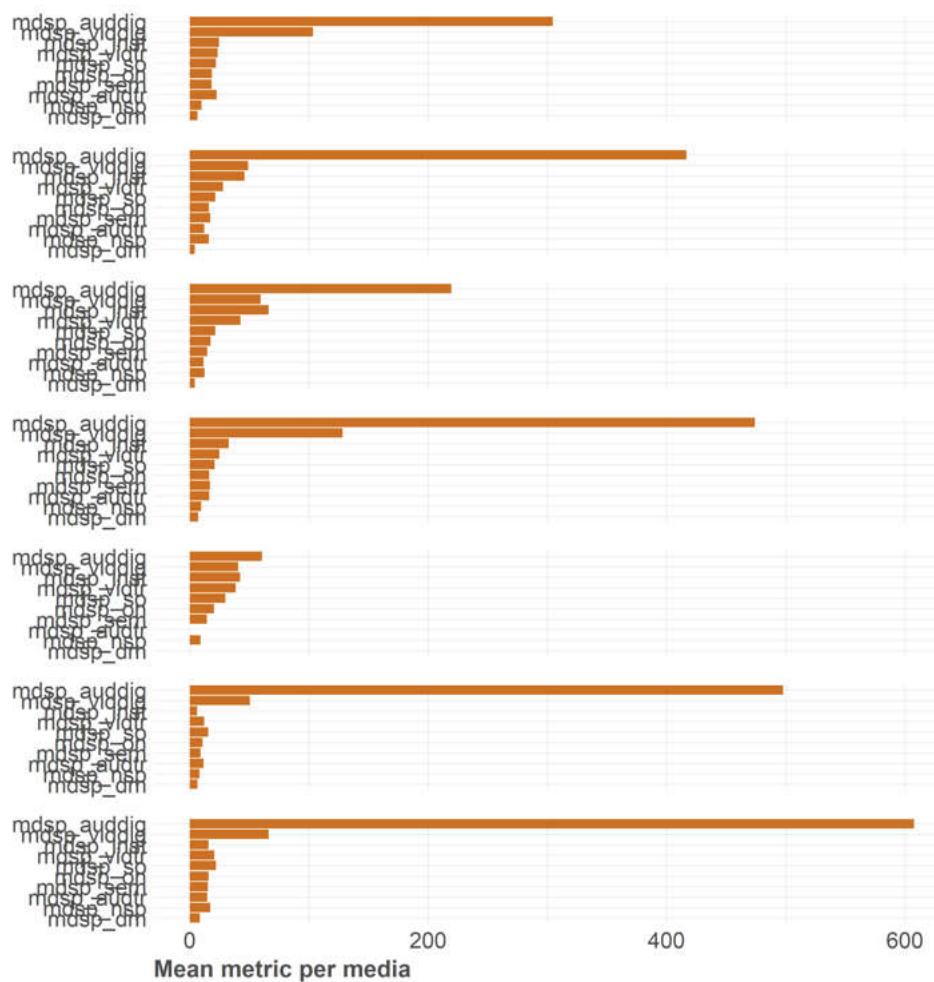
Selecting Top 1 Performing Models by Cluster

Based on minimum (weighted) distance to origin



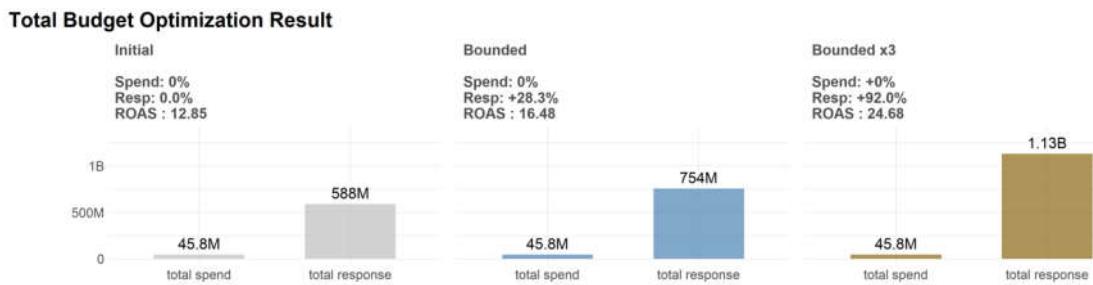
Como resultado de la segmentación en los clústers se identifican los medios más importantes como se muestra en el gráfico, siendo el canal de audio digital el más importante por su comportamiento con los usuarios con diferencia.

Top Performing Models



8. Asignación del presupuesto

Una vez analizado los gráficos con el modelo optimo, obtenemos los resultados de la asignación del presupuesto optimizado para cada canal:



Budget Allocation per Channel*

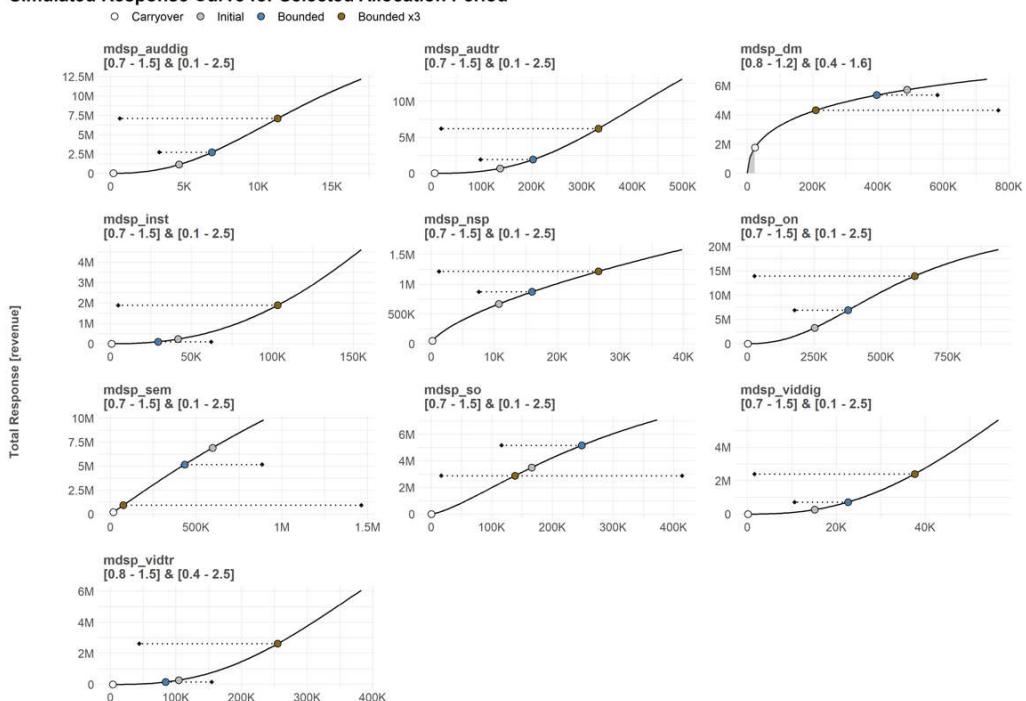
Paid Channels	Initial				Bounded				Bounded x3			
	spend%	response%	ROAS	mROAS	spend%	response%	ROAS	mROAS	spend%	response%	ROAS	mROAS
mdsp_auddig	0.2%	5%	252	565	0.4%	9.4%	406	848	0.6%	16.3%	636	1.02K
mdsp_audtr	7.4%	3%	5.26	13.3	11.1%	6.5%	9.7	24	18.5%	14.3%	19.1	40
mdsp_dm	26.5%	25.3%	12.3	3.59	21.2%	18.5%	14.4	4.33	10.6%	10%	23.2	7.35
mdsp_inst	2.3%	1%	5.44	12.8	1.6%	0.3%	3.37	7.87	5.8%	4.3%	18.4	41.7
mdsp_nsp	0.6%	2.9%	63.5	41.7	0.9%	3%	55.2	36.2	1.5%	2.8%	46.2	30
mdsp_on	14.2%	14.4%	13	25.8	21.3%	23.7%	18.3	30.7	35.5%	32%	22.3	23.7
mdsp_sem	32.8%	30.5%	11.9	10.4	23.6%	17.8%	12.4	11.1	3.3%	2.1%	15.9	12.4
mdsp_so	9.4%	15.5%	21.2	22	14.1%	17.8%	20.8	18.1	7.8%	6.6%	20.9	23
mdsp_viddig	0.9%	1.2%	17.5	43.5	1.3%	2.5%	31.7	77.5	2.1%	5.5%	63.8	144
mdsp_vidtr	5.7%	1.2%	2.74	6.95	4.6%	0.5%	1.95	4.92	14.3%	6%	10.4	23.6

Spend** per week (Mean Adstock Zone in Grey)

* ROAS = total response / raw spend | mROAS = marginal response / marginal spend
 * When reallocating budget, mROAS converges across media within respective bounds
 ** Dotted lines show budget optimization lower-upper ranges per media

Se comprueba la asignación presupuestaria inicial frente a optimizada, donde se puede observar la cuota de inversión original frente a la nueva cuota optimizada recomendada. Si la cuota optimizada es mayor que la original, esto significa que tendrá que aumentar proporcionalmente los presupuestos para ese canal en función de la diferencia entre ambas cuotas. Y se reducirían los presupuestos en el caso de que la cuota de gasto fuera mayor que la cuota optimizada.

Después de la optimización vemos que los tres canales principales de inversión en medios son las campañas online con un 32%, el audio digital con un 16% y la radio con un 14%. Siendo el medio que peor comportamiento ha tenido en la optimización el SEM, de ser el medio principal con un 30% a un 2,1%.

Simulated Response Curve for Selected Allocation Period

Y en este gráfico observamos la curva de respuesta y gasto medio por canal, que representan el grado de saturación de un canal y, por lo tanto, pueden sugerir posibles estrategias de reasignación del presupuesto. Cuanto más rápido lleguen las curvas a un punto de inflexión y a una pendiente horizontal/plana, más rápido se saturarán con cada gasto adicional.

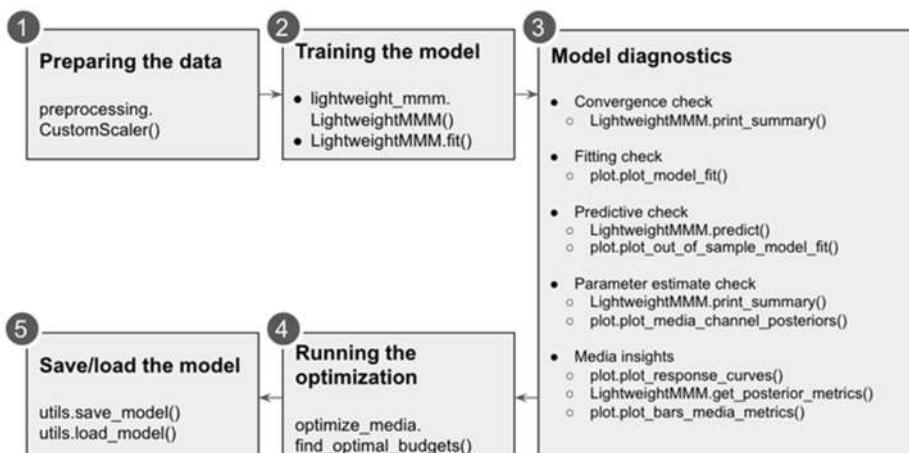
Para interpretar los resultados del modelo de marketing mix se utiliza el coeficiente R-cuadrado, siendo:

1. $R^2 < 0,8$ = no es ideal, intente mejorarlo;
2. $0,8 < R^2 < 0,9$ = aceptable, si es posible, intentar mejorar;
3. $R^2 > 0,9$ = ideal.

4.8.2. Modelo Bayesiano con Carryover y Shape Effects

El modelo bayesiano que se utilizará es el de LightweightMMM, un desarrollo de Google, que utiliza el lenguaje de python para optimizar la inversión en marketing en todos los canales de medios, utilizando las librerías de Numpyro y JAX para la programación probabilística, lo que hace que el proceso de modelado sea mucho más rápido que otros algoritmos (https://github.com/google/lightweight_mmm).

Los pasos del modelo son los siguientes:



Fuente Google

Ilustración 25 - Fases del Modelo LightweightMMM

1. Preparación de los datos

En este paso hay que convertir los datos a un vector o tensor que permita el procesamiento en el framework de Tensorflow que usa el modelo.

En esta fase hay que escalar los datos mediante un modelo estándar, aunque dependiendo de los datos, si contienen muchos ceros, se debería escalar usando sólo los valores distintos de cero.

Es importante escalar todas las variables, tanto para obtener una estimación correcta del efecto de los medios como para ayudar al optimizador cuando está tomando medidas para encontrar los coeficientes correctos para el modelo.

2. Transformaciones de Medios

En la mayoría de las ocasiones se formula la hipótesis de que la inversión en medios tiene un impacto sostenido en las ventas a lo largo de varios días, semanas o incluso meses. Además, en ocasiones se plantea que los efectos de la inversión en medios sobre las ventas experimentan un cierto retraso. Las transformaciones que consideran este efecto de “Carryover”(de arrastre) se pueden incorporar en este modelo, mediante la transformación de decaimiento geométrico “Geo Decay”, el “adstock” y el “Hill-Adstock” .

- **Decaimiento Geométrico “Geo Decay”**

Las transformaciones de decaimiento geométrico nos permiten modelar el impacto prolongado de la inversión en marketing:

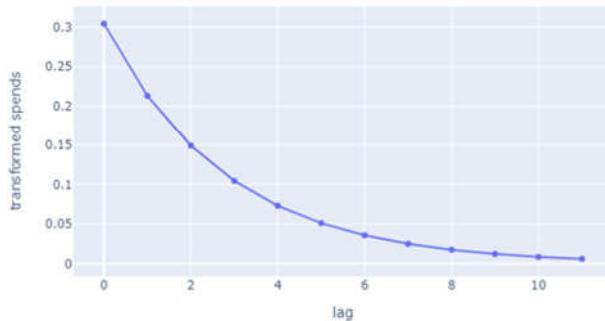
$$\text{geodecay}(x_{t-L+1,m}, \dots, x_{t,m}; a_m, L) = \frac{\sum_{l=0}^{L-1} a_m^l \cdot x_{t-l,m}}{\sum_{l=0}^{L-1} a_m^l}$$

$x_{t,m}$: the spend for media m at time t

a_m : the retain rate for media m ($0 < a_m < 1$)

L : maximum duration of effect

Geodecay effect on spends with retain_rate = 0.7



Fuente:

<https://github.com/leopoldavezac>

- **Efecto “Adstock”**

La transformación adstock nos permite modelizar el impacto duradero y el efecto retardado del gasto en marketing. El modelo aplica un lag infinito que va disminuyendo su peso a medida que pasa el tiempo.

$$\text{adstock}(x_{t-L+1,m}, \dots, x_{t,m}; a_m, \theta_m, L) = \frac{\sum_{l=0}^{L-1} a_m^{(l-\theta_m)^2} \cdot x_{t-l,m}}{\sum_{l=0}^{L-1} a_m^{(l-\theta_m)^2}}$$

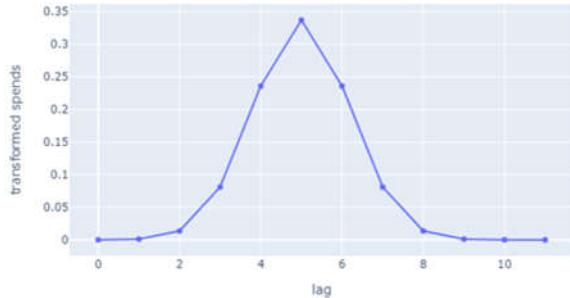
$x_{t,m}$: the spend for media m at time t

a_m : the retain rate for media m ($0 < a_m < 1$)

θ_m : the delay of the peak effect for media m ($0 < \theta_m < L - 1$)

L : maximum duration of effect

Adstock effect on spends with retain_rate = 0.7, delay = 5



- **Rendimientos decrecientes “Hill-Adstock”**

También podríamos plantear la hipótesis de que el impacto de los gastos en las ventas sigue una curva en S. Podemos integrar explícitamente esta hipótesis en nuestro modelo utilizando la transformación de colina o de alcance. El modelo aplica una función de tipo sigmoide para rendimientos decrecientes a la salida de la función “adstock”.

- Efecto “Hill”:

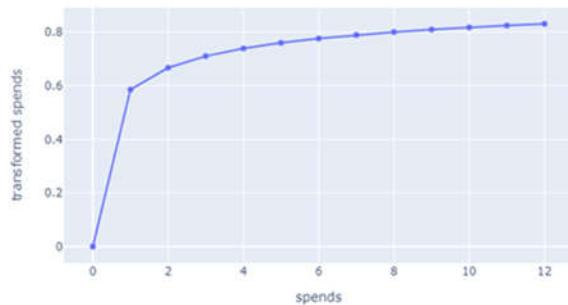
$$Hill(x_{t,m}; K_m, S_m) = \frac{1}{1 + (x_{t,m}/K_m)^{-S_m}}$$

$x_{t,m}$: the spend for media m at time t

S_m : the shape for media m ($S_m > 0$)

K_m : the half saturation for media m ($K_m > 0$)

Hill effect on spends with $S = 0.5, K = 0.5$



- Efecto “Reach”:

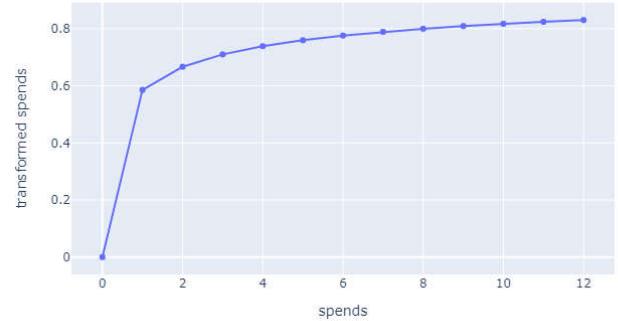
Hill effect on spends with $S = 0.5, K = 0.5$

$$Hill(x_{t,m}; K_m, S_m) = \frac{1}{1 + (x_{t,m}/K_m)^{-S_m}}$$

$x_{t,m}$: the spend for media m at time t

S_m : the shape for media m ($S_m > 0$)

K_m : the half saturation for media m ($K_m > 0$)



3. Estimación de la Tendencia y la Estacionalidad:

La Estacionalidad se recoge de forma nativa, con un parámetro sinusoidal con un patrón repetitivo. También hay una intercepción (una variable de referencia) que es una práctica estándar, así como términos de tendencia y error. Estos se incluyen como variables en el modelo para controlar la estacionalidad.

$$y_t = \tau + trend_t + seast + \sum_m \beta_m z_{t,m} + \sum_c \gamma_c d_{t,c} + \epsilon_t$$

Baseline (Intercept)

$$\tau \sim N(0, 2)$$

Trend

$$trend_t = \beta_{trend} t^{expotrend+0.5}$$

$$\beta_{trend} \sim N(0, 1)$$

$$expotrend \sim Beta(1, 1)$$

Sesonalidad

$$seast = \sum_{k=1}^d (\gamma_{1,k} \cos \frac{2\pi k}{s} + \gamma_{2,k} \sin \frac{2\pi k}{s}), \text{ where } s = 52 \text{ and } d = 2$$

$$\gamma_{1,k}, \gamma_{2,k} \sim N(0, 1)$$

Extra feature (Control variable) effect

$$\beta_c \sim N(0, 1)$$

Noise

$$\epsilon_t \sim N(0, \sigma)$$

$$\sigma \sim gamma(1, 1)$$

Fuente Google



ADTP

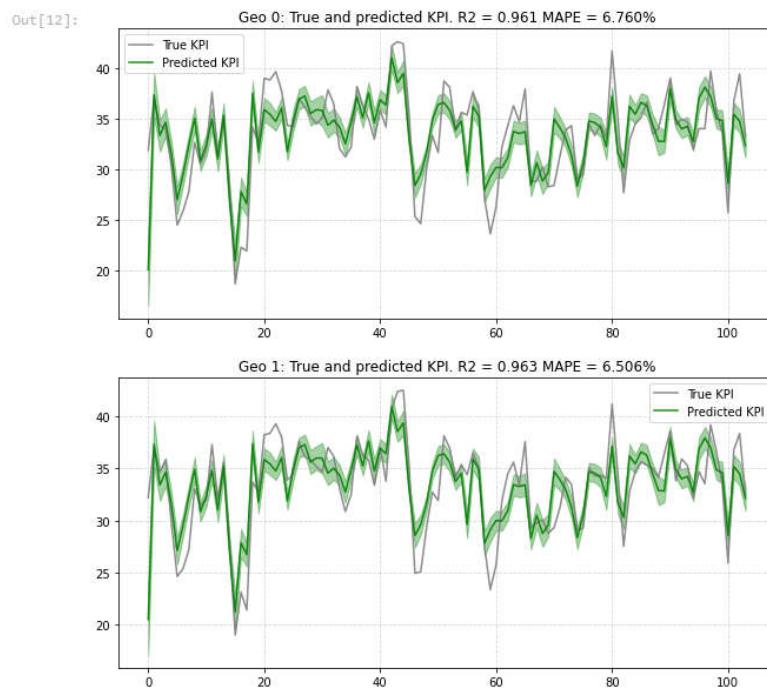
4. Optimización de los presupuestos de marketing:

El modelo se optimiza basándose únicamente en la precisión, medida por el MAPE (Mean Absolute Percentage Error), es decir, el porcentaje medio de error del modelo en un día o una semana. Pero para conseguir que las asignaciones de gastos actuales sean adecuadas y creíbles, el modelo utiliza un marco bayesiano, donde los valores a priori que se establecen actúan como guardarráfiles contra resultados improbables para los coeficientes de marketing. Además, el optimizador del presupuesto de medios de Google sólo recomienda por defecto hasta un 20% de cambios en el gasto, aunque esto es ajustable.

5. Calibración del modelo

Se realiza mediante un algoritmo llamado Cadena Bayesiana de Markov Monte-Carlo que puede usar los valores previos de cada parámetro para informar al modelo de las opiniones sólidas o débiles que tiene sobre la naturaleza de cómo se desempeñará cada canal.

- **Funciones adicionales:** Permite utilizar información procedente de la experiencia en el sector o de modelos anteriores mediante el uso de priors bayesianos. Y construir modelos jerárquicos, con intervalos de credibilidad generalmente más ajustados, utilizando dimensiones de desglose como la geografía (GEOs).
- **Toma de decisiones:** El modelo proporciona tanto dentro de la muestra (datos que alimentamos al modelo para entrenarlo) como fuera de la muestra (datos que el modelo aún no ha visto) gráficos y métricas de precisión, lo que ayuda a diagnosticar rápidamente qué tan confiable es el modelo.



Fuente Google

Lo interesante de la salida del modelo es que admite regiones geográficas de forma nativa. Entonces, por ejemplo, si pasa varias regiones geográficas, genera gráficos para cada una, además de acumular métricas de rendimiento y precisión general, siendo esta una característica excelente del modelo.

4.8.3. Resultados del Modelo Bayesiano en el proyecto:

Los resultados que nos da el modelo bayesiano de LightweightMMM se dividen por estos pasos:

1. Verificación de la calidad de los datos:

Paso 1: Comprobación de la matriz de correlaciones

El modelo calcula la matriz de correlaciones entre todas las características, y entre cada característica y el objetivo. Con el objetivo de seleccionar las correlaciones muy positivas o negativas (con un valor absoluto superior, por ejemplo, a 0,7) y evaluar la posibilidad de suprimir o fusionar los elementos muy correlacionados.

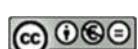
En nuestro caso como ya lo vimos en el análisis exploratorio EDA, las correlaciones que se dan entre estas variables están por debajo del valor recomendado, con lo que se vio que no existía multicolinealidad y todas son adecuadas para el modelo.

Paso 2: Comprobación de Varianzas:

Se realiza el cálculo son inferiores al ude la varianza de cada característica a lo largo del tiempo, identificando las varianzas que son inferiores al umbral “low_variance_threshold” especificado o superiores al umbral” high_variance_threshold” especificado marcadas en **rojo**.

	geo_0
feature_0	0.6182
feature_1	0.5518
feature_2	1.1634
feature_3	0.9629
feature_4	0.5059
feature_5	0.8723
feature_6	1.2885
feature_7	0.8319
feature_8	0.3252
feature_9	0.2964
extra_feature_0	0.1940
extra_feature_1	0.4089
extra_feature_2	0.2572

Los resultados que se obtenidos verifican que las varianzas obtenidas son inferiores al umbral especificado:



Paso 3: Comprobación de las partidas de gasto:

Esta comprobación es muy sencilla. El LMMM utiliza el gasto total de cada canal para establecer la prioridad del coeficiente de contribución a los medios del canal, así como para calcular el ROI más adelante en el análisis. Por lo tanto, el gasto de cada canal debe ser positivo (no negativo ni cero) y lo ideal es que cada canal no represente una fracción insignificante del gasto total.

En nuestro vaso los resultados obtenidos observamos que hay dos medios en los que la inversión está por debajo de los valores recomendados, para tener en consideración en los resultados.

	fraction of spend
feature_0	0.3226
feature_1	0.0338
feature_2	0.1084
feature_3	0.0016
feature_4	0.0522
feature_5	0.0716
feature_6	0.0079
feature_7	0.0434
feature_8	0.0919
feature_9	0.2666

Paso 4: Comprobación de los factores de inflación de la varianza

Aunque la comprobación de la matriz de correlaciones en el paso 1 suele ser suficiente para detectar una multicolinealidad evidente en un conjunto de datos, el factor de inflación de la varianza es técnicamente la mejor métrica para identificar características multicolineales.

Si el número es demasiado alto, se utiliza un umbral de 7, para considerar eliminar o fusionar características con factores de inflación de varianza altos.

Observamos que en nuestro caso las variables de los medios no presentan ninguna multicolinealidad por lo que es recomendable mantener todas las variables para el modelo.

	geo_θ
feature_0	1.1922
feature_1	1.9091
feature_2	2.1417
feature_3	1.5289
feature_4	1.9891
feature_5	2.5334
feature_6	1.4769
feature_7	1.8248
feature_8	1.7648
feature_9	2.3510
extra_feature_0	1.6638
extra_feature_1	1.1050
extra_feature_2	1.9892

Los **resultados del entrenamiento del modelo** son los siguientes:

El entrenamiento del modelo se realiza con los siguientes parámetros:

- medios
- total_costs (un valor por canal)
- objetivo

No se han considerado los parámetros adicionales como:

- extra_features: Otras variables a añadir al modelo.
- grados_estacionalidad: Número de grados a utilizar para la estacionalidad.
- seasonality_frequency: Frecuencia del periodo de tiempo utilizado.
- media_names: Nombres de los canales de medios pasados.
- number_warmup: Número de muestras de calentamiento.
- number_samples: Número de muestras durante el muestreo.
- número_cadenas: Número de cadenas a muestrear.

Visualizamos las distribuciones posteriores de los efectos de los medios, que nos muestra una estimación de los coeficientes de cada canal de medios. Los números altos significan que el canal influyó más en los ingresos.



El eje de abscisas es el coeficiente estimado y el eje de ordenadas indica el grado de confianza del modelo en que el valor del eje de abscisas es el correcto para el efecto de los medios de comunicación.

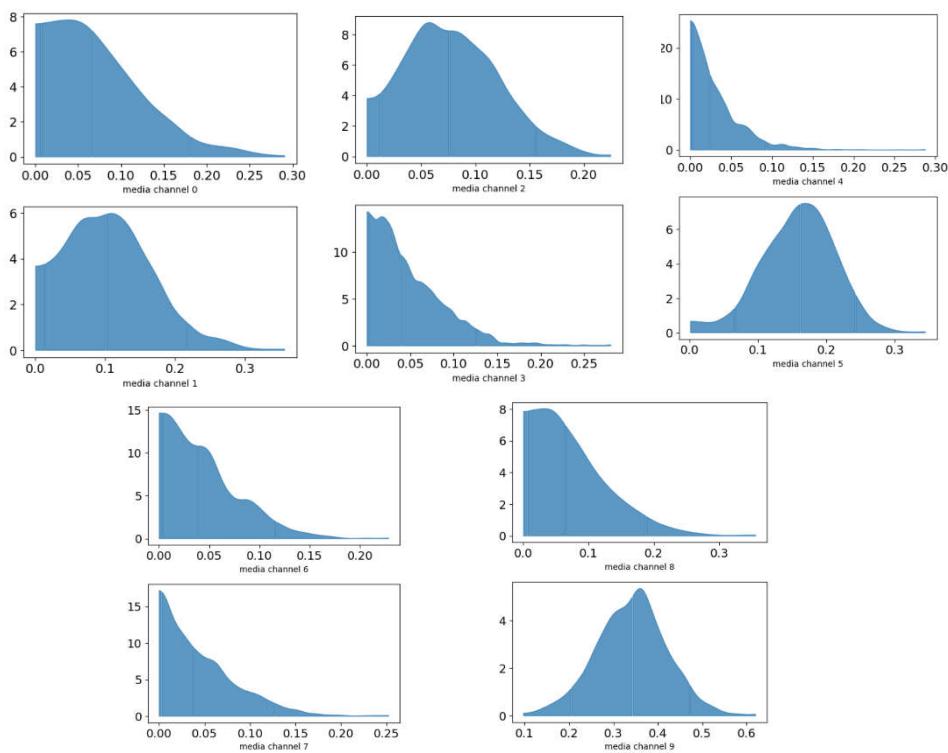


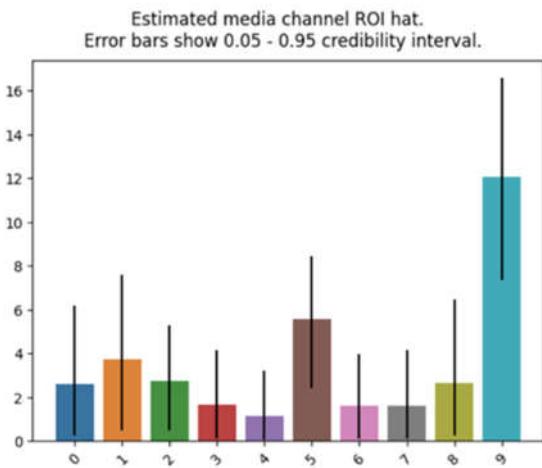
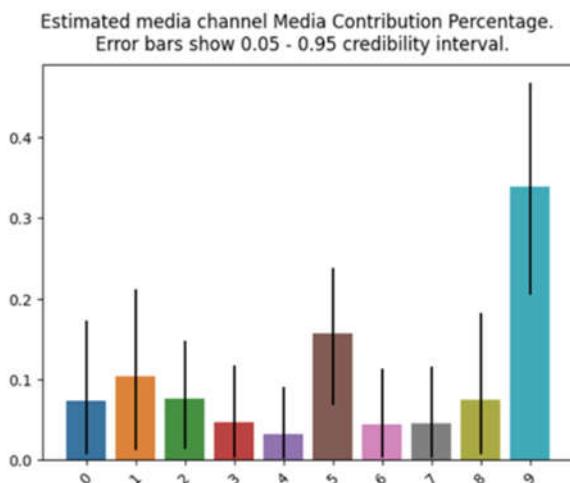
Ilustración 26 - gráfica con distribuciones posteriores de los efectos de los medios

En nuestro caso vemos que el modelo cree que el efecto de estos canales son adecuados, pero sólo conocer este efecto es limitado, ya que un canal puede ser muy eficaz pero muy caro. Así que es necesario complementarlo con el gráfico ROI.

Este gráfico tiene en cuenta no sólo el efecto de los medios, sino también cuánto cuesta conseguir ese efecto. La regla general sería asignar más presupuesto a los canales de alto ROI y menos a los de bajo ROI.

Esto no siempre es posible, ya que un canal puede tener muy buen ROI pero no ser escalable.

El retargeting es un ejemplo: es muy fácil obtener un buen ROI cuando la publicidad se dirige a personas que ya conocen tu producto, pero es un grupo muy limitado.

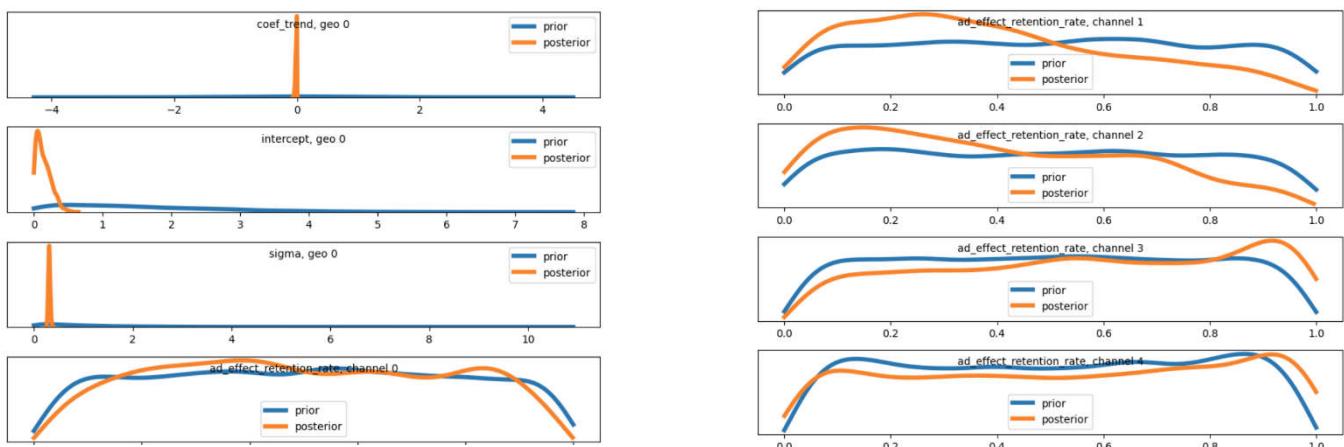


También visualizamos las distribuciones a priori y a posteriori de cada parámetro del modelo a la vez, para comprobar si mantiene la misma distribución. Este gráfico nos muestra de forma intuitiva cómo nuestras creencias sobre las variables cambian con la incorporación de nuevos datos.

Las distribuciones a priori representan nuestras creencias iniciales sobre los parámetros antes de observar los datos. Esto es, nos muestra la distribución a priori de las variables de los canales de medios, mostrando la eficacia del canal basado en los datos históricos.

Una vez que observamos los datos, actualizamos nuestras creencias iniciales utilizando el teorema de Bayes para obtener las distribuciones a posteriori. Estas distribuciones a posteriori representan nuestras creencias actualizadas sobre los parámetros después de haber tenido en cuenta los datos.

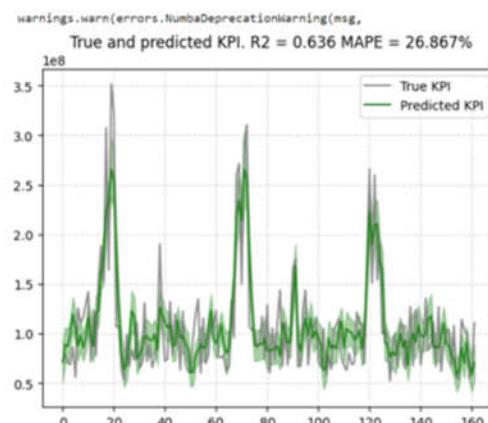
Por lo tanto, un gráfico que muestra tanto las distribuciones a priori como las a posteriori para las variables del MMM nos permite ver visualmente cómo nuestros datos han influido en nuestras creencias sobre estos parámetros. En este caso se ha utilizado un estimador de densidad kernel para suavizar estas distribuciones para facilitar su interpretación.



Comprobamos que si se mantienen la distribuciones a priori y la posteriori, con lo que se puede proceder a la optimización del cálculo de la contribución por canales.

Comprobamos el ajuste del entrenamiento del modelo:
 Estos Kpis nos permite sacar las siguientes conclusiones:

- **Ajuste moderado:** El coeficiente de determinación (R^2) de 0.636 indica que el modelo puede explicar aproximadamente el 63.6% de la variabilidad en los datos de respuesta. Esto sugiere que el modelo tiene un ajuste moderado a los datos, pero aún queda una parte significativa de la variabilidad sin explicar.
- **Error absoluto promedio:** El MAPE (Mean Absolute Percentage Error) de 26.86% indica que, en promedio, las predicciones del modelo tienen un error absoluto del 26.86% en relación con los valores reales. Esto significa que el modelo tiene un error promedio considerable en las predicciones y puede haber cierta falta de precisión.



Podemos visualizar la contribución estimada de los medios de comunicación y la línea de base a lo largo del tiempo:

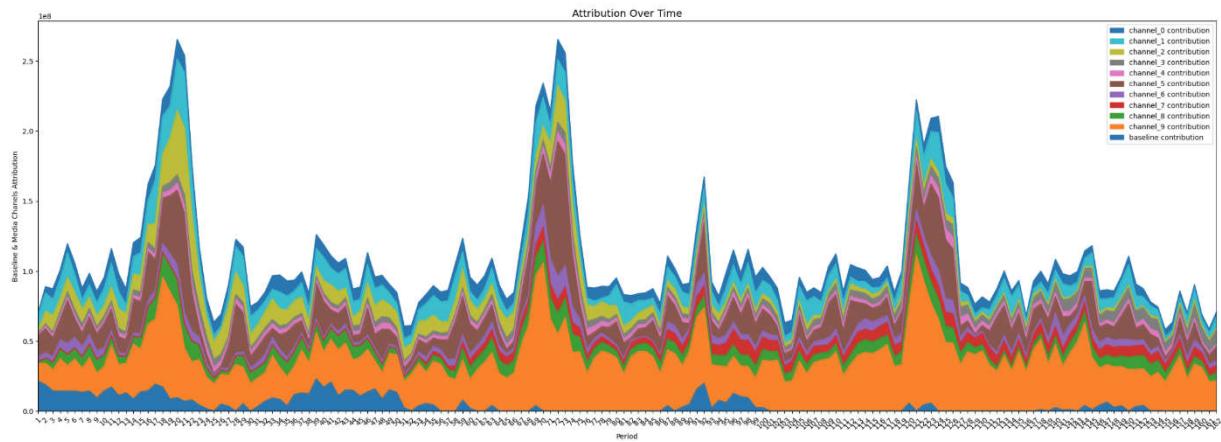


Ilustración 27 - Estimación de la contribución de los medios a las ventas en 6 meses

Como resultado del modelo podemos ver la comparación de la asignación presupuestaria antes y después de la optimización para cada canal:

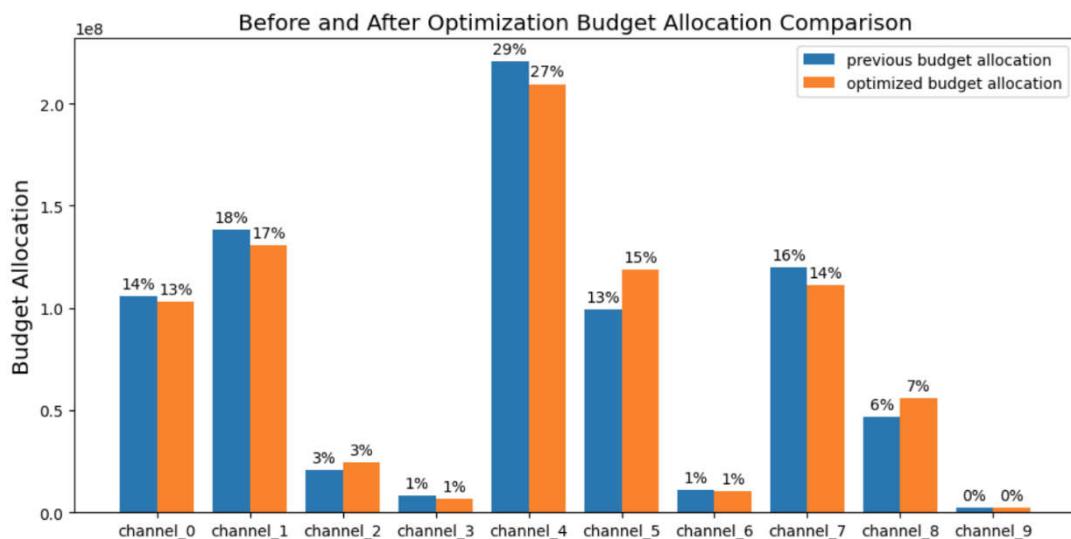


Ilustración 28 - Estimación de la inversión por cada canal

Y la comparación de las variables objetivo previstas antes y después de la optimización:

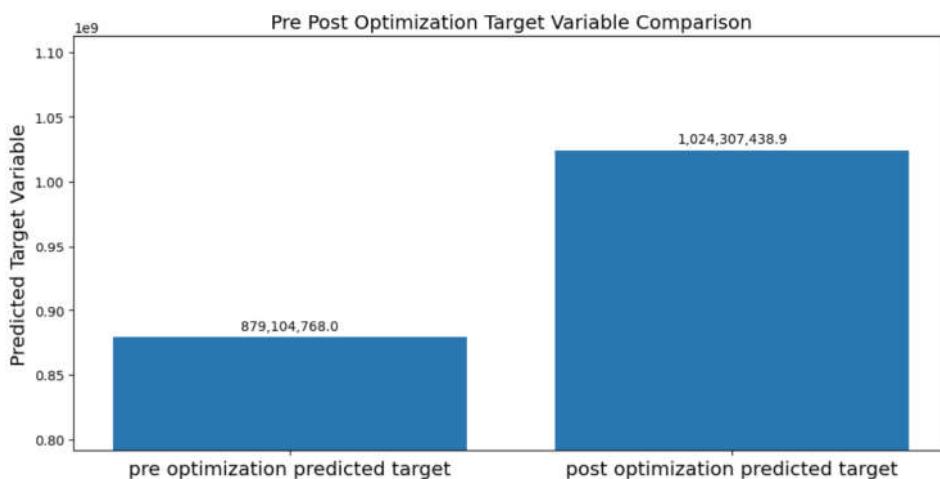


Ilustración 29 - Estimación de la inversión total

4.8.4. Modelo Bayesiano con Stan

Basado en la metodología del modelo LightweightMMM y comparar con un modelo bayesiano diferente, se ha elegido Stan, que es un lenguaje de programación para realizar análisis estadísticos bayesianos y diseñado para implementar una amplia gama de modelos estadísticos.

Stan implementa una forma de Muestreo de Monte Carlo Hamiltoniano, que es un método eficiente para obtener muestras de la distribución a posteriori de los parámetros del modelo, lo que es especialmente útil en contextos donde los modelos son complejos y la distribución a posteriori no puede ser calculada directamente.

Además, Stan también proporciona herramientas para el diagnóstico de la convergencia del muestreo, lo que es crucial para asegurarse de que las estimaciones que obtienes son confiables. También proporciona una sintaxis flexible y fácil de usar para la especificación de modelos, lo que lo hace adecuado para una amplia gama de aplicaciones, con las siguientes características:

- Utiliza un modelo multiplicativo (comentado anteriormente) de tipo logarítmico:

$$\log y = \beta_0 + \beta_{TV} \log x_{TV} + \beta_{SEM} \log x_{SEM} + \dots + \beta_{ctrl} \log x_{ctrl}$$

↗ ↓ ↗ ↗ ↗
 In(Sales) Intercept Media Control

Ajustando un modelo de regresión con coeficientes a priori, utilizando las impresiones (o el gasto) de los canales de comunicación y las variables de control para predecir las ventas, de la siguiente forma:

- Descomponiendo las ventas en función de la contribución de cada canal de medios. La contribución del canal se calcula comparando las ventas originales y las ventas previstas tras la eliminación del canal.
- Para el modelo de rentabilidad decreciente se basa en encontrar la relación entre el gasto y la contribución, esto es ajustando una función de Hill, de modo que puedan calcularse el ROAS y el ROAS marginal.

$$y = \beta \text{Hill}(x; K, S) = \beta \cdot \frac{1}{1 + (x/K)^{-S}}$$

↓ ↓
 Media Adstocked Media
 Contribution Spending

- El modelo se configura utilizando la librería de Stan, una biblioteca de modelización bayesiana, que constituye el núcleo de la forma en que se construye la optimización. Esta basado en un backend C++ que permite un cálculo muy rápido, especialmente en hardware básico como ordenadores de sobremesa y portátiles. Además, el análisis sintáctico automatizado de un programa Stan para convertirlo en un programa C++ y, a continuación, en un programa ejecutable, ofrece a los usuarios la ventaja de ese rendimiento sin limitar su libertad de modelado ni requerir ningún conocimiento del propio C++.

