UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Aprendizaje automático para la predicción de métricas y KPI en campañas de marketing digital

INFORME DE PASANTÍA PARA OPTAR POR EL TÍTULO DE MATEMÁTICO
PROYECTO CURRICULAR DE MATEMÁTICAS

Jorge Alexander Fajardo Muñoz
Jefe inmediato: Laura Carolina Padilla Hernández
Profesor director: Luis Alejandro Másmela Caita

Bogotá DC
Mayo de 2022

Resumen

En el presente documento se expone el desarrollo del proyecto de pasantía realizado con el equipo *Advanced Analytics* de TAAG (*Technology And Activation Group*) que hace parte de *Publicis Groupe*, una de las multinacionales más importantes de publicidad y relaciones públicas del mundo. En la pasantía se realizan distintos tipos de análisis de datos y análisis estadísticos, haciendo uso del lenguaje de programación Python para la implementación de modelos para la posterior predicción de métricas y KPI de algunas campañas de marketing digital.

Palabras clave: Machine learning, aprendizaje automático, marketing digital, predicción, KPI.

Agradecimientos:

A cada una de las personas que hicieron parte de mi desarrollo tanto profesional como académico. A todo el equipo TAAG por darme la oportunidad de crecer junto a ellos, en especial a Carolina Padilla por sus enseñanzas tanto profesionales como personales. Al profesor Alejandro Masmela por impulsarme a aprender y siempre mejorar académicamente. A mis compañeros y demás profesores del proyecto curricular de matemáticas, a mis amigos y familia, por acompañarme y apoyarme en este enriquecedor proceso.

Índice

1. Objetivo de la pasantía

1.1. Objetivo general

Buscar relacionar las matemáticas con el campo del marketing digital, haciendo uso de modelamiento estadístico y matemático, programación y análisis de datos.

1.2. Objetivos específicos

- Comprender las dinámicas y conceptos básicos del marketing digital.
- Aplicar modelos estadísticos y de aprendizaje automático para la predicción de métricas e indicadores de rendimiento en marketing digital.
- Analizar los datos de los clientes y dar conclusiones y recomendaciones de valor para el negocio.
- Analizar la viabilidad de implementar y desplegar en producción un modelo de predicción de métricas e indicadores de rendimientos (KPI)

2. Preliminares

En un mundo globalizado como el actual, el internet y la tecnología se han vuelto una parte intrínseca de nuestra sociedad. Por tal motivo, las matemáticas cobran especial relevancia dadas sus diversas aplicaciones en dichos campos, ya que permiten optimizar y mejorar los procesos de la industria, además de mejorar la experiencia de los usuarios. En la presente sección, se contextualizará al lector sobre los aspectos básicos y más importantes de las campañas en marketing digital, así como una breve introducción al aprendizaje automático.

2.1. Marketing digital

El marketing digital es la parte del marketing que realiza e implementa sus estrategias por medio de las nuevas tecnologías y medios digitales, como lo son la televisión, el internet o los teléfonos celulares.

Esta conlleva varias ventajas respecto al marketing tradicional, ya que al implementarse en distintos medios tecnológicos que se reinventan constantemente, esta también tendrá que estar en constante actualización, adoptando características modernas que permitan la optimización de los procesos y una mejora en los resultados de las campañas.

2.1.1. Campaña

Una campaña de marketing digital es una estrategia ejecutada por medio de los diferentes canales digitales (correo electrónico, redes sociales, motores de búsqueda en internet) con los cuales los clientes interactúan, esto con el objetivo de generar beneficios, mejorar sus KPI (*Key Performance Indicator* o indicador clave de rendimiento) y aumentar su tasa de conversiones (suscripciones a un servicio, venta de un producto). Se usará la frase pautar para referirse a la acción de ejecutar una campaña sobre cierto lapso de tiempo.

2.1.2. Métrica

Una métrica en marketing digital se define como una variable que permite medir alguna acción en concreto, los siguientes son ejemplos de algunas métricas populares [8]:

- Clics - acción en la cual se oprime un botón del mouse encima de alguna sección o elemento de una página web
- Conversiones - acción deseada que realice el usuario (como una compra en un *e-commerce* o comercio electrónico)

- Leads - clientes potenciales
- Engagement - porcentaje de interacciones obtenidas con respecto del total de veces que un contenido fue visualizado
- Impresiones - frecuencia con la que se muestra un anuncio

2.1.3. KPI

Un KPI, es una medida o indicador sobre el rendimiento de algún proceso. En el área del marketing digital, los KPI indican el rendimiento de algunas métricas importantes en nuestra campaña digital, algunos ejemplos de KPI pueden ser:

- ROI - rendimiento económico que se tiene al realizar una inversión
- CPM - Costo por mil, en este contexto se calcula con la fórmula $1000 \times \frac{\text{Inversión}}{\text{\# Impresiones}}$
- CPC - costo por click, se calcula con la formula $\frac{\text{Inversión}}{\text{\#Clicks}}$
- CPL - costo por lead, se calcula con la formula $\frac{\text{Inversión}}{\text{\#Leads}}$
- CPA - costo por conversión, se calcula con la formula $\frac{\text{Inversión}}{\text{\#Conversiones}}$

En el presente proyecto, se trabajará con métricas y KPI específicos para la plataforma Google Ads. Google Ads es una plataforma que pertenece a Google, en la cual se puede generar publicidad en linea, esta es especializada en campañas pautadas en el motor de búsqueda de Google, a nivel general [8].

2.2. Conceptos del aprendizaje automático - *Machine Learning*

El aprendizaje automático es una rama de la inteligencia artificial que usa distintos tipos de datos, técnicas matemáticas y de programación con el objetivo de lograr que un computador aprenda a realizar cierta tarea. En muchos casos esto se hace en función de crear un modelo que resuelva una tarea o acción, dicho modelo se debe entrenar y probar con el conjunto de datos que corresponda, junto con el ajuste de sus hiperparámetros (parámetros del modelo que ayudan a controlar y mejorar el proceso de aprendizaje) hasta lograr un buen nivel de precisión.

Existen dos tipos de algoritmos y modelos principales: modelos de regresión y los modelos de clasificación. Cada uno de estos tiene aplicaciones que dependen del problema que se busque abordar, por ejemplo, se podría realizar un modelo el cual permita clasificar los anuncios que se le presentan a una persona en internet según sus búsquedas más recientes o comunes [6]. Por otro lado, si el objetivo fuera predecir el costo que tiene una vivienda dependiendo de los metros cuadrados que esta posee, un algoritmo apropiado sería aquel de tipo regresión. El problema que se plantea implica predecir los KPI para ciertas campañas, conociendo su inversión, su fecha, el objetivo y canal por el cual se pautó dicha campaña, esto no es más que un problema de regresión.

Los conceptos que se enunciarán a continuación son fundamentales y deben tenerse en cuenta para el desarrollo del proyecto.

2.2.1. Aprendizaje supervisado

En el aprendizaje automático se encuentran diferentes tipos de algoritmos, como el aprendizaje supervisado, no supervisado, semisupervisado y por refuerzo [6]. En este caso se trabajará con aprendizaje supervisado, cuyas características principales son establecer una función entre las