El mayor problema del uso de Stan es que para que pueda compilar modelos fácilmente, se deben instalar un conjunto de herramientas C++ en el ordenador, y unir a un interfaz a ese conjunto de herramientas, siendo una tarea relativamente de usuario avanzado.

- La biblioteca de algoritmos Stan contiene tres algoritmos C++ basados en gradientes que implementan alguna forma de cálculo probabilístico para modelos especificados por programas Stan. Una implementación de alto rendimiento de Hamiltonian Monte Carlo dinámico sirve como centro de trabajo de la Biblioteca de Algoritmos Stan. También incluye un optimizador de memoria limitada Broyden-Fletcher-Goldfarb-Shanno para la estimación puntual y una versión experimental de la inferencia variacional de diferenciación automática o ADVI.

4.8.5. Resultados del Modelo Bayesiano Stan en el proyecto



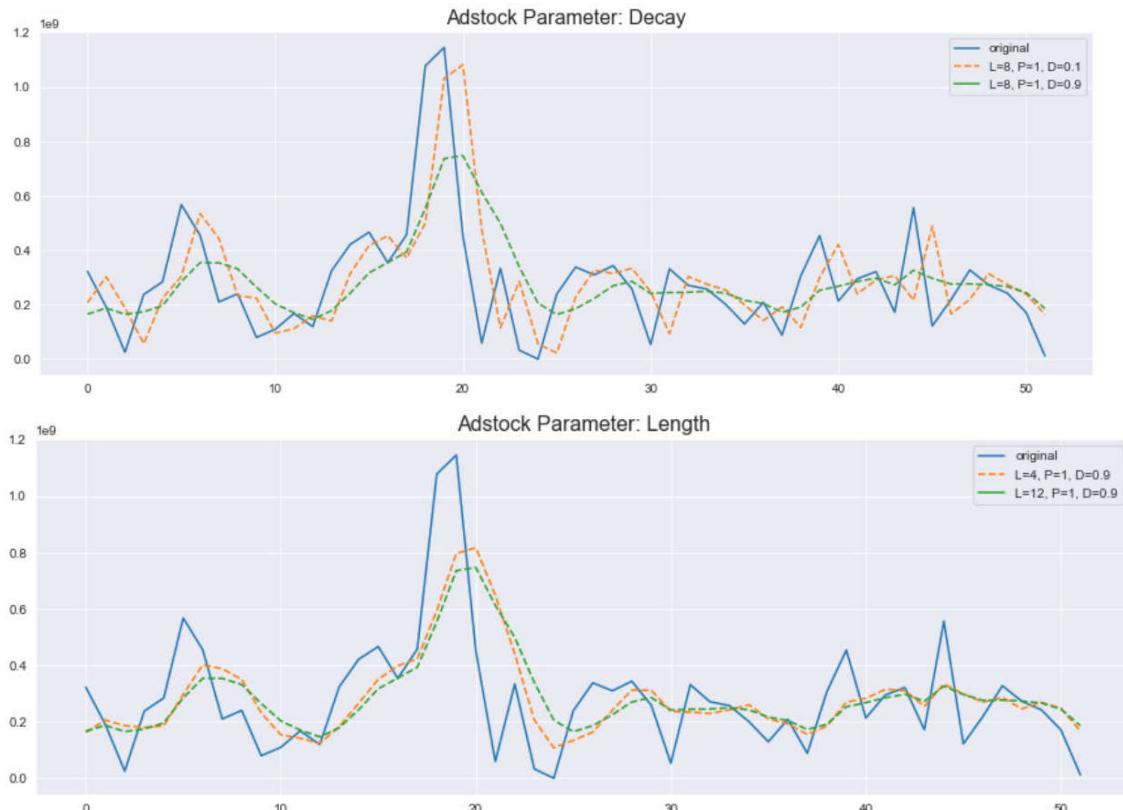
- El modelo de regresión bayesiano utiliza los coeficientes a priori de más impresiones (o el gasto) de los canales de comunicación y las variables de control para predecir las ventas.

Los parámetros de las variables de control son los siguientes:

```
{'dm': {'L': 8, 'P': 0.8147057071636012, 'D': 0.5048365638721349},
 'inst': {'L': 8, 'P': 0.6339321363933637, 'D': 0.40532404247040194},
 'nsp': {'L': 8, 'P': 1.1076944292039324, 'D': 0.4612905130128658},
 'auddig': {'L': 8, 'P': 1.8834110997525702, 'D': 0.5117823761413419},
 'audtr': {'L': 8, 'P': 1.9892680621155827, 'D': 0.5046141055524362},
 'vidtr': {'L': 8, 'P': 0.05520253973872224, 'D': 0.0846136627657064},
 'viddig': {'L': 8, 'P': 1.862571613911107, 'D': 0.5074553132446618},
 'so': {'L': 8, 'P': 1.7027472358912694, 'D': 0.5046386226501091},
 'on': {'L': 8, 'P': 1.4169662215350334, 'D': 0.4907407637366824},
 'em': {'L': 8, 'P': 1.0590065753144235, 'D': 0.44420264450045377},
 'sms': {'L': 8, 'P': 1.8487648735160152, 'D': 0.5090970201714644},
 'aff': {'L': 8, 'P': 0.6018657109295106, 'D': 0.39889023002777724},
 'sem': {'L': 8, 'P': 1.34945185610011, 'D': 0.47875793676213835}}
```

Obteniendo los siguientes resultados en el modelo para:

- Efecto Adstock con decaimiento variable:



- El ajuste de los parámetros nos permite ver el decaimiento más optimo en base al tipo de canal de publicidad, haciendo que su efecto sea más disperso a mayor valor.
-



- Efecto Hill o saturación de medios:
- Se obtiene para cada canal la relación (ajustando la función de Hill) entre el gasto y la contribución, para poder calcular el ROAS y el ROAS marginal, basado en esta fórmula:

$$y = \beta \text{Hill}(x; K, S) = \beta \cdot \frac{1}{1 + (x/K)^{-S}}$$

↓ ↓
 Media Contribution Adstocked Media Spending

- x : gasto en canales de medios de comunicación
- K : media saturación
- S : forma
- Variable objetivo: contribución del canal de medios
- Las variables se centralizan por la media.

Parametros	Prior	Descripción
K_{ec}	beta(2, 2)	Half saturation point
S_{slope}	gamma(3, 1)	Shape
β_{beta_hill}	half normal(0, 1)	Coefficient
Residual noise_var	inverse gamma(0.05, 0.05 * 0.01)	Residual variance

- Observamos los resultados del modelo 'Hill' del entrenamiento para todos los canales de medios, observando si el ajuste es el adecuado en cada una de las disminuciones para los retornos adicionales.



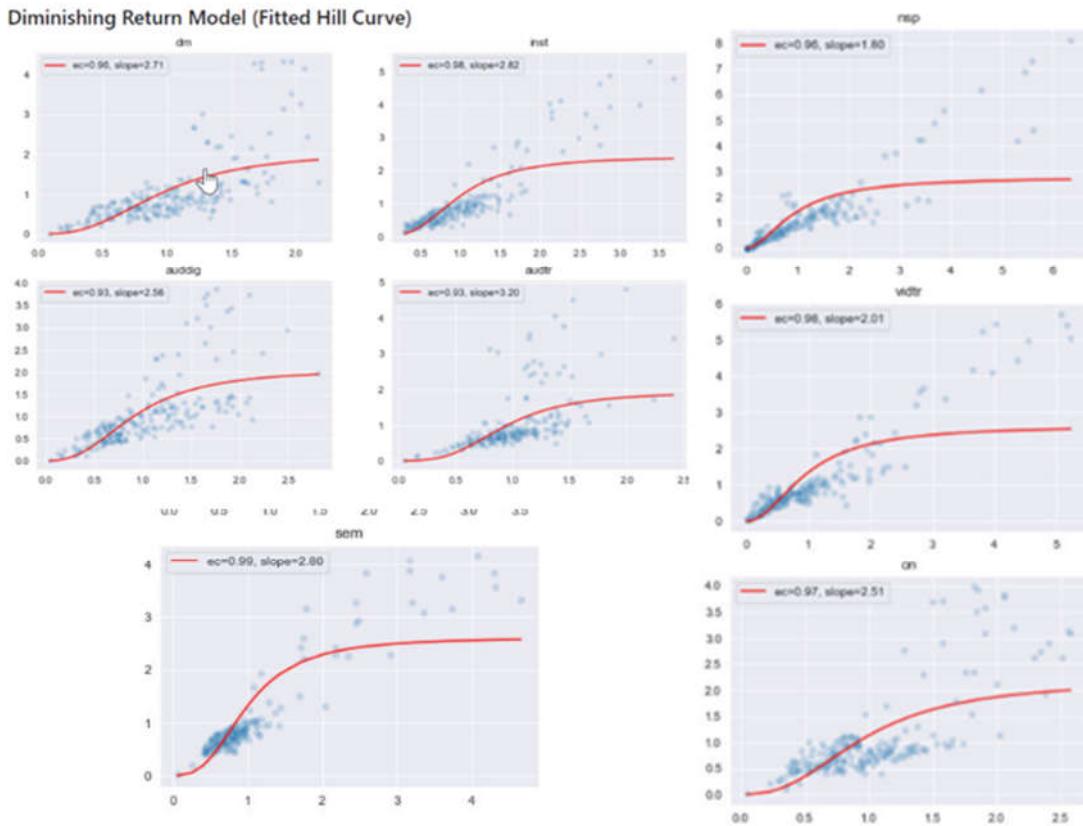


Ilustración 30 - Gráficas con los efectos de marketing

- Observamos que el modelo en la función Hill de cada canal, representa adecuadamente la saturación de los medios, permitiendo determinar el gasto óptimo en marketing en base al impacto en las ventas, obteniendo el punto de equilibrio donde cada gasto adicional impacta cada vez menos.

Con la optimización del modelo se obtiene el ROAS global y el ROAS semanal, siendo:

- ROAS global = contribución total a los medios / gasto total en medios
- ROAS semanal = contribución semanal a los medios / gasto semanal en medios

red line: mean, green line: median

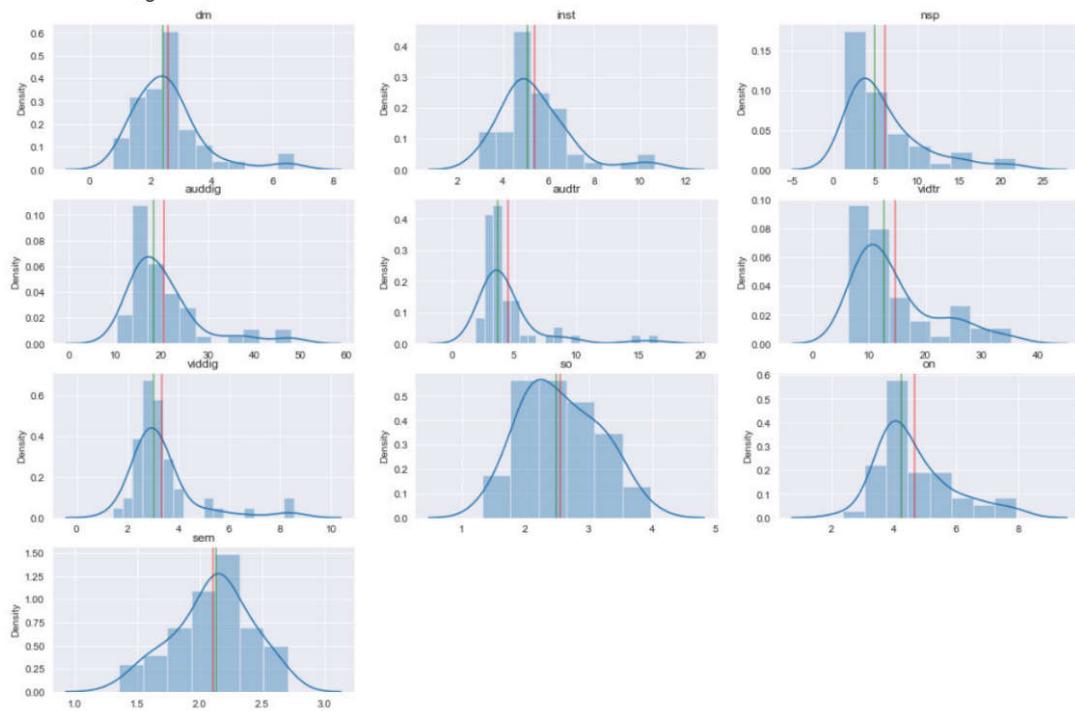


Ilustración 31 - función Hill de cada canal

- Una vez optimizado el modelo del presupuesto de marketing se obtienen los siguientes resultados:

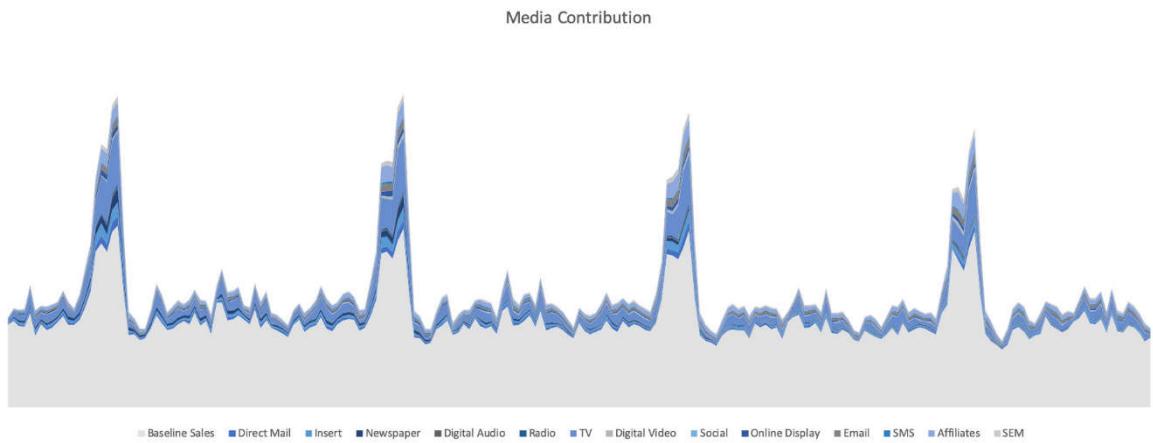


Ilustración 32 - Estimación de la contribución para cada canal de los medios en las ventas

Podemos observar que aproximadamente el 80% de las ventas no se han producido por el efecto de las acciones de marketing, siendo su contribución del 20%.

Siendo los canales de marketing que han tenido mayor contribución a este 20% son la TV y los afiliados y las campañas de SEM.

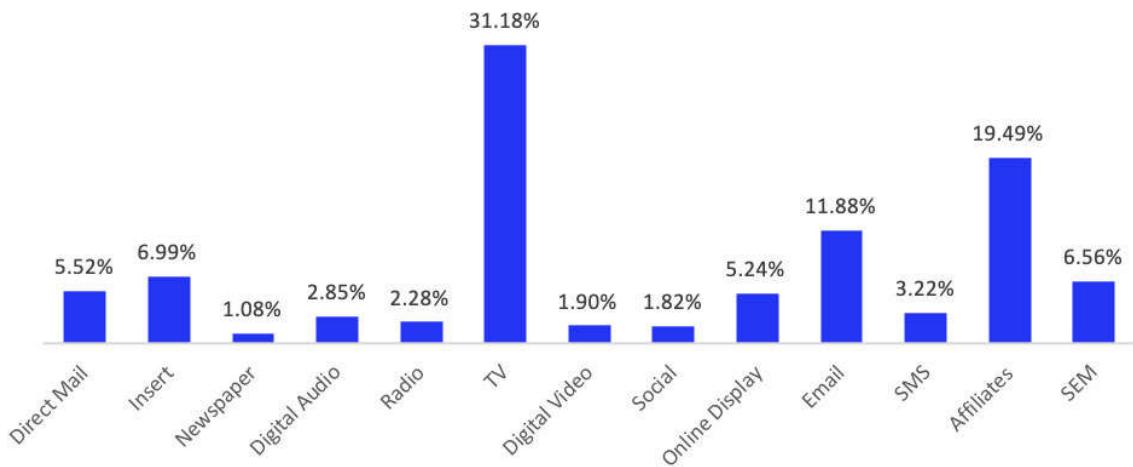


Ilustración 33 - Contribución Canales de Marketing

Y si medimos la rentabilidad de los anuncios (ROAS), observamos que también la TV muestra una alta rentabilidad o canales como la publicidad online. Pero en el caso del SEM, que tenía una alta contribución, observamos que su contribución a la rentabilidad es muy baja:

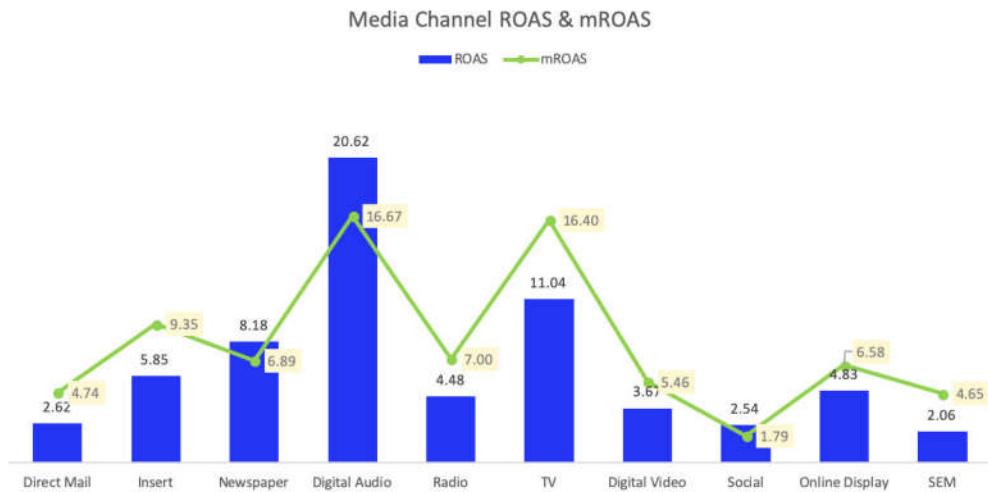


Ilustración 34 - Contribución Canales de Marketing con el ROAS

Y finalmente podemos observar cómo es la contribución de los canales a las ventas y su rentabilidad, siendo el canal de marketing más importante con diferencia el de TV. Y el resto de canales en menor medida por su contribución y rentabilidad:

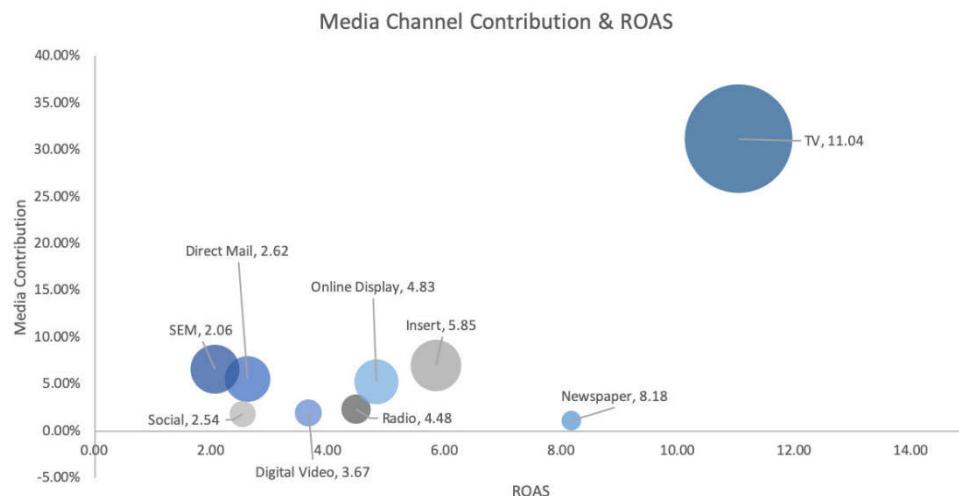


Ilustración 35 - Contribución por canal y ROAS

4.8.6. Comparativa de los Modelos

Se han realizado en total cinco modelos de MMM:

- Modelo de Regresión lineal Multivariante
- Modelo Regresión Multiplicativo
- Modelo Robyn
- Modelo Bayesiano LightweightMMM
- Modelo Bayesiano Stan

Estas diferentes metodologías y enfoques nos permiten entender como impactan las variables en el ROI del mix de marketing, pudiendo determinar qué modelo se ajusta más a la realidad compleja de las acciones de marketing y así poder utilizarlo en el primer prototipo del simulador, siendo el objetivo de este proyecto.

4.8.6.1. Comparativa de los modelos tradicionales:

Primero comparamos los modelos tradicionales, el de Regresión lineal Multivariante y Multiplicativo, que no incluyen los efectos del marketing, la estacionalidad y la tendencia, así como otras variables externas. Hemos evaluado los modelos de Regresión lineal multivariante y multiplicativa, con los siguientes resultados:

	MAPE	MSE	RMSE	MAE	R^2
REGRESIÓN LINEAL	0,2935	1.38e+15	3.72e+07	2.97e+07	0.214
MULPLICATIVO	0,2819	1,11E+21	3,33E+13	2,71E+13	0.625

Observamos que el modelo Multiplicativo muestra un MAPE ligeramente más bajo que el modelo de Regresión Lineal, lo cual indica una menor proporción de error absoluto en las predicciones.

El modelo Multiplicativo tiene un MSE y RMSE significativamente más altos que el modelo de Regresión Lineal, lo que indica una mayor dispersión y varianza en los errores entre las

predicciones y los valores reales. El MAE del modelo Multiplicativo es considerablemente mayor que el del modelo de Regresión Lineal, lo que indica una mayor magnitud promedio del error absoluto entre las predicciones y los valores reales. Y el modelo Multiplicativo muestra un R² más alto que el modelo de Regresión Lineal, lo que indica que es capaz de explicar una mayor proporción de la varianza en la variable objetivo en comparación con el modelo de Regresión Lineal.

En general, el modelo Multiplicativo parece tener un mejor rendimiento en términos de MAPE y R², pero presenta valores más altos de MSE, RMSE y MAE.

Como conclusión vemos que no son modelos óptimos para nuestro objetivo de usarlos en el prototipo del simulador de Marketing Mix, aunque son relativamente fáciles de entender e interpretar, son rápidos de calcular y no requieren mucha capacidad de procesamiento. No son realmente eficientes en la interpretación de las situaciones de incertidumbre y factores complejos del marketing, pues asumen una relación lineal, lo cual puede no ser realista en muchas situaciones. Y no pueden capturar fácilmente efectos no lineales o interacciones entre variables a menos que se especifiquen explícitamente.

Por todo lo anterior, no se consideran estos modelos como válidos en la implementación del simulador del proyecto.

4.8.6.2. Comparativa de los modelos de nueva generación:

Los modelos de nueva generación evaluados en este proyecto han sido, el modelo Robyn, los bayesianos Lightweight MMM y Stan, que son modelos complejos incluyendo los efectos del marketing, la estacionalidad y la tendencia, así como otras variables externas.

Estos modelos utilizan diferentes lenguajes y algoritmos, así en el caso de Robyn MMM utiliza el lenguaje R y Nevergrad Python de Facebook, mientras que los otros dos modelos utilizan Numpyro o Stan y Python.

R es un lenguaje popular para la computación estadística y el análisis de datos, y proporciona una amplia gama de herramientas para el modelado lineal y no lineal, cálculo, pruebas, visualización y análisis. En cambio, Nevergrad es una biblioteca de Python para la optimización que utiliza técnicas modernas de aprendizaje automático, como la optimización bayesiana y los algoritmos genéticos, para encontrar los mejores parámetros para un modelo dado.

Además, Robyn MMM utiliza una técnica denominada optimización de hiper parámetros, que consiste en ejecutar el modelo miles de veces con diferentes parámetros para encontrar la mejor combinación de parámetros que produzca los resultados más precisos.

Aunque Python es cada vez más popular debido a su disponibilidad de código abierto, R sigue siendo más valorado para estadísticos y analistas de datos debido a sus potentes herramientas y funciones para el análisis y la visualización de datos.

En el caso de Lightweight MMM utiliza Numpyro y JAX para la programación probabilística, lo que le permite procesar los datos más rápidamente que Robyn. En general, Lightweight MMM es una solución más rápida y ligera que Robyn, por lo que es un punto a favor para su implementación como modelos del simulador de marketing mix en el proyecto.

En términos de resultados, los tres modelos proporcionan estimaciones del impacto de las actividades de marketing en las ventas, así como estimaciones del retorno de la inversión (ROI) de cada actividad. Se pueden comparar la eficacia de los distintos canales de marketing y realizar simulaciones del impacto de los cambios en el marketing mix.

En el caso de Robyn, ofrece mayor detalle en los resultados y proporciona información sobre la importancia relativa de los diferentes canales de marketing, con simulaciones basadas en diferentes escenarios.

Pero la diferencia clave entre estas soluciones es el enfoque de modelado, así Lightweight MMM y Stan utilizan un enfoque de modelado bayesiano, que se basa en probabilidades y permite una mayor flexibilidad a la hora de modelar relaciones complejas. Por el contrario, Robyn MMM, utiliza un enfoque de gradient boosting, que es una técnica de aprendizaje automático que permite relaciones más complejas y no lineales entre variables.

Con los resultados analizados se observa que el modelo Lightweight MMM permite una mayor flexibilidad, ya que incorpora mejor las variables complejas del marketing, por el modelo bayesiano que permite estimaciones previas de la elasticidad y actualizar las creencias previas con nuevos datos. Por lo que el modelo puede adaptarse mejor y aprender con el tiempo, es un enfoque más potente y flexible, ya que puede manejar una gama más amplia de tipos de datos, incorporar conocimientos previos y proporcionar un marco probabilístico para modelar los datos de marketing.

Los tres modelos ofrecen una función de optimización que ayuda a asignar la inversión en medios, con el fin de maximizar el ROI, dado que su enfoque de optimización es diferente, los resultados son ligeramente diferentes, sin grandes diferencias.

El punto más desfavorable para los modelos Lightweight MMM y Stan, es que no consideran ningún análisis de la tendencia y la estacionalidad de las ventas como serie temporal. Mientras que Robyn usa Prophet, ya comentado anteriormente, que permite una descomposición entre la estacionalidad, los eventos y los días de la semana.

Con esta valoración, la comparativa indica que los tres modelos de nueva generación son adecuados para la creación del prototipo de simulador de MMM, pero al ser este una primera versión de aprendizaje y validación de los modelos con la realidad del marketing, se ha seleccionado el modelo Lightweight MMM, tanto por su forma de aplicar el modelo bayesiano, como por su análisis explicativo y ser más ligero y rápido en su procesamiento.

5. Desarrollo Aplicación Web: “Nebula Navigator”

Se ha seleccionado como nombre de esta aplicación web del simulador de marketing mix el nombre de “Nebula Navigator”, que transmiten las siguientes ideas:

Nebula: Una nebulosa es una formación de polvo y gas en el espacio. Las nebulosas son conocidas por su belleza y complejidad, y muchas de las imágenes más icónicas del espacio son de nebulosas. En este contexto, "nebula" puede transmitir la idea de explorar el incierto y desconocido universo del marketing mix.

Navigator: Un navegante es alguien que encuentra el camino, especialmente a través de rutas complicadas o desconocidas. En este contexto, "navigator" puede transmitir la idea de



guiar a los usuarios a través del complejo mundo de los datos para ayudarles a encontrar la información o las respuestas que buscan.

Por lo tanto, el nombre "Nebula Navigator" podría transmitir la idea de una herramienta que ayuda a los usuarios a explorar y encontrar su camino a través de la incertidumbre y complejo universo del marketing mix.

La aplicación permite que cualquier usuario pueda realizar una simulación de su modelo de marketing mix, solamente subiendo un data set y siguiendo los pasos descritos en el menú de navegación.

Para ello se ha definido la siguiente arquitectura:

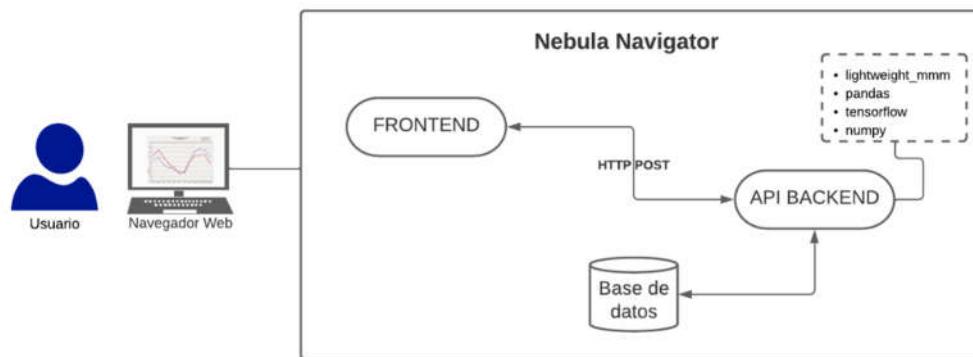


Ilustración 36 - Arquitectura aplicación

Para el desarrollo de los procesos se ha establecido el siguiente diagrama de flujos:

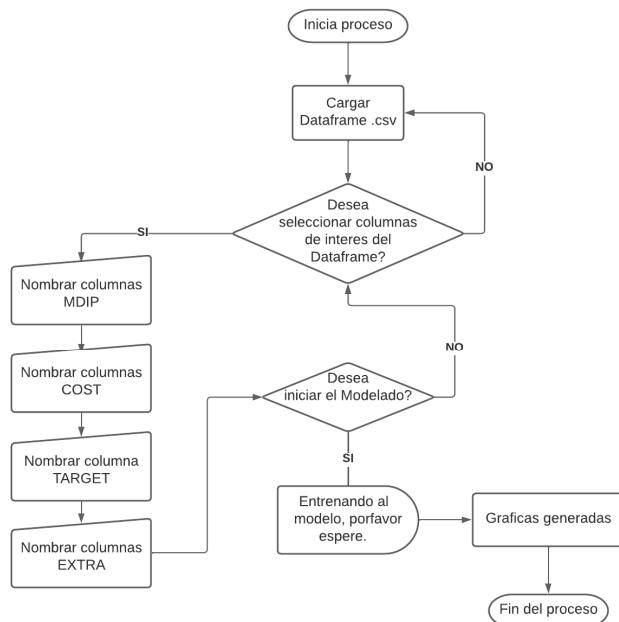


Diagrama de flujo, Uso del sistema

La arquitectura de nuestra aplicación del simulador de Marketing Mix Modeling (MMM) se ha diseñado con el objetivo de optimizar la interacción del usuario, así como asegurar la eficiencia y robustez de los cálculos en el backend.

En el **front-end**, se ha optado por un stack de tecnología basado en HTML, CSS, JavaScript y Bootstrap. HTML y CSS forman la base de la estructura y el diseño de la aplicación, mientras que JavaScript se utiliza para proporcionar funcionalidades interactivas. Bootstrap, un marco de diseño de código abierto, se utiliza para garantizar una presentación consistente y adaptable en una variedad de dispositivos y tamaños de pantalla.

HTML (HyperText Markup Language): Es el lenguaje estándar para la creación de páginas web. HTML proporciona la estructura básica de las páginas, permitiéndonos definir elementos como encabezados, párrafos, listas, enlaces, imágenes y otros bloques de construcción de una página web.

CSS (Cascading Style Sheets): CSS es un lenguaje de hojas de estilo utilizado para describir la apariencia de los elementos HTML. Permite controlar aspectos como colores, fuentes, diseño de la página, transiciones y animaciones. Con CSS, podemos crear diseños coherentes y atractivos en toda la aplicación.

JavaScript: Este es un lenguaje de programación del lado del cliente que permite añadir interactividad y comportamiento dinámico a las páginas web. JavaScript nos permite crear funciones como formularios interactivos, animaciones, actualizaciones en tiempo real, y mucho más.

Bootstrap: Es un popular marco de trabajo de código abierto para el diseño de páginas web y aplicaciones móviles. Bootstrap proporciona una colección de plantillas de diseño basadas en HTML y CSS, así como plugins de JavaScript opcionales. Con Bootstrap, podemos crear interfaces de usuario responsivas y compatibles con múltiples navegadores de manera eficiente.

El **back-end** de la aplicación se ha desarrollado utilizando Django, un marco de trabajo de alto nivel para Python que fomenta el desarrollo rápido y el diseño limpio y pragmático. Además de permitir una integración fluida con nuestra base de código de Python, Django proporciona numerosas utilidades para manejar aspectos comunes del desarrollo web, como el enrutamiento de URL, la seguridad y la gestión de bases de datos.

Django: Ofrece una serie de herramientas listas para usar, como un sistema de plantillas, un mapeador objeto-relacional (ORM) para manejar las bases de datos, y un sistema de enrutamiento de URL. Además, Django fomenta el reusó y la "no repetición" de componentes, lo que permite escribir menos código y minimizar la duplicación.

Python: Este es un lenguaje de programación de alto nivel conocido por su código limpio y legible. Python es muy versátil y se utiliza en una variedad de dominios, desde el desarrollo web hasta la ciencia de datos y el aprendizaje automático. En el caso de nuestro simulador de MMM, Python se utiliza para implementar la lógica de negocio y los modelos de aprendizaje automático.

Finalmente, el motor de análisis de la aplicación utiliza Python y TensorFlow, un marco de trabajo de aprendizaje automático de código abierto desarrollado por Google. TensorFlow nos permite construir y entrenar modelos de aprendizaje automático de alta eficiencia, lo cual es esencial para las funcionalidades de simulación de MMM de nuestra aplicación.

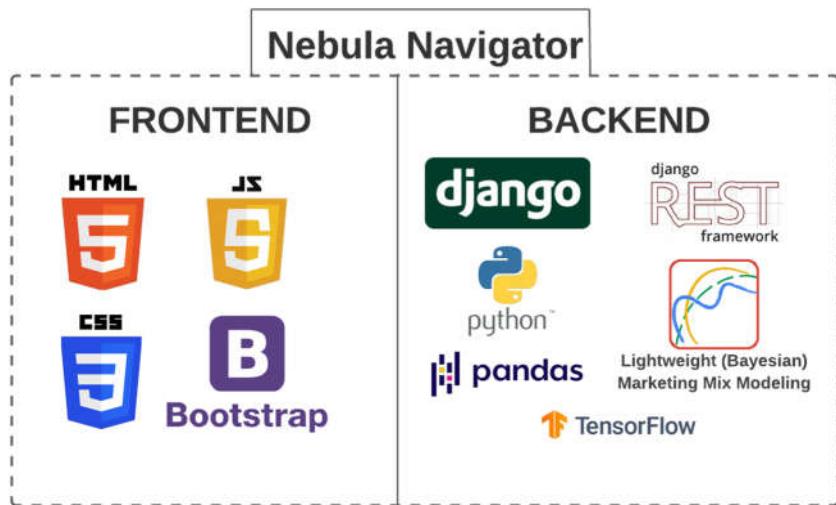


Ilustración 37 - Tecnologías de la aplicación

5.1. Aplicación Web: “Nebula Navigator”

La aplicación “nebula navigator”, nace con el propósito de ser la herramienta para que los profesionales de marketing puedan realizar diferentes simulaciones que les ayuden a cumplir sus objetivos. Por lo que este prototipo esta diseñado para poder escalar en el futuro con otros modelos de marketing mix, por ejemplo incorporar el de Robyn u otros. Y no ser una herramienta exclusiva de MMM, sino poder incorporar otras soluciones como modelos de Customer Lifetime Value (CLV), análisis de sentimiento o un sistema de recomendación, entre otros.

La aplicación se compone de las siguientes partes que permiten la interacción con los usuarios y su administración:

- **Home:**

Se ha creado un sistema de login para que cada usuario tenga su propia cuenta con sus modelos, así se garantiza la privacidad y seguridad de la información:

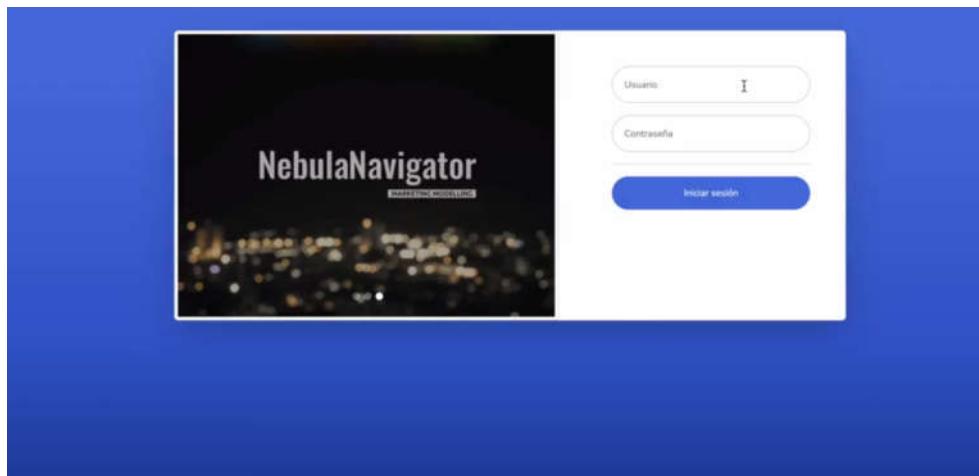


Ilustración 38 - Pantalla de Login

En esta versión, los usuarios se generan desde el área de administración, para evitar su uso abierto y que sea exclusivamente mediante invitación.

- **Menú principal:**

Una vez que el usuario ha introducido sus credenciales, se abre el menú principal, donde se pueden visualizar los elementos de la aplicación, se muestran los elementos necesarios para la simulación de los modelos:

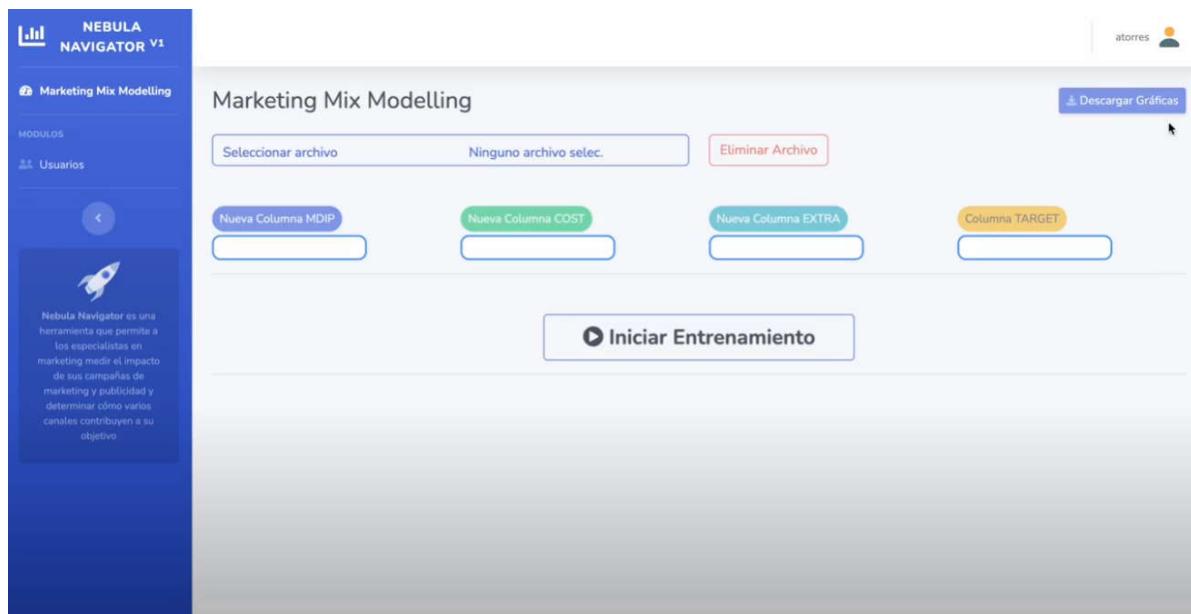


Ilustración 39 - Home de la aplicación

En el banner de la izquierda aparece el modelo elegido, en este caso solo hay uno que es el de MMM, pero en esta área aparecerán los modelos de marketing disponibles. Si el usuario que se ha logado tiene perfil de administrador, le aparecería como en esta imagen el botón de “usuarios”, que haciendo clic aparece el área de gestión de usuarios:

ID	USERNAME	NOMBRE	APELLIDOS	EMAIL	ADMIN	ACCIONES
1	alejojr	Alejandro	Caicedo Palacios	alejo@gmail.com	true	<button>Edit</button> <button>Delete</button>
2	atorres	Alberto	de Torres Pachón	atorres@gmail.com	false	<button>Edit</button> <button>Delete</button>
3	jrcalcedo	Junior	Caicedo Palacios	jrcalcedo@gmail.com	false	<button>Edit</button> <button>Delete</button>
4	jramirez	Juan	Ramirez	jramirez@gmail.com	true	<button>Edit</button> <button>Delete</button>
5	jsalavertt	Javier	Salavert	jsalavert@gmail.com	true	<button>Edit</button> <button>Delete</button>
6	cgarciia	Camila	Garciaaa	cgarcia@gmail.com	false	<button>Edit</button> <button>Delete</button>

Ilustración 40 - Pantalla gestión usuarios

En esta área se encuentra las gestión de altas y mantenimiento de usuarios. Como funcionalidad tiene el alta de nuevos usuarios, editar, dar rol de usuarios o admin, eliminar usuarios y generar las contraseñas:

ID	USERNAME	FIRST NAME	LAST NAME	EMAIL	ADMIN	ACTIONS
1	alejorj				true	<button>Edit</button> <button>Delete</button>
2	atorres				false	<button>Edit</button> <button>Delete</button>
3	jraicedo				false	<button>Edit</button> <button>Delete</button>
4	jramirez				true	<button>Edit</button> <button>Delete</button>
5	jsalavertt	Javier	Salavert	jsalavert@gmail.com	true	<button>Edit</button> <button>Delete</button>
6	cgarcia	Camila	Garcisaa	cgarcia@gmail.com	false	<button>Edit</button> <button>Delete</button>

Ilustración 41 - Gestión de usuarios alta

En la pantalla principal, para comenzar la simulación el primer paso es el de subir un data set con las variables de marketing:

Ilustración 42 - Carga datos y variables

Para ello se debe hacer clic en la pestaña de “seleccionar archivo” y subir el archivo:

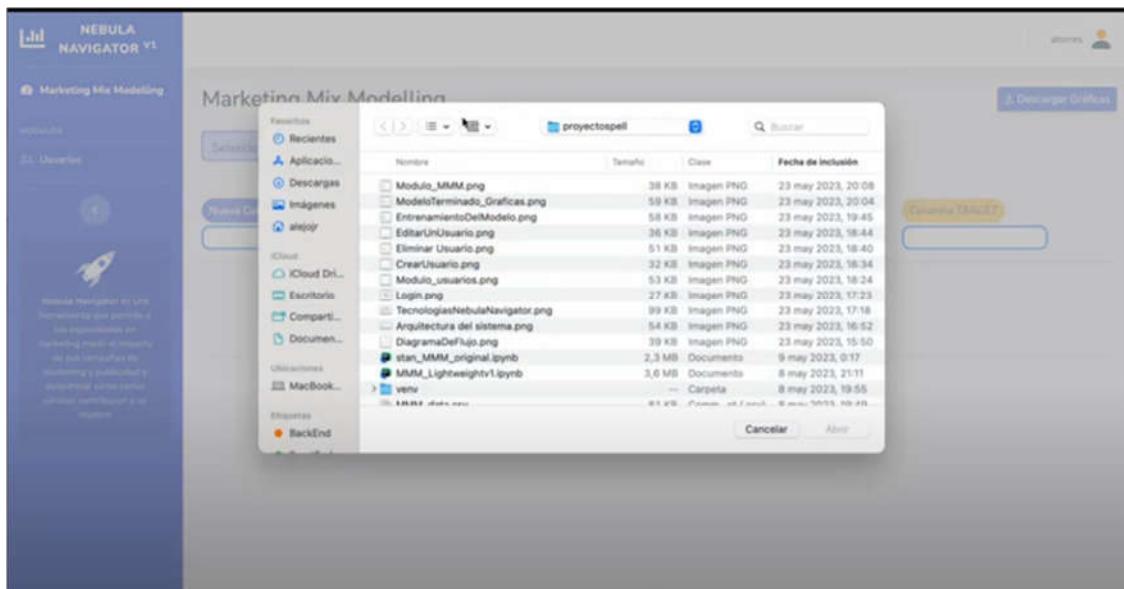


Ilustración 43 - Carga Datos

A continuación se deben incluir los nombres de las variables, según la tipología. Estas corresponden a las variables necesarias para un modelo bayesiano MMM y son las correspondientes a las impresiones de medios, las inversiones por cada medio, otro tipo de variables, como pueden ser las variables macroeconómicas,... y la variable target que se quiere predecir, como las ventas. La aplicación permite introducir hasta 10 diferentes variables para cada caso, excepto para la variable de ventas , que solo permite una. Es imprescindible que cuando se introducen los nombres de las variables estas deben estar escritas de la misma forma, sino saldría un error en el modelo. Como se pude ver en esta imagen:

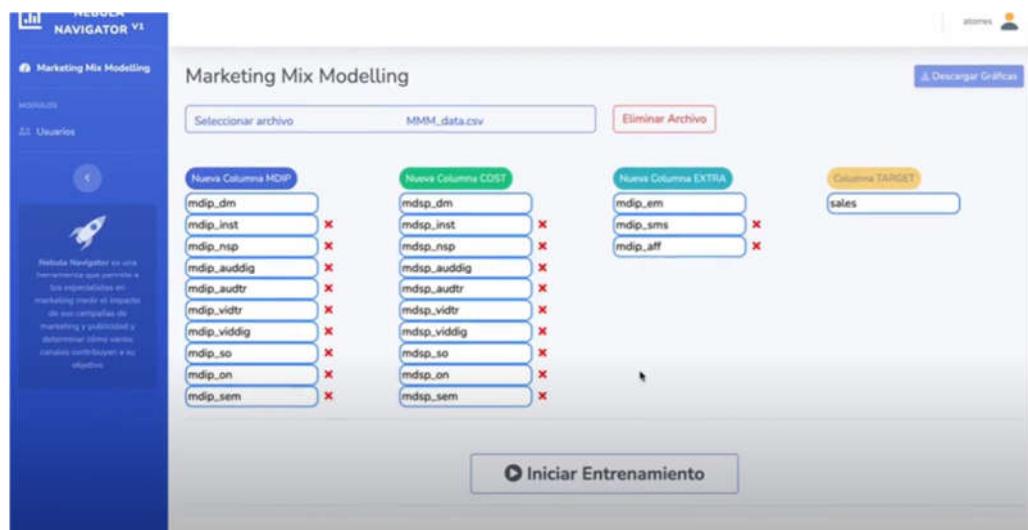


Ilustración 44 - Carga Variables

Una vez introducidas las variables ya se está en disposición de ejecutar el modelo, para ello se hace clic en la pestaña de “iniciar entrenamiento”, y el modelo se empieza a ejecutar:

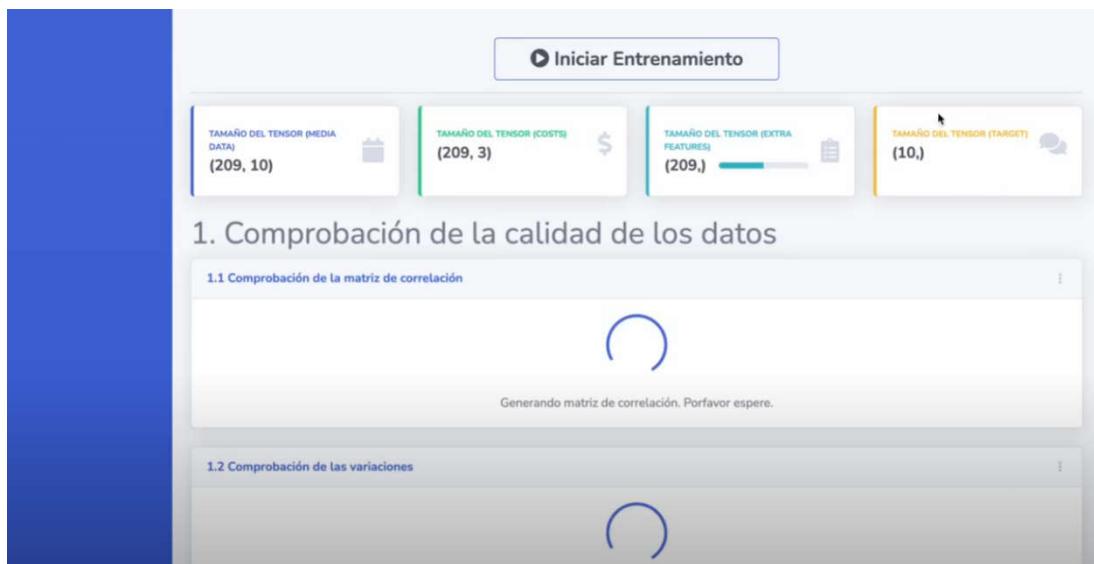


Ilustración 45 - Modelo en ejecución

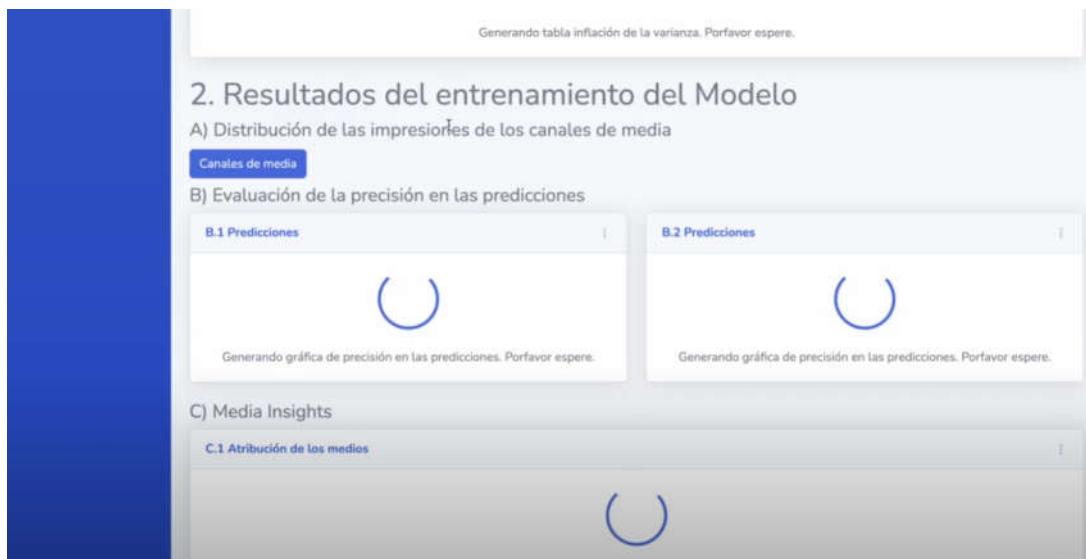


Ilustración 46 - Modelo en ejecución

La ejecución suele durar unos cinco minutos, con la actual cloud instalada, si esta aplicación se pusiera en producción, con un cloud adecuado el tiempo de ejecución sería menor de dos minutos.

Una vez que ha terminado la ejecución se pueden visualizar los resultados y además se ha dispuesto de la posibilidad de descarga de los informes y gráficas en formato pdf, con un botón en la página principal “ Descargar Gráficas”, que descarga todos resultados un archivo, o también se pueden descargar igualmente en formato pdf, cualquiera de las gráficas que se han obtenido, para ello se ha dispuesto un botón con tres puntos donde se habilita la función de descarga.

Una vez que el modelo se ha ejecutado se tiene acceso a los gráficos generados, estos son: El modelo hace una verificación de la calidad de los datos y genera una matriz de correlaciones con las variables para poder comprobar si se da el efecto de multicolinealidad y se deberían eliminar alguna variable.



Ilustración 47 - Verificación calidad datos

Se genera una lista con las diferentes varianzas de las variables para evaluar su variabilidad.

The figure shows a user interface for training a machine learning model. It displays two tables. The first table, titled '1.2 Comprobación de las variaciones', lists the standard deviation (geo_0) for various features: feature_0 (0.6182), feature_1 (0.5518), feature_2 (1.1634), feature_3 (0.9629), feature_4 (0.5059), feature_5 (0.8723), feature_6 (1.2885), feature_7 (0.8319), feature_8 (0.3252), feature_9 (0.2964), extra_feature_0 (0.1940), extra_feature_1 (0.4089), and extra_feature_2 (0.2572). The second table, titled '1.3 Comprobación de las fracciones de gasto', lists the fraction of spend for each feature: feature_0 (0.3226), feature_1 (0.0338), feature_2 (0.1084), and feature_3 (0.0016).

	geo_0
feature_0	0.6182
feature_1	0.5518
feature_2	1.1634
feature_3	0.9629
feature_4	0.5059
feature_5	0.8723
feature_6	1.2885
feature_7	0.8319
feature_8	0.3252
feature_9	0.2964
extra_feature_0	0.1940
extra_feature_1	0.4089
extra_feature_2	0.2572

	fraction of spend
feature_0	0.3226
feature_1	0.0338
feature_2	0.1084
feature_3	0.0016

Ilustración 48 - Comprobación Variaciones

Resultados del modelo después del entrenamiento, aquí se generan varias gráficas con varios ratios, como el r^2 o mape (media average precision error). También se grafica la curva de ventas predicha con la real.



Ilustración 49 - - Distribución impresiones de canales

Se generan todos los gráficos de las variables, tanto de impresiones, como de los gastos de las variables de los medios de marketing con el resultado de las distribuciones bayesianas, comparando las a priori y las posteriores para mostrar de forma intuitiva cómo nuestras creencias sobre las variables cambian con la incorporación de nuevos datos.

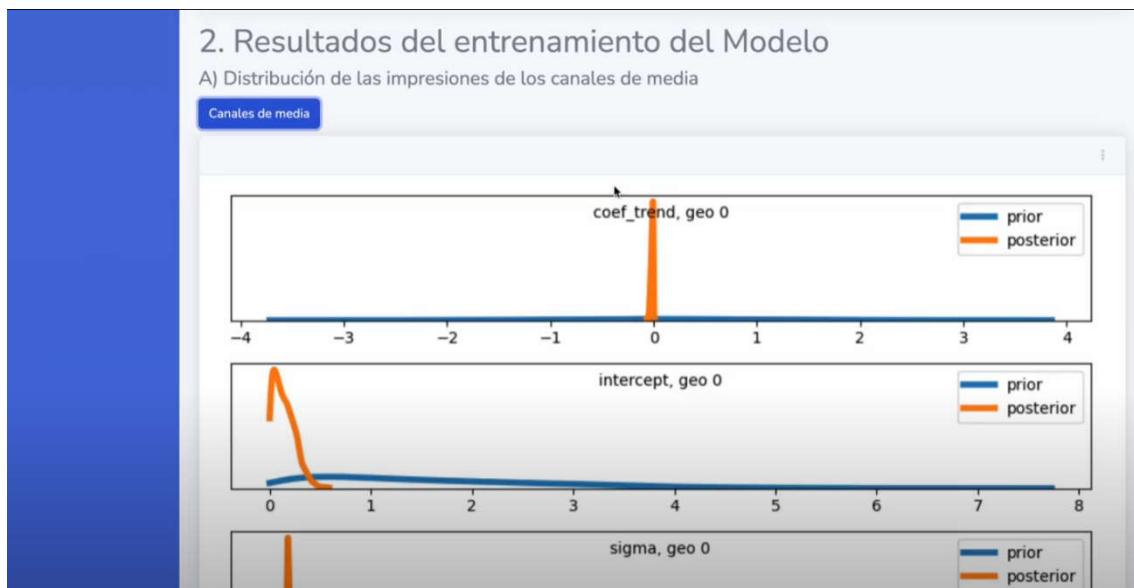


Ilustración 50 - Curvas bayesianas

Se genera una gráfica con la contribución estimada de los medios de comunicación a las ventas por cada canal y la línea de base que no está influida por la acción de los medios, a lo largo del tiempo

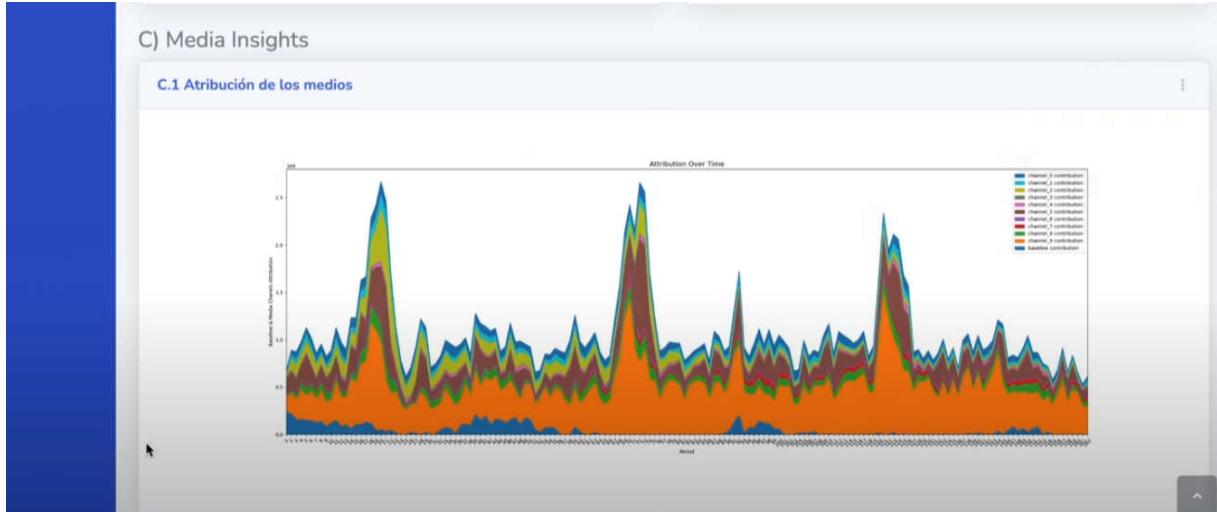


Ilustración 51 - - Estimación a 6 meses de la contribución de los canales en las ventas.

Y finalmente el modelo realiza la predicción ajustando las inversiones por casa canal de los medios, comparando con la inversión real.

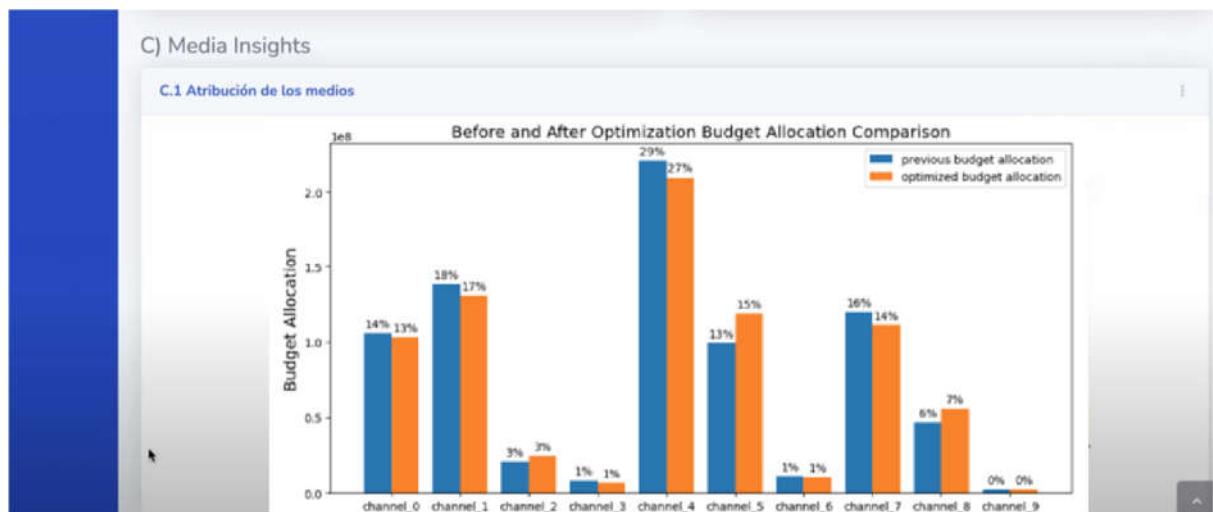


Ilustración 52 - estimación de la inversión por canales

6. Conclusiones

En este último capítulo se realiza la valoración del desarrollo del proyecto y se exponen las principales aportaciones de este proyecto, así como la continuidad en el futuro.

6.1. Valoración del proyecto

La valoración general de este proyecto de Trabajo Fin de Grado en el área de Marketing Mix Modeling (MMM) es extremadamente positiva. Esta experiencia ha brindado la oportunidad de poner en práctica los conocimientos adquiridos a lo largo de los años de estudio. Es un orgullo ser uno de los primeros científicos de datos, en esta universidad y en el país, lo que demuestra el crecimiento y la evolución personal en el campo de la ciencia de datos.

Durante el desarrollo de este proyecto, se ha conseguido consolidar mucho de los conceptos que se han ido explicando en el grado, esto ha permitido poder desarrollar una idea y conseguir modelar un campo tan complejo e incierto como el marketing mix. Además, la asignatura de Contextualización del Trabajo de Fin de Grado ha desempeñado un papel fundamental al brindarnos las herramientas necesarias para comprender las necesidades del proyecto, delimitar su alcance y ajustar los diferentes aspectos en función de esas necesidades.

Es importante destacar que, este trabajo de fin de grado ha sido una excelente experiencia para aprender la planificación y gestión de proyectos, así como afrontar los retos para conseguir su finalización en el tiempo requerido. Esto no hubiera sido posible, sin los valiosos consejos y la guía proporcionada por el tutor académico, con su ayuda se ha permitido aprender, aplicar y mejorar en cada etapa del proceso. Estas orientaciones han sido cruciales para cumplir con los hitos y tareas establecidas, evitando contratiempos y asegurando un progreso constante.

En este sentido, cabe resaltar la importancia de contar con una planificación sólida y una gestión efectiva del proyecto. La experiencia adquirida durante el desarrollo del proyecto ha permitido superar obstáculos, ajustar los enfoques y lograr resultados satisfactorios. Además, el uso de las herramientas adecuadas, en combinación con la orientación recibida, ha sido clave para alcanzar los objetivos.

Durante el desarrollo de este proyecto, varias han sido las dificultades que se han tenido que ir solventando. En primer lugar, uno de los mayores obstáculos fue la recopilación de información, ya que requería una exhaustiva investigación en diversos artículos y documentos académicos sobre el marketing mix. Esta etapa inicial permitió adentrarse en el tema y comprender en profundidad los conceptos clave.

Además, se han tenido que estudiar y aplicar diferentes metodologías más avanzadas que las estudiadas durante el grado. Especialmente, uno de los mayores desafíos fue al utilizar librerías científicas de optimización, lo cual representó un verdadero reto tanto en su comprensión como en su integración en el proyecto. La compatibilidad entre estas librerías se convirtió en un verdadero dolor de cabeza, lo que llevó a explorar y utilizar diferentes entornos para solucionar los problemas de incompatibilidad de requisitos.

El segundo gran desafío fue el diseño y desarrollo de una aplicación web en un tiempo limitado, con el objetivo de crear un producto mínimo viable. En este sentido, se tomó la decisión acertada de utilizar Django, un framework nativo de Python, para la construcción del backoffice de la aplicación. Esta elección permitió unos resultados buenos en términos de integración de APIs y facilitó enormemente la comunicación entre el código y el frontend.



Sin embargo, durante el desarrollo de la aplicación surgieron nuevos desafíos con la integración de las librerías necesarias en la máquina virtual, lo cual requirió tiempo y esfuerzo adicional para asegurar su correcto funcionamiento. Además, de contar con suficiente capacidad de memoria RAM en la GPU para llevar a cabo el procesamiento de datos de manera eficiente.

A pesar de estos obstáculos, se buscaron soluciones efectivas, consiguiendo desarrollar una aplicación web prototipo capaz de cumplir con los requisitos establecidos y servir de base para futuros desarrollos.

En resumen, el desarrollo de este proyecto ha brindado la oportunidad de aprender y aplicar conocimientos adquiridos a lo largo del grado y superar diversas dificultades, consiguiendo afianzar las habilidades necesarias para un científico de datos. A través de la adaptación, el aprendizaje constante y la superación de desafíos, se ha logrado alcanzar los objetivos propuestos y obtener resultados satisfactorios. Este proyecto sienta las bases para futuras investigaciones y aplicaciones prácticas en el campo del Marketing Mix Modeling.

6.2. Conclusiones de los resultados

Este proyecto se materializa en una aplicación web completamente operativa, la cual logra alcanzar los propósitos establecidos:

- Una solución que contribuye de forma positiva en las organizaciones, al reducir los sesgos y maximizar el retorno de la inversión en marketing.
- Una plataforma web con la capacidad de llevar a cabo simulaciones del modelo de Marketing Mix (MMM).

Además, se han logrado alcanzar los diferentes objetivos específicos planteados:

- Obtener un modelo que refleje de manera precisa la realidad de un modelo de marketing.
- Realizar simulaciones del retorno de la inversión en marketing.
- Evaluar diferentes enfoques metodológicos de Modelos de Marketing Mix (MMM).
- Crear modelos de optimización del MMM que incluyan los efectos del marketing.
- Desarrollar y evaluar una herramienta de simulaciones (producto final).

En relación al impacto ético-social del proyecto del Modelo de Marketing Mix (MMM), la aplicación se ha diseñado de forma que no exista ninguna violación de la privacidad o el uso indebido de datos personales. La aplicación no almacena ningún tipo de datos que utilicen en el modelo de marketing mix, solamente se ejecutan en el modelo.

Y en el caso de las direcciones de email que se usan para el registro, El hecho de recopilar el correo electrónico de los usuarios en la aplicación de MMM para fines de registro y uso de la aplicación está sujeto a la Ley de Protección de Datos Personales. El correo electrónico se considera información personal identificable, ya que puede utilizarse para identificar a una persona específica. Conforme establece la "Ley Orgánica de Protección de Datos Personales y garantía de los derechos digitales" (LOPDGDD). Esta ley fue aprobada el 5 de diciembre de 2018 y establece las normas para la protección de los datos personales de los ciudadanos españoles y garantiza sus derechos en el ámbito digital. La LOPDGDD es la adaptación española al Reglamento General de Protección de Datos (RGPD) de la Unión Europea y complementa las disposiciones contenidas en dicho reglamento para su aplicación en el territorio nacional. para recopilar, almacenar y procesar datos personales, incluido el correo electrónico.

Según esta ley, se requiere obtener el consentimiento explícito y voluntario de los usuarios para recopilar y procesar su correo electrónico. Además, se deben tomar medidas de seguridad



adecuadas para proteger los datos personales recopilados y garantizar que se utilicen únicamente para los fines previstos y de acuerdo con las políticas de privacidad establecidas.

Como los usuarios deben solicitar el alta para el uso de la aplicación, se les proporcionará en ese momento, un texto que destinado a informar sobre cómo se utilizará su correo electrónico, cómo se protegerá y si se compartirá con terceros, así como proporcionar a los usuarios la opción de revocar su consentimiento y solicitar la eliminación de sus datos en cualquier momento.

Por otro lado, el uso de esta aplicación puede tener un impacto significativo en la sostenibilidad de las empresas al maximizar la inversión en medios de publicidad y contribuir positivamente a varios Objetivos de Desarrollo Sostenible (ODS). Al utilizar técnicas de Marketing Mix Modeling (MMM), las empresas pueden optimizar sus estrategias de marketing y lograr un impacto positivo en los siguientes ODS:

- **ODS 9 - Industria, Innovación e Infraestructura:** Al maximizar la inversión en medios de publicidad de manera eficiente, las empresas pueden impulsar la innovación en la industria y mejorar su competitividad en el mercado.
- **ODS 12 - Producción y Consumo Responsables:** Al identificar los medios publicitarios más efectivos, las empresas pueden reducir el desperdicio de recursos y promover una producción y consumo más responsable. Esto implica minimizar el uso de recursos naturales y limitar la generación de residuos asociados a campañas ineficientes.
- **ODS 13 - Acción por el Clima:** Al optimizar las estrategias de marketing, las empresas pueden reducir su huella de carbono al minimizar la necesidad de recursos adicionales y disminuir la emisión de gases de efecto invernadero asociados a actividades de marketing ineficientes.
- **ODS 17 - Alianzas para lograr los Objetivos:** Al utilizar esta aplicación y adoptar enfoques sostenibles en sus estrategias de marketing, las empresas pueden fortalecer alianzas con otras organizaciones comprometidas con la sostenibilidad, promoviendo colaboraciones que impulsen un impacto positivo en la sociedad y el medio ambiente.

En resumen, al maximizar la inversión en medios de publicidad y utilizar técnicas de Marketing Mix Modeling, las empresas pueden contribuir positivamente a varios ODS. Al tomar decisiones informadas y estratégicas, minimizar el desperdicio de recursos y promover la sostenibilidad, las empresas pueden desempeñar un papel clave en la consecución de los objetivos establecidos por la Agenda 2030 de las Naciones Unidas.

Y en lo referente a la dimensión de diversidad, el uso de los MMM de nueva generación vistos en el punto 4.8. de esta memoria, demuestra que el uso de datos objetivos y automatizados en lugar del conocimiento subjetivo de los analistas de marketing, contribuye de forma eficaz para reducir el sesgo.

La aplicación de estos MMM permiten considerar múltiples variables demográficas al desarrollar estrategias publicitarias, posibilitando la creación de campañas más inclusivas, que reflejan la diversidad de la sociedad y se adaptan a los diferentes segmentos de la población.

En este TFG, se ha evidenciado cómo el MMM desempeña un papel fundamental en la reducción del sesgo de los analistas y en la promoción de la diversidad en el marketing. Al basar las decisiones en datos objetivos y contemplar de manera equitativa los perfiles de la población, este enfoque ha demostrado su capacidad para generar resultados más inclusivos y efectivos en las estrategias de marketing.



El resultado global de este proyecto es sumamente satisfactorio, ya que ha alcanzado un nivel cercano al de un proyecto de producción. Se ha logrado cumplir con todos los objetivos establecidos, lo que valida la solidez y viabilidad de la propuesta.

Por todo se puede afirmar que en términos de resultados, este proyecto ha sentado las bases para futuras aplicaciones profesionales o investigaciones en el campo del Marketing Mix Modeling (MMM). La metodología desarrollada, las herramientas implementadas y los conocimientos adquiridos constituyen un valioso legado que puede ser aprovecharse en diferentes ámbitos.

La capacidad de este proyecto para entender y poder simular la inversión en medios de publicidad, considerar la sostenibilidad y la diversidad, y promover la toma de decisiones basadas en datos, lo posiciona como un recurso valioso para profesionales del marketing y especialistas en ciencia de datos.

Asimismo, este proyecto ha demostrado su potencial para ser aplicado en entornos profesionales y adaptado a diversas industrias y sectores. Su enfoque integral y la versatilidad de sus resultados lo convierten en una herramienta valiosa para mejorar la toma de decisiones estratégicas en marketing y optimizar la inversión en publicidad.

En resumen, el éxito alcanzado en este proyecto, tanto en términos de cumplimiento de objetivos como de potencial aplicabilidad, respalda su relevancia y su contribución al campo del MMM. Los resultados obtenidos y las lecciones aprendidas brindan una base sólida para futuros proyectos y aplicaciones profesionales en este campo, y abren la puerta a nuevas investigaciones en la intersección entre el marketing y la ciencia de datos.

6.3. Posibles proyectos derivados

El resultado obtenido en este Trabajo Fin de Grado abre las puertas a una serie de proyectos derivados que podrían ampliar aún más el alcance y la utilidad de la aplicación desarrollada en el marco del Marketing Mix Modeling (MMM).

Una posible dirección para futuros proyectos sería utilizar la aplicación como plataforma de investigación colaborativa, permitiendo evaluar y comparar diferentes modelos de MMM con empresas de cualquier parte del mundo. Mediante acuerdos de colaboración, estas empresas podrían utilizar la aplicación y, a cambio, compartir los resultados obtenidos, lo que enriquecería la herramienta y permitiría seguir investigando nuevos enfoques y mejoras en los modelos de machine learning avanzados.

Además, existe la oportunidad de explorar la posibilidad de comercializar la aplicación desarrollada como un producto real. La robustez y efectividad de la herramienta la convierten en un recurso valioso para empresas y profesionales del marketing, que podrían beneficiarse de su capacidad para maximizar la inversión en publicidad y reducir los sesgos en sus estrategias de marketing.

Otra perspectiva interesante sería utilizar la aplicación como base para crear una plataforma de Modelos de Marketing más amplia, donde el MMM sería una categoría específica. Esta plataforma permitiría la integración de otros modelos de análisis, como RFM (Recency, Frequency, Monetary Value), CLTV (Customer Lifetime Value), grafos de redes sociales, sistemas de recomendación y muchos otros. Esto ampliaría las funcionalidades y aplicaciones



de la herramienta, brindando un entorno más completo para el análisis y la toma de decisiones en marketing.

Además, el conocimiento adquirido durante el desarrollo de este proyecto podría servir de base para profundizar en el campo del MMM y desarrollar nuevos modelos y enfoques. La experiencia acumulada y los desafíos superados ofrecen una base sólida para seguir explorando áreas de mejora y expansión en el ámbito del marketing y la ciencia de datos.

Por último, este proyecto también podría ser el punto de partida para la redacción de un paper científico o incluso para continuar estudios a nivel de doctorado. La originalidad, el enfoque innovador y los resultados obtenidos son elementos clave que respaldan la validez y el potencial de este proyecto como base para investigaciones académicas más avanzadas.

En conclusión, las posibilidades derivadas de este TFG son amplias y emocionantes. Desde seguir investigando y mejorando el MMM, hasta su comercialización como producto, su integración en una plataforma más amplia, el desarrollo de nuevos modelos y la continuidad académica, el futuro de este proyecto promete continuar generando impacto y contribuir al avance del marketing y la ciencia de datos.

7. Glosario

DJANGO: Django es un marco de desarrollo web en Python que facilita la creación rápida de aplicaciones seguras y mantenibles, incluyendo funcionalidades preconstruidas y siguiendo el patrón Modelo-Vista-Controlador (MVC).

HTML (HyperText Markup Language): Es el lenguaje de marcado estándar para crear páginas web y definir su estructura.

JS (JavaScript): Es un lenguaje de programación utilizado para agregar interactividad y funcionalidad avanzada a las páginas web.

CSS (Cascading Style Sheets): Es un lenguaje de estilo que se utiliza para describir cómo se debe mostrar el contenido HTML en la pantalla.

Serie temporal: Es una secuencia de datos recogidos a intervalos regulares de tiempo. Se utilizan en muchos campos para analizar tendencias, ciclos y patrones en los datos a lo largo del tiempo.

Estadística bayesiana: Es un enfoque de la estadística que combina los datos observados con información previa para obtener conclusiones más completas. Utiliza la Regla de Bayes para actualizar las creencias previas basándose en los nuevos datos, resultando en una distribución a posteriori.

Regresión Ridge: Es un método de regresión regularizado que ayuda a evitar el sobreajuste al introducir una penalización a los coeficientes del modelo. Este enfoque controla la complejidad del modelo y mejora su generalización limitando la magnitud de los coeficientes a través de un parámetro de ajuste.

MVP: Producto Mínimo Viable es un producto con suficientes características para satisfacer a los clientes iniciales, y proporcionar retroalimentación para el desarrollo futuro.

CRISP-DM: se trata de un modelo estándar abierto del proceso que describe los enfoques comunes que utilizan los expertos en minería de datos.

PMBOK: Project Management Body of Knowledge, es una guía fundamental para los gestores de proyectos que desean mejorar las actividades y los resultados empresariales.

EDA: En estadística, el análisis exploratorio de datos es un enfoque de análisis de conjuntos de datos para resumir sus características principales, a menudo utilizando gráficos estadísticos y otros métodos de visualización de datos

BootStrap: es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web.

API (Interfaz de Programación de Aplicaciones): Es un conjunto de reglas y protocolos que permite a las aplicaciones de software interactuar entre sí. Proporciona un método para integrar diferentes sistemas de software, permitiendo que las funcionalidades de un sistema sean utilizadas por otro.

Plotly: antes de ser una conocida librería Python, es una empresa con sede en Montreal, su objetivo es desarrollar herramientas de visualización y analizar datos.

Jupyter Notebook: Jupyter es el acrónimo de Julia, Python y R, siendo un entorno informático interactivo, que permite a los usuarios experimentar con el código y compartirlo.

ETL: Extract, Transform and Load es el proceso que permite a las organizaciones mover datos desde múltiples fuentes, reformatearlos y limpiarlos, y cargarlos en otra base de datos, data mart, o data warehouse para analizar, o en otro sistema operacional para apoyar un proceso de negocio.

8. Bibliografía

- Leeflang, P. S. H., Wittink, D. R., Wedel, M., & Naert, P. A. (2000). Building Models for Marketing Decisions. Kluwer Academic Publishers.
- Gujarati, D. N. (2003). Basic Econometrics. McGraw Hill.
- Wooldridge, J. M. (2012). Introductory Econometrics: A Modern Approach. South-Western Cengage Learning.
- James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). An Introduction to Statistical Learning. Springer.
- Tellis, G. J. (1988). The Price Elasticity of Selective Demand: A Meta-Analysis of Econometric Models of Sales. Journal of Marketing Research, 25(4), 331-341.
- Hanssens, D. M., Parsons, L. J., & Schultz, R. L. (2001). Market Response Models: Econometric and Time Series Analysis. Kluwer Academic Publishers.
- Leeflang, P. S. H., Wittink, D. R., Wedel, M., & Naert, P. A. (2000). Building Models for Marketing Decisions. Kluwer Academic Publishers.
- Durbin, J., & Koopman, S. J. (2012). Time Series Analysis by State Space Methods: Second Edition. Oxford University Press.
- Kim, C. J., & Nelson, C. R. (1999). State-Space Models with Regime Switching: Classical and Gibbs-Sampling Approaches with Applications. The MIT Press.
- Hanssens, D. M., & Pauwels, K. H. (2016). Demonstrating the Value of Marketing. Journal of Marketing, 80(6), 173-190.
- Gregory Piatetsky, KDnuggets on October 28, 2014 in CRISP-DM, Data Mining, James Taylor, Methodology, Poll
- PMBOK® Guide – Fifth Edition (2013). A guide to the project management body of knowledge. Project Management Institute, Inc.
- Amaratunga, D., Cabrera, J., Fernholz, L. T., & Morgenthaler, S. (2018). Exploratory Data Analysis. John Wiley & Sons.
- Batini, C., & Scannapieco, M. (2010). Data Quality: Concepts, Methodologies and Techniques. Springer.
- Belenguer, L. AI bias: exploring discriminatory algorithmic decision-making models and the application of possible machine-centric solutions adapted from the pharmaceutical industry. AI Ethics 2, 771–787 (2022). <https://doi.org/10.1007/s43681-022-00138-8>
- Kumar, V. (2017). A theory of marketing: Outline of a social systems perspective. Routledge.



- Kumar, V., & Leone, R. P. (1988). Measuring the effect of retail store promotions on brand and store substitution. *Journal of Marketing Research*, 25(2), 178-185.
- R.J. Freund, W.J. Wilson, and P. Sa. *Regression Analysis: Statistical Modeling of a Response Variable*. Elsevier Academic Press, 2006. isbn: 9780120885978. url: <https://books.google.se/books?id=Qtx2IAEACAAJ>.

Fuentes académicas:

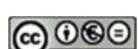
- The double-edged razor of machine learning algorithms in marketing: benefits vs. ethical concerns (2018), Author: Tanser Karakash, University of Twente P.O.
 - Bayesian Time Varying Coefficient Model with Applications to Marketing Mix Modeling (2015). Edwin Ng, Zhishi Wang, Athena Dai,
 - The Aggregate Marketing System Simulator(2017). Zhang, S. and Vaver.
 - Inferring causal impact using bayesian structural time-series models (2016).k.brodersen, f.gallusser, j.koehler, n. remy and s. scott
 - Challenges And Opportunities In Media Mix Modeling (2017). David Chan and Michael Perry Google Inc.
 - Geo-level Bayesian Hierarchical Media Mix Modeling Yunting (2.017). Sun, Yueqing Wang, Yuxue Jin, David Chan, Jim Koehler. Google Inc.
- Taylor SJ, Letham B. 2017. Forecasting at scale. PeerJ Preprints 5:e3190v2
<https://doi.org/10.7287/peerj.preprints.3190v2>

Fuentes profesionales:

- Ethics guidelines for trustworthy ai, European Commission, Document made public on 8 April 2019
- <https://towardsdatascience.com/marketing-mix-modeling-101-d0e24306277d>
- <https://medium.com/@skillcate/marketing-strategy-project-using-machine-learning-6c3101372a04>
- <https://medium.com/analytics-vidhya/marketing-mix-model-guide-with-dataset-using-python-r-and-excel-4e319be47b4>
- Python/STAN Implementation of Multiplicative Marketing Mix Model | by Sibyl He | Towards Data Science
- 21.26 Marketing Mix Optimization Models | Marketing Research (bookdown.org)
- Using R to build a simple marketing mix model (MMM) and make predictions | Towards Data Science
- Facebook Robyn – variable transformations, mission
- Bayesian Media Mix Modeling using PyMC3, for Fun and Profit – bayesian methods, process
- Bayesian Methods for Media Mix Modeling with Carryover and Shape Effects – bayesian methods, variable transformations
- A Hierarchical Bayesian Approach to Improve Media Mix Models Using Category Data – hierarchical models
- How to create a basic Marketing Mix Model in scikit-learn – using multiple algos, scikit learn
- Python/STAN Implementation of Multiplicative Marketing Mix Model – multiplicative models, seasonality

Webgrafía

- <https://www.microprediction.com/blog/prophet>
- <https://www.youtube.com/watch?v=B7ZWehBHVw0>
- <https://www.latticeworkinsights.com/press/we-evaluated-3-media-mix-models-so-you-dont-have-to>
- https://ekimetrics.com/wp-content/uploads/2020/05/Ekimetrics_Facebook_White-paper.pdf?fbclid=IwAR1mvLJ8zcVO567q-3nv21c2DF57kA_eAQWRp1KI4a56eDYGMPIQ1iedul
- <https://www2.deloitte.com/content/dam/Deloitte/es/Documents/estrategia/Deloitte-es-estrategia-y-operaciones-combinacion-mmm-cle.pdf>
- <https://sd-group.com.au/en/blog/market-mix-vs-multi-touch-attribution-model>
- <https://uk.sganalytics.com/case-study/analytics/market-mix-modeling-what-if-simulator-insurance/>
- <https://www.pymc-labs.io/blog-posts/reducing-customer-acquisition-costs-how-we-helped-optimizing-hellofreshs-marketing-budget/>
- <https://towardsdatascience.com/introduction-to-bayesian-linear-regression-e66e60791ea7>



9. Anexos

- Anexo I – Diccionario de Datos
- Anexo II – Preprocesamiento
- Anexo III – Modelo de Regresión Lineal Multivariante
- Anexo IV – Modelo de Regresión Multiplicativo
- Anexo V – Modelo Robyn
- Anexo VI – Modelo Bayesiano con Efectos de Marketing
- Anexo VII – Modelo Bayesiano Stan
- Anexo VIII – Primer Informe de seguimiento
- Anexo IX – Segundo Informe de seguimiento
- Anexo X – Documentación Técnica Aplicación Web “Nebula Navigator”
- Anexo XI – Fichero LICENSE.MD – Licencia de aplicación web
- Anexo XII - Link GITHUB Proyecto



Anexo I: Diccionario de datos



GRADO DE CIENCIA DE DATOS APLICADA

Diccionario de Datos

18 Junio 2023

Descripción breve

Diccionario de datos



Alberto de Torres Pachón

Dirección académica: Xavier Florit

Responsable académico: Elena Rodríguez

Trabajo final de grado 22536



ADTP

Variables	Nombre	Descripción
wk strt_dt	week	Semanas desde 3/08/2014 hasta 29/07/2018
UOC-GCDA-TFG	year	Años de la muestra
yr_nbr	quarter	Trimestre, empezando por febrero como el primero
qtr_nbr	period	Periodo empezando en Febrero
prd	nu_week_mo	Numero de la semana en cada mes del 1 al 5
wk_nbr	nu_week_yr	Numero de la semana en el año
wk_in_yr_nbr	dm_imp	Numero de impresiones de direct mail
mdip_dm	insert_imp	Numero de impresiones de inserciones de materiales promocionales para insertar en activos digitales
mdip_inst	prensa_imp	Numero de impresiones en prensa
mdip_nsp	audiodig_imp	Numero de impresiones del audio digital
mdip_auddig	radio_imp	Numero de impresiones de radio
mdip_audtr	tv_imp	Numero de impresiones de tv
mdip_vidtr	video_imp	Numero de impresiones de video digital
mdip_viddig	somedia_imp	Numero de impresiones de social media
mdip_so	ondisplay_imp	Numero de impresiones de display online
mdip_on	email_imp	Numero de impresiones de emails
mdip_em	sms_imp	Numero de impresiones de los SMS
mdip_sms	afiliado_imp	Numero de impresiones de afiliados
mdip_aff	sem_imp	Numero de imprssiones del sem
mdip_sem	dm_inv	Inversión en direct mail
mdsp_dm	insert_inv	Inversión en inserción de materiales promocionales para insertar en activos digitales
mdsp_inst	prensa_inv	Inversión en campañas de prensa
mdsp_nsp	audiodig_inv	Inversión en campañas de audio digital
mdsp_auddig	radio_inv	Inversión en campañas de radio
mdsp_audtr	tv_inv	Inversión en campañas de tv
mdsp_vidtr	video_inv	Inversión en campañas de video digital
mdsp_viddig	somedia_inv	Invdrsión en campañas de social media
mdsp_so	ondisplay_inv	Invrsión en campañas de display online
mdsp_on	sem_inv	Invrsión en campañas de sem
mdsp_sem	ventas	Ventas de la empresa
sales	me_inflaccion	Variable macroeconómica: Ratio de la inflación
me_ics_all	me_gas	Variable macroeconómica: Precio galon gasolina
me_gas_dpg	tiendas	Numero de tiendas
st_ct	descuento_1	Reducción precios permanente
mrkdn_valadd_edw	descuento_2	Reducción precios permanente
va_pub_0.15		
va_pub_0.2		
va_pub_0.25		
va_pub_0.3		
hldy_Black Friday	hldy_Black Friday	Campaña del Black Friday
hldy_Christmas Day	hldy_Christmas Day	Campaña Christmas Day
hldy_Christmas Eve	hldy_Christmas Eve	Campaña Christmas Eve
hldy_Columbus Day	hldy_Columbus Day	Campaña Columbus Day
hldy_Cyber Monday	hldy_Cyber Monday	Campaña Cyber Monday
hldy_Day after Christmas	hldy_Day after Christmas	Campaña Day after Christmas
hldy_Easter	hldy_Easter	Campaña _Easter
hldy_Father's Day	hldy_Father's Day	Campaña Father's Day
hldy_Green Monday	hldy_Green Monday	Campaña _Green Monday
hldy_July 4th	hldy_July 4th	Campaña July 4th
hldy_Labor Day	hldy_Labor Day	Campaña Labor Day



hldy_MLK	hldy_MLK	Campaña MLK
hldy_Memorial Day	hldy_Memorial Day	Campaña Memorial Day
hldy_Mother's Day	hldy_Mother's Day	Campaña Mother's Day
hldy_NYE	hldy_NYE	Campaña NYE
hldy_New Year's Day	hldy_New Year's Day	Campaña New Year's Day
hldy_Pre Thanksgiving	hldy_Pre Thanksgiving	Campaña Pre Thanksgiving
hldy_Presidents Day	hldy_Presidents Day	Campaña Presidents Day
hldy_Prime Day	hldy_Prime Day	Campaña Prime Day
hldy_Thanksgiving	hldy_Thanksgiving	Campaña Thanksgiving
hldy_Valentine's Day	hldy_Valentine's Day	Campaña Valentine's Day
hldy_Veterans Day	hldy_Veterans Day	Campaña Veterans Day
seas_prd_1	seas_prd_1	Estacionalidad de Febrero
seas_prd_2	seas_prd_2	Estacionalidad de Marzo
seas_prd_3	seas_prd_3	Estacionalidad de Abril
seas_prd_4	seas_prd_4	Estacionalidad de Mayo
seas_prd_5	seas_prd_5	Estacionalidad de Junio
seas_prd_6	seas_prd_6	Estacionalidad de Julio
seas_prd_7	seas_prd_7	Estacionalidad de Agosto
seas_prd_8	seas_prd_8	Estacionalidad de Septiembre
seas_prd_9	seas_prd_9	Estacionalidad de Octubre
seas_prd_12	seas_prd_12	Estacionalidad de Enero
seas_week_40	seas_week_40	Estacionalidad de Noviembre semana 1
seas_week_41	seas_week_41	Estacionalidad de Noviembre semana 2
seas_week_42	seas_week_42	Estacionalidad de Noviembre semana 3
seas_week_43	seas_week_43	Estacionalidad de Noviembre semana 4
seas_week_44	seas_week_44	Estacionalidad de Noviembre semana 5
seas_week_45	seas_week_45	Estacionalidad de Diciembre semana 1
seas_week_46	seas_week_46	Estacionalidad de Diciembre semana 2
seas_week_47	seas_week_47	Estacionalidad de Diciembre semana 3
seas_week_48	seas_week_48	Estacionalidad de Diciembre semana 4

Variables	Nombre	Descripcion
wk strt_dt	week	Semanas desde 3/08/2014 hasta 29/07/2018
yr_nbr	year	Años de la muestra
qtr_nbr	quarter	Trimestre, empezando por febrero como el primero
prd	period	Periodo empezando en Febrero
wk_nbr	nu_week_mo	Numero de la semana en cada mes del 1 al 5
wk_in_yr_nbr	nu_week_yr	Numero de la semana en el año
mdip_dm	dm_imp	Numero de impresiones de direct mail
mdip_inst	insert_imp	Numero de impresiones de inserciones de materiales promocionales digitales
mdip_nsp	prensa_imp	Numero de impresiones en prensa
mdip_auddig	audiodig_imp	Numero de impresiones del audio digital
mdip_audtr	radio_imp	Numero de impresiones de radio
mdip_vidtr	tv_imp	Numero de impresiones de tv
mdip_viddig	video_imp	Numero de impresiones de video digital
mdip_so	somedia_imp	Numero de impresiones de social media



mdip_on	ondisplay_imp	Numero de impresiones de display online
mdip_em	email_imp	Numero de impresiones de emails
mdip_sms	sms_imp	Numero de impresiones de los SMS
mdip_aff	afiliado_imp	Numero de impresiones de afiliados
mdip_sem	sem_imp	Numero de imprssiones del sem
mdsp_dm	dm_inv	Inversión en direct mail
mdsp_inst	insert_inv	Inversión en inserción de materiales promocionales para insertar en acti
mdsp_nsp	prensa_inv	Inversión en campañas de prensa
mdsp_auddig	audiodig_inv	Inversión en campañas de audio digital
mdsp_audtr	radio_inv	Inversión en campañas de radio
mdsp_vidtr	tv_inv	Inversión en campañas de tv
mdsp_viddig	video_inv	Inversión en campañas de video digital
mdsp_so	somedia_inv	Invdrsión en campañas de social media
mdsp_on	ondisplay_inv	Invrsión en campañas de display online
mdsp_sem	sem_inv	Invrsión en campañas de sem
sales	ventas	Ventas de la empresa
me_ics_all	me_inflaccion	Variable macroeconómica: Ratio de la inflación
me_gas_dpg	me_gas	Variable macroeconómica: Precio galon gasolina
st_ct	tiendas	Numero de tiendas
mrkdn_valadd_edw	descuento_1	Reducción precios permanente
mrkdn_pdm	descuento_2	Reducción precios permanente
va_pub_0.15		
va_pub_0.2		
va_pub_0.25		
va_pub_0.3		
hldy_Black Friday	hldy_Black Friday	Campaña del Black Friday
hldy_Christmas Day	hldy_Christmas Day	Campaña Christmas Day
hldy_Christmas Eve	hldy_Christmas Eve	Campaña Christmas Eve
hldy_Columbus Day	hldy_Columbus Day	Campaña Columbus Day
hldy_Cyber Monday	hldy_Cyber Monday	Campaña Cyber Monday
hldy_Day after Christmas	hldy_Day after Christmas	Campaña Day after Christmas
hldy_Easter	hldy_Easter	Campaña _Easter
hldy_Father's Day	hldy_Father's Day	Campaña Father's Day
hldy_Green Monday	hldy_Green Monday	Campaña _Green Monday
hldy_July 4th	hldy_July 4th	Campaña July 4th
hldy_Labor Day	hldy_Labor Day	Campaña Labor Day
hldy_MLK	hldy_MLK	Campaña MLK
hldy_Memorial Day	hldy_Memorial Day	Campaña Memorial Day
hldy_Mother's Day	hldy_Mother's Day	Campaña Mother's Day
hldy_NYE	hldy_NYE	Campaña NYE
hldy_New Year's Day	hldy_New Year's Day	Campaña New Year's Day
hldy_Pre Thanksgiving	hldy_Pre Thanksgiving	Campaña Pre Thanksgiving
hldy_Presidents Day	hldy_Presidents Day	Campaña Presidents Day
hldy_Prime Day	hldy_Prime Day	Campaña Prime Day
hldy_Thanksgiving	hldy_Thanksgiving	Campaña Thanksgiving
hldy_Valentine's Day	hldy_Valentine's Day	Campaña Valentine's Day
hldy_Veterans Day	hldy_Veterans Day	Campaña Veterans Day
seas_prd_1	seas_prd_1	Estacionalidad de Febrero



seas_prd_2	seas_prd_2	Estacionalidad de Marzo
seas_prd_3	seas_prd_3	Estacionalidad de Abril
seas_prd_4	seas_prd_4	Estacionalidad de Mayo
seas_prd_5	seas_prd_5	Estacionalidad de Junio
seas_prd_6	seas_prd_6	Estacionalidad de Julio
seas_prd_7	seas_prd_7	Estacionalidad de Agosto
seas_prd_8	seas_prd_8	Estacionalidad de Septiembre
seas_prd_9	seas_prd_9	Estacionalidad de Octubre
seas_prd_12	seas_prd_12	Estacionalidad de Enero
seas_week_40	seas_week_40	Estacionalidad de Noviembre semana 1
seas_week_41	seas_week_41	Estacionalidad de Noviembre semana 2
seas_week_42	seas_week_42	Estacionalidad de Noviembre semana 3
seas_week_43	seas_week_43	Estacionalidad de Noviembre semana 4
seas_week_44	seas_week_44	Estacionalidad de Noviembre semana 5
seas_week_45	seas_week_45	Estacionalidad de Diciembre semana 1
seas_week_46	seas_week_46	Estacionalidad de Diciembre semana 2
seas_week_47	seas_week_47	Estacionalidad de Diciembre semana 3
seas_week_48	seas_week_48	Estacionalidad de Diciembre semana 4



Anexo II: Preprocesamiento de los datos



GRADO DE CIENCIA DE DATOS APLICADA

Análisis Exploratorio de Datos Preprocesamiento

EDA Preprocesado de Datos

18 Junio 2023

Descripción breve

Informe con el contenido completo del EDA donde se han procesado los datos



Alberto de Torres Pachón

Dirección académica: Xavier Florit

Responsable académico: Elena Rodríguez
Trabajo final de grado 22536



ADTP

1. Introducción

Este análisis se realizó utilizando un conjunto de datos cargado desde Google Drive. El conjunto de datos, denominado MMM_data.csv, incluye diversas variables que fueron exploradas a lo largo de este análisis.

```
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
from sktime.utils.plotting import plot_series
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as ex
import plotly.graph_objs as go
import plotly.figure_factory as ff
from plotly.subplots import make_subplots
import plotly.offline as pyo
pyo.init_notebook_mode()
sns.set_style('darkgrid')
from sklearn import preprocessing

plt.rc('figure', figsize=(18, 9))
```

2. Carga de Datos

Los datos fueron cargados utilizando la biblioteca pandas de Python. Se obtuvo una vista preliminar de los datos mediante el método head() que mostró las primeras 20 entradas.

```
df = pd.read_csv('/content/drive/MyDrive/Datos/MMM_data.csv')
```

	wk_start_dt	yr_nbr	qtr_nbr	prd	wk_nbr	wk_in_yr_nbr	mdip_dm	mdip_inst	mdip_nsp	mdip_audig	...	seas_prd_12	seas_week_40	seas_week_41	seas_week_42	seas_week_43	seas_week_44	seas_week_45
0	2014-08-03	2014	3	7	1	27	4863885	29087520	2421933	692315	...	0	0	0	0	0	0	
1	2014-08-10	2014	3	7	2	28	20887502	8345120	3984494	475810	...	0	0	0	0	0	0	
2	2014-08-17	2014	3	7	3	29	11097724	17276800	1846832	784732	...	0	0	0	0	0	0	
3	2014-08-24	2014	3	7	4	30	1023446	18468480	2394834	1032301	...	0	0	0	0	0	0	
4	2014-08-31	2014	3	8	1	31	21109811	26659920	3312008	400456	...	0	0	0	0	0	0	
5	2014-09-07	2014	3	8	2	32	5965633	6255666	3215276	461272	...	0	0	0	0	0	0	
6	2014-09-14	2014	3	8	3	33	10034343	4232000	1365872	497614	...	0	0	0	0	0	0	
7	2014-09-21	2014	3	8	4	34	12764686	15474013	2024228	270547	...	0	0	0	0	0	0	
8	2014-09-28	2014	3	8	5	35	11745748	7490000	1715439	294330	...	0	0	0	0	0	0	

3. Análisis del Conjunto de Datos

Se realizó un análisis en profundidad del conjunto de datos para comprender mejor su estructura y contenido. Este análisis incluyó lo siguiente:

a) Tipo de Datos



Se evaluaron los tipos de datos presentes en el conjunto de datos, lo cual es esencial para entender qué tipos de operaciones se pueden realizar en cada columna.

```
wk strt_dt      object
yr_nbr          int64
qtr_nbr         int64
prd             int64
wk_nbr          int64
...
seas_week_44    int64
seas_week_45    int64
seas_week_46    int64
seas_week_47    int64
seas_week_48    int64
Length: 80, dtype: object
```

b) Dimensionalidad de los datos

Se examinó la forma del conjunto de datos para entender cuántas observaciones y características (columnas) contiene.

```
1 print ('La dimensionalidad es de: ', df.shape)
```

```
La dimensionalidad es de: (209, 80)
```

c) Análisis de Datos Duplicados

Se comprobó la presencia de entradas duplicadas y se observó cuántas de estas existen.

```
[ ] 1 print(df.shape[0], 'filas,', df.drop_duplicates().shape[0],
2           'distintas entradas ->',
3           str(df.shape[0]-df.drop_duplicates().shape[0]),
4           'duplicadas.')
```

```
209 filas, 209 distintas entradas -> 0 duplicadas.
```

d) Indicadores Estadísticos Principales

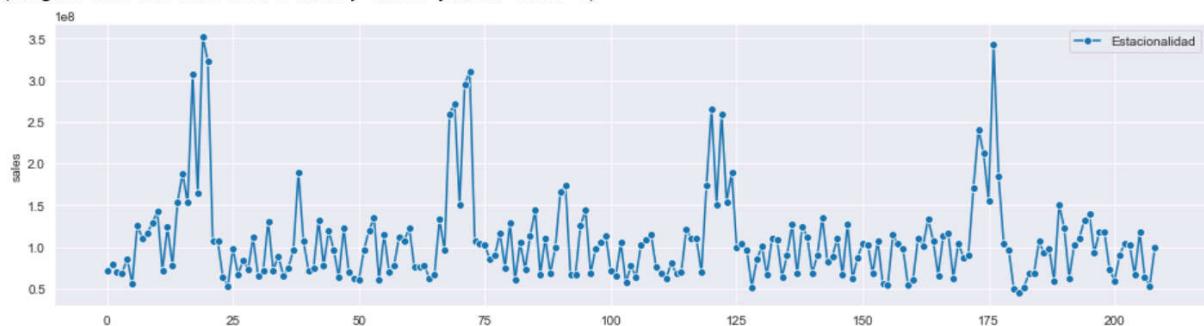
Se generó un resumen estadístico de las variables numéricas en el conjunto de datos, que incluye medidas como el valor medio, mínimo, máximo, la mediana (percentil 50) y los percentiles 25 y 75.

	yr_nbr	qtr_nbr	prd	wk_nbr	wk_in_yr_nbr	mdip_dm	mdip_inst	mdip_nsp	mdip_auddig	mdip_audtr	...	seas_prd_12	seas_week_40	seas_week_41
count	209.000000	209.000000	209.000000	209.000000	209.000000	2.090000e+02	2.090000e+02	2.090000e+02	2.090000e+02	2.090000e+02	...	209.000000	209.000000	209.000000
mean	2016.004785	2.507177	6.526316	2.703349	26.626794	9.544510e+06	1.247717e+07	1.616957e+06	1.002816e+06	2.295103e+07	...	0.081340	0.019139	0.019139
std	1.226697	1.122838	3.465562	1.274015	15.119856	8.293082e+06	1.024959e+07	2.203341e+06	8.122848e+05	1.567124e+07	...	0.274012	0.137342	0.137342
min	2014.000000	1.000000	1.000000	1.000000	1.000000	0.000000e+00	4.853300e+04	0.000000e+00	1.561800e+04	0.000000e+00	...	0.000000	0.000000	0.000000
25%	2015.000000	2.000000	4.000000	2.000000	14.000000	2.087021e+06	5.304240e+06	2.542340e+05	4.577220e+05	1.236705e+07	...	0.000000	0.000000	0.000000
50%	2016.000000	3.000000	7.000000	3.000000	27.000000	7.664954e+06	8.911466e+06	8.870720e+05	8.061170e+05	1.910160e+07	...	0.000000	0.000000	0.000000
75%	2017.000000	4.000000	10.000000	4.000000	40.000000	1.533852e+07	1.786920e+07	2.248483e+06	1.344765e+06	2.956004e+07	...	0.000000	0.000000	0.000000
max	2018.000000	4.000000	12.000000	5.000000	53.000000	3.979871e+07	6.545146e+07	1.553181e+07	5.418819e+06	9.066538e+07	...	1.000000	1.000000	1.000000



```
1 plot_series(df.sales.astype('float64'), labels=["Estacionalidad"])
```

```
(<Figure size 1152x288 with 1 Axes>, <Axes: ylabel='sales'>)
```



f) Análisis de Valores Nulos

Se buscó en el conjunto de datos cualquier valor nulo, luego se evaluó su proporción en relación con los datos completos, tanto en términos de columnas como de filas.

```
1 pd_null_columnas = pd.DataFrame(df.isnull().sum().sort_values(ascending=False), columns=['nulos_columnas'])
2 pd_null_filas = pd.DataFrame(df.isnull().sum(axis=1).sort_values(ascending=False), columns=['nulos_filas'])
3 pd_null_columnas['Proporcion_columnas'] = pd_null_columnas['nulos_columnas']/df.shape[0]
4 pd_null_filas['Proporcion_filas']= pd_null_filas['nulos_filas']/df.shape[1]
5 print(pd_null_filas)
6 print(pd_null_columnas)

    nulos_filas  Proporcion_filas
0            0           0.0
105          0           0.0
133          0           0.0
134          0           0.0
135          0           0.0
..
73            ...         ...
74            0           0.0
75            0           0.0
76            0           0.0
208          0           0.0

[209 rows x 2 columns]
              nulos_columnas  Proporcion_columnas
wk strt_dt                  0           0.0
yr nbr                      0           0.0
hldy Thanksgiving             0           0.0
hldy Prime Day                0           0.0
hldy Presidents Day           0           0.0
...
mdsp viddig                   0           0.0
mdsp vidtr                     0           0.0
mdsp audtr                     0           0.0
mdsp auddig                     0           0.0
seas week_48                   0           0.0
```

g) Análisis de Valores Atípicos

Se analizó la presencia de valores atípicos (outliers) en cada variable numérica del conjunto de datos. Los outliers pueden ser problemáticos para muchos modelos de

aprendizaje automático, por lo que es importante identificarlos durante la etapa de análisis de datos.

4. Transformación del Conjunto de Datos

Se realizaron varias transformaciones en el conjunto de datos para prepararlo para un análisis posterior o para la modelización. Estas transformaciones incluyeron:

a) Eliminación de Variables No Necesarias

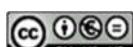
Se eliminaron algunas variables no necesarias para simplificar el análisis.

```
1 def calcular_outliers(df, columnas):
2     """
3         Función que devuelve el número de filas atípicas para cada variable numérica en un dataframe
4         utilizando la regla del rango intercuartil (IQR) ampliado.
5
6         :param df: Un dataframe de Pandas con variables numéricas
7         :return: Un diccionario con el número de filas atípicas para cada variable numérica
8         """
9
10    for columna in columnas:
11        Q1 = df[columna].quantile(0.25)
12        Q3 = df[columna].quantile(0.75)
13        IQR = Q3 - Q1
14        outliers = df[(df[columna] < (Q1 - 1.5 * IQR)) | (df[columna] > (Q3 + 1.5 * IQR))]
15        num_outliers = len(outliers)
16        pct_outliers = round(num_outliers / len(df[columna]) * 100, 2)
17        print(f'{columna}: {num_outliers} outliers ({pct_outliers}%)')
```



```
1 calcular_outliers(df, columnas)
2

mdip_dm: 2 outliers (0.96%)
mdip_inst: 6 outliers (2.87%)
mdip_nsp: 10 outliers (4.78%)
mdip_audig: 9 outliers (4.31%)
mdip_audtr: 8 outliers (3.83%)
mdip_vidtr: 17 outliers (8.13%)
mdip_viddig: 15 outliers (7.18%)
mdip_so: 2 outliers (0.96%)
mdip_on: 9 outliers (4.31%)
mdip_em: 7 outliers (3.35%)
mdip_sms: 5 outliers (2.39%)
mdip_aff: 10 outliers (4.78%)
mdip_sem: 9 outliers (4.31%)
mdsp_dm: 9 outliers (4.31%)
mdsp_inst: 9 outliers (4.31%)
mdsp_nsp: 8 outliers (3.83%)
mdsp_audig: 13 outliers (6.22%)
mdsp_audtr: 5 outliers (2.39%)
mdsp_vidtr: 14 outliers (6.7%)
mdsp_viddig: 10 outliers (4.78%)
mdsp_so: 8 outliers (3.83%)
mdsp_on: 5 outliers (2.39%)
mdsp_sem: 16 outliers (7.66%)
```



```
1 df2=df.iloc[:,6:30]
```

b) Cambio de Tipo de Datos

Se cambiaron los tipos de datos de algunas columnas para adaptarlas mejor al análisis posterior.

```
1 df2.dtypes
```

```
mdip_dm          int64
mdip_inst        int64
mdip_nsp         int64
mdip_auddig      int64
mdip_audtr       int64
mdip_vidtr       int64
mdip_viddig      int64
mdip_so          int64
mdip_on          int64
mdip_em          int64
mdip_sms         int64
mdip_aff         int64
mdip_sem         int64
mdsp_dm          float64
mdsp_inst        float64
mdsp_nsp         float64
mdsp_auddig      float64
mdsp_audtr       float64
mdsp_vidtr       float64
mdsp_viddig      float64
mdsp_so          float64
mdsp_on          float64
mdsp_sem         float64
sales            float64
dtype: object
```

c) Reescalado de Variables Numéricas

Las variables numéricas se reescalaron utilizando un MinMaxScaler para asegurarse de que todas las variables estuvieran en la misma escala. Esto es especialmente útil para ciertos tipos de modelos de aprendizaje automático.

```
1 x = df2.values #returns a numpy array
2 min_max_scaler = preprocessing.MinMaxScaler()
3 x_scaled = min_max_scaler.fit_transform(x)
4 df_scaled = pd.DataFrame(x_scaled, columns= df2.columns)
```

d) Verificación de Outliers en las Variables

Se volvió a revisar la presencia de outliers en las variables utilizando gráficos de dispersión y boxplots.



```

1 import colorsys
2
3 # Number of colors to generate
4 num_colors = 7
5
6 # List to store the colors
7 super_bright_colors = []
8
9 # Generate colors with maximum saturation and value
10 for i in range(num_colors):
11     hue = i / num_colors # hue varies from 0 to 1
12     saturation = 1.0      # maximum saturation
13     value = 1.0          # maximum value
14     rgb = colorsys.hsv_to_rgb(hue, saturation, value)
15     super_bright_colors.append(rgb)
16
17 # Display the colors
18 print(super_bright_colors)

[(1.0, 0.0, 0.0), (1.0, 0.8571428571428571, 0.0), (0.2857142857142858, 1.0, 0.0), (0.0, 1.0, 0.5714285714285712), (0.0, 0.5714285714285716, 1.0), (0.2857142857142856, 0.0, 1.0), (1.0, 0.0, 0.8

```

5. Análisis Gráfico

Se generaron varios gráficos para entender mejor las relaciones entre las variables y su distribución. Esto incluyó histogramas, gráficos de dispersión, gráficos de caja, y matrices de correlación.

```

import colorsys

# Number of colors to generate
num_colors = 7

# List to store the colors
super_bright_colors = []

# Generate colors with maximum saturation and value
for i in range(num_colors):
    hue = i / num_colors # hue varies from 0 to 1
    saturation = 1.0      # maximum saturation
    value = 1.0          # maximum value
    rgb = colorsys.hsv_to_rgb(hue, saturation, value)
    super_bright_colors.append(rgb)

# Display the colors
print(super_bright_colors)

def plot_sales(df, df2, column, date, target):

```

```

fig, ax = plt.subplots(figsize=(40, 20))
ax.plot(df2[date], df[target], color='red', label='Sales')
ax.plot(df2[date], df[column], color=(0.12, 0.12, 0.12), label=column)

# Ajustar el tamaño de la fuente de la leyenda
ax.legend(fontsize=20)

# Ajustar el tamaño de la fuente de los ejes
ax.tick_params(axis='both', which='major', labelsize=20)

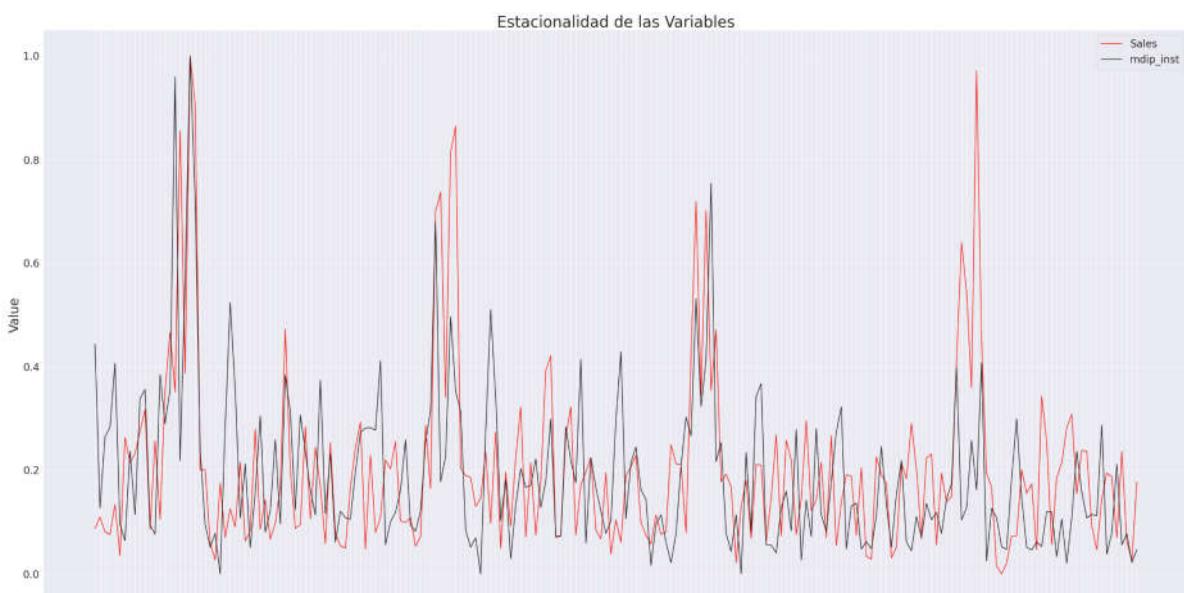
# Ajustar el tamaño de la fuente del título
ax.set_title('Estacionalidad de las Variables', fontsize=30)

# Ajustar el tamaño de la fuente de los ejes x e y
ax.set_xlabel('Index', fontsize=25)
ax.set_ylabel('Value', fontsize=25)

plt.show()
matplotlib.rcParams['figure.max_open_warning'] = 50

for column in columns:
    plot_sales(df_scaled, df, column, 'wk strt dt' , 'sales')

```



Principales indicadores del data set resultante transformado:

```

# Resumen estadístico de las variables numéricas
print(df2.describe())

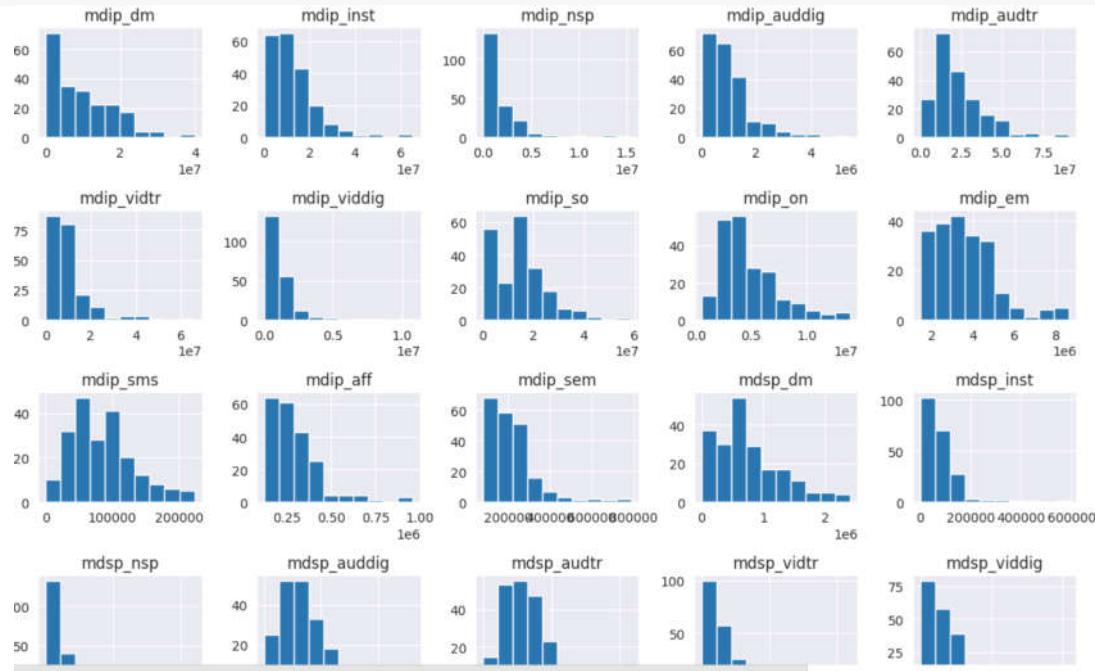
```



	mdip_dm	mdip_inst	mdip_nsp	mdip_auddig	mdip_audtr
count	2.090000e+02	2.090000e+02	2.090000e+02	2.090000e+02	2.090000e+02
mean	9.544510e+06	1.247717e+07	1.616957e+06	1.002816e+06	2.295103e+07
std	8.293082e+06	1.024959e+07	2.203341e+06	8.122848e+05	1.567124e+07
min	0.000000e+00	4.853300e+04	0.000000e+00	1.561800e+04	0.000000e+00
25%	2.087021e+06	5.304240e+06	2.542340e+05	4.577220e+05	1.236705e+07
50%	7.664954e+06	8.911466e+06	8.870720e+05	8.061170e+05	1.910160e+07
75%	1.533852e+07	1.786920e+07	2.248483e+06	1.344765e+06	2.956004e+07
max	3.979871e+07	6.545146e+07	1.553181e+07	5.418819e+06	9.066538e+07
	mdip_vidtr	mdip_viddig	mdip_so	mdip_on	mdip_em
count	2.090000e+02	2.090000e+02	2.090000e+02	2.090000e+02	2.090000e+02
mean	9.846007e+06	1.096360e+06	1.382028e+07	4.742089e+06	3.585528e+06
std	9.254274e+06	1.247631e+06	1.091767e+07	2.669165e+06	1.467660e+06
min	0.000000e+00	0.000000e+00	0.000000e+00	5.028230e+05	1.446695e+06
25%	4.378348e+06	3.931700e+05	4.305911e+06	2.893497e+06	2.416255e+06
50%	7.562079e+06	8.357480e+05	1.381069e+07	4.100627e+06	3.452070e+06
75%	1.139192e+07	1.298726e+06	1.974306e+07	6.108048e+06	4.431058e+06
max	6.582956e+07	1.078819e+07	5.882166e+07	1.394427e+07	8.614296e+06
	...	mdsp_inst	mdsp_nsp	mdsp_auddig	mdsp_audtr \
count	...	209.00000	2.090000e+02	209.00000	209.00000
mean	...	79474.858947	2.545628e+05	3844.330287	122606.298373
std	...	69706.379634	3.415438e+05	2574.853370	66082.688970
min	...	1138.73000	0.000000e+00	1.620000	0.000000
25%	...	36827.56000	5.102103e+04	2014.370000	77073.280000
50%	...	61208.05000	1.286947e+05	3237.400000	116023.470000
75%	...	99921.33000	3.619149e+05	4891.120000	158288.110000
max	...	590148.110000	2.198467e+06	13064.830000	435614.540000
	mdsp_vidtr	mdsp_viddig	mdsp_so	mdsp_on	\
count	2.090000e+02	209.00000	209.00000	209.00000	
mean	1.681586e+05	18495.923349	102010.544498	215863.998038	
std	1.685799e+05	17093.178098	95765.738144	124662.993635	
min	0.000000e+00	0.000000	0.000000	40324.230000	
25%	6.029077e+04	6840.090000	35081.440000	125161.210000	
50%	1.123532e+05	12991.090000	70799.440000	186763.140000	
75%	2.185535e+05	24837.910000	143463.910000	281687.620000	
max	1.100083e+06	104352.440000	573355.550000	695750.180000	

```
# Histograma de las variables numéricas
```

```
df2.hist(figsize=(12, 10))
plt.tight_layout()
plt.show()
```

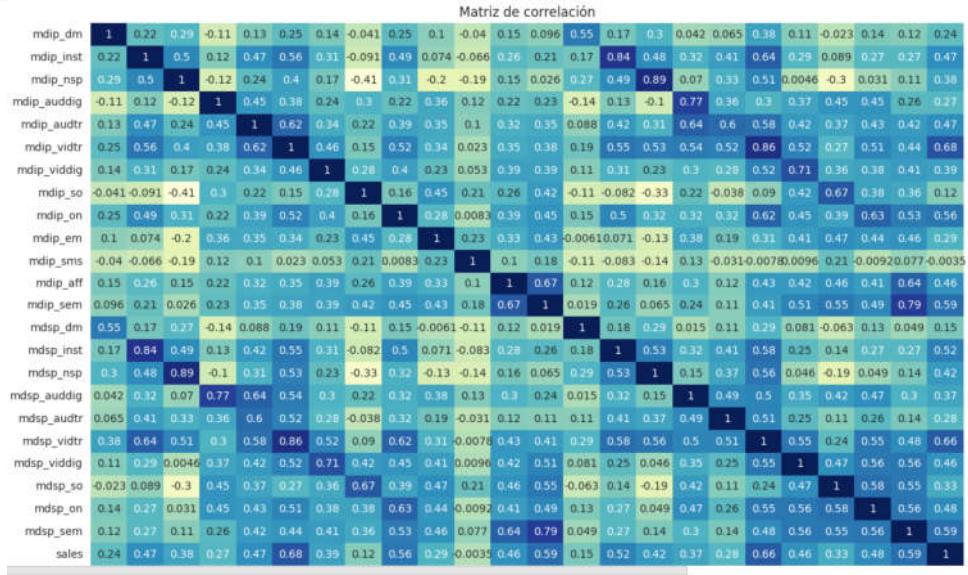


```
# Matriz de correlación
```

```
correlation_matrix = df2.corr()
```

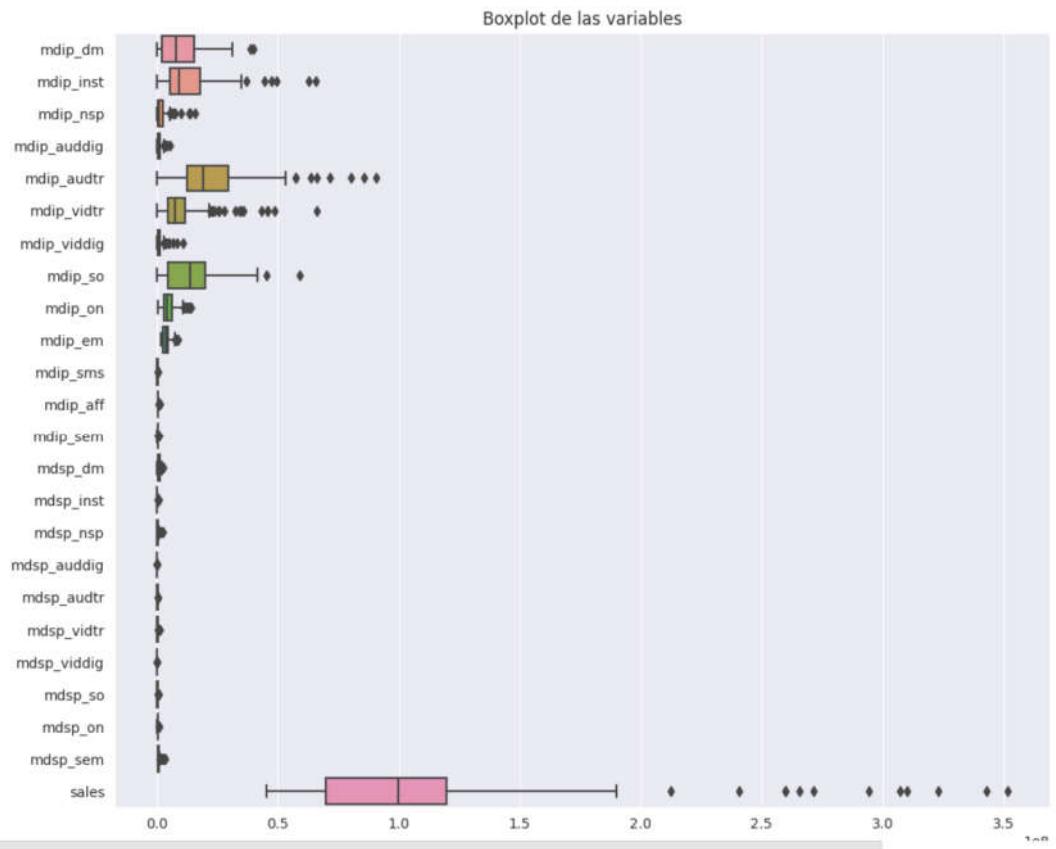
```
sns.heatmap(correlation_matrix, annot=True, cmap="YlGnBu")
plt.title("Matriz de correlación")
plt.show()
```

```
# Boxplot de las variables numéricas
plt.figure(figsize=(12, 10))
sns.boxplot(data=df2, orient="h")
plt.title("Boxplot de las variables")
plt.show()
```

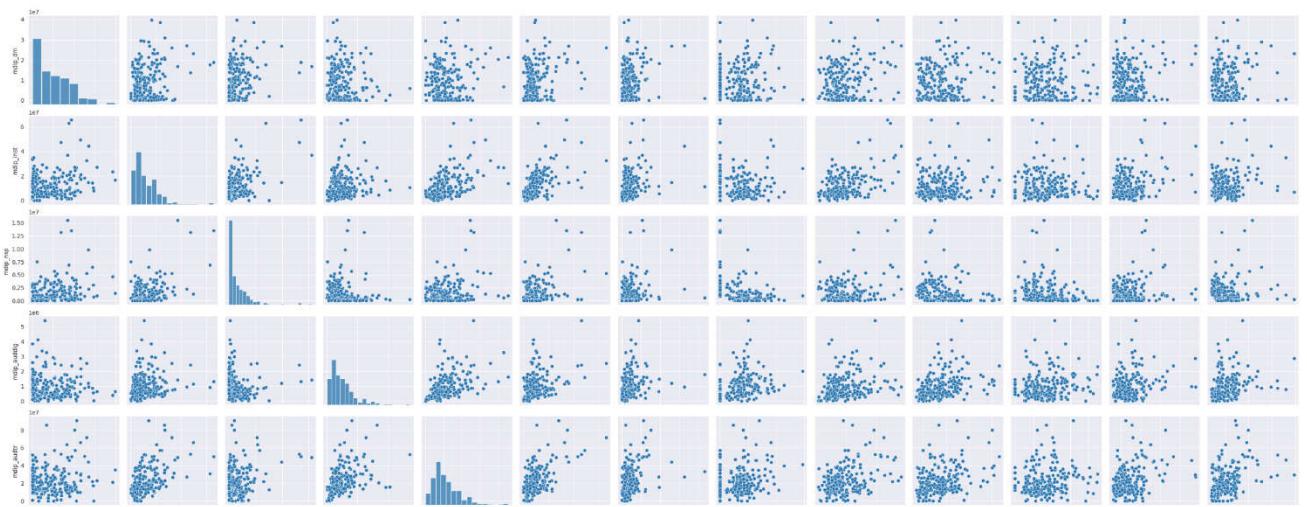


```
# Boxplot de las variables numéricas
plt.figure(figsize=(12, 10))
sns.boxplot(data=df2, orient="h")
plt.title("Boxplot de las variables")
plt.show()
```





```
# Gráfico de dispersión entre variables numéricas
sns.pairplot(df2)
plt.show()
```



En conclusión, este análisis inicial proporciona una visión detallada del conjunto de datos. Los próximos pasos pueden incluir una limpieza de datos más profunda, la ingeniería de características y la aplicación de modelos de aprendizaje automático.

Anexo III: Modelo de Regresión Lineal Multivariante



GRADO DE CIENCIA DE DATOS APLICADA

Modelo de Regresión Lineal Multivariante

18 Junio 2023

Descripción breve

Informe con el Modelo de Regresión Multivariante



Alberto de Torres Pachón

Dirección académica: Xavier Flor Responsable académico:
Trabajo final de grado 22536



ADTP

Este script está destinado a implementar un modelo de regresión lineal en un conjunto de datos de ventas. El objetivo es predecir las ventas en función de un conjunto de características. El código realiza una serie de operaciones de procesamiento de datos, análisis exploratorio y modelado de datos.

Carga de datos: El código comienza importando las bibliotecas necesarias y montando Google Drive. A continuación, carga un archivo CSV de Google Drive en un DataFrame de pandas.

```
1 def calcular_outliers(df, columnas):
2     """
3         Función que devuelve el número de filas atípicas para cada variable numérica en un dataframe
4         utilizando la regla del rango intercuartil (IQR) ampliado.
5     """
6     :param df: Un dataframe de Pandas con variables numéricas
7     :return: Un diccionario con el número de filas atípicas para cada variable numérica
8     """
9     for columna in columnas:
10         Q1 = df[columna].quantile(0.25)
11         Q3 = df[columna].quantile(0.75)
12         IQR = Q3 - Q1
13         outliers = df[(df[columna] < (Q1 - 1.5 * IQR)) | (df[columna] > (Q3 + 1.5 * IQR))]
14         num_outliers = len(outliers)
15         pct_outliers = round(num_outliers / len(df[columna]) * 100, 2)
16         print(f"{columna}: {num_outliers} outliers ({pct_outliers}%)")
17 
```



```
1 calcular_outliers(df, columnas)
2

mdip_dm: 2 outliers (0.96%)
mdip_inst: 6 outliers (2.87%)
mdip_nsp: 10 outliers (4.78%)
mdip_audig: 9 outliers (4.31%)
mdip_audtr: 8 outliers (3.83%)
mdip_vidtr: 17 outliers (8.13%)
mdip_viddig: 15 outliers (7.18%)
mdip_so: 2 outliers (0.96%)
mdip_on: 9 outliers (4.31%)
mdip_em: 7 outliers (3.35%)
mdip_sms: 5 outliers (2.39%)
mdip_aff: 10 outliers (4.78%)
mdip_sem: 9 outliers (4.31%)
mdsp_dm: 9 outliers (4.31%)
mdsp_inst: 9 outliers (4.31%)
mdsp_nsp: 8 outliers (3.83%)
mdsp_audig: 13 outliers (6.22%)
mdsp_audtr: 5 outliers (2.39%)
mdsp_vidtr: 14 outliers (6.7%)
mdsp_viddig: 10 outliers (4.78%)
mdsp_so: 8 outliers (3.83%)
mdsp_on: 5 outliers (2.39%)
mdsp_sem: 16 outliers (7.66%)
```



```

import warnings
warnings.filterwarnings("ignore")

import numpy as np
import pandas as pd
import sys
import time
from datetime import datetime
from datetime import timedelta
import matplotlib.pyplot as plt
import seaborn as sns

```

Análisis exploratorio de datos (EDA): Realiza un análisis exploratorio inicial de los datos, mostrando las primeras entradas del DataFrame, describiendo las estadísticas descriptivas, y chequeando los tipos de datos.

```
1 data = pd.read_csv('/content/drive/MyDrive/Datos/MMM_data.csv')
```

```
1 df1.describe()
```

	mdip_dm	mdip_inst	mdip_nsp	mdip_auddig	mdip_audtr	mdip_vidtr	mdip_viddig	mdip_so	mdip_on	mdip_em
count	2.090000e+02									
mean	9.544510e+06	1.247717e+07	1.616957e+06	1.002816e+06	2.295103e+07	9.846007e+06	1.096360e+06	1.382028e+07	4.742089e+06	3.585528e+06
std	8.293082e+06	1.024959e+07	2.203341e+06	8.122848e+05	1.567124e+07	9.254274e+06	1.247631e+06	1.091767e+07	2.669165e+06	1.467660e+06
min	0.000000e+00	4.853300e+04	0.000000e+00	1.561800e+04	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	5.028230e+05	1.446695e+06
25%	2.087021e+06	5.304240e+06	2.542340e+05	4.577220e+05	1.236705e+07	4.378348e+06	3.931700e+05	4.305911e+06	2.893497e+06	2.416255e+06
50%	7.664954e+06	8.911466e+06	8.870720e+05	8.061170e+05	1.910160e+07	7.562079e+06	8.357480e+05	1.381069e+07	4.100627e+06	3.452070e+06
75%	1.533852e+07	1.786920e+07	2.248483e+06	1.344765e+06	2.956004e+07	1.139192e+07	1.298726e+06	1.974306e+07	6.108048e+06	4.431058e+06
max	3.979871e+07	6.545146e+07	1.553181e+07	5.418819e+06	9.066538e+07	6.582956e+07	1.078819e+07	5.882166e+07	1.394427e+07	8.614296e+06

8 rows × 29 columns

Preprocesamiento de datos: Los datos se dividen en conjuntos de características y etiquetas (ventas). Luego, se divide este conjunto de datos en conjuntos de entrenamiento y prueba usando una división de 80-20.

```

1 # Splitting predictor data and target data:
2 x = df1.drop(['sales'], axis = 1)
3 y = df1.sales

1 from sklearn.model_selection import train_test_split
2
3 X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
4

```



Modelado: Utiliza un modelo de Regresión Lineal de la biblioteca scikit-learn. Entrena este modelo con los datos de entrenamiento y luego utiliza el modelo entrenado para hacer predicciones en el conjunto de datos de prueba.

```

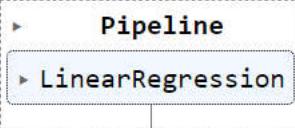
1 # Fitting the Logistic Regression model
2 from sklearn.pipeline import Pipeline
3 from sklearn.linear_model import LinearRegression

1 LR_model = LinearRegression()

1 LR_model = Pipeline([
2     ("Reg", LinearRegression())])

1 # Fitting the Logistic Regression model
2 LR_model.fit(X_train, y_train)

```



Evaluación: Evalúa el rendimiento del modelo utilizando métricas como el error cuadrático medio (MSE), la raíz del error cuadrático medio (RMSE), el error absoluto medio (MAE) y el coeficiente de determinación (R^2). También se calcula el error porcentual absoluto medio (MAPE).

```

1 # Crear un dataframe para las predicciones
2 df_predictions = pd.DataFrame(predictions, columns=['Predictions'])
3
4 # Crear un dataframe para las verdaderas etiquetas
5 df_true = pd.DataFrame(y_test.values, columns=['True Labels'])
6
7 # Concatenar ambos dataframes
8 result = pd.concat([df_true, df_predictions], axis=1)
9

1 # Seleccionar las columnas mdsp_ de xtest
2 mdsp_columns = X_test.filter(regex='mdsp_')
3
4 # Concatenar el dataframe mdsp_columns, df_true y df_predictions
5 result = pd.concat([mdsp_columns.reset_index(drop=True), df_true, df_predictions], axis=1)
6

```

```

] 1  from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
2
3  # Obtener las predicciones en el conjunto de prueba
4  predictions = LR_model.predict(X_test)
5
6  # Calcular el MSE
7  mse = mean_squared_error(y_test, predictions)
8
9  # Calcular el RMSE
10 rmse = np.sqrt(mse)
11
12 # Calcular el MAE
13 mae = mean_absolute_error(y_test, predictions)
14
15 # Calcular el R^2
16 r2 = r2_score(y_test, predictions)
17
18 # Imprimir los resultados
19 print("MSE:", mse)
20 print("RMSE:", rmse)
21 print("MAE:", mae)
22 print("R^2:", r2)
23

```

MSE: 1383008583597725.2
RMSE: 37188823.36936361
MAE: 29675654.46125685
R^2: 0.2135468032923713

Interpretación del modelo: Extrae los coeficientes y la intercepción del modelo de regresión lineal.

```

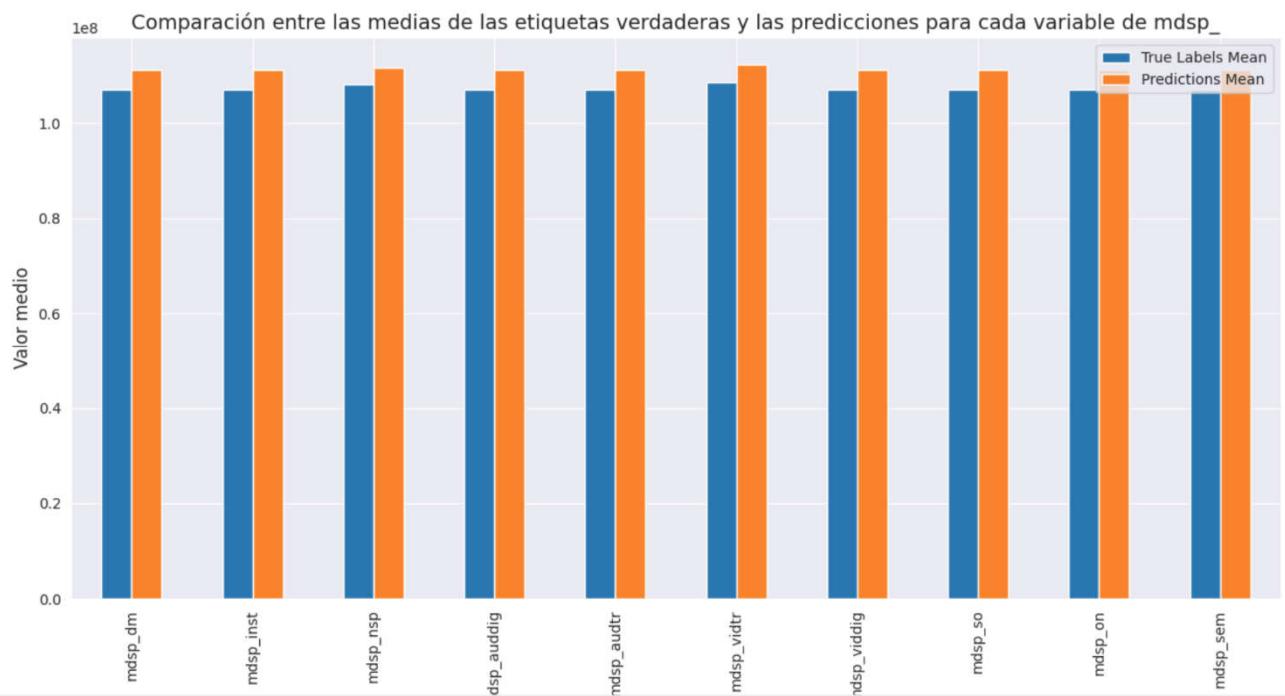
1 # Crear un DataFrame para almacenar los resultados por cada variable de mdsp_
2 mdsp_results = pd.DataFrame()
3
4 # Calcular los resultados por cada variable de mdsp_
5 for col in mdsp_columns.columns:
6     temp_df = result[['True Labels', 'Predictions', col]]
7     mdsp_results[col] = [temp_df[temp_df[col] != 0]['True Labels'].mean(), temp_df[temp_df[col] != 0]['Predictions'].mean()]
8
9 # Transponer el DataFrame para facilitar la lectura
10 mdsp_results = mdsp_results.transpose()
11
12 # Nombrar las columnas
13 mdsp_results.columns = ['True Labels Mean', 'Predictions Mean']
14
15
16

```

```

1 import matplotlib.pyplot as plt
2
3 # Crear un gráfico de barras
4 mdsp_results.plot(kind='bar', figsize=(15,7))
5
6 # Añadir título y etiquetas a los ejes
7 plt.title('Comparación entre las medias de las etiquetas verdaderas y las predicciones para cada variable de mdsp_', fontsize=14)
8 plt.xlabel('Variables de mdsp_', fontsize=12)
9 plt.ylabel('Valor medio', fontsize=12)
10
11 # Mostrar el gráfico
12 plt.show()
13

```



Análisis adicional: Realiza un análisis adicional donde se compara el valor medio de las etiquetas verdaderas y las predicciones para cada variable mdsp_. También calcula el retorno de la inversión publicitaria (ROAS) para cada canal y visualiza estas métricas utilizando gráficos de barras. Además, calcula la correlación entre las impresiones y las ventas para cada canal.

```

] 1 # Crear un DataFrame para almacenar los resultados de ROAS
2 roas_results = pd.DataFrame()
3
4 # Calcular el ROAS para cada canal
5 for col in mdsp_columns.columns:
6     roas_results[col] = result[col] / df1['sales']
7
8 # Ver los primeros registros del DataFrame de ROAS
9 print(roas_results.head())
10

```

```
    mdsp_dm  mdsp_inst  mdsp_nsp  mdsp_auddig  mdsp_audtr  mdsp_vidtr \
0  0.011861  0.000886  0.005686  0.000037  0.001131  0.002200
1  0.012366  0.000953  0.000551  0.000072  0.002239  0.004164
2  0.017174  0.000609  0.002529  0.000026  0.001322  0.001896
3  0.000042  0.000674  0.000002  0.000067  0.003175  0.002814
4  0.018500  0.001046  0.002328  0.000031  0.001373  0.001341

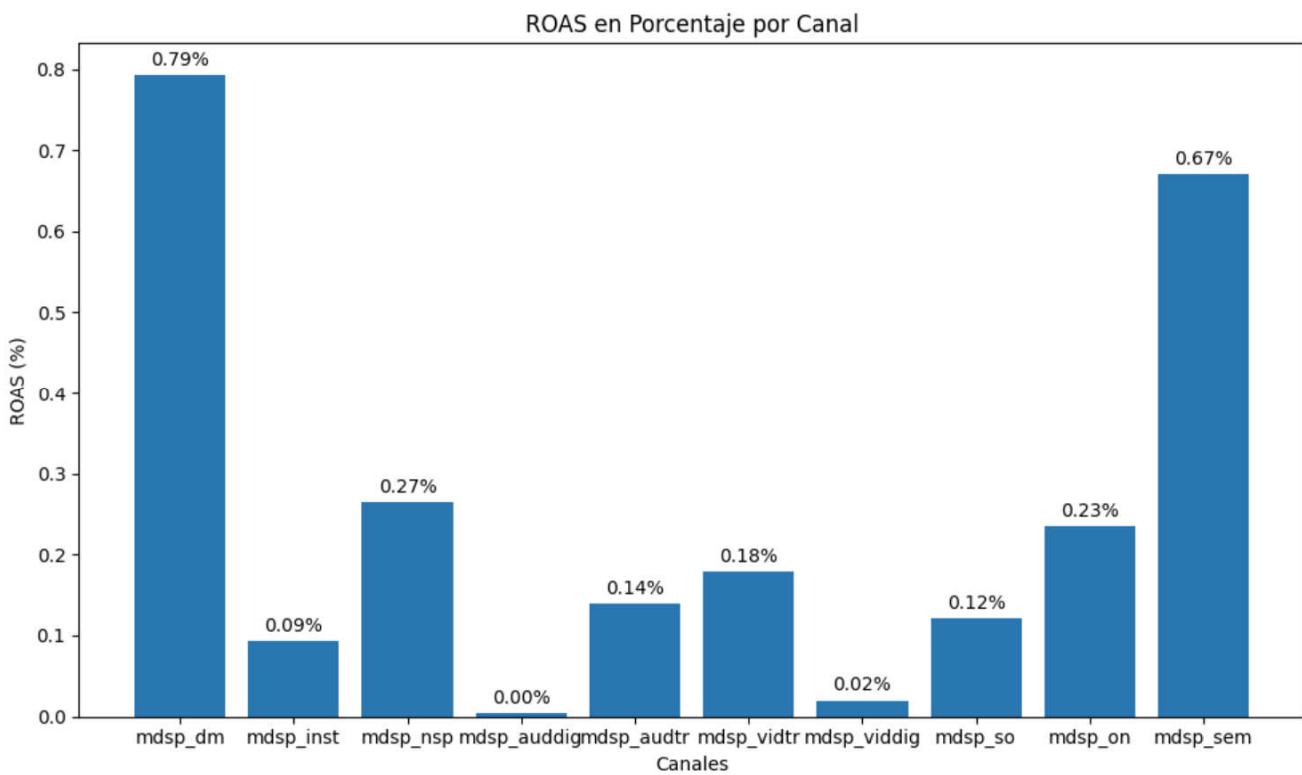
    mdsp_viddig  mdsp_so  mdsp_on  mdsp_sem
0    0.000326  0.000864  0.001427  0.005484
1    0.001106  0.003621  0.003862  0.013638
2    0.000208  0.000645  0.002656  0.004545
3    0.000411  0.002861  0.003921  0.006756
4    0.000118  0.000407  0.002608  0.005060
```

```
1 # Calcular el ROAS en porcentaje para cada canal
2 roas_results_percentage = roas_results.mean() * 100
3
4 # Ver los resultados
5 print(roas_results_percentage)
6
```

```
mdsp_dm      0.792256
mdsp_inst    0.092988
mdsp_nsp     0.265144
mdsp_auddig  0.004467
mdsp_audtr   0.139329
mdsp_vidtr   0.178610
mdsp_viddig  0.019735
mdsp_so      0.121668
mdsp_on      0.234684
mdsp_sem     0.671001
dtype: float64
```



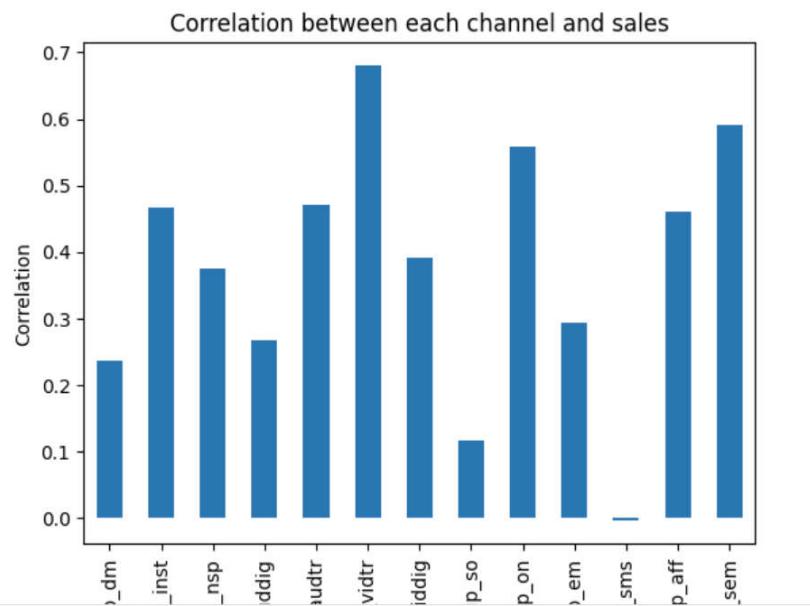
```
1 import matplotlib.pyplot as plt
2
3 # Calcular el ROAS en porcentaje para cada canal
4 roas_results_percentage = roas_results.mean() * 100
5
6 # Crear una figura y un conjunto de ejes
7 fig, ax = plt.subplots(figsize=(10, 6))
8
9 # Obtener los nombres de los canales
10 canales = roas_results_percentage.index
11
12 # Obtener los valores de ROAS en porcentaje
13 porcentajes = roas_results_percentage.values
14
15 # Crear el gráfico de barras
16 barplot = ax.bar(canales, porcentajes)
17
18 # Añadir etiquetas de porcentaje a las barras
19 for rect in barplot:
20     height = rect.get_height()
21     ax.annotate(f'{height:.2f}%', xy=(rect.get_x() + rect.get_width() / 2, height),
22                 xytext=(0, 3), textcoords='offset points', ha='center', va='bottom')
23
24 # Añadir etiquetas a los ejes
25 ax.set_xlabel('Canales')
26 ax.set_ylabel('ROAS (%)')
27 ax.set_title('ROAS en Porcentaje por Canal')
28
29 # Ajustar el espacio entre las barras
30 plt.tight_layout()
31
32 # Mostrar el gráfico
33 plt.show()
34
```



```

1 # Calcular correlación entre impresiones y ventas
2 corr = df1.filter(regex='mdip_').apply(lambda x: x.corr(df1['sales']))
3 corr = corr.to_frame(name='Correlation')
4
5 # Visualizar la correlación
6 corr.plot(kind='bar', legend=False)
7 plt.title('Correlation between each channel and sales')
8 plt.xlabel('Channels')
9 plt.ylabel('Correlation')
10 plt.show()
11

```



Modelado y evaluación adicional: Finalmente, realiza un nuevo entrenamiento y evaluación del modelo de regresión lineal con un conjunto de características reducido, seguido de un análisis de los porcentajes predichos para las columnas mdsp_.

```

1 # Obtener las predicciones para X_test
2 predictions = regression_model.predict(X_test)
3
4 # Crear un DataFrame con las predicciones
5 df_predictions = pd.DataFrame(predictions, columns=['Predictions'])
6
7 # Calcular los porcentajes para las columnas de mdsp en X_test
8 mdsp_columns_values = X_test[mdsp_columns_originales].mean()
9 valor_predicho = df_predictions.loc[0, 'Predictions']
10 porcentajes = mdsp_columns_values / sum(mdsp_columns_values) * valor_predicho
11
12 # Crear una tabla con los porcentajes
13 tabla_porcentajes = pd.DataFrame(porcentajes, columns=['Porcentaje Predicho'])
14
15 # Mostrar la tabla de porcentajes
16 print(tabla_porcentajes)
17

```

	Porcentaje Predicho
mdsp_dm	3.203444e+07
mdsp_inst	3.390185e+06
mdsp_nsp	1.173666e+07
mdsp_auddig	1.669445e+05
mdsp_audtr	5.337658e+06
mdsp_vidtr	7.226338e+06
mdsp_viddig	7.378800e+05
mdsp_so	4.616247e+06
mdsp_on	9.092798e+06
mdsp_sem	2.607698e+07

```
1 # Calcular los porcentajes de cada columna de mdsp
2 porcentajes = (mdsp_columns_values / sum(mdsp_columns_values)) * 100
3
4 # Crear una tabla con los porcentajes
5 tabla_porcentajes = pd.DataFrame(porcentajes, columns=['Porcentaje Predicho'])
6
7 # Convertir los porcentajes a formato de cadena con el simbolo %
8 tabla_porcentajes['Porcentaje Predicho'] = tabla_porcentajes['Porcentaje Predicho'].map('{:.2f}% del valor total predicho'.format)
9
10 # Mostrar la tabla de porcentajes
11 print(tabla_porcentajes)
12
```

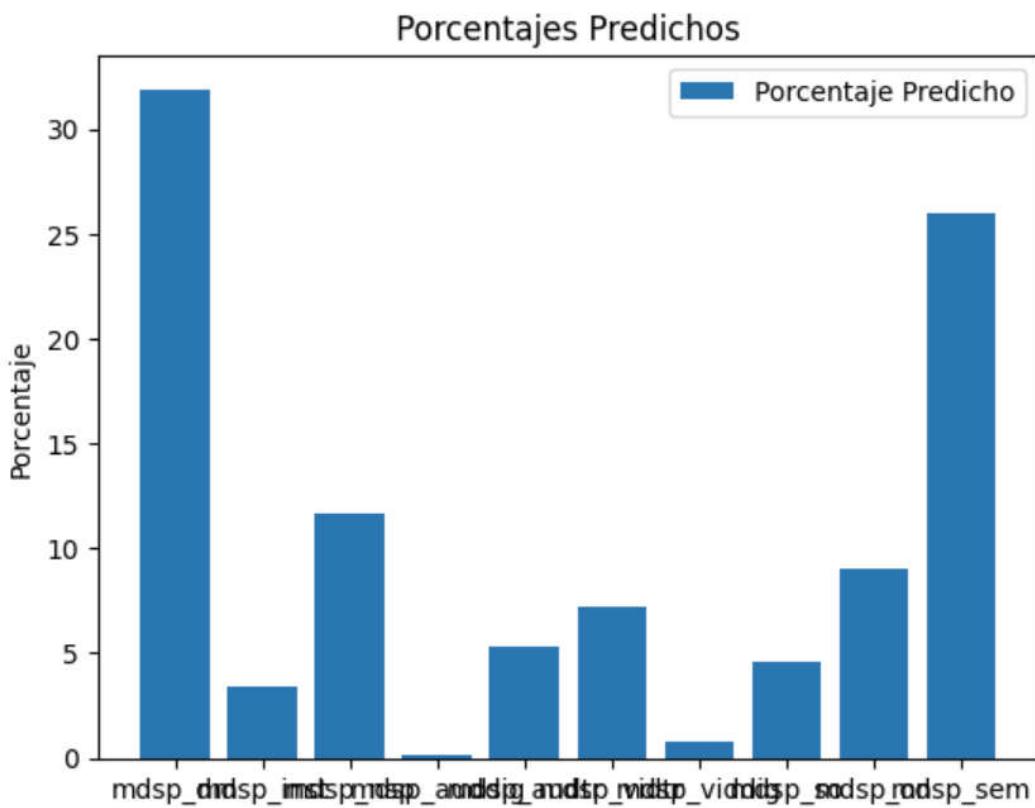
	Porcentaje Predicho
mdsp_dm	31.90% del valor total predicho
mdsp_inst	3.38% del valor total predicho
mdsp_nsp	11.69% del valor total predicho
mdsp_auddig	0.17% del valor total predicho
mdsp_audtr	5.32% del valor total predicho
mdsp_vidtr	7.20% del valor total predicho
mdsp_viddig	0.73% del valor total predicho
mdsp_so	4.60% del valor total predicho
mdsp_on	9.06% del valor total predicho
mdsp_sem	25.97% del valor total predicho

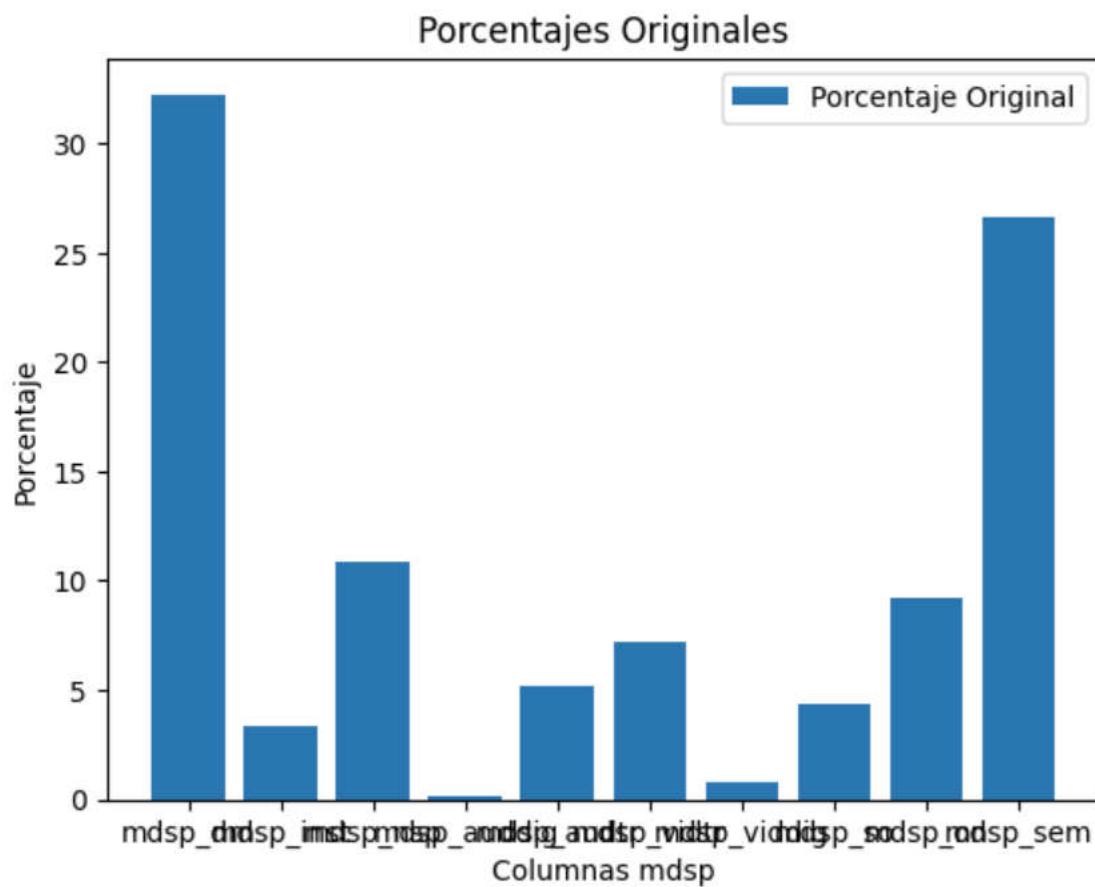
```
] 1 # Seleccionar las columnas correctas para el cálculo de los porcentajes
2 mdsp_columns_values = df_mdsp.mean()
3
4 # Obtener el valor predicho
5 valor_predicho = 100
6
7 # Calcular el porcentaje de cada columna de mdsp
8 porcentajes = mdsp_columns_values / sum(mdsp_columns_values) * valor_predicho
9
10 # Crear una tabla con los porcentajes
11 tabla_porcentajes = pd.DataFrame(porcentajes, columns=['Porcentaje Predicho'])
12
13 # Mostrar la tabla de porcentajes
14 print(tabla_porcentajes)
15
16
```

	Porcentaje Predicho
mdsp_dm	32.260263
mdsp_inst	3.383466
mdsp_nsp	10.837447
mdsp_auddig	0.163664
mdsp_audtr	5.219691
mdsp_vidtr	7.158981
mdsp_viddig	0.787423
mdsp_so	4.342873
mdsp_on	9.189931
mdsp_sem	26.656260



```
[ ] 1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 # Obtener los valores de las columnas originales de mdsp
5 valores_mdsp_originales = df1[mdsp_columns_originales].mean()
6
7 # Calcular los porcentajes de los valores originales de mdsp
8 porcentajes_originales = (valores_mdsp_originales / sum(valores_mdsp_originales)) * 100
9
10 # Crear una tabla con los porcentajes originales
11 tabla_porcentajes_originales = pd.DataFrame(porcentajes_originales, columns=['Porcentaje Original'])
12
13 # Calcular los porcentajes de cada columna de mdsp
14 porcentajes_predichos = (mdsp_columns_values / sum(mdsp_columns_values)) * 100
15
16 # Crear una tabla con los porcentajes predichos
17 tabla_porcentajes_predichos = pd.DataFrame(porcentajes_predichos, columns=['Porcentaje Predicho'])
18
19 # Graficar los porcentajes originales
20 fig, ax = plt.subplots()
21 ax.bar(tabla_porcentajes_originales.index, tabla_porcentajes_originales['Porcentaje Original'], label='Porcentaje Original')
22
23 # Agregar etiquetas y leyenda al gráfico de porcentajes originales
24 ax.set_xlabel('Columnas mdsp')
25 ax.set_ylabel('Porcentaje')
26 ax.set_title('Porcentajes Originales')
27 ax.legend()
28
29 # Mostrar el gráfico de porcentajes originales
30 plt.show()
31
32 # Graficar los porcentajes predichos
33 fig, ax = plt.subplots()
34 ax.bar(tabla_porcentajes_predichos.index, tabla_porcentajes_predichos['Porcentaje Predicho'], label='Porcentaje Predicho')
35
36 # Agregar etiquetas y leyenda al gráfico de porcentajes predichos
37 ax.set_xlabel('Columnas mdsp')
```





```

1 import pandas as pd
2
3 # Obtener los valores de las columnas originales de mdsp
4 valores_mdsp_originales = df1[mdsp_columns_originales].mean()
5
6 # Calcular los porcentajes de los valores originales de mdsp
7 porcentajes_originales = (valores_mdsp_originales / sum(valores_mdsp_originales)) * 100
8
9 # Crear una tabla con los porcentajes originales
10 tabla_porcentajes_originales = pd.DataFrame(porcentajes_originales, columns=['Porcentaje Original'])
11
12 # Calcular los porcentajes de cada columna de mdsp
13 porcentajes_predichos = (mdsp_columns_values / sum(mdsp_columns_values)) * 100
14
15 # Crear una tabla con los porcentajes predichos
16 tabla_porcentajes_predichos = pd.DataFrame(porcentajes_predichos, columns=['Porcentaje Predicho'])
17
18 # Combinar las dos tablas en una sola
19 tabla_completa = pd.concat([tabla_porcentajes_originales, tabla_porcentajes_predichos], axis=1)
20
21 # Mostrar el DataFrame con los resultados
22 print(tabla_completa)
23

```

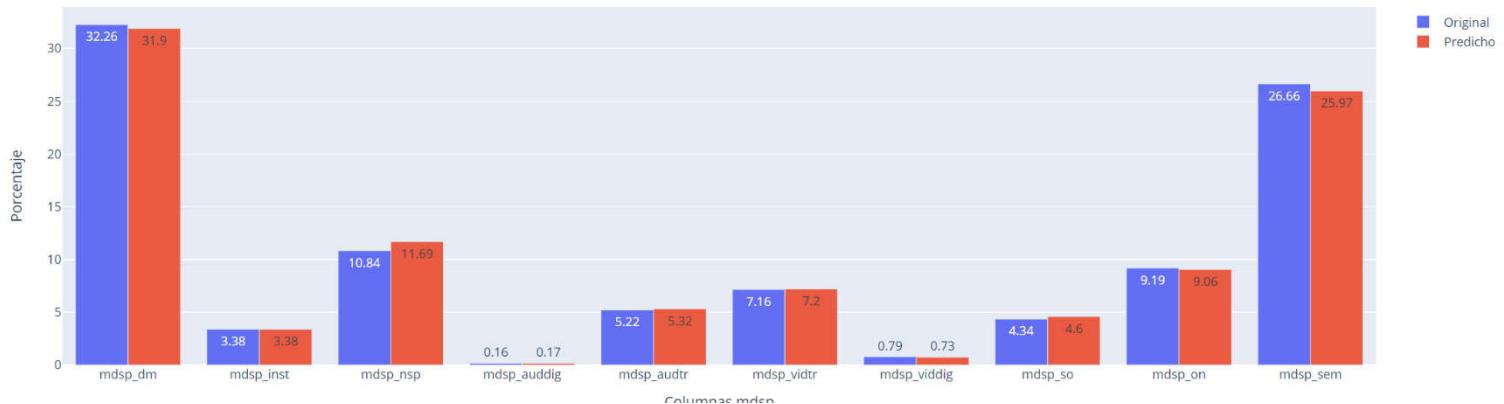
	Porcentaje Original	Porcentaje Predicho
mdsp_dm	32.260263	31.901687
mdsp_inst	3.383466	3.376135
mdsp_nsp	10.837447	11.688026
mdsp_auddig	0.163664	0.166253
mdsp_audtr	5.219691	5.315538
mdsp_vidtr	7.158981	7.196391
mdsp_viddig	0.787423	0.734822
mdsp_so	4.342873	4.597116
mdsp_on	9.189931	9.055117
mdsp_sem	26.656260	25.968915

```

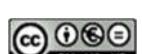
1 import plotly.graph_objects as go
2
3 # Crear una lista con los nombres de las columnas de mdsp
4 columnas_mdsp = list(tabla_completa.index)
5
6 # Obtener los valores de los porcentajes originales y predichos
7 porcentajes_originales = tabla_completa['Porcentaje Original']
8 porcentajes_predichos = tabla_completa['Porcentaje Predicho']
9
10 # Crear la figura y los subplots
11 fig = go.Figure()
12
13 # Agregar las barras para los porcentajes originales
14 fig.add_trace(go.Bar(x=columnas_mdsp, y=porcentajes_originales, name='Original', text=porcentajes_originales.round(2), textposition='auto'))
15
16 # Agregar las barras para los porcentajes predichos
17 fig.add_trace(go.Bar(x=columnas_mdsp, y=porcentajes_predichos, name='Predicho', text=porcentajes_predichos.round(2), textposition='auto'))
18
19 # Configurar el título y las etiquetas de los ejes
20 fig.update_layout(title='Porcentajes Originales y Predichos de mdsp',
21                   xaxis_title='Columnas mdsp',
22                   yaxis_title='Porcentaje')
23
24 # Mostrar la gráfica
25 fig.show()
26

```

Porcentajes Originales y Predichos de mdsp



En resumen, este código es un ejemplo de un flujo de trabajo de ciencia de datos completo, que incluye la carga y limpieza de datos, el análisis exploratorio de datos, la construcción de un modelo de machine learning, la evaluación del rendimiento del modelo y el análisis adicional basado en los resultados del modelo.



Anexo IV: Modelo de Regresión Lineal Multivariante



GRADO DE CIENCIA DE DATOS APLICADA

Modelo de Regresión Multiplicativo

18 Junio 2023

Descripción breve

Informe con el Modelo de Regresión Multiplicativo



Alberto de Torres Pachón

Dirección académica: Xavier Florit

Responsable académico: Elena Rodríguez

Trabajo final de grado 22536

117



ADTP

1. Introducción

El análisis en cuestión se basa en un conjunto de datos denominado MMM_data.csv, que se cargó desde Google Drive. Este conjunto de datos contiene diferentes variables de ventas y marketing, las cuales se utilizaron para crear un modelo matemático con el fin de predecir las ventas futuras basándose en el gasto en marketing.

```

 1 import numpy as np
 2 import pandas as pd
 3 import statsmodels.api as sm
 4 from scipy.optimize import least_squares
 5

[ ] 1 from google.colab import drive
2 drive.mount('/content/drive')

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount)

[ ] 1 df = pd.read_csv('/content/drive/MyDrive/Datos/MMM_data.csv')

```

2. Preparación y exploración de los datos

Se exploraron las columnas del conjunto de datos y se crearon nuevos subconjuntos que se centraron solo en las columnas que contienen datos de marketing (las que contienen 'mdsp' en el nombre de la columna). Además, se verificó la existencia de ciertas columnas en el conjunto de datos.

```

[ ] 1 df.columns

Index(['wk strt_dt', 'yr_nbr', 'qtr_nbr', 'prd', 'wk_nbr', 'wk_in_yr_nbr',
       'mdip_dm', 'mdip_inst', 'mdip_nsp', 'mdip_auddig', 'mdip_audtr',
       'mdip_vidtr', 'mdip_viddig', 'mdip_so', 'mdip_on', 'mdip_em',
       'mdip_sms', 'mdip_aff', 'mdip_sem', 'mdsp_dm', 'mdsp_inst', 'mdsp_nsp',
       'mdsp_auddig', 'mdsp_audtr', 'mdsp_vidtr', 'mdsp_viddig', 'mdsp_so',
       'mdsp_on', 'mdsp_sem', 'sales', 'me_ics_all', 'me_gas_dpg', 'st_ct',
       'mrkdn_valadd_edw', 'mrkdn_pdm', 'va_pub_0.15', 'va_pub_0.2',
       'va_pub_0.25', 'va_pub_0.3', 'hldy_Black Friday', 'hldy_Christmas Day',
       'hldy_Christmas Eve', 'hldy_Columbus Day', 'hldy_Cyber Monday',
       'hldy_Day after Christmas', 'hldy_Easter', 'hldy_Father's Day',
       'hldy_Green Monday', 'hldy_July 4th', 'hldy_Labor Day', 'hldy_MLK',
       'hldy_Memorial Day', 'hldy_Mother's Day', 'hldy_NYE',
       'hldy_New Year's Day', 'hldy_Pre Thanksgiving', 'hldy_Presidents Day',
       'hldy_Prime Day', 'hldy_Thanksgiving', 'hldy_Valentine's Day',
       'hldy_Veterans Day', 'seas_prd_1', 'seas_prd_2', 'seas_prd_3',
       'seas_prd_4', 'seas_prd_5', 'seas_prd_6', 'seas_prd_7', 'seas_prd_8',
       'seas_prd_9', 'seas_prd_12', 'seas_week_40', 'seas_week_41',
       'seas_week_42', 'seas_week_43', 'seas_week_44', 'seas_week_45',
       'seas_week_46', 'seas_week_47', 'seas_week_48'],
      dtype='object')

[ ] 1 mdsp_columns = [col for col in df.columns if 'mdsp' in col]
2 mdsp_columns.append('sales') # Añadimos la columna 'sales' a la lista
3 df_mdsp = df[mdsp_columns]

[ ] 1 mdsp_columns = [col for col in df.columns if 'mdsp' in col]
2 mdsp_columns.append('sales') # Añadimos la columna 'sales' a la lista
3 df_mdsp = df[mdsp_columns]

[ ] 1 df_mdsp.columns
2

Index(['mdsp_dm', 'mdsp_inst', 'mdsp_nsp', 'mdsp_auddig', 'mdsp_audtr',
       'mdsp_vidtr', 'mdsp_viddig', 'mdsp_so', 'mdsp_on', 'mdsp_sem', 'sales'],
      dtype='object')

```

3. Modelado y validación del modelo



Se utilizó el método de los mínimos cuadrados para ajustar un modelo multiplicativo a los datos. Este modelo multiplica cada variable de entrada por un coeficiente y suma todos los productos para generar una predicción. El objetivo del modelo es minimizar la suma de las diferencias cuadradas entre las predicciones del modelo y los valores reales.

El modelo se entrenó utilizando un subconjunto de los datos (80% de los datos), y luego se validó con el subconjunto de datos restante (20% de los datos). La métrica de validación utilizada fue el error porcentual absoluto medio (MAPE), que indica el porcentaje medio en el que el modelo se desvía de los valores reales.

También se calcularon otras métricas de rendimiento del modelo, como el error cuadrático medio (MSE), la raíz del error cuadrático medio (RMSE), el error absoluto medio (MAE) y el coeficiente de determinación (R^2).

```
[ ] 1 from sklearn.metrics import mean_absolute_error
2 import numpy as np
3 from scipy.optimize import least_squares
4 from sklearn.model_selection import train_test_split
5
6 def multiplicative_func(params, x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, x11, x12, x13):
7     return params[0] + params[1]*x1 + params[2]*x2 + params[3]*x3 + params[4]*x4 + params[5]*x5 + params[6]*x6 + params[7]*x7 + params[8]*x8 + params[9]*x9 + params[10]*x10 + params[11]*x11 + params[12]*x12 + params[13]*x13
8
9 def residuals_func(params, y, x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, x11, x12, x13):
10    return y - multiplicative_func(params, x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, x11, x12, x13)
11
12 # Parámetros iniciales para la optimización
13 initial_guess = [1 for _ in range(14)]
14
15 variables = ['mdip_dm', 'mdip_inst', 'mdip_nsp', 'mdip_auddig', 'mdip_audtr', 'mdip_vidtr', 'mdip_viddig', 'mdip_so', 'mdip_viss', 'mdip_viss']
16 X = df[variables]
17 y = df['sales']
18
19 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
20
21 # Ajustar el modelo
22 result = least_squares(residuals_func, initial_guess, args=(y_train, *X_train.T.values))
23
24 # Hacer las predicciones
25 y_pred = multiplicative_func(result.x, *X_test.T.values)
26 X_pred = df[variables]
27
28 # Calcular el MAPE
29 def mean_absolute_percentage_error(y_true, y_pred):
30     return np.mean(np.abs((y_true - y_pred) / y_true)) * 100
31
32 mape = mean_absolute_percentage_error(y_test, y_pred)
33 print(f'MAPE: {mape}%')
34
```

MAPE: 28.191176636439213%

4. Interpretación del modelo y predicciones

Después de ajustar el modelo, se utilizó para hacer predicciones. Además, se calculó la contribución relativa de cada canal de marketing a las ventas totales. Estas contribuciones se calcularon multiplicando las predicciones del modelo por el ROAS (retorno de la inversión publicitaria) de cada canal.

Se compararon las contribuciones relativas calculadas de cada canal con las contribuciones relativas originales en el conjunto de datos. Estas comparaciones se presentaron en un gráfico de barras que muestra las contribuciones relativas tanto predichas como originales.

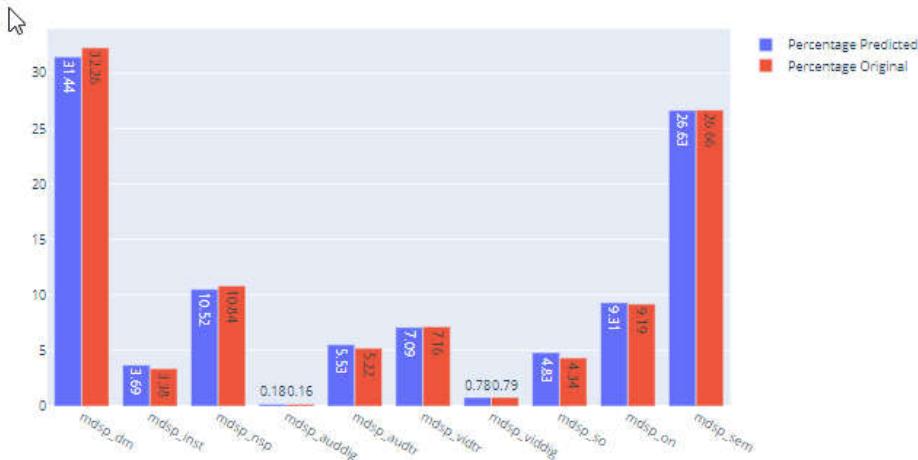


```

[ ] 1 import plotly.graph_objects as go
2
3 # Crear gráfico de barras para porcentaje predicho
4 bar1 = go.Bar(
5     x=mdsp_total.index,
6     y=mdsp_total['percentage'],
7     name='Percentage Predicted',
8     text=mdsp_total['percentage'].round(2), # Añade el texto (porcentaje) a las barras
9     textposition='auto' # Posiciona automáticamente el texto dentro de las barras
10 )
11
12 # Crear gráfico de barras para porcentaje original
13 bar2 = go.Bar(
14     x=mdsp_total.index,
15     y=mdsp_total['percentage_original'],
16     name='Percentage Original',
17     text=mdsp_total['percentage_original'].round(2), # Añade el texto (porcentaje) a las barras
18     textposition='auto' # Posiciona automáticamente el texto dentro de las barras
19 )
20
21 # Combina los gráficos
22 data = [bar1, bar2]
23
24 # Define la disposición del gráfico
25 layout = go.Layout(
26     title='Comparativa de porcentajes: Predicho vs Original',
27     barmode='group'
28 )
29
30 # Crea la figura y añade la disposición y los datos
31 fig = go.Figure(data=data, layout=layout)
32
33 # Muestra la figura
34 fig.show()
35

```

Comparativa de porcentajes: Predicho vs Original



5. Predicciones para nuevos datos

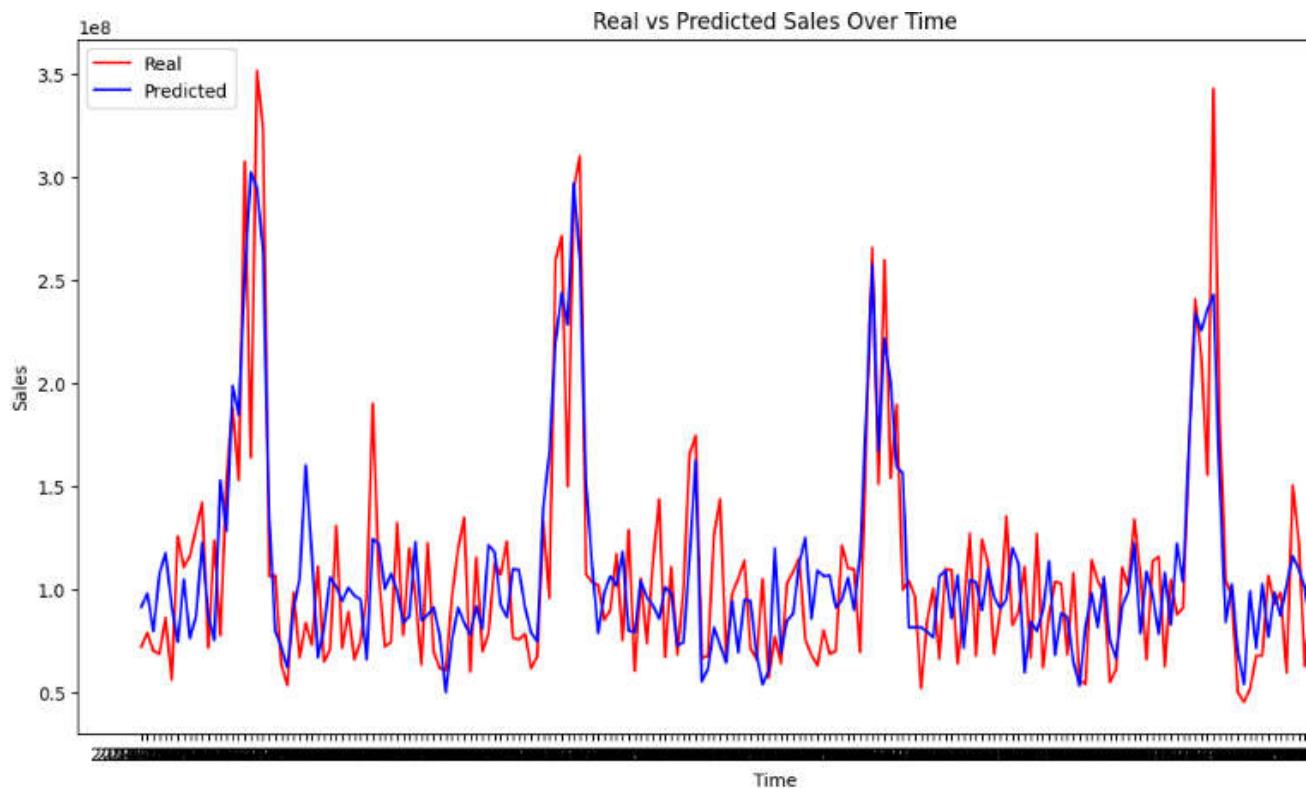
Se demostró cómo el modelo puede usarse para hacer predicciones con nuevos datos. Suponiendo que los nuevos datos estén en el mismo formato que los datos originales, simplemente se pueden alimentar al modelo para generar predicciones de ventas.

```
[ ] 1 # Asumiendo que result.x contiene los coeficientes optimizados y que new_data es un DataFrame que contiene los nuevos datos
2
3 # Poner los nuevos datos en el mismo formato que los datos originales
4 X_new = new_data[['mdip_dm', 'mdip_inst', 'mdip_nsp', 'mdip_auddig', 'mdip_audtr', 'mdip_vidtr', 'mdip_viddig', 'mdip_so', 'mdip_vnsp']]
5
6 # Generar las predicciones
7 y_pred = multiplicative_func(result.x, *X_new.T.values)
8
9 # y_pred ahora contiene las predicciones de ventas para los nuevos datos
10
```

6. Visualización de las ventas reales y predichas a lo largo del tiempo

Finalmente, se visualizaron las ventas reales y predichas a lo largo del tiempo. Este gráfico proporciona una visión clara de cómo las predicciones del modelo se comparan con los valores reales a lo largo del tiempo.

```
1 import matplotlib.pyplot as plt
2
3 # Ordenar el DataFrame original 'df' por la columna 'wk strt_dt'
4 df_sorted = df.sort_values(by='wk strt_dt')
5
6 # Filtrar las columnas relevantes para el conjunto de prueba
7 X_test = df_sorted[['mdip_dm', 'mdip_inst', 'mdip_nsp', 'mdip_auddig', 'mdip_audtr', 'mdip_vidtr', 'mdip_viddig', 'mdip_so', 'mdip_vnsp']]
8 y_test = df_sorted['sales']
9
10 # Predicciones en el conjunto de prueba
11 y_pred = multiplicative_func(result.x, *X_test.T.values)
12
13 # Crear un gráfico de las ventas reales y predichas a lo largo del tiempo
14 plt.figure(figsize=(14, 7))
15 plt.plot(df_sorted['wk strt_dt'], y_test, label='Real', color='red')
16 plt.plot(df_sorted['wk strt_dt'], y_pred, label='Predicted', color='blue')
17 plt.xlabel('Time')
18 plt.ylabel('Sales')
19 plt.title('Real vs Predicted Sales Over Time')
20 plt.legend(loc='upper left')
21 plt.show()
```



Anexo V: Modelo Robyn



GRADO DE CIENCIA DE DATOS APLICADA

Modelo Robyn

18 Junio 2023

Descripción breve

Informe con el Modelo del Modelo Robyn



Alberto de Torres Pachón

Dirección académica: Xavier Florit

Responsable académico: Elena Rodríguez
Trabajo final de grado 22536



ADTP

Este código está implementando un modelo de Marketing Mix Modeling (MMM) utilizando la biblioteca Robyn de Facebook. Aquí hay un resumen de lo que hace cada sección de código:

Instalación de paquetes y configuración del entorno: Esta sección instala los paquetes necesarios, incluyendo Robyn, y configura el entorno para que el modelo pueda correr.

```
# install.packages
library(Robyn)

install.packages("extrafont")

# load library
library(extrafont)

# Import fonts available on your system. It might take a few minutes.
font_import()

# Load the fonts into R
loadfonts(device = "win")

## Force multi-core use when running RStudio
Sys.setenv(R_FUTURE_FORK_ENABLE = "true")
options(future.fork.enable = TRUE)

# Set to FALSE to avoid the creation of files locally
create_files <- TRUE
```

Importación de datos: Este bloque de código carga los datos del archivo MMM_data.csv, los formatea adecuadamente y carga los datos de vacaciones.

```
#### Step 1: Load data

## dataset
setwd("C:/Users/AlbertodeTorres/OneDrive - Nektiu S.L/UOC/TFG_1/PEC 2")
library(readr)
MMM_data <- read_csv("MMM_data.csv")
#View(MMM_data)

# Convert the variable to a date format
library(dplyr)
df <- MMM_data %>% mutate(date = as.Date(wk strt_dt, format = "%Y-%m-%d"))
#View(df)

## holidays from Prophet library

data("dt_prophet_holidays")
head(dt_prophet_holidays)

# Directory to export results
#robyn_object <- "~/Desktop"
```



Definición de las variables de entrada: Aquí se definen las variables de entrada para el modelo, incluyendo las variables dependientes e independientes, las variables de los medios pagados, las variables orgánicas y las variables de contexto. También se establece el rango de fechas para el modelo y el método de adstock (cómo se distribuye el impacto de la publicidad a lo largo del tiempo).

Definición y visualización de los hiperparámetros: Los hiperparámetros son las variables que el algoritmo de aprendizaje puede ajustar para mejorar la precisión del modelo. En este bloque de código, se definen los rangos para los hiperparámetros y se proporcionan visualizaciones para entender cómo los hiperparámetros afectan las curvas de adstock y saturación. También se establece el tamaño de la muestra de entrenamiento.

```
df <- df[,1:35]
```

```
InputCollect <- robyn_inputs(
  dt_input = df,
  dt_holidays = dt_prophet_holidays,
  date_var = "wk strt_dt", # date format must be "2020-01-01"
  dep_var = "sales", # there should be only one dependent variable
  dep_var_type = "revenue", # "revenue" (ROI) or "conversion" (CPA)
  prophet_vars = c("trend", "season", "holiday"), # "trend", "season", "weekday" & "holiday"
  prophet_country = "US", # input one country. dt_prophet_holidays includes 59 countries by default
  context_vars = c("mrkdn_valadd_edw", "mrkdn_pdm", 'me_ics_all','me_gas_dpg'), # e.g. competitors, discount, unemployment etc
  paid_media_spends = c("mdsp_dm", "mdsp_inst", "mdsp_nsp", "mdsp_auddig", "mdsp_audtr",
  'mdsp_vidtr','mdsp_viddig', 'mdsp_so', 'mdsp_on', 'mdsp_sem'), # mandatory input
  paid_media_vars = c("mdip_dm", "mdip_inst", "mdip_nsp", "mdip_auddig", "mdip_audtr",
  'mdip_vidtr', 'mdip_viddig', 'mdip_so', 'mdip_on', 'mdip_sem'), # mandatory.
  # paid_media_vars must have same order as paid_media_spends. Use media exposure metrics like
  # impressions, GRP etc. If not applicable, use spend instead.
  organic_vars = c("mdip_em", 'mdip_sms', 'mdip_aff'), # marketing activity without media spend
  # factor_vars = c("events"), # force variables in context_vars or organic_vars to be categorical
  window_start = "2014-08-03",
  window_end = "2018-07-29",
  adstock = "geometric" # geometric, weibull_cdf or weibull_pdf.
)
print(InputCollect)
```

2a-2: Second, define and add hyperparameters

```
## hyperparameter names are based on paid_media_spends names too.
hyper_names(adstock = InputCollect$adstock, all_media = InputCollect$all_media)
```

Guide to setup & understand hyperparameters

```
## hyperparameters have four components:
## Adstock parameters (theta or shape/scale).
```



```

## Saturation parameters (alpha/gamma).
## Regularisation parameter (lambda). No need to specify manually.
## Time series validation parameter (train_size).

## 1. IMPORTANT: set plot = TRUE to create example plots for adstock & saturation
## hyperparameters and their influence in curve transformation
plot_adstock(plot = TRUE)
plot_saturation(plot = TRUE)

```

Comprobación del ajuste al gasto: Esta sección comprueba si las curvas de adstock y saturación se ajustan bien a los gastos en medios pagados.

```

##### Check spend exposure fit if available
if (length(InputCollect$exposure_vars) > 0) {
  lapply(InputCollect$modNLS$plots, plot)
}
# Library and environment for Reticulate/Nevergrad.
library("reticulate")
conda_create("r-reticulate")
conda_install("r-reticulate", "nevergrad", pip=TRUE)
use_condaenv("r-reticulate")

```

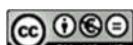
Configuración de Python y Nevergrad: Este bloque de código configura el entorno Python para que pueda ser utilizado en R a través del paquete reticulate. En particular, este código está configurado para utilizar el optimizador Nevergrad, una biblioteca de optimización de código abierto de Facebook.

1. load reticulate
- library("reticulate")
- # 2. Install conda if not available
- #install_miniconda()
- # 3. create virtual environment
- conda_create("r-reticulate")
- # 4. use the environment created
- use_condaenv("r-reticulate")
- # 5. point Python path to the python file in the virtual environment. Below is
- # an example for MacOS M1 or above. The "~" is my home dir "/Users/gufengzhou".
- # Show hidden files in case you want to locate the file yourself

```

Sys.setenv(RETICULATE_PYTHON = "C:/Users/alber/anaconda3/envs/r-reticulate/python.exe")
# 6. Check python path
py_config() # If the first path is not as 5, do 7
# 7. Restart R session, run #5 first, then load library("reticulate"), check
# py_config() again, python should have path as in #5
# 8. Install numpy if py_config shows it's not available
conda_install("r-reticulate", "numpy", pip=TRUE)
# 9. Install nevergrad
conda_install("r-reticulate", "nevergrad", pip=TRUE)
# 10. If successful, py_config() should show numpy and nevergrad with installed paths
# 11. Everytime R session is restarted, you need to run #4 first to assign python
# path before loading Robyn

```



```
py_config()
```

En la segunda parte del script de Robyn, el flujo de trabajo se centra en la asignación de presupuesto, la actualización del modelo y la obtención de retornos marginales basados en los resultados del modelo.

```
# Run all trials and iterations.
OutputModels <- robyn_run(
  InputCollect = InputCollect, # feed in all model specification
  cores = NULL, # NULL defaults to (max available - 1)
  iterations = 2000, # 2000 recommended for the dummy dataset with no calibration
  trials = 5, # 5 recommended for the dummy dataset
  ts_validation = TRUE, # 3-way-split time series for NRMSE validation.
  add_penalty_factor = FALSE # Experimental feature. Use with caution.
)
print(OutputModels)
```

Obtener la asignación de presupuesto basada en el modelo seleccionado

Se utiliza la función robyn_allocator para asignar el presupuesto a los canales de medios pagados en función del modelo seleccionado. Se presentan tres escenarios en los que se pueden asignar presupuestos con base en diferentes parámetros, como el rango de fechas y el presupuesto total.

```
OutputModels$convergence$moo_distrb_plot
OutputModels$convergence$moo_cloud_plot

## Check time-series validation plot (when ts_validation == TRUE)
# Read more and replicate results: ?ts_validation
if (OutputModels$ts_validation) OutputModels$ts_validation_plot

## Calculate Pareto fronts, cluster and export results and plots. See ?
OutputCollect <- robyn_outputs(
  InputCollect, OutputModels,
  pareto_fronts = "auto", # automatically pick how many pareto-fronts to fill min_candidates
  (100)
  # min_candidates = 100, # top pareto models for clustering. Default to 100
  # calibration_constraint = 0.1, # range c(0.01, 0.1) & default at 0.1
  csv_out = "pareto", # "pareto", "all", or NULL (for none)
  clusters = TRUE, # Set to TRUE to cluster similar models by ROAS. See ?robyn_clusters
  export = create_files, # this will create files locally
  #plot_folder = robyn_object, # path for plots exports and files creation
  plot_pareto = create_files # Set to FALSE to deactivate plotting and saving model one-pagers
)
print(OutputCollect)
```

Actualizar el modelo basado en el modelo seleccionado y los resultados guardados

El modelo puede ser actualizado o refrescado usando la función robyn_refresh. Esta función es útil para actualizar el modelo en "períodos razonables". Si la mayor parte de los datos es nueva, puede ser mejor reconstruir el modelo desde cero.

```
## Compare all model one-pagers and select one that mostly reflects your business reality
print(OutputCollect)
select_model <- "1_256_7" # Pick one of the models from OutputCollect to proceed
```



```
##### Version >=3.7.1: JSON export and import (faster and lighter than RDS files)
ExportedModel <- robyn_write(InputCollect, OutputCollect, select_model, export = create_files)
print(ExportedModel)
```

Obtener la recomendación de asignación de presupuesto basada en las carreras seleccionadas

Una vez actualizado el modelo, se puede obtener una recomendación de asignación de presupuesto para un escenario en el que se espera una cantidad específica de gasto.

```
AllocatorCollect1 <- robyn_allocator(
  InputCollect = InputCollect,
  OutputCollect = OutputCollect,
  select_model = select_model,
  date_range = NULL, # When NULL, will set last month (30 days, 4 weeks, or 1 month)
  channel_constr_low = 0.7,
  channel_constr_up = c(1.2, 1.5, 1.5, 1.5, 1.5),
  channel_constr_multiplier = 3,
  scenario = "max_response_expected_spend",
  export = create_files
)
# Print the allocator's output summary
print(AllocatorCollect1)
# Plot the allocator one-pager
plot(AllocatorCollect1)

# Case 2: date_range defined, total_budget NULL (mean spend of date_range as initial spend)
AllocatorCollect2 <- robyn_allocator(
  InputCollect = InputCollect,
  OutputCollect = OutputCollect,
  select_model = select_model,
  date_range = "last_26", # Last 26 periods, same as c("2018-07-09", "2018-12-31")
  channel_constr_low = c(0.8, 0.7, 0.7, 0.7, 0.7),
  channel_constr_up = c(1.2, 1.5, 1.5, 1.5, 1.5),
  channel_constr_multiplier = 3,
  scenario = "max_historical_response",
  export = create_files
)
print(AllocatorCollect2)
plot(AllocatorCollect2)
```

Obtener los retornos marginales

Finalmente, la función `robyn_response` se utiliza para obtener las curvas de respuesta de saturación para los canales de medios pagados y orgánicos. También se puede utilizar para obtener el ROI marginal del próximo gasto en un nivel específico de gasto para un canal de medios específico.

```
AllocatorCollect3 <- robyn_allocator(
  InputCollect = InputCollect,
  OutputCollect = OutputCollect,
  select_model = select_model,
  # date_range = "last_4",
  total_budget = 5000000,
  channel_constr_low = 0.7,
```



```

channel_constr_up = c(1.2, 1.5, 1.5, 1.5, 1.5),
channel_constr_multiplier = 5,
scenario = "max_historical_response",
export = create_files
)
print(AllocatorCollect3)
plot(AllocatorCollect3)

## A csv is exported into the folder for further usage. Check schema here:
## https://github.com/facebookexperimental/Robyn/blob/main/demo/schema.R

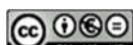
## QA optimal response
# Pick any media variable: InputCollect$all_media
select_media <- "search_S"
# For paid_media_spends set metric_value as your optimal spend
metric_value <- AllocatorCollect1$dt_optimOut$optmSpendUnit[
  AllocatorCollect1$dt_optimOut$channels == select_media
]; metric_value
## For paid_media_vars and organic_vars, manually pick a value
# metric_value <- 10000

## Saturation curve for adstocked metric results (example)
robyn_response(
  InputCollect = InputCollect,
  OutputCollect = OutputCollect,
  select_model = select_model,
  metric_name = select_media,
  metric_value = metric_value,
  metric_ds = "last_5"
)

#####
En este punto, también es posible recrear modelos antiguos y replicar resultados utilizando la función robyn_recreate. Esto es útil si se desea reproducir los resultados obtenidos previamente.

json_file <- "~/Desktop/Robyn_202211211853_init/RobynModel-1_100_6.json"
RobynRefresh <- robyn_refresh(
  json_file = json_file,
  dt_input = dt_simulated_weekly,
  dt_holidays = dt_prophet_holidays,
  refresh_steps = 13,
  refresh_iters = 1000, # 1k is an estimation
  refresh_trials = 1
)
# Now refreshing a refreshed model, following the same approach
json_file_rf1 <- "~/Desktop/Robyn_202208231837_init/Robyn_202208231841_rf1/RobynModel-1_12_5.json"
RobynRefresh <- robyn_refresh(
  json_file = json_file_rf1,
  dt_input = dt_simulated_weekly,
  dt_holidays = dt_prophet_holidays,
  refresh_steps = 7,
)

```



```
refresh_iters = 1000, # 1k is an estimation  
refresh_trials = 1  
)
```

Al final, se guardan los resultados, se imprimen resúmenes de los modelos y se crean gráficos para la visualización de los resultados.



Anexo VI: Modelo Bayesiano con efectos de marketing



GRADO DE CIENCIA DE DATOS APLICADA

Modelo Bayesiano con Efectos de Marketing

18 Junio 2023

Descripción breve

Informe con el Modelo Bayesiano con efectos de Marketing

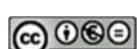


Alberto de Torres Pachón

Dirección académica: Xavier Florit

Responsable académico: Elena Rodríguez

Trabajo final de grado 22536



ADTP

Este script primero instala e importa los módulos necesarios, incluyendo lightweight_mmm, un paquete de Python para modelar la mezcla de medios.

```
[ ] 1 # First would be to install lightweight_mmm
[ ] 2 !pip install --upgrade git+https://github.com/google/lightweight_mmm.git

[ ] 1 # Import jax.numpy and any other library we might need.
[ ] 2 import jax.numpy as jnp
[ ] 3 import numpyro
[ ] 4 import pandas as pd
[ ] 5 import numpy as np
[ ] 6
[ ] 7 import tensorflow as tf

[ ] 1 # Import the relevant modules of the library
[ ] 2 from lightweight_mmm import lightweight_mmm
[ ] 3 from lightweight_mmm import optimize_media
[ ] 4 from lightweight_mmm import plot
[ ] 5 from lightweight_mmm import preprocessing
[ ] 6 from lightweight_mmm import utils
```

Luego, se cargan los datos a través de un archivo CSV. Este archivo parece contener varias columnas correspondientes a diferentes medios de publicidad y ventas. Los datos se dividen en conjuntos de prueba y de entrenamiento.



```

1  from google.colab import drive
2  drive.mount('/content/drive')

[ ] 1  # Four years' (209 weeks) records of sales, media impression and media spending at weekly level.
2  df = pd.read_csv('/content/drive/MyDrive/Datos/MMM_data.csv')

[ ] 1  SEED= 105
2  data_size = len(df)
3  data_size

209

[ ] 1  # Seleccionar las columnas de interés del dataframe
2  mdip_cols = ['mdip_dm', 'mdip_inst', 'mdip_nsp', 'mdip_auddig', 'mdip_audtr', 'mdip_vidtr', 'mdip_viddig', 'mdip_so', 'mdip_on', 'extra_cols = ['mdip_em', 'mdip_sms', 'mdip_aff']

5

6  # Convertir los datos seleccionados en tensores de NumPy
7  media_data = np.array(df[mdip_cols])
8  extra_features = np.array(df[extra_cols])
9  target = np.array(df['sales'])

11 # Imprimir los tamaños de los tensores resultantes
12 print(f'Tamaño del tensor media_data: {media_data.shape}')
13 print(f'Tamaño del tensor extra_features: {extra_features.shape}')
14 print(f'Tamaño del tensor target: {target.shape}')

15

[ ] Tamaño del tensor media_data: (209, 10)
Tamaño del tensor extra_features: (209, 3)
Tamaño del tensor target: (209,)

[ ] 1  # Agregar una nueva columna "costs" al dataframe
2  cost_cols = ['mdsp_dm', 'mdsp_inst', 'mdsp_nsp', 'mdsp_auddig', 'mdsp_audtr', 'mdsp_vidtr', 'mdsp_viddig', 'mdsp_so', 'mdsp_on', 'costs = np.array(df[cost_cols].sum())
4
5  print(f'Tamaño del tensor costs: {costs.shape}')

Tamaño del tensor costs: (10,)

[ ] 1  costs = costs.astype('float32')
2  costs

array([1.5837336e+08, 1.6610246e+07, 5.3203628e+07, 8.0346500e+05,
       2.5624716e+07, 3.5145152e+07, 3.8656480e+06, 2.1320204e+07,
       4.5115576e+07, 1.3086197e+08], dtype=float32)

[ ] 1  media_data = media_data.astype('float32')
2  media_data

array([[4.8638850e+06, 2.9087520e+07, 2.4219330e+06, ..., 0.0000000e+00,
       3.2710070e+06, 8.3054000e+04],
       [2.0887502e+07, 8.3451200e+06, 3.9844940e+06, ..., 0.0000000e+00,
       4.2607150e+06, 8.3124000e+04],
       [1.1097724e+07, 1.7276800e+07, 1.8468320e+06, ..., 0.0000000e+00,
       4.4059920e+06, 7.9768000e+04],
       ...,
       [2.2250920e+06, 5.1023530e+06, 0.0000000e+00, ..., 1.3695379e+07,
       3.7190320e+06, 1.6252000e+05],
       [1.7544332e+07, 1.4785660e+06, 1.8910000e+03, ..., 2.2415692e+07,
       3.1098110e+06, 1.3414500e+05],
       [3.0800000e+04, 3.10867080e+06, 4.5000000e+02, ..., 1.2657494e+07,
       5.2603820e+06, 2.5090700e+05]], dtype=float32)

[ ] 1  extra_features = extra_features.astype('float32')
2  extra_features

array([[1514755.,    27281.,   197828.],
       [2234569.,    27531.,   123688.],
       [1616990.,    55267.,   186781.],
       [1897998.,    32470.,   122389.],
       [3.7190320e+06, 1.6252000e+05],
       [1.7544332e+07, 1.4785660e+06, 1.8910000e+03, ..., 2.2415692e+07,
       3.1098110e+06, 1.3414500e+05],
       [3.0800000e+04, 3.10867080e+06, 4.5000000e+02, ..., 1.2657494e+07,
       5.2603820e+06, 2.5090700e+05]], dtype=float32)

[ ] 1  extra_features = extra_features.astype('float32')
2  extra_features

[3919402.,    19453.,   171538.],
[3994731.,    68106.,   455603.],
[5004684.,    73392.,   644982.],
[4792193.,    76833.,   627599.],
[3011480.,    78243.,   454282.],
[2725976.,    47346.,   443262.],
[2614827.,    58661.,   415309.],
[3348174.,    49829.,   384458.],
[3324061.,    75335.,   332465.],
[1643881.,    50603.,   321684.],
[2538628.,    86999.,   272314.],
[2222585.,    25552.,   322420.],
[2882195.,    51349.,   262676.],
[1733570.,    133039.,   243343.],
[1791431.,    26063.,   293187.],
[1524411.,    89467.,   147659.],
[1723349.,    52347.,   182776.],
[2730029.,    52601.,   172944.],
[3179412.,    52840.,   152907.],
[3389875.,    91415.,   153482.],
[1818036.,    137757.,   211464.],
[3377012.,    92239.,   156635.],
[3192477.,    93108.,   165370.],
[1446695.,    93515.,   272759.],
[4060529.,    54819.,   291715.],

```

Posteriormente, los datos se escalan utilizando la clase CustomScaler proporcionada por el paquete lightweight_mmm. Este paso es crucial ya que el escalamiento de los datos puede mejorar significativamente el rendimiento de los modelos de aprendizaje automático.

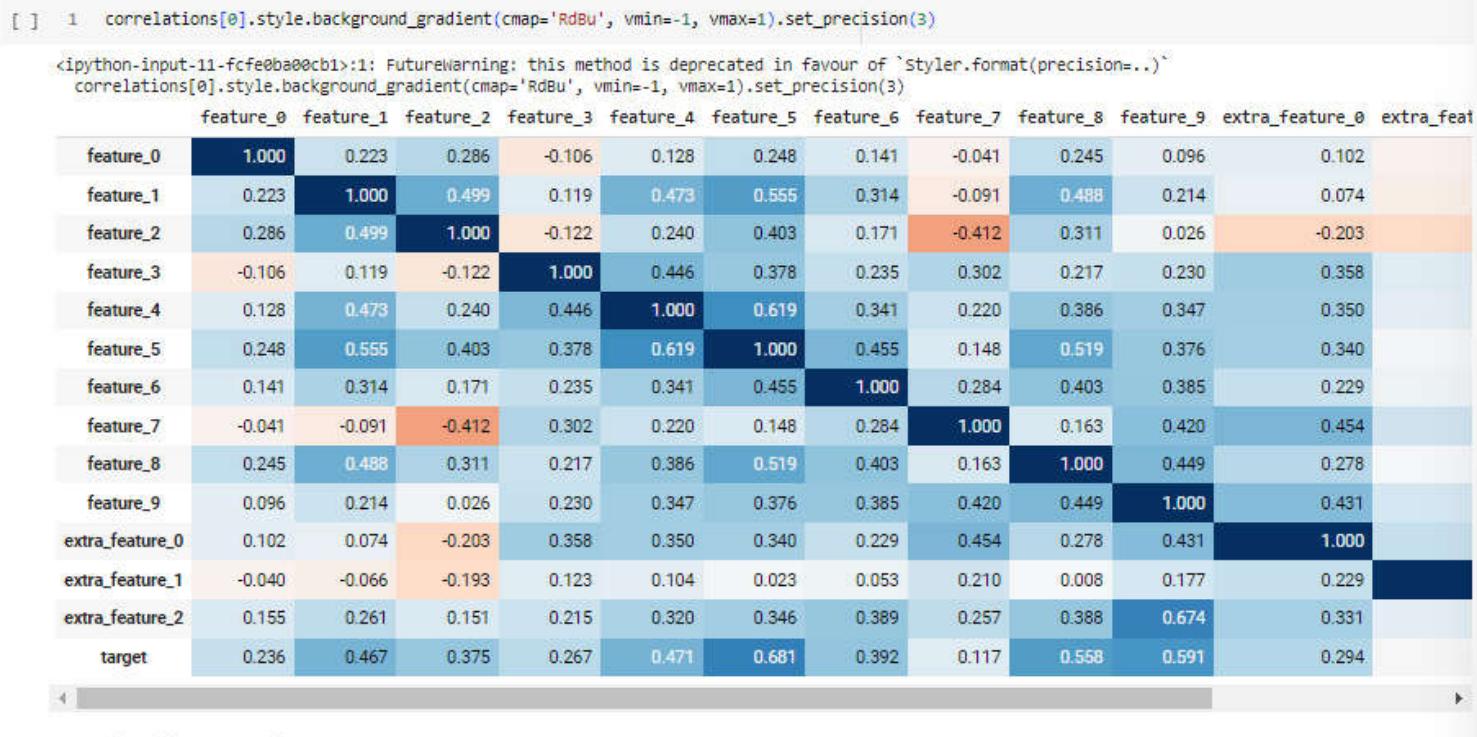
```
[ ] 1 # Split and scale data.
2 split_point = data_size - 47
3 # Media data
4 media_data_train = media_data[:split_point, ...]
5 media_data_test = media_data[split_point:, ...]
6 # Extra features
7 extra_features_train = extra_features[:split_point, ...]
8 extra_features_test = extra_features[split_point:, ...]
9 # Target
10 target_train = target[:split_point]

[ ] 1 media_scaler = preprocessing.CustomScaler(divide_operation=jnp.mean)
2 extra_features_scaler = preprocessing.CustomScaler(divide_operation=jnp.mean)
3 target_scaler = preprocessing.CustomScaler(divide_operation=jnp.mean)
4 cost_scaler = preprocessing.CustomScaler(divide_operation=jnp.mean, multiply_by=0.15)
5
6 media_data_train = media_scaler.fit_transform(media_data_train)
7 extra_features_train = extra_features_scaler.fit_transform(extra_features_train)
8 target_train = target_scaler.fit_transform(target_train)
9 costs = cost_scaler.fit_transform(costs)
```

Después, se realiza una comprobación de calidad de los datos. Esto implica calcular las correlaciones, las varianzas, las fracciones de gasto y los factores de inflación de la varianza de los datos. Estas estadísticas proporcionan información útil sobre la calidad de los datos y pueden ayudar a identificar problemas potenciales antes de que los datos se ajusten al modelo.

```
[ ] 1 correlations, variances, spend_fractions, variance_inflation_factors = preprocessing.check_data_quality(
2     media_data=media_scaler.transform(media_data),
3     target_data=target_scaler.transform(target),
4     cost_data=costs,
5     extra_features_data=extra_features_scaler.transform(extra_features))
```





Una vez que los datos han sido preprocesados y examinados, se ajusta un modelo de mezcla de medios utilizando la clase LightweightMMM del paquete lightweight_mmm. El modelo elegido aquí es el modelo 'carryover', pero se podrían usar otros modelos según la naturaleza de los datos y el objetivo del análisis.



```
[ ] 1 def highlight_variances(x: float,
2                               low_variance_threshold: float=1.0e-3,
3                               high_variance_threshold: float=3.0) -> str:
4
5     if x < low_variance_threshold or x > high_variance_threshold:
6         weight = 'bold'
7         color = 'red'
8     else:
9         weight = 'normal'
10        color = 'black'
11    style = f'font-weight: {weight}; color: {color}'
12    return style
13
14 variances.style.set_precision(4).applymap(highlight_variances)
15

<ipython-input-19-0e1f2bc3fdd6>:14: FutureWarning: this method is deprecated in favour of `Styler.format(pr
variances.style.set_precision(4).applymap(highlight_variances)

      geo_0
feature_0    0.6182
feature_1    0.5518
feature_2    1.1634
feature_3    0.9629
feature_4    0.5059
feature_5    0.8723
feature_6    1.2885
feature_7    0.8319
feature_8    0.3252
feature_9    0.2964
extra_feature_0  0.1940
extra_feature_1  0.4089
extra_feature_2  0.2572
```

Finalmente, el modelo se ajusta a los datos de entrenamiento utilizando el método fit de la clase LightweightMMM. Este método también permite especificar varios parámetros adicionales, como el número de muestras durante el muestreo y el número de cadenas de muestreo.



```
[ ] 1 mmm = lightweight_mmm.LightweightMMM(model_name="carryover")
```

El entrenamiento del modelo se realiza con los siguientes parámetros:

- medios
- total_costs (un valor por canal)
- objetivo

No se han considerado los parámetros adicionales como:

- extra_features: Otras variables a añadir al modelo.
- grados_estacionalidad: Número de grados a utilizar para la estacionalidad.
- seasonality_frequency: Frecuencia del periodo de tiempo utilizado.
- media_names: Nombres de los canales de medios pasados.
- number_warmup: Número de muestras de calentamiento.
- number_samples: Número de muestras durante el muestreo.
- número_cadenas: Número de cadenas a muestrear.

```
[ ] 1 number_warmup=1000
2 number_samples=1000
```

```
[ ] 1 # For replicability in terms of random number generation in sampling
2 # reuse the same seed for different trainings.
3 mmm.fit(
4     media=media_data_train,
5     media_prior=costs,
6     target=target_train,
7     extra_features=extra_features_train,
8     number_warmup=number_warmup,
9     number_samples=number_samples,
10    seed=SEED)
```

```
/usr/local/lib/python3.10/dist-packages/lightweight_mmm/lightweight_mmm.py:358: UserWarning: There are not enough devices to run pa
  mcmc = numpyro.infer.MCMC(
sample: 100%|██████████| 2000/2000 [07:30<00:00,  4.44it/s, 127 steps of size 3.20e-02. acc. prob=0.95]
sample: 100%|██████████| 2000/2000 [06:55<00:00,  4.81it/s, 63 steps of size 4.74e-02. acc. prob=0.90]
```

La segunda parte del código continúa con el proceso de evaluación y optimización del modelo, después de la etapa de entrenamiento del Modelo de Mezcla de Medios

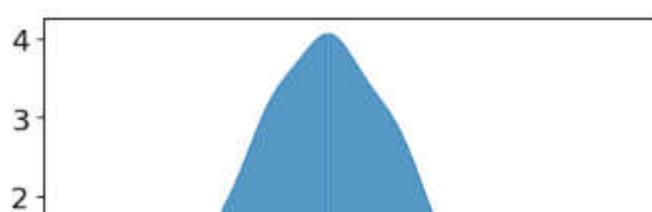
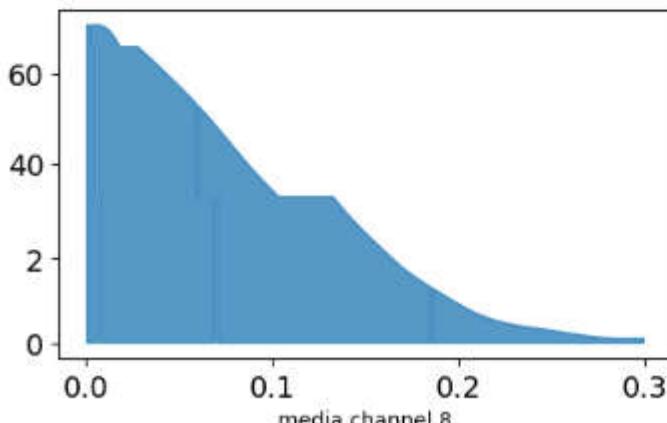
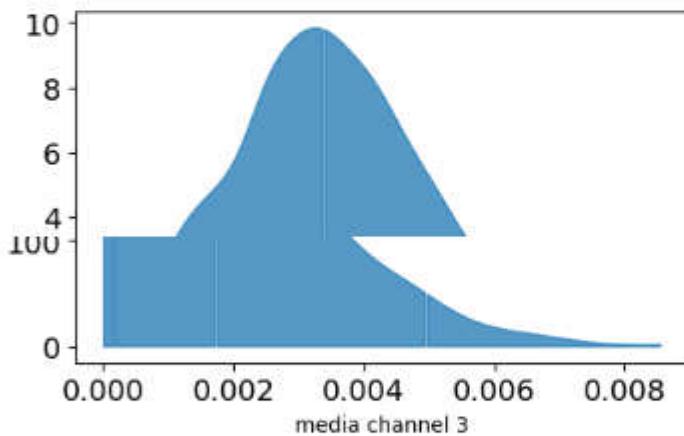
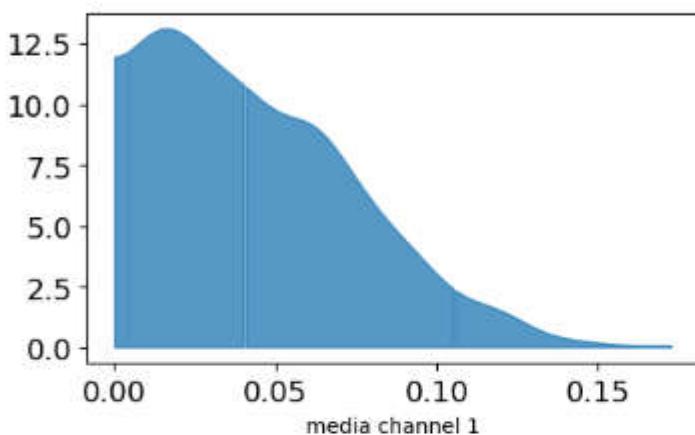
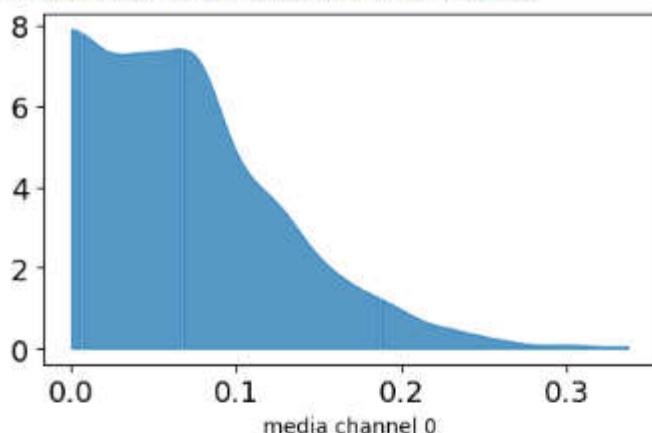
Después del proceso de entrenamiento, se imprime el resumen del modelo utilizando el método `.print_summary()`.

```
[ ] 1 plot.plot_media_channel_posteriors(media_mix_model=mmm)
```

A continuación, se crea una serie de visualizaciones para evaluar el rendimiento del modelo, utilizando varios métodos del paquete Lightweight MMM.



```
warnings.warn(errors.NumbaDeprecationWarning(msg,
```

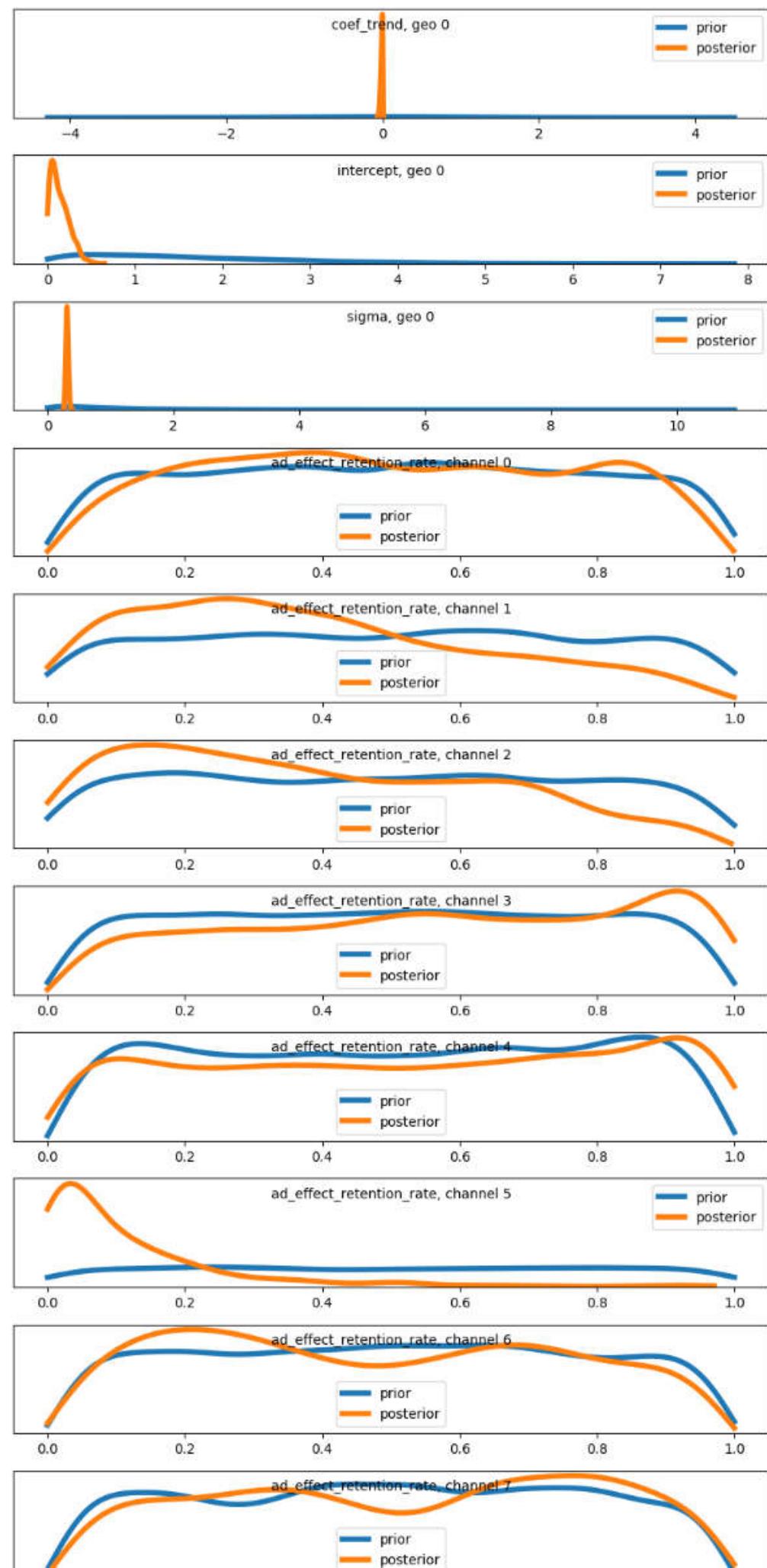


La primera, `plot_media_channel_posteriors()`, visualiza las distribuciones posteriores de los efectos de los medios.

```
[ ] 1 plot.plot_prior_and_posterior(media_mix_model=mmm)
```

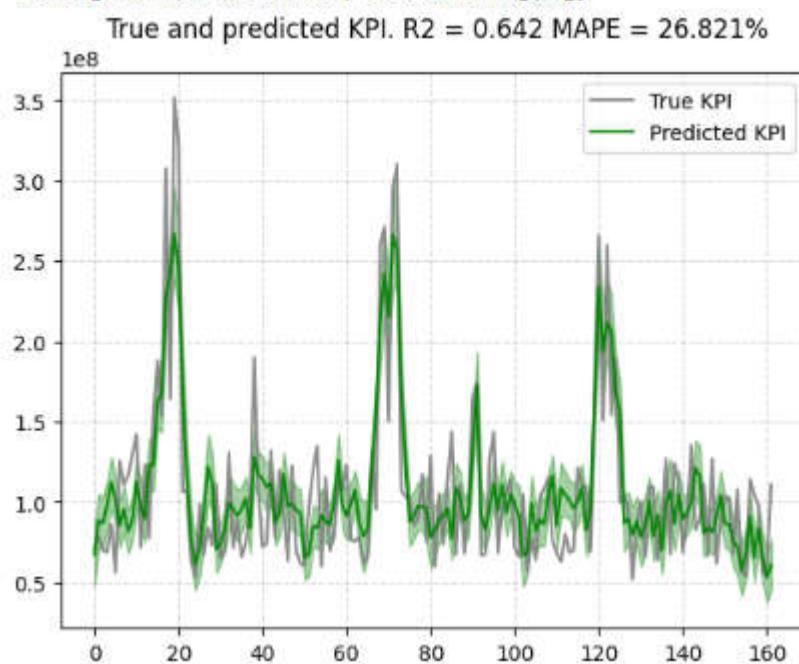
Luego, `plot_prior_and_posterior()` proporciona la visualización de las distribuciones previas y posteriores para cada parámetro del modelo.





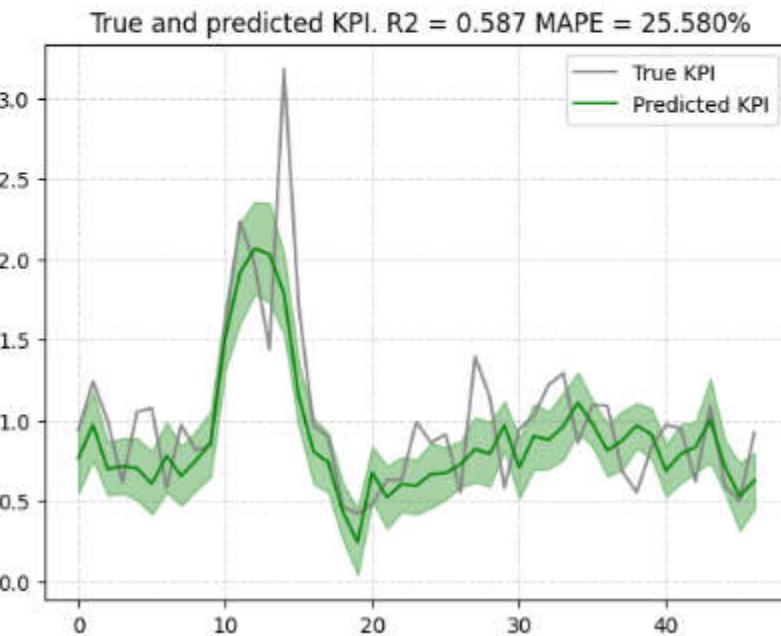
El método `plot_model_fit()` verifica el ajuste del modelo a los datos de entrenamiento.

```
[ ] 1 # Here is another example where we can pass the target scaler if you want the plot to be in the "not scale"
2 plot.plot_model_fit(mmm, target_scaler=target_scaler)
```



Luego, puedes ejecutar predicciones sobre datos no vistos utilizando el método `predict`. Los resultados predichos se visualizan luego utilizando la función `plot_out_of_sample_model_fit()`.

```
[ ] 1 # We have to scale the test media data if we have not done so before.  
2 new_predictions = mmm.predict(media=media_scaler.transform(media_data_test),  
3                                extra_features=extra_features_scaler.transform(extra_features_test),  
4                                seed=105)  
5 new_predictions.shape  
  
(2000, 47)  
  
[ ] 1 plot.plot_out_of_sample_model_fit(out_of_sample_predictions=new_predictions,  
2                                     out_of_sample_target=target_scaler.transform(target[split_point:]))
```



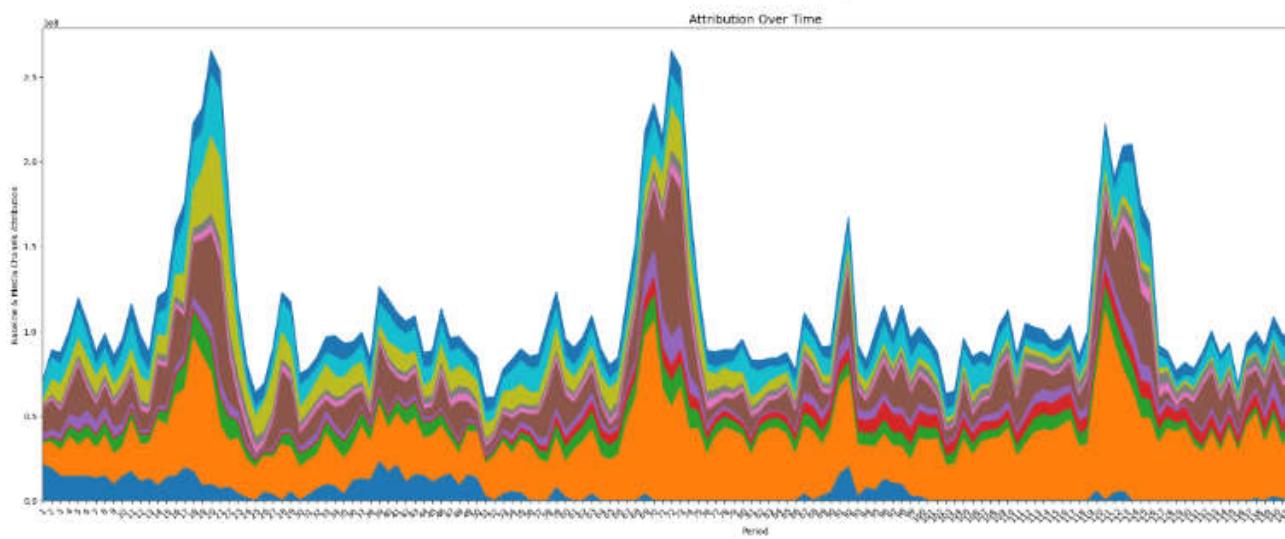
Avanzando, el método `get_posterior_metrics()` calcula la contribución de los medios y el retorno de la inversión (ROI). La contribución de los medios y la línea de base en el tiempo se visualiza utilizando `plot_media_baseline_contribution_area_plot()`, y las contribuciones de los medios y el ROI se muestran con gráficos de barras utilizando la función `plot_bars_media_metrics()`.

La curva de respuesta para cada canal de medios, que muestra cómo se comporta a medida que aumenta la inversión, se traza utilizando `plot_response_curves()`.

```
[ ] 1 media_contribution, roi_hat = mmm.get_posterior_metrics(target_scaler=target_scaler, cost_scaler=cost_scaler)
```

/visualizamos la contribución estimada de los medios de comunicación y la línea de base a lo largo del tiempo

```
[ ] 1 plot.plot_media_baseline_contribution_area_plot(media_mix_model=mmm,
2 target_scaler=target_scaler,
3 fig_size=(30,10))
```



Después de esta fase de evaluación, el script pasa a la fase de optimización.

Este proceso está destinado a responder preguntas de asignación de presupuesto. El usuario debe definir el periodo de tiempo para el cual quiere optimizar su presupuesto, y establecer precios y limitaciones de presupuesto.

El método `find_optimal_budgets()` del módulo `optimize_media` calcula el presupuesto óptimo para cada canal de medios.

```
[ ] 1 # Both numbers should be almost equal
2 budget, jnp.sum(solution.x * prices)
```

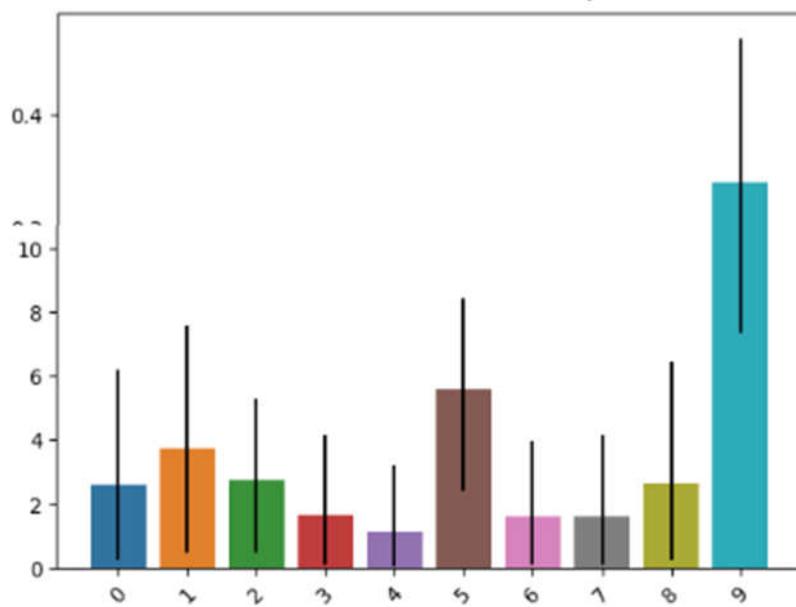
```
(Array(7.7314234e+08, dtype=float32), Array(7.7314234e+08, dtype=float32))
```

```
[ ] 1 # Plot out pre post optimization budget allocation and predicted target variable comparison.
2 plot.plot_pre_post_budget_allocation_comparison(media_mix_model=mmm,
3 kpi_with_optim=solution['fun'],
4 kpi_without_optim=kpi_without_optim,
5 optimal_buget_allocation=optimal_buget_allocation,
6 previous_buget_allocation=previous_buget_allocation,
7 figure_size=(10,10))
```

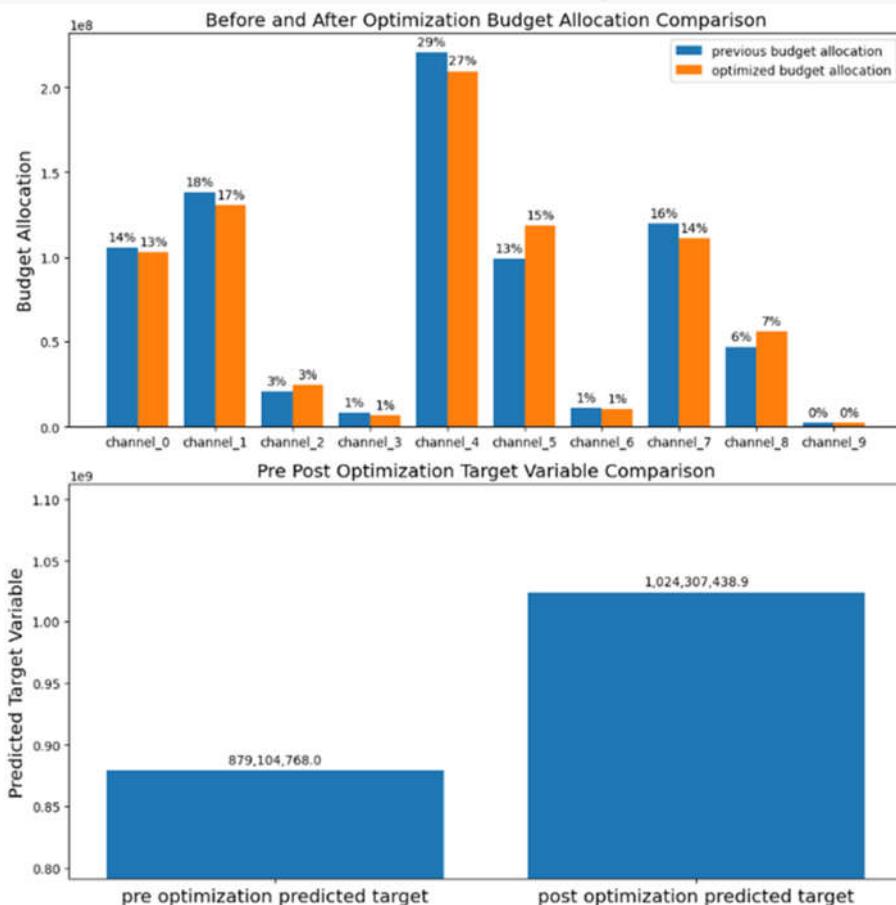
Finalmente, se comparan y visualizan las asignaciones de presupuesto y las variables objetivo predichas antes y después de la optimización.

```
[ ] 1 plot.plot_bars_media_metrics(metric=media_contribution, metric_name="Media Contribution")
```

Estimated media channel Media Contribution Percentage.
Error bars show 0.05 - 0.95 credibility interval.



La última parte del script muestra cómo guardar el modelo entrenado en el disco para su uso futuro. La función `save_model()` del módulo `utils` se utiliza para guardar el modelo.



Anexo VII: Modelo Bayesiano Stan



Universitat Oberta
de Catalunya

GRADO DE CIENCIA DE DATOS APLICADA

Modelo Bayesiano Stan

18 Junio 2023

Descripción breve

Informe con el Modelo Bayesiano Stan



Alberto de Torres Pachón

Dirección académica: Xavier Florit

Responsable académico: Elena Rodríguez

Trabajo final de grado 22536

144



ADTP

Importación de bibliotecas y datos: En primer lugar, se importan las bibliotecas necesarias y se carga el conjunto de datos.

```
|: import warnings
warnings.filterwarnings("ignore")

import numpy as np
import pandas as pd
import sys
import time
from datetime import datetime
from datetime import timedelta
import matplotlib.pyplot as plt
import seaborn as sns
# get_ipython().run_line_magic('matplotlib', 'inline')

sns.color_palette("husl")
sns.set_style('darkgrid')
```

Preparación de datos: A continuación, se lleva a cabo una exploración y preparación de datos en el conjunto de datos. Se definen las variables de interés, como variables de medios, variables de control y variables de ventas.



```
# Data
# Four years' (209 weeks) records of sales, media impression and media spending at weekly L
df = pd.read_csv('data.csv')

# 1. media variables
# media impression
mdip_cols=[col for col in df.columns if 'mdip_' in col]
# media spending
mdsp_cols=[col for col in df.columns if 'mdsp_' in col]

# 2. control variables
# macro economics variables
me_cols = [col for col in df.columns if 'me_' in col]
# store count variables
st_cols = ['st_ct']
# markdown/discount variables
mrkdn_cols = [col for col in df.columns if 'mrkdn_' in col]
# holiday variables
hldy_cols = [col for col in df.columns if 'hldy_' in col]
# seasonality variables
seas_cols = [col for col in df.columns if 'seas_' in col]
base_vars = me_cols+st_cols+mrkdn_cols+hldy_cols+seas_cols

# 3. sales variables
sales_cols =[['sales']]

df[['wk strt_dt']+mdip_cols+['sales']].head()

# EDA - correlation, distribution plots
#plt.figure(figsize=(24,20))
#sns.heatmap(df[mdip_cols+['sales']].corr(), square=True, annot=True, vmax=1, vmin=-1, cmap='viridis')

#plt.figure(figsize=(50,50))
#sns.pairplot(df[mdip_cols+['sales']], vars=mdip_cols+['sales'])
```

Adstock Transformation: La transformación de Adstock se utiliza para modelar el efecto retrasado y decreciente de la publicidad en las ventas. Esta función se aplica a las variables de medios en los datos.



```

|: # 2.2 Marketing Mix Model
df_mmm, sc_mmm = mean_log1p_transform(df, ['sales', 'base_sales'])
mu_mdip = df[mdip_cols].apply(np.mean, axis=0).values
max_lag = 8
num_media = len(mdip_cols)
# padding zero * (max_lag-1) rows
X_media = np.concatenate((np.zeros((max_lag-1, num_media)), df[mdip_cols].values), axis=0)
X_ctrl = df_mmm['base_sales'].values.reshape(len(df),1)
model_data2 = {
    'N': len(df),
    'max_lag': max_lag,
    'num_media': num_media,
    'X_media': X_media,
    'mu_mdip': mu_mdip,
    'num_ctrl': X_ctrl.shape[1],
    'X_ctrl': X_ctrl,
    'y': df_mmm['sales'].values
}

model_code2 = '''
functions {
    // the adstock transformation with a vector of weights
    real Adstock(vector t, row_vector weights) {
        return dot_product(t, weights) / sum(weights);
    }
}

```

Transformación de la función de Hill (Diminishing Return): Se utiliza la transformación de la función de Hill para modelar la ley de rendimientos decrecientes. Esto asume que a medida que se incrementa la inversión en publicidad, el impacto incremental en las ventas disminuye.

Modelo de Control/Base Sales Model: Aquí, se ajusta un modelo para estimar las ventas básicas que habrían ocurrido sin ninguna actividad de medios. Se utiliza el modelo Stan para esto.

```

}
data {
    // the total number of observations
    int N;
    // the vector of sales
    real y[N];
    // the maximum duration of lag effect, in weeks
    int max_lag;
    // the number of media channels
    int num_media;
    // matrix of media variables
    matrix[N*max_lag-1, num_media] X_media;
    // vector of media variables' mean
    real mu_mdip[num_media];
    // the number of other control variables
    int num_ctrl;
    // a matrix of control variables
    matrix[N, num_ctrl] X_ctrl;
}
parameters {
    // residual variance
    real noise_var;
    // the intercept
    real tau;
    // the coefficients for media variables and base sales
    vector[num_media+num_ctrl] beta;
    // the decay and peak parameter for the adstock transformation of
    // each media
    vector[num_media] decay;
    vector[num_media] peak;
}
```

```

}
transformed parameters {
  // the cumulative media effect after adstock
  real cum_effect;
  // matrix of media variables after adstock
  matrix[N, num_media] X_media_adstocked;
  // matrix of all predictors
  matrix[N, num_media+num_ctrl] X;

  // adstock, mean-center, log1p transformation
  row_vector[max_lag] lag_weights;
  for (nn in 1:N) {
    for (media in 1 : num_media) {
      for (lag in 1 : max_lag) {
        lag_weights[max_lag-lag+1] <- pow(decay[media], (lag - 1 - peak[media]) ^ 2);
      }
      cum_effect <- Adstock(sub_col(X_media, nn, media, max_lag), lag_weights);
      X_media_adstocked[nn, media] <- log1p(cum_effect/mu_mdip[media]);
    }
    X <- append_col(X_media_adstocked, X_ctrl);
  }
}
model {
  decay ~ beta(3,3);
  peak ~ uniform(0, cell(max_lag/2));
  tau ~ normal(0, 5);
  for (i in 1 : num_media+num_ctrl) {
    beta[i] ~ normal(0, 1);
  }
  noise_var ~ inv_gamma(0.05, 0.05 * 0.01);
  y ~ normal(tau + X * beta, sqrt(noise_var));
}
...
sm2 = pystan.StanModel(model_code=model_code2, verbose=True)
fit2 = sm2.sampling(data=model_data2, iter=1000, chains=3)
fit2_result = fit2.extract()

```

Entrenamiento del modelo de regresión: Finalmente, se divide el conjunto de datos en conjuntos de entrenamiento y prueba, se ajusta un modelo de regresión lineal a los datos de entrenamiento y luego se usa el modelo para predecir las ventas en los datos de prueba.

```

# extract mmm parameters
def extract_mmm(fit_result, max_lag=max_lag,
                media_vars=mdip_cols, ctrl_vars=['base_sales'],
                extract_param_list=True):
    mmm = {}

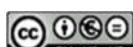
    mmm['max_lag'] = max_lag
    mmm['media_vars'], mmm['ctrl_vars'] = media_vars, ctrl_vars
    mmm['decay'] = decay = fit_result['decay'].mean(axis=0).tolist()
    mmm['peak'] = peak = fit_result['peak'].mean(axis=0).tolist()
    mmm['beta'] = fit_result['beta'].mean(axis=0).tolist()
    mmm['tau'] = fit_result['tau'].mean()

    if extract_param_list:
        mmm['decay_list'] = fit_result['decay'].tolist()
        mmm['peak_list'] = fit_result['peak'].tolist()
        mmm['beta_list'] = fit_result['beta'].tolist()
        mmm['tau_list'] = fit_result['tau'].tolist()

    adstock_params = {}
    media_names = [col.replace('mdip_', '') for col in media_vars]
    for i in range(len(media_names)):
        adstock_params[media_names[i]] = {
            'L': max_lag,
            'P': peak[i],
            'D': decay[i]
        }
    mmm['adstock_params'] = adstock_params
    return mmm

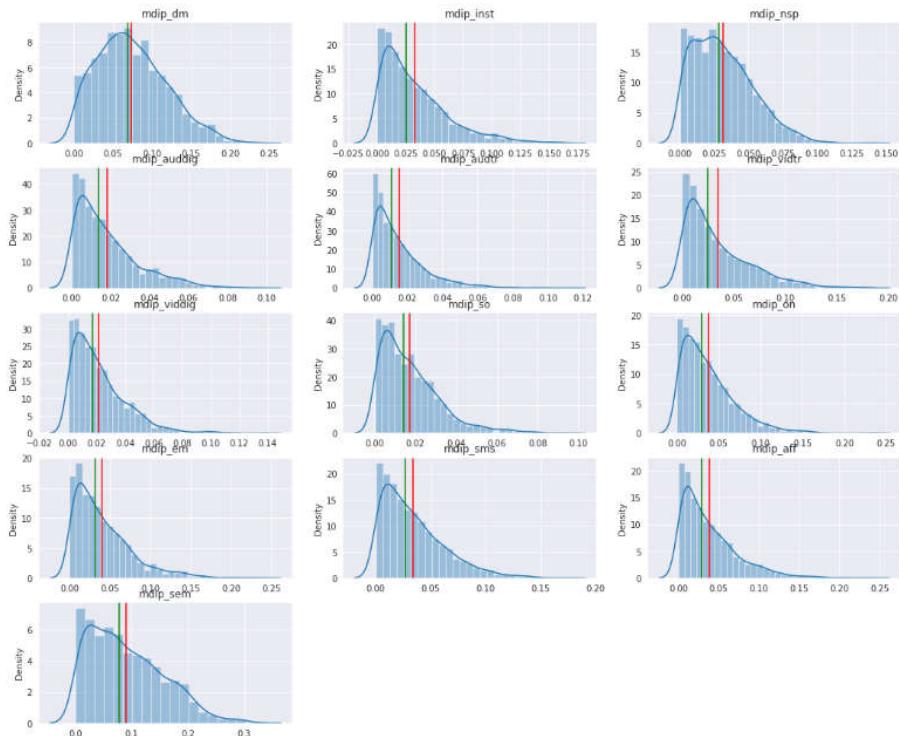
mmm = extract_mmm(fit2, max_lag=max_lag,
                  media_vars=mdip_cols, ctrl_vars=['base_sales'])
# save_json(mmm, 'mmml1.json')

```



```
# plot media coefficients' distributions
# red line: mean, green line: median
beta_media = {}
for i in range(len(mmm['media_vars'])):
    md = mmm['media_vars'][i]
    betas = []
    for j in range(len(mmm['beta_list'])):
        betas.append(mmm['beta_list'][j][i])
    beta_media[md] = np.array(betas)

f = plt.figure(figsize=(18,15))
for i in range(len(mmm['media_vars'])):
    ax = f.add_subplot(5,3,i+1)
    md = mmm['media_vars'][i]
    x = beta_media[md]
    mean_x = x.mean()
    median_x = np.median(x)
    ax = sns.distplot(x)
    ax.axvline(mean_x, color='r', linestyle='--')
    ax.axvline(median_x, color='g', linestyle='--')
    ax.set_title(md)
```



Evaluación del modelo: Se evalúa el rendimiento del modelo de regresión utilizando la métrica de error porcentual absoluto medio (MAPE).

```

# Decompose sales to media channels' contribution
# Each media channel's contribution = total sales - sales upon removal the channel

# decompose sales to media contribution
def mmm_decompose_contrib(mmm, df, original_sales=df['sales']):
    # adstock params
    adstock_params = mmm['adstock_params']
    # coefficients, intercept
    beta, tau = mmm['beta'], mmm['tau']
    # variables
    media_vars, ctrl_vars = mmm['media_vars'], mmm['ctrl_vars']
    num_media, num_ctrl = len(media_vars), len(ctrl_vars)
    # X_media2: adstocked, mean-centered media variables + 1
    X_media2 = adstock_transform(df, media_vars, adstock_params)
    X_media2, sc_mmm2 = mean_center_trandform(X_media2, media_vars)
    X_media2 = X_media2 + 1
    # X_ctrl2, mean-centered control variables + 1
    X_ctrl2, sc_mmm2_1 = mean_center_trandform(df[ctrl_vars], ctrl_vars)
    X_ctrl2 = X_ctrl2 + 1
    # y_true2, mean-centered sales variable + 1
    y_true2, sc_mmm2_2 = mean_center_trandform(df, ['sales'])
    y_true2 = y_true2 + 1
    sc_mmm2.update(sc_mmm2_1)
    sc_mmm2.update(sc_mmm2_2)
    # X2 <- media variables + ctrl variable
    X2 = pd.concat([X_media2, X_ctrl2], axis=1)

    # 1. compute each media/control factor:
    # log-log model: log(sales) = log(X[0])*beta[0] + ... + log(X[13])*beta[13] + tau
    # multiplicative model: sales = X[0]^beta[0] * ... * X[13]^beta[13] * e^tau
    # each factor = X[i]^beta[i]
    # intercept = e^tau
    factor_df = pd.DataFrame(columns=media_vars+ctrl_vars+[ 'intercept'])
    for i in range(num_media):
        colname = media_vars[i]
        factor_df[colname] = X2[colname] ** beta[i]
    for i in range(num_ctrl):
        colname = ctrl_vars[i]
        factor_df[colname] = X2[colname] ** beta[num_media+i]
    factor_df['intercept'] = np.exp(tau)

```



```

# 2. calculate the product of all factors -> y_pred
# baseline = intercept * control factor = e^tau * X[13]^beta[13]
y_pred = factor_df.apply(np.prod, axis=1)
factor_df['y_pred'], factor_df['y_true2'] = y_pred, y_true2
factor_df['baseline'] = factor_df[['intercept']+ctrl_vars].apply(np.prod, axis=1)

# 3. calculate each media factor's contribution
# media contribution = total volume - volume upon removal of the media factor
mc_df = pd.DataFrame(columns=media_vars+['baseline'])
for col in media_vars:
    mc_df[col] = factor_df['y_true2'] - factor_df['y_true2']/factor_df[col]
mc_df['baseline'] = factor_df['baseline']
mc_df['y_true2'] = factor_df['y_true2']

# 4. scale contribution
# predicted total media contribution: product of all media factors
mc_df['mc_pred'] = mc_df[media_vars].apply(np.sum, axis=1)
# true total media contribution: total volume - baseline
mc_df['mc_true'] = mc_df['y_true2'] - mc_df['baseline']
# predicted total media contribution is slightly different from true total media contribution
# scale each media factor's contribution by removing the delta volume proportionally
mc_df['mc_delta'] = mc_df['mc_pred'] - mc_df['mc_true']
for col in media_vars:
    mc_df[col] = mc_df[col] - mc_df['mc_delta']*mc_df[col]/mc_df['mc_pred']

# 5. scale mc_df based on original sales
mc_df['sales'] = original_sales
for col in media_varst+['baseline']:
    mc_df[col] = mc_df[col]*mc_df['sales']/mc_df['y_true2']

print('rmse (log-log model): ',
      mean_squared_error(np.log(y_true2), np.log(y_pred)) ** (1/2))
print('mape (multiplicative model): ',
      mean_absolute_percentage_error(y_true2, y_pred))
return mc_df

# calculate media contribution percentage
def calc_media_contrib_pct(mc_df, media_vars=mdip_cols, sales_col='sales', period=52):
    """
    returns:
    mc_pct: percentage over total sales
    mc_pct2: percentage over incremental sales (sales contributed by media channels)
    """
    mc_pct = {}
    mc_pct2 = {}
    s = 0
    if period is None:
        for col in (media_varst+['baseline']):
            mc_pct[col] = (mc_df[col]/mc_df[sales_col]).mean()
    else:
        for col in (media_varst+['baseline']):
            mc_pct[col] = (mc_df[col]/mc_df[sales_col])[-period:].mean()
    for m in media_vars:
        s += mc_pct[m]
    for m in media_vars:
        mc_pct2[m] = mc_pct[m]/s
    return mc_pct, mc_pct2

    mc_df = mmm_decompose_contrib(mmm, df, original_sales=df['sales'])
adstock_params = mmm['adstock_params']
mc_pct, mc_pct2 = calc_media_contrib_pct(mc_df, period=52)
# mc_df.to_csv('mc_df1.csv', index=False)
# save_json(adstock_params, 'adstock_params1.json')
# pd.concat([
#     pd.DataFrame.from_dict(mc_pct, orient='index', columns=['mc_pct']),
#     pd.DataFrame.from_dict(mc_pct2, orient='index', columns=['mc_pct2'])
# ], axis=1).to_csv('mc_pct_df1.csv')

rmse (log-log model): 0.13647800963534104

```

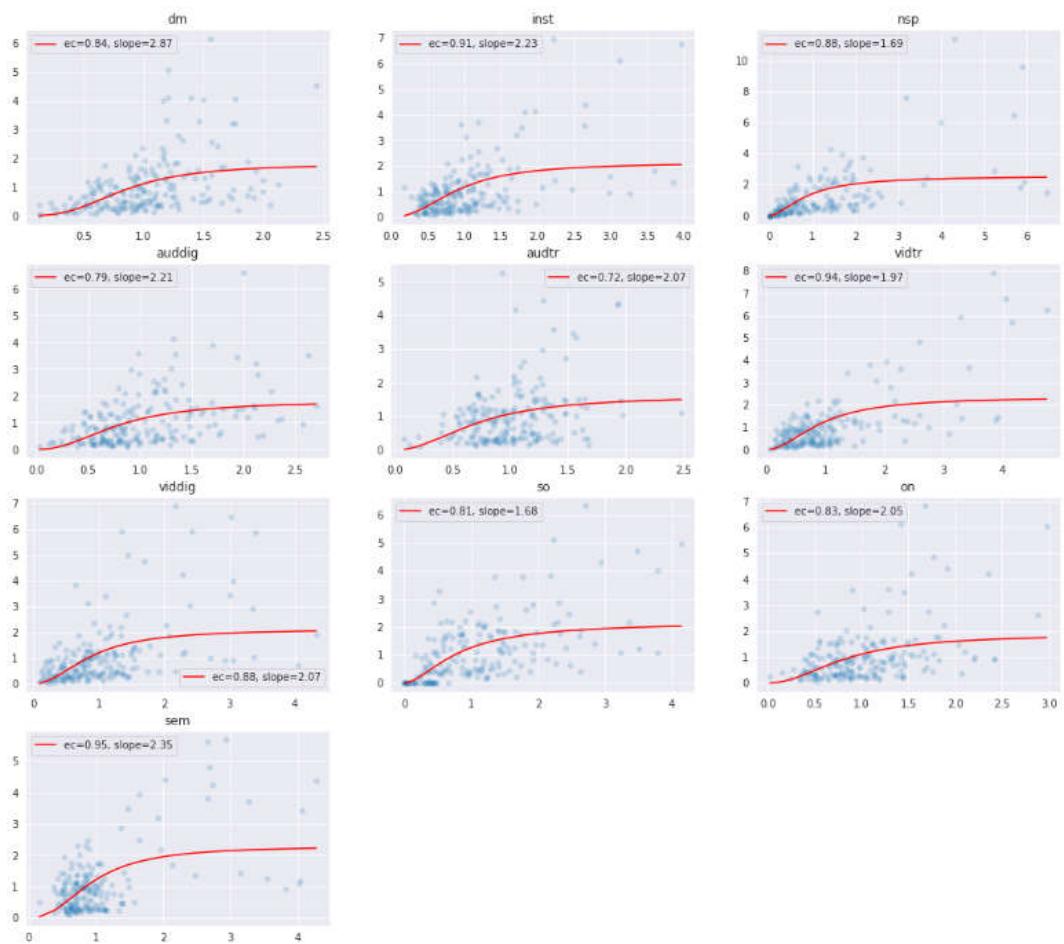
```
In [16]: # train hill models for all media channels
sm3 = pystan.StanModel(model_code=model_code3, verbose=True)
hill_models = {}
to_train = ['dm', 'inst', 'nsp', 'auddig', 'audtr', 'vidtr', 'viddig', 'so', 'on', 'sem']
for media in to_train:
    print('training for media: ', media)
    hill_model = train_hill_model(df, mc_df, adstock_params, media, sm3)
    print("trained for media: ", media)
    hill_models[media] = hill_model

# extract params by mean
hill_model_params_mean, hill_model_params_med = {}, {}
for md in list(hill_models.keys()):
    print("extracting " + md)
    hill_model = hill_models[md]
    params1 = extract_hill_model_params(hill_model, method='mean')
    params1['sc'] = hill_model['sc']
    hill_model_params_mean[md] = params1
    # params2 = extract_hill_model_params(hill_model, method='median')
    # params2['sc'] = hill_model['sc']
    # hill_model_params_med[md] = params2
    # save_json(hill_model_params_med, 'hill_model_params_med.json')
    # save_json(hill_model_params_mean, 'hill_model_params_mean.json')

    # evaluate model params extracted by mean
    for md in list(hill_models.keys()):
        print('evaluating media: ', md)
        hill_model = hill_models[md]
        hill_model_params = hill_model_params_mean[md]
        _ = evaluate_hill_model(hill_model, hill_model_params)
    # evaluate model params extracted by median
    # for md in list(hill_models.keys()):
    #     print('evaluating media: ', md)
    #     hill_model = hill_models[md]
    #     hill_model_params = hill_model_params_med[md]
    #     _ = evaluate_hill_model(hill_model, hill_model_params)
```

```
In [17]: # plot fitted hill function
f = plt.figure(figsize=(18,16))
hm_keys = list(hill_models.keys())
for i in range(len(hm_keys)):
    ax = f.add_subplot(4,3,i+1)
    md = hm_keys[i]
    hm = hill_models[md]
    hmp = hill_model_params_mean[md]
    x, y = hm['data']['X'], hm['data']['y']
    #mu_x, mu_y = hm['sc']['x'], hm['sc']['y']
    ec, slope = hmp['ec'], hmp['slope']
    x_sorted = np.array(sorted(x))
    y_fit = hill_model_predict(hmp, x_sorted)
    ax = sns.scatterplot(x=x, y=y, alpha=0.2)
    ax = sns.lineplot(x=x_sorted, y=y_fit, color='r',
                      label='ec=%2f, slope=%2f%(ec, slope)')
    ax.set_title(md)
```





```
# Calculate overall ROAS and weekly ROAS
# - Overall ROAS = total contribution / total spending
# - Weekly ROAS = weekly contribution / weekly spending

# adstocked media spending
ms_df = pd.DataFrame()
for md in list(hill_models.keys()):
    hill_model = hill_models[md]
    x = np.array(hill_model['data']['X']) * hill_model['sc']['x']
    ms_df['mdsp_'+md] = x
# ms_df.to_csv('ms_df1.csv', index=False)

# calc overall ROAS of a given period
def calc_roas(mc_df, ms_df, period=None):
    roas = {}
    md_names = [col.split('_')[-1] for col in ms_df.columns]
    for i in range(len(md_names)):
        md = md_names[i]
        sp, mc = ms_df['mdsp_'+md], mc_df['mdip_'+md]
        if period is None:
            md_roas = mc.sum()/sp.sum()
        else:
            md_roas = mc[-period:].sum()/sp[-period:].sum()
        roas[md] = md_roas
    return roas

# calc weekly ROAS
def calc_weekly_roas(mc_df, ms_df):
    weekly_roas = pd.DataFrame()
    md_names = [col.split('_')[-1] for col in ms_df.columns]
    for md in md_names:
        weekly_roas[md] = mc_df['mdip_'+md]/ms_df['mdsp_'+md]
    weekly_roas.replace([np.inf, -np.inf, np.nan], 0, inplace=True)
    return weekly_roas

roas_1y = calc_roas(mc_df, ms_df, period=52)
weekly_roas = calc_weekly_roas(mc_df, ms_df)
roas1y_df = pd.DataFrame(index=weekly_roas.columns.tolist())
roas1y_df['roas_mean'] = weekly_roas[-52:].apply(np.mean, axis=0)
roas1y_df['roas_median'] = weekly_roas[-52:].apply(np.median, axis=0)
```



```

def calc_mroas(hill_model, hill_model_params, period=52):
    """
    calculate mROAS for a media
    params:
    hill_model: a dict containing model data and scaling factor
    hill_model_params: a dict containing beta_hill, ec, slope
    period: in weeks, the period used to calculate ROAS and mROAS. 52 is last one year
    return:
    mROAS value
    """
    mu_x, mu_y = hill_model['sc']['x'], hill_model['sc']['y']
    # get current media spending level over the period specified
    cur_sp = np.asarray(hill_model['data']['X'])
    if period is not None:
        cur_sp = cur_sp[-period:]
    cur_mc = sum(hill_model_predict(hill_model_params, cur_sp) * mu_y)
    # next spending level: increase by 1%
    next_sp = cur_sp * 1.01
    # media contribution under next spending level
    next_mc = sum(hill_model_predict(hill_model_params, next_sp) * mu_y)

    # mROAS
    delta_mc = next_mc - cur_mc
    delta_sp = sum(next_sp * mu_x) - sum(cur_sp * mu_x)
    mroas = delta_mc/delta_sp
    return mroas

# calc mROAS of recent 1 year
mroas_1y = {}
for md in list(hill_models.keys()):
    hill_model = hill_models[md]
    hill_model_params = hill_model_params_mean[md]
    mroas_1y[md] = calc_mroas(hill_model, hill_model_params, period=52)

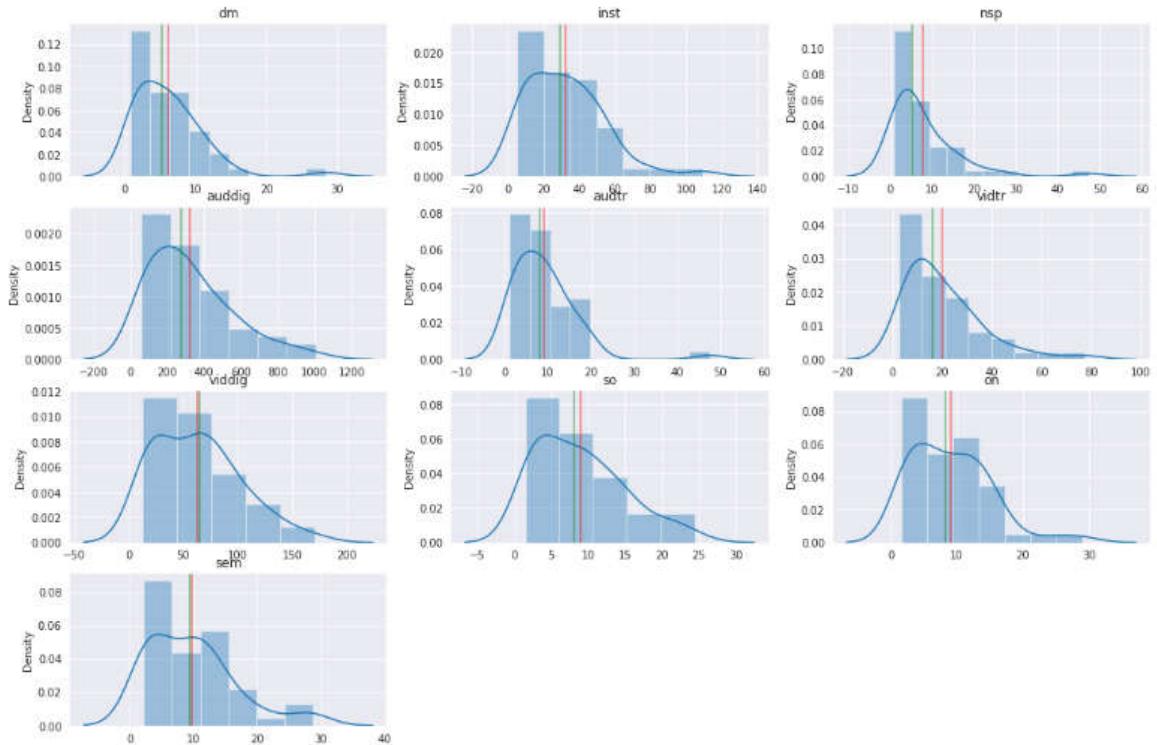
roas1y_df = pd.concat([
    roas1y_df[['roas_mean', 'roas_median']],
    pd.DataFrame.from_dict(mroas_1y, orient='index', columns=['mroas']),
    pd.DataFrame.from_dict(roas_1y, orient='index', columns=['roas_avg'])
], axis=1)
# roas1y_df.to_csv('roas1y_df1.csv')

roas1y_df

```

	roas_mean	roas_median	mroas	roas_avg
dm	6.067648	5.200908	8.176664	6.305869
inst	32.120422	29.527071	37.829563	32.800853
nsp	7.804195	5.160552	11.177744	10.364456
auddig	325.145473	273.614418	149.047283	313.753330
audtr	9.117158	8.268813	5.769686	8.846472
vidtr	20.146669	16.259589	12.025508	17.251446
viddig	63.139437	64.798941	53.981645	69.496131
so	8.899403	8.030128	3.907974	8.786031
on	9.055892	8.207717	6.848882	9.185394
sem	9.769199	9.507649	9.902280	9.299544





Anexo VIII: Primer Informe de Seguimiento



GRADO DE CIENCIA DE DATOS APLICADA

Primer Informe de Seguimiento

18 Junio 2023

Descripción breve
Primer Informe de Seguimiento



Alberto de Torres Pachón

Dirección académica: Xavier Florit
Responsable académico: Elena Rodríguez
Trabajo final de grado 22536



ADTP

PEC2: INFORME SEGUIMIENTO TFG MARKETING MIX MODELING

1. Identificación del trabajo y fecha del informe:

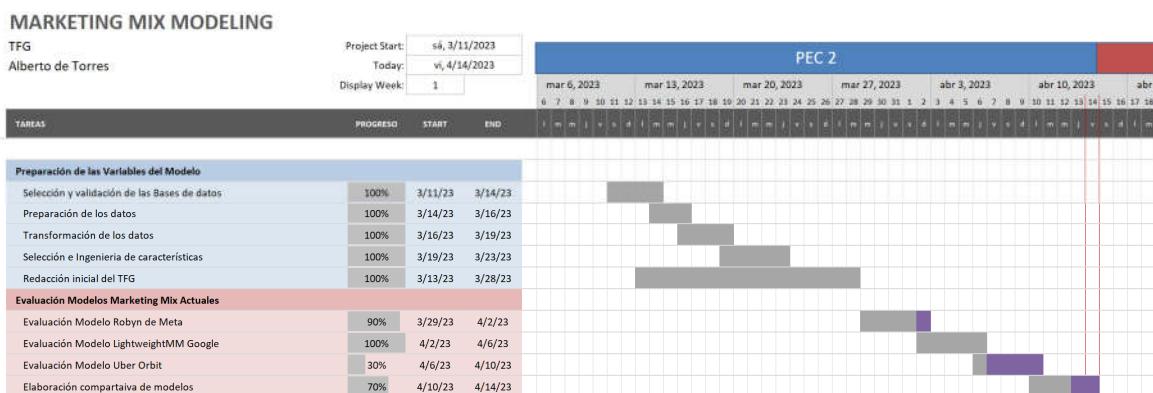
El trabajo corresponde al TFG sobre la creación de un Modelo de Marketing Mix que permita simular la optimización de la inversión de marketing.

Este informe se realiza para la PEC 2 con fecha 14 de Abril del 2023

2. Descripción del avance del proyecto

A. Grado de cumplimiento de los objetivos y resultados previstos en el plan de trabajo.

El grado de avance del proyecto se ha actualizado en el Cronograma (se adjunta Cronograma Gannt actualizado).



En global el avance en esta PEC2 ha sido del 89%, con lo que la desviación sobre la planificación es asumible recuperarla en la siguiente PEC.

3. Relación de las actividades realizadas

A. Actividades previstas en el plan de trabajo:

I. Preparación de las Variables del Modelo:

✓ Selección y validación de las Bases de datos:

Después de una evaluación de bastantes bases de datos, he elegido en github uno que es público ([mmm_stan/main · sibylhe/mmm_stan \(github.com\)](https://github.com/sibylhe/mmm_stan)) y es muy cercano a las variables que un departamento de marketing utiliza en la vida real, con una dimensionalidad de los datos de 209 filas por 80 columnas:



- Variables de medios de comunicación:
 - Impresión en medios (prefijo='mdip_'): impresiones de 13 canales de medios: publicidad directa, encartes, periódicos, audio digital, radio, televisión, vídeo digital, medios sociales, visualización en línea, correo electrónico, SMS, afiliados, SEM.
 - Gasto en medios (prefix='mdsp_'): gasto de los canales de medios.
 - Variables de control:
 - Macroeconomía (prefijo='me_'): IPC, precio del gas.
 - Rebaja (prefijo='mrkdn_'): rebaja/descuento.
 - Recuento de tiendas ('st_ct')
 - Días festivos en el comercio minorista (prefijo='hldy_')
 - Estacionalidad (prefijo='seas_'): mes, con noviembre y diciembre divididos en semanas.
 - Ventas

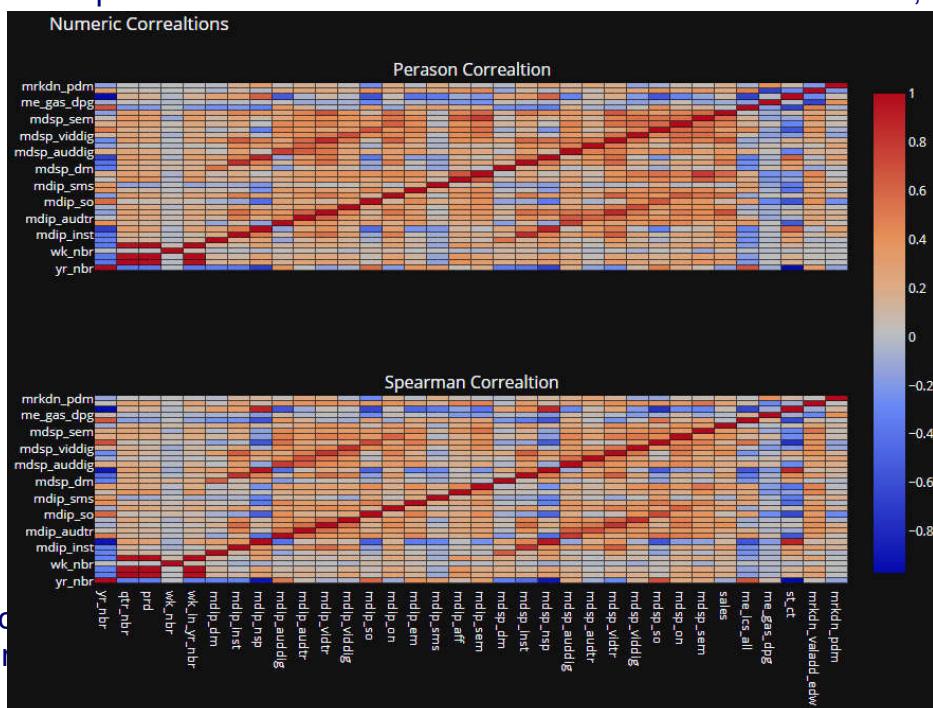
Para este set de datos he desarrollado un diccionario con la descripción y tipo de variables que tiene este dataset (Se adjunta en el entregable).

Este objetivo se ha cumplido al 100%.

✓ Preparación y transformación de los datos:

He realizado un análisis de las variables, para preparar los datos para su posterior procesamiento, haciendo las siguientes fases:

- Conversión de las variables en su formato adecuado: Fecha, Número entero, real.
 - Verificación que no hay datos duplicados.
 - Histograma de la variable dependiente de ventas, estacionalidad y valores extremos.
 - Se ha realizado un análisis de la correlación de las variables , con el coeficiente de correlación de Pearson para medir la relación lineal entre dos variables continuas;



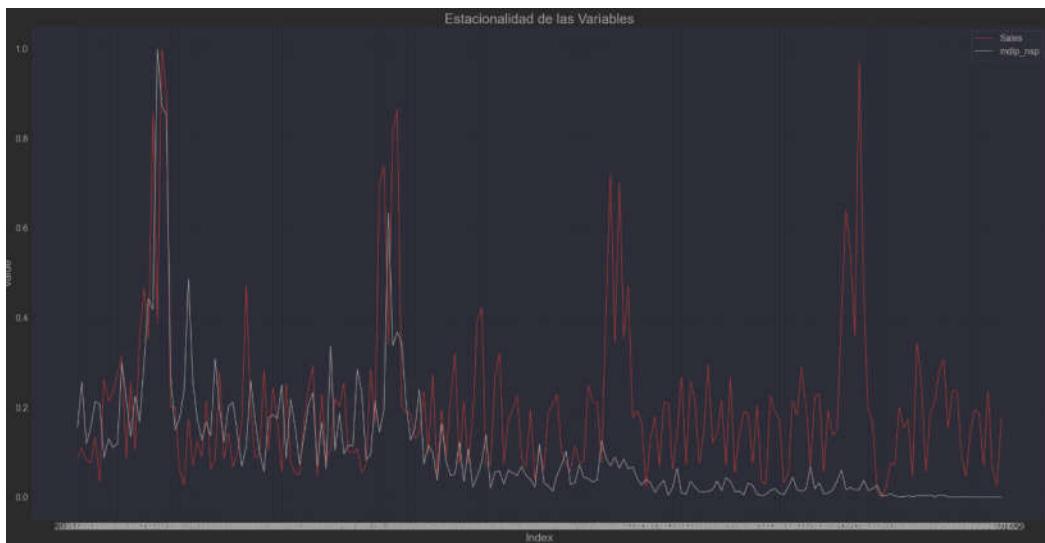


Observamos que las correlaciones no son muy altas, por lo que en principio no podemos descartar ninguna de las variables.

- Verificamos si existen valores nulo, que se comprueba que todos los datos tiene valores.
- Hacemos un análisis de los valores extremos y se verifica que son valores que no se deben eliminar o tratar ya que son pocos entre un 2 y un 8% de cada variable.

✓ Selección e ingeniería de características:

- Reducimos las variables a 24 variables para desarrollar un primer modelo de regresión multivariante.
- Para analizar la estacionalidad de las variables, primero hacemos una transformación de las variables numéricas para que tengan una escala específica. Y definimos una función que nos grifique todas las variables que contienen los impactos de los medios de publicidad con las ventas. Estas gráficas nos permiten visualizar la estacionalidad y como los impactos de la publicidad impactan en las ventas y en los picos estacionales, a modo de ejemplo este sería uno de los 20 gráficos obtenidos (se adjunta en los anexos, con el notebook):



Este objetivo se ha cumplido al 100% sobre lo planificado.

II. Evaluación Modelos Marketing Mix Actuales

En este apartado voy a adaptar los modelos más importantes que existen actualmente con los datos de nuestro proyecto, el objetivo es entender como se construyen y que aportan a un MMM, así como evaluar con los otros modelos para obtener un modelo personalizado que permita una mejor predicción.

✓ Evaluación Modelo Robyn de Meta:

Robyn, de Facebook Experimental, es un código automatizado de modelado de marketing mix (MMM) que se encuentra actualmente en versión beta.

He utilizado de base el repositorio del modelo de Robin experimental en R (<https://github.com/facebookexperimental/Robyn>).

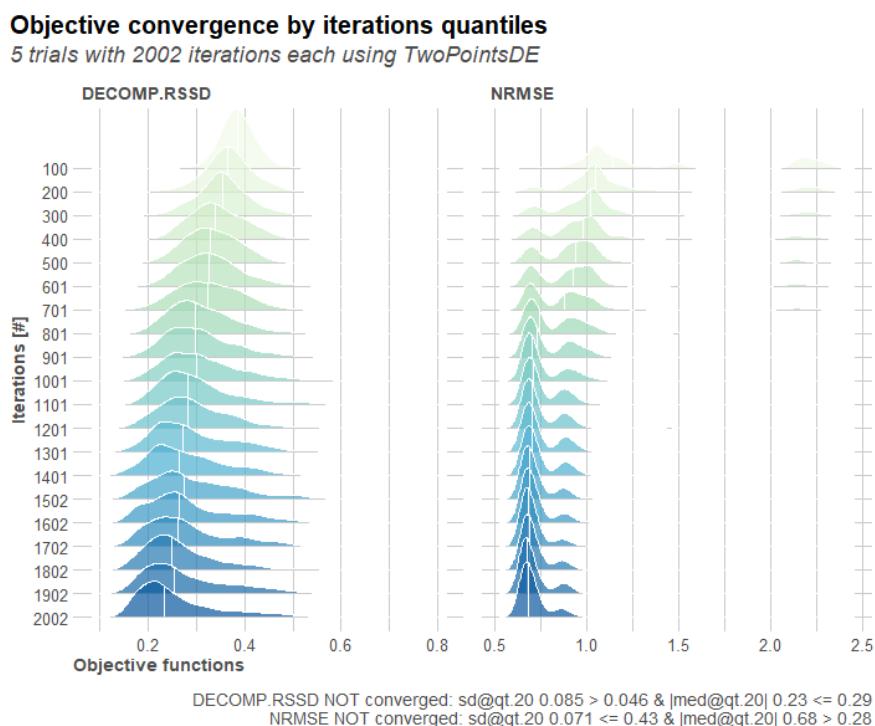
Este modelo es bastante complejo y me ha llevado un tiempo el entender como se compone y como ir construyendo cada fase, pero es una buena base para la construcción de un MMM con mi criterio final.

Los pasos del modelo son los siguientes;

- 1) Preparación de los datos
- 2) Aislamos tendencia y estacionalidad
- 3) Estimamos ad-stock (*) y curvas de saturación
- 4) Elegimos variables en el modelo que explican la variable objetivo
- 5) Nos quedamos con la solución que más nos convenza

Sobre lo planificado he tenido que emplear más tiempo, entender el modelo y adaptarlo, así como la instalación de varias librerías, ya que usa “Nevergrad” , librería de Python para optimizaciones, superando un 40% más del tiempo estimado.

Genera un conjunto de soluciones de modelo óptimas de Pareto utilizando la plataforma de optimización sin gradiente Never de Facebook.



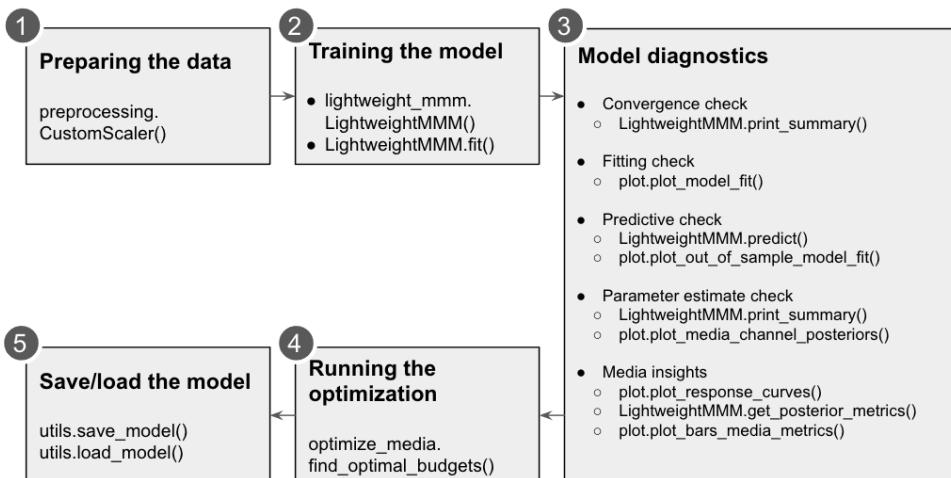
Para aumentar la precisión del modelo, permite incluir resultados de experimentos controlados aleatorios.

(*) AdStock es un concepto utilizado en la publicidad y el marketing para describir cómo los efectos de la publicidad en la memoria del consumidor se desvanece gradualmente con el tiempo. En otras palabras, es una medida de la duración del impacto de la publicidad en la mente del consumidor después de que ha sido vista o escuchada. Esta memoria acumulativa se puede representar en forma de curva de adopción o curva de respuesta publicitaria, que muestra la tasa de pérdida del impacto publicitario en el tiempo.

✓ Evaluación Modelo LightweightMM Google:

LMMM utiliza python para optimizar el gasto en marketing en todos los canales de medios, a nivel de modelo utiliza Numpyro y JAX para la programación probabilística, lo que hace que el proceso de modelado sea mucho más rápido (https://github.com/google/lightweight_mmm)

Estos son los pasos del modelo:



Cuadro: Fuente Google

Igual que en el modelo anterior la instalación del modelo bayesiano que es bastante complejo y al instalar la librería pystan, con problemas de instalación por el compilador de Visual Basic, superando un 30% más del tiempo estimado.

En este modelo esta terminado al 100% llegando a calcular el ROAS (Return on Advertising Spend)(*). En el contexto del marketing mix, ROAS se utiliza para evaluar la rentabilidad de la inversión publicitaria dentro del mix de marketing. Se ha medido el ROAS de cada canal publicitario utilizado dentro del mix de marketing, podemos determinar qué canales son más efectivos para alcanzar sus objetivos de marketing y asignar su presupuesto publicitario de manera más eficiente.

	roas_mean	roas_median	mroas	roas_avg
dm	6.067648	5.200908	8.176664	6.305869
inst	32.120422	29.527071	37.829563	32.800853
nsp	7.804195	5.160552	11.177744	10.364456
auddig	325.145473	273.614418	149.047283	313.753330
audtr	9.117158	8.268813	5.769686	8.846472
vidtr	20.146669	16.259589	12.025508	17.251446
viddig	63.139437	64.798941	53.981645	69.496131
so	8.899403	8.030128	3.907974	8.786031
on	9.055892	8.207717	6.848882	9.185394
sem	9.769199	9.507649	9.902280	9.299544

(*)métrica utilizada en el marketing digital para medir la efectividad de una campaña publicitaria. Se calcula dividiendo el ingreso generado por la campaña publicitaria por el coste total de la campaña.

✓ Evaluación Modelo Uber Orbit:

Este modelo está basado en el método de modelización que se denomina Bayesian Time-Varying Coefficients (BTVC), permite comprender mejor el efecto de cada medio de publicidad, viendo cómo varía el efecto a lo largo del tiempo con intervalos de confianza.

✓ Elaboración comparativa de modelos



Las conclusiones iniciales de estos modelos son:

- ✓ Robyn:
 - Los valores de ad-stock para cada medio y los niveles de saturación también para cada medio se calculan de manera óptima.
 - Este modelo utiliza la regression ridge que permite que los coeficientes de las variables de los medios sean pequeños y estén más distribuidos equitativamente.
 - Como modelo de machine learning usa prophet (desarrollado por Facebook) que permite descomponer la serie estacional y tendencia.
- ✓ LightweightMMM:
 - El modelo no tiene características específicas para incorporar el conocimiento del dominio, a diferencia de Robyn, que tiene esto como una característica específica y un objetivo de optimización para el algoritmo evolutivo de Nevergrad. Sin embargo, esto es algo que funciona bien de forma nativa dentro de los marcos bayesianos, ya que se pueden utilizar las priors de cada parámetro para informar al modelo de las opiniones fuertes o débiles que se tienen sobre la naturaleza de cómo funcionará cada canal. La forma en que está construido LightweightMMM permite conocer lo que está haciendo el modelo y poder utilizar unos valores propios a priori.
- ✓ Orbit:

Permite ajustar varios modelos, como LightweightMMM, Robyn y BTVC, y combinar sus resultados para obtener estimaciones más estables de los verdaderos efectos de los medios.

Orbit utiliza Stan para el modelado probabilístico, así como Pyro para el cálculo, que es similar al utilizado en LightweightMMM de Google. Al ser un modelo bayesiano, todos los parámetros se estiman a la vez, por lo que no hay un paso significativo de optimización de hiperparámetros como el que hay que esperar con Meta Robyn, que utiliza una biblioteca evolutiva independiente llamada Nevergrad para realizar esta tarea. Al igual que Google, aunque no como Meta, Uber tampoco utiliza herramientas como Prophet para predecir por separado la estacionalidad, sino que se gestiona de forma nativa como parte del modelo. Prophet utiliza técnicas bayesianas y es una herramienta de previsión de series temporales como Orbit.

Orbit tiene unos resultados mínimos para un MMM, pero le faltan bastantes visualizaciones en comparación con Robyn, ayudando a una mejor interpretación de resultados. Robyn aporta no solo aporta precisión frente a la predicción, sino métricas de precisión dentro de la muestra (datos que hemos introducido en el modelo para entrenarlo) y fuera de la muestra (datos que el modelo aún no se han visto), lo que ayuda a diagnosticar rápidamente la fiabilidad del modelo.

En el caso de Orbit el modelo no utiliza la transformación de adstock a las entradas, sino como hiperparámetro en su proceso final, este modelo para casos de productos que el proceso de compra es bajo funciona mejor.

Conclusión

Este es un ejercicio inicial para entender los diferentes modelos, como están construidos, que aportan y como se pueden utilizar. El análisis de estos diferentes enfoques y diferentes pruebas, me permitirá construir un modelo de marketing mix que sea más útil para el conjunto de datos y caso de uso.

B. Actividades no previstas y realizadas o programas:

- He creado un repositorio en Github para los entregables del TFG.
- Preparación del entorno de R y el IDE de Dataspell con diferentes librerías de complicada instalación
- Lectura de los manuales y videos de Robyn, LightweightMMM y Orbit para entender la metodología de estos modelos.

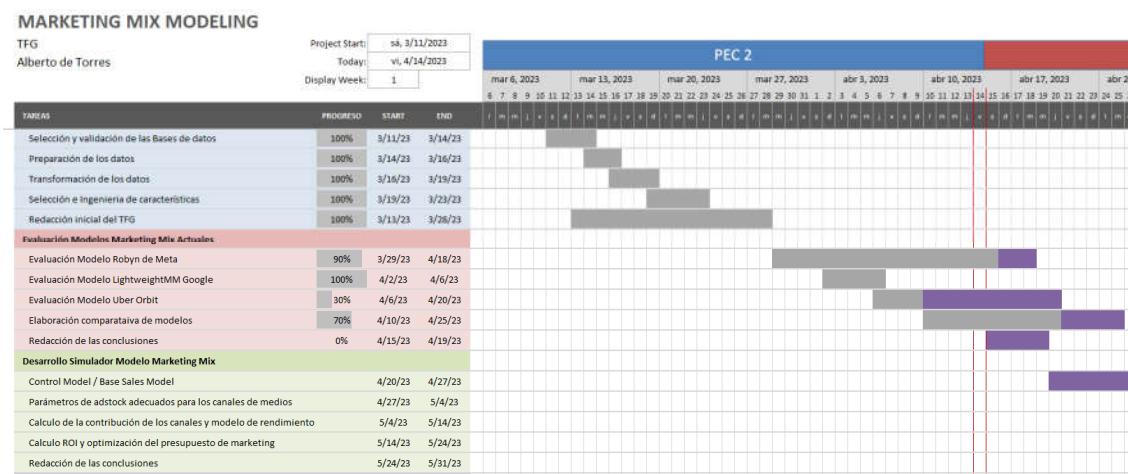
4. Relación de las desviaciones en la temporización y acciones de mitigación si procede y actualización del cronograma si procede:

Las desviaciones que he tenido han sido en los siguientes puntos:



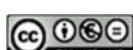
- Evaluación Modelo Robyn de Meta que he alcanzado un 90% sobre el objetivo, solamente me queda las simulaciones del ROI. Espero poder realizarlo en unos dos días de trabajo, asumibles dentro del próximo sprint del proyecto.
- Evaluación Modelo Uber Orbit que he alcanzado un 30% del objetivo, motivado porque los otros modelos me han llevado más tiempo que la estimación. Espero poder terminarlo en unos cinco días de trabajo, asumibles también dentro del próximo sprint del proyecto.
- Elaboración comparativa de modelos que he alcanzado un 70% del objetivo, motivado porque al no terminar los modelos no he podido hacer la tabla de comparación de los resultados. Espero poder terminarlo en unos tres días de trabajo, asumibles también dentro del próximo sprint del proyecto.
- El resto de los seis objetivos marcados se han alcanzado al 100% sin desviaciones.

He reprogramado el cronograma con los objetivos no alcanzados en los días que he estimado, pero mantengo las mismas acciones iniciales para este segundo sprint, con las mismas fechas de entrega:



5. Listado de los resultados parciales obtenidos hasta el momento:

- Base de datos seleccionada
- Análisis Exploratorio de Datos, notebook con análisis y transformación de los datos:
 1. Introducción al conjunto de datos
 - Descripción del conjunto de datos
 - Fuente de datos
 - Objetivos del análisis
- 2. Análisis de las variables numéricas
 - Estadísticas descriptivas (media, mediana, desviación estándar, etc.)
 - Histogramas, boxplots y densidad de distribución
 - Correlación entre variables numéricas
- 3. Análisis de las variables categóricas
 - Tablas de frecuencia y porcentajes
 - Gráficos de barras
- 4. Análisis de las relaciones entre variables
 - Matriz de correlación
 - Scatter plots y heatmaps
- 5. Análisis de correlación y regresión
- 6. Análisis de valores atípicos y datos faltantes
 - Identificación de valores atípicos
 - Tratamiento de valores atípicos
 - Identificación de datos faltantes
 - Tratamiento de datos faltantes
- 7. Análisis de resultados y conclusiones
 - Interpretación de los resultados del EDA
 - Conclusiones y recomendaciones para el análisis posterior
- Diccionario de datos:



Archivo que describe el conjunto de datos, incluyendo el significado, formato, tipo y rango de valores para cada variable. El diccionario de datos nos permite mejorar la gestión de datos, ya que ayuda a asegurar la consistencia y la calidad de los datos al proporcionar una descripción clara y precisa de cada variable en el conjunto de datos. Se ha incluido la siguiente información:

- Nombre de la variable: El nombre dado a cada variable en el conjunto de datos.
 - Tipo de variable: El tipo de variable, como numérica, categórica, binaria, de fecha, de texto, etc.
 - Descripción: Una descripción detallada de lo que representa la variable.
 - Rango de valores: Los valores mínimos y máximos que pueden tomar las variables numéricas, y los posibles valores que pueden tomar las variables categóricas.
 - Unidades: Las unidades de medida utilizadas en los datos numéricos.
- Modelo de Marketing Mix con la metodología de Robyn. Notebook con el modelo del set de datos seleccionado. Informe en pdf con los resultados del modelo
 - Modelo de Marketing Mix con la metodología LightweightMMM. Notebook con el modelo del set de datos seleccionado.
 - Borrador de la memoria del TFM con los apartados parciales según avance del proyecto.
 - Cronograma Gannt con los hitos del proyecto.

6. Comentarios de vuestro director particular si lo consideráis necesario

En general he trabajado bastante esta PEC, que ha sido de mucha documentación, trabajo de código en los notebooks y crear el entorno. Ahora ya tengo todo organizado y avanzado en la base del MMM por lo que este primer esfuerzo ha sido positivo para que en la tercera PEC tenga mi simulador de MMM terminado en el código y pueda dedicar el tiempo a escribir la memoria y preparar la presentación.



Anexo IX: Segundo Informe de Seguimiento



GRADO DE CIENCIA DE DATOS APLICADA

Segundo Informe de Seguimiento

18 Junio 2023

Descripción breve

Segundo Informe de Seguimiento



Alberto de Torres Pachón

Dirección académica: Xavier Florit

Responsable académico: Elena Rodríguez

Trabajo final de grado 22536



ADTP

PEC3: INFORME SEGUIMIENTO TFG MARKETING MIX MODELING

1. Identificación del trabajo y fecha del informe:

El trabajo corresponde al TFG sobre la creación de un Simulador de Modelo de Marketing Mix que permita simular la optimización de la inversión de marketing.

Este informe se realiza para la PEC 3 con fecha 26 de Mayo del 2023

2. Descripción del avance del proyecto

Grado de cumplimiento de los objetivos y resultados previstos en el plan de trabajo.

El grado de avance del proyecto se ha actualizado en el Cronograma (se adjunta Cronograma Gannt actualizado).

En global el avance en esta PEC3 ha sido del 100% y además he incluido dos tareas nuevas no planificadas, creación de un nuevo modelo basado en el método multiplicativo y el desarrollo de la aplicación web simulador denominado “Nebula Navigator”.

3. Relación de las actividades realizadas

A. Actividades previstas en el plan de trabajo:

I. Finalización del análisis EDA:

He terminado el análisis del data set y organizado el código para su interpretación. Con esto ya he podido incluir el análisis en la memoria del TFG.

Ver el borrador de la memoria del TFG con todos los análisis y conclusiones del análisis EDA.

II. Creación Modelo Multiplicativo:

Este es un objetivo que he añadido en esta PEC, como resultado de la investigación realizada sobre los tipos de modelos para el marketing mis, el asociativo es un modelo que aunque se basa en modelos tradicionales es más completo que la regresión multivariante. El resultado del modelo se puede ver en el borrador del TFG.

Este nuevo objetivo se ha cumplido en la PEC al 100%.

III. Finalización del Modelo Robyn:

Después de los problemas que me han dado la librería reticulate, que permite trabajar con Python en R. He conseguido estabilizar el modelo y ahora ya funciona bien.

Ver los resultados del modelo están en el borrador de la memoria del TFG.

Este objetivo se ha cumplido al 100% sobre lo planificado.

IV. Finalización del Modelo Bayesiano con LighweightMMM

He conseguido solucionar uno de los problemas que me daba este modelo, en la integración de cualquier data set, para poder convertirlo en vectores (tensores) para que se pueda utilizar en el framework de Tensorflow, que es el que está basado LightweightMMM. Los resultados del modelo se pueden ver en el borrador de la memoria del TFG.

Gracias a haber podido solucionar esto, me he decidido a empezar la creación de la aplicación web del simulador de MMM, que he llamado “Nebula Navigator”. Ya que este modelo LightweightMMM es el que después del análisis y evaluación entre todos los modelos es el que ha resultado mejor.

Este objetivo se ha cumplido al 100% sobre lo planificado.

V. Finalización del Modelo Stan



Este modelo que ya lo trabajo en la PEC anterior y que daba problemas con la librería stan de optimización en el entorno Windows. Para ello me he creado una maquina virtual en Linux e instalado un entorno de ejecución que ha permitido que el modelo se pueda ejecutar, aunque es poco estable por necesitar bastante procesamiento.

Este modelo que inicialmente tenía planificado usar para la aplicación web, por los buenos resultados y la explicabilidad del modelo, pero su alta demanda de procesamiento y dificultad en el entorno, ha motivado la no selección.

Los resultados del modelo se pueden ver en el borrador del TFG.

Este objetivo se ha cumplido al 100% sobre lo planificado.

VI. Desarrollo de una aplicación web de simulación del MMM

Como he comentado antes, al solucionar los problemas con los modelos y conseguir que el modelo bayesiano funcione bien y ser la mejor opción, me he decidido a incluir en el proyecto este nuevo objetivo que aunque implica una sobrecarga alta adicionales en el proyecto, me permitirá visualizar y dar un mayor valor a los resultados de la investigación y creación de los modelos. Además de ser el producto mínimo viable de un futuro producto de datos para su comercialización, permitiendo hacer pruebas reales con empresas.

El diseño de la aplicación, así como los principales wiframes de las pantallas a desarrollar se pueden ver en el borrador de la memoria del TFG.

En esta PEC se han conseguido avances de un 100% en el diseño de la aplicación y un 15% en el desarrollo del código, dejando esta parte para la siguiente PEC.

VII. Redacción de la memoria del TFG

VIII. Actividades no previstas y realizadas o programas:

- o Incorporación de un nuevo modelo de marketing mix, multiplicativo.
- o Estudio de diferentes diseños y herramientas para la aplicación Web.
- o Desarrollo de la aplicación Web del simulador “Nebula Navigator”.

IX. Relación de las desviaciones en la temporización y acciones de mitigación si procede y actualización del cronograma si procede:

En esta PEC he dedicado más tiempo que el planificado, pero me ha permitido recuperar las actividades pendientes de la PEC anterior y no solo completar las actividades planificadas para esta PEC, sino añadir nuevos objetivos. Espero se puedan cumplir estos nuevos objetivos en la próxima PEC, ya que he avanzado también en la redacción de la memoria, para así poder terminar todo en la próxima PEC.

He reprogramado el cronograma con los objetivos nuevos y ajustado las fechas al nuevo escenario. Creando nuevas acciones sobre las planificadas para segundo y tercer sprint, con las mismas fechas de entrega:

X. Listado de los resultados parciales obtenidos hasta el momento:

Los documentos y resultados parciales conseguidos hasta este momento son:



- Base de datos.
- Notebook con Análisis Exploratorio de Datos(EDA) y transformación de los datos:
 - Introducción al conjunto de datos
 - Descripción del conjunto de datos
 - Fuente de datos
 - Objetivos del análisis
 - Análisis de las variables numéricas
 - Estadísticas descriptivas (media, mediana, desviación estándar, etc.)
 - Histogramas, boxplots y densidad de distribución
 - Correlación entre variables numéricas
 - Análisis de las variables categóricas
 - Tablas de frecuencia y porcentajes
 - Gráficos de barras
 - Análisis de las relaciones entre variables
 - Matriz de correlación
 - Scatter plots y heatmaps
 - Análisis de correlación y regresión
 - Análisis de valores atípicos y datos faltantes
 - Identificación de valores atípicos
 - Tratamiento de valores atípicos
 - Identificación de datos faltantes
 - Tratamiento de datos faltantes
 - Análisis de resultados y conclusiones
 - Interpretación de los resultados del EDA
 - Conclusiones y recomendaciones para el análisis posterior
- Diccionario de datos:

Archivo que describe el conjunto de datos, incluyendo el significado, formato, tipo y rango de valores para cada variable. El diccionario de datos nos permite mejorar la gestión de datos, ya que ayuda a asegurar la consistencia y la calidad de los datos al proporcionar una descripción clara y precisa de cada variable en el conjunto de datos.

Se ha incluido la siguiente información:

- Nombre de la variable: El nombre dado a cada variable en el conjunto de datos.
- Tipo de variable: El tipo de variable, como numérica, categórica, binaria, de fecha, de texto, etc.
- Descripción: Una descripción detallada de lo que representa la variable.
- Rango de valores: Los valores mínimos y máximos que pueden tomar las variables numéricas, y los posibles valores que pueden tomar las variables categóricas.
- Unidades: Las unidades de medida utilizadas en los datos numéricos.
- Notebook con Modelo de Marketing Mix con metodología de Regresión multivariante

- Notebook con Modelo de Marketing Mix con metodología de Multiplicativa
- Notebook con Modelo de Marketing Mix con la metodología de Robyn.
- Notebook con Modelo de Marketing Mix con la metodología Bayesiana

LightweightMMM.

- Notebook con Modelo de Marketing Mix con la metodología Optimización Stan.
- Borrador de la memoria del TFM con los apartados parciales según avance del proyecto.
- Diseño y wiframes de la aplicación web “Nebula Navigator”.

- Cronograma Gannt con los hitos del proyecto.

XI. Comentarios de vuestro director particular si lo consideráis necesario

Esta PEC ha sido realmente muy importante, pues he podido avanzar mucho y consolidar todos los análisis, investigaciones y tiempo empleado en los modelos de las PECs anteriores. Aunque el esfuerzo y la dedicación ha sido mucho mayor a la planificación, creo que ha merecido la pena y espero que el MVP “Nebula Navigator” sea el comienzo de un producto funcional en el futuro y seguir avanzando en la investigación de los modelos matemáticos de camino al Máster próximo y doctorado.



Anexo X: Documentación Técnica de la aplicación web

GRADO DE CIENCIA DE DATOS APLICADA

Documentación Técnica

Descripción de la aplicación web

18 Junio 2023

Descripción breve

Descripción técnica de la Aplicación web



Alberto de Torres Pachón

Dirección académica: Xavier Florit

Responsable académico: Elena Rodríguez

Trabajo final de grado 22536



ADTP

Funcionalidades

Se presenta un listado con las funcionalidades agrupadas que han sido tomadas como requisitos de las diferentes reuniones con el señor Alberto. Se identifican 2 elementos principales y para cada uno de ellos se enumeran las funcionalidades necesarias.

- **Elemento Usuario**

Este elemento representa el modelo del objeto Usuario del cual nos interesa conocer su nombre completo, nombre de usuario, contraseña, y rol de usuario. Las funciones que este modelo tiene son las siguientes.

- Crear
- Editar
- Listar
- Eliminar

- **Elemento Training**

Este elemento representa el objetivo principal de Nebula Navigator, este se encarga de crear todo el modelo necesario para un DataSet, generando las gráficas del análisis de los datos las cuales se agrupan en 2 Fases del entrenamiento. A continuación, se mencionan.

1. Comprobación de la calidad de los datos

- a) Comprobación de la matriz de correlación
- b) Comprobación de las variaciones
- c) Comprobación de las fracciones de gasto
- d) Comprobación de los factores de inflación de la varianza

2. Resultados del entrenamiento del Modelo

- a) Distribución de las impresiones de los canales de media
- b) Evaluación de la precisión en las predicciones
- c) Media Insights
- d) Optimización

Las funciones del este elemento son las siguientes:

- Importar Dataset
- Eliminar Dataset
- Iniciar entrenamiento.

3. Arquitectura del sistema

Nebula Navigator es una aplicación web y está basada en una arquitectura cliente/servidor. Este software está compuesto por dos aplicaciones, la primera es una aplicación web encargada de toda la funcionalidad del Frontend y la segunda una API REST como servicio Backend encargada de devolver los datos del entrenamiento.

La arquitectura está conformada por los siguientes elementos:

- **Frontend**

Se encarga de ser la parte visible con la que los usuarios interactuarán con la aplicación, esta se ejecutara en el navegador web del cliente y se comunicara con el Backend por medio de peticiones GET y POST.

- **Backend**

Es la parte oculta al usuario y se encargará de ser la capa de acceso a los datos, se construyó como una API REST con el fin de ser consumida por la aplicación Frontend.

- **Base de datos local**

Nebula Navigator cuenta con una base de datos local, la cual se encargará de almacenar toda la información de los diferentes modelos que requiera de la aplicación.

4. Historias de usuario

Con el fin de presentar de manera ordenada las historias de usuario, estas serán agrupadas en las diferentes funcionalidades de la aplicación, si se requiere también estarán acompañadas de la documentación del Endpoint utilizado en la API.

4.1. ENDPOINTS PRINCIPALES

CRUD Modelo User

Se detallan los requisitos necesarios para el modelo de usuarios y sus endpoint empleados.

Historia de usuario	Crear usuario
ID	HU01
Como	
Usuario administrador	
Quiero	
Dar de alta a un usuario en la aplicación	
Para	
<i>Permitirle el ingreso al sistema por medio de usuario y contraseña.</i>	

Endpoint Crear Usuario.

Nebula Navigator API

download

read

GET /nebulanavigator/v1/download/{nameFile}/

 Interact

Class para descargar una grafica

Path Parameters

The following parameters should be included in the URL path.

Parameter	Description
nameFile required	

login

create

POST /nebulanavigator/v1/login

 Interact

Loguearse en la API y retornar un Token de Sesión :param request: (username, password) :return: Token sesión

startModel

create

POST /nebulanavigator/v1/startModel/

 Interact

Class que recibe un dataset .csv y genera el modelo de entrenamiento

training

create

POST /nebulanavigator/v1/training/

 Interact

Class que recibe un dataset .csv y genera el modelo de entrenamiento

CRUD Modelo User

Se detallan los requisitos necesarios para el modelo de usuarios y sus endpoint empleados.

Historia de usuario	Actualizar usuario
ID	HU02
Como	
Usuario administrador	
Quiero	
Editar la información de un usuario	
Para	

Corregir posibles cambios en el futuro.

Endpoint Crear Usuario.

update

 Interact

PUT `/nebulanavigator/v1/users/{id}/`

Clase que contiene los métodos genéricos para la vista de los User (Finca) ['get', 'post', 'put', 'delete']

Path Parameters

The following parameters should be included in the URL path.

Parameter	Description
<code>id</code> <small>required</small>	

Request Body

The request body should be a `"application/json"` encoded object, containing the following items.

Parameter	Description
<code>username</code> <small>required</small>	Requerido. 150 caracteres como máximo. Únicamente letras, dígitos y @./+/-/_
<code>first_name</code>	
<code>last_name</code>	
<code>email</code>	
<code>is_superuser</code>	Indica que este usuario tiene todos los permisos sin asignárselos explícitamente.
<code>password</code> <small>required</small>	

Historia de usuario	Listar usuarios
ID	HU03
Como	



Usuario administrador
Quiero
Listar todos los usuarios que están dados de alta en la aplicación
Para
Tener un control sobre ellos, realizando alguna acción o visualizar su información.

Endpoint Listar Usuarios.

users

list

 Interact

GET </nebulanavigator/v1/users/>

Clase que contiene los métodos genéricos para la vista de los User (Finca) ['get', 'post', 'put', 'delete']

Query Parameters

The following parameters should be included as part of a URL query string.

Parameter	Description
page	A page number within the paginated result set.



ADTP

list

Page

GET /nebulanavigator/v1/users/ 200

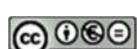
A page number within the paginated result set.

```
{
  "count": 11,
  "next": null,
  "previous": null,
  "results": [
    {
      "id": 1,
      "username": "alejojr",
      "first_name": "Alejandro",
      "last_name": "Caicedo Palacios",
      "email": "alejo@gmail.com",
      "is_superuser": true,
      "password": "pbkdf2_sha256$60000$7OpFUrKE17c"
    },
    {
      "id": 8,
      "username": "amartinez",
      "first_name": "Andres",
      "last_name": "martinez",
      "email": "amartinez@gmail.com",
      "is_superuser": true,
      "password": "pbkdf2_sha256$60000$WFSpoumlh3"
    },
    {
      "id": 2,
      "username": "atorres",
      "first_name": "Alberto"
    }
  ]
}
```

Close **Send Request**

Historia de usuario	Eliminar usuario
ID	HU04
Como	
Usuario administrador	
Quiero	
Dar de baja a un usuario en la aplicación	
Para	
Corregir errores o no permitir el acceso a la aplicación.	

Endpoint Eliminar Usuario.



delete

[Interact](#)

DELETE /nebulanavigator/v1/users/{id}/

Clase que contiene los metodos genericos para la vista de los User (Finca) ['get', 'post', 'put', 'delete']

Path Parameters

The following parameters should be included in the URL path.

Parameter	Description
id <small>required</small>	

CLASES IMPORTANTES DEL CODIGO



ADTP

```
class TrainingModel(APIView):
    """Class que recibe un dataset .csv y genera el modelo de entrenamiento"""

    def post(self, request):
        try:
            base_dir_media = settings.MEDIA_ROOT
            path_temp = os.path.join(base_dir_media, 'tmp/')

            file_dataset = request.FILES['document']
            objInputs = json.loads(request.data['objInputs'])

            destination = crateFileDataset(file_dataset)
            df = pd.read_csv(destination.name)

            SEED= 105
            data_size = len(df)

            # Seleccionar las columnas de interés del dataframe
            mdip_cols = objInputs['cols_mdip']
            extra_cols = objInputs['cols_extra']
            target_col = objInputs['cols_target']

            # Convertir los datos seleccionados en tensores de NumPy
            media_data = np.array(df[mdip_cols])
            extra_features = np.array(df[extra_cols])
            target = np.array(df[target_col[0]])

            # Agregar una nueva columna "costs" al dataframe
            cost_cols = objInputs['cols_cost']
            costs = np.array(df[cost_cols].sum())

            costs = costs.astype('float32')
            media_data = media_data.astype('float32')
            extra_features = extra_features.astype('float32')
            target= target.astype('float32')

            # Split and scale data.
            split_point = data_size - 47
            # Media data
            media_data_train = media_data[:split_point, ...]
            media_data_test = media_data[split_point:, ...]
            # Extra features
            extra_features_train = extra_features[:split_point, ...]
            extra_features_test = extra_features[split_point:, ...]
            # Target
            target_train = target[:split_point]
```

Generación de graficas

```
media_data_train = media_scaler.fit_transform(media_data_train)
extra_features_train = extra_features_scaler.fit_transform(extra_features_train)
target_train = target_scaler.fit_transform(target_train)
costs = cost_scaler.fit_transform(costs)

correlations, variances, spend_fractions, variance_inflation_factors = preprocessing.check_data_quality(
    media_data=media_scaler.transform(media_data),
    target_data=target_scaler.transform(target),
    cost_data=costs,
    extra_features_data=extra_features_scaler.transform(extra_features))

#<< TABLA - CORRELACIÓN -->
styled_matrix = correlations[0].style.background_gradient(cmap='RdBu', vmin=-1, vmax=1).format(precision=3)
html = styled_matrix.to_html()

with open(path_temp + 'tbl_correlacion.html', 'w') as f:
    f.write(html)

imgkit.from_file(path_temp + 'tbl_correlacion.html', base_dir_media + '/tbl_correlacion.png')


#<< TABLA - VARIANCIAS -->
variances_matrix = variances.style.format(precision=4).applymap(highlight_variances)
html = variances_matrix.to_html()

with open(path_temp + 'tbl_variances.html', 'w') as f:
    f.write(html)

imgkit.from_file(path_temp + 'tbl_variances.html', base_dir_media + '/tbl_variances.png')


#<< TABLA - GASTO -->
spend_matrix = spend_fractions.style.format(precision=4).applymap(highlight_low_spend_fractions)
html = spend_matrix.to_html()

with open(path_temp + 'tbl_gasto.html', 'w') as f:
    f.write(html)

imgkit.from_file(path_temp + 'tbl_gasto.html', base_dir_media + '/tbl_gasto.png')


#<< TABLA - INFILACIÓN -->
inflation_matrix = variance_inflation_factors.style.format(precision=4).applymap(highlight_high_vif_values)
html = inflation_matrix.to_html()

with open(path_temp + 'tbl_inflacion.html', 'w') as f:
    f.write(html)

imgkit.from_file(path_temp + 'tbl_inflacion.html', base_dir_media + '/tbl_inflacion.png')
```

Entrenamiento



```
#«— E N T R E N A M I E N T O —»
mmm = Lightweight_mmm.LightweightMMM(model_name="carryover")

number_warmup=1000
number_samples=1000

mmm.fit(
    media=media_data_train,
    media_prior=costs,
    target=target_train,
    extra_features=extra_features_train,
    number_warmup=number_warmup,
    number_samples=number_samples,
    seed=SEED)

mmm.print_summary()
```

Cargar un DATASET

```
def crateFileDataset(file_dataset):
    base_dir = settings.MEDIA_ROOT
    path = os.path.join(base_dir, 'datasets/')
    destination = open(path + file_dataset.name, 'wb+')

    for chunk in file_dataset.chunks():
        destination.write(chunk)
    destination.close()

    return destination
```

Descargar una Grafica

```

class FileDownloadImg(generics.ListAPIView):
    """Class para descargar una grafica"""

    authentication_classes = [TokenAuthentication, SessionAuthentication]
    permission_classes = [IsAuthenticated]

    def get(self, request, nameFile, format=None):
        base_dir = settings.MEDIA_ROOT
        pathFile = base_dir + '/' + nameFile
        print('Archivo a descargar: ', pathFile)
        document = open(pathFile, 'rb')
        response = HttpResponse(FileWrapper(document), content_type='application/png')
        response['Content-Disposition'] = 'attachment; filename="%s"' % nameFile
        return response

```

CLASE USUARIO

CRUD completo de Usuarios

```

class UserViewSet(viewsets.ModelViewSet):
    """Clase que contiene los métodos genéricos para la vista de los User (Finca)
    ['get', 'post', 'put', 'delete']"""

    authentication_classes = [TokenAuthentication, SessionAuthentication]
    permission_classes = [IsAuthenticated]

    serializer_class = UserSerializer

    def get_queryset(self, pk=None):
        if pk is None:
            return self.get_serializer().Meta.model.objects.all().order_by('username')
        return self.get_serializer().Meta.model.objects.filter(id=pk).first()

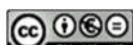
    def create(self, request, *args, **kwargs):
        try:
            user = User.objects.create_user(first_name=request.data['name'],
                                            last_name=request.data['lastname'],
                                            username=request.data['username'],
                                            password=request.data['password'],
                                            email=request.data['email'],
                                            is_superuser=request.data['isSuperUser'])

            respJson = [{"status": "OK", "message": "Created"}]
            return Response(respJson, status=status.HTTP_201_CREATED)
        except Exception:
            e = sys.exc_info()[1]
            if e.args[0] == 1062:
                fieldError = e.args[1].split('\\')
                respJson = [{"status": "ERROR", "message": "El campo: <b>' + fieldError[1]+ '</b> ya existe en el sistema!", "error": ""}]
                return Response(respJson, status=status.HTTP_226_IM_USED)
            else:
                return Response(e.args[0], status=status.HTTP_400_BAD_REQUEST)

    def update(self, request, pk=None, *args, **kwargs):
        if self.get_queryset(pk):
            user_serializer = self.serializer_class(self.get_queryset(pk), data=request.data)
            if user_serializer.is_valid():
                updateUser = User()
                updateUser.id=user_serializer.initial_data['id'],
                updateUser.username=user_serializer.initial_data['username'],
                updateUser.first_name=user_serializer.initial_data['first_name'],
                updateUser.last_name=user_serializer.initial_data['last_name'],
                updateUser.email=user_serializer.initial_data['email'],
                updateUser.password=make_password(user_serializer.initial_data['password']),
                updateUser.is_superuser=user_serializer.initial_data['is_superuser']
                updateUser.save()
                respJson = [{"status": "OK", "message": "Created"}]
                return Response(respJson, status=status.HTTP_200_OK)
            return Response(user_serializer.errors, status=status.HTTP_400_BAD_REQUEST)

```

Login en la aplicación



```
class LoginNebulaNavigatorApi(APIView):
    """ Class para autenticar un usuario via local y crear un login session """

    authentication_classes = ()

    def post(self, request):
        """Loguearse en la API y retornar un Token de Sesión
        :param request: (username, password)
        :return: Token sesión
        """
        user_obj = authenticate(username=request.data['username'],
                               password=request.data['password'])
        if user_obj:
            auth_login(request, user_obj)
            token, _ = Token.objects.get_or_create(user=user_obj)

            data = {
                'id': user_obj.id,
                'username': user_obj.username,
                'is_superuser': user_obj.is_superuser,
                'token': token.key
            }

            return Response(data, status=status.HTTP_200_OK)
        else:
            data = {'No Authenticate': 'Nombre de usuario y/o contraseña incorrectos'}
            return Response(data, status=status.HTTP_401_UNAUTHORIZED)
```

Acceso a la aplicación web

Para acceder a la aplicación web son necesarias credenciales que se deben facilitar solicitándolas y a continuación se puede acceder mediante el siguiente enlace:



Anexo XI: Fichero – Licencia aplicación web



GRADO DE CIENCIA DE DATOS APLICADA

Licencia de Aplicación web

Nebula Navigator

18 Junio 2023

Descripción breve

Texto de Licencia Software



Alberto de Torres Pachón

Dirección académica: Xavier Florit

Responsable académico: Elena Rodríguez

185



ADTP

Trabajo final de grado 22536

Licencia de Usuario Final del Software

Copyright (c) [2023] [Alberto de Torres Pachón]

Al instalar, copiar o usar este software, usted acepta los términos de esta Licencia. Si no está de acuerdo con los términos de esta Licencia, no instale, copie ni use este software.

1. LICENCIA

[Alberto de Torres Pachón] le concede una licencia no exclusiva, no transferible y no sublicenciable para usar este Software para su uso personal solamente. Este Software no puede ser copiado, distribuido, vendido, sublicenciado, alquilado, arrendado ni prestado. No se le permite utilizar este Software para fines comerciales.

2. PROPIEDAD INTELECTUAL

Este Software es propiedad de [Alberto de Torres Pachón] y está protegido por las leyes de propiedad intelectual y tratados internacionales. Usted no adquiere ningún derecho de propiedad sobre este Software.

3. TERMINACIÓN

Esta Licencia está vigente hasta que sea terminada. [Alberto de Torres Pachón] puede terminar esta Licencia en cualquier momento si usted viola cualquiera de sus términos. Al terminar, usted debe destruir todas las copias del Software.



ADTP

Anexo XII: Fichero – Licencia aplicación web



GRADO DE CIENCIA DE DATOS APLICADA

Repositorio GITHUB con el proyecto Nebula Navigator

18 Junio 2023

Descripción breve

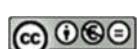
Dirección del GITHUB del proyecto nebula navigator



Alberto de Torres Pachón

Dirección académica: Xavier Florit

Responsable académico: Elena Rodríguez



ADTP

El trabajo final de grado presentado aquí, incluyendo todo el código, los datos y los análisis, se ha subido a GitHub, una plataforma de desarrollo colaborativo. GitHub se utiliza para versionar el código y los archivos asociados al proyecto, lo que permite mantener un registro detallado de todos los cambios realizados, además de facilitar la colaboración y el intercambio de información entre diferentes usuarios.

El proyecto se encuentra alojado en el siguiente enlace:
<https://github.com/detorrespa/TFG>

Al compartir este proyecto en GitHub, se busca promover la transparencia y la reproducibilidad en la investigación. Cualquier persona interesada en este trabajo puede revisar y descargar el código y los datos para reproducir los resultados, o incluso extender y adaptar el análisis para sus propios propósitos.

Además, el uso de GitHub permite que otros puedan contribuir a mejorar el código, corrigiendo errores o añadiendo nuevas funcionalidades, lo que puede resultar en una mejora continua del proyecto.

Invito a cualquier persona interesada a visitar el repositorio en GitHub, descargar y utilizar el código y los datos, y proporcionar cualquier comentario o sugerencia que pueda ayudar a mejorar este trabajo.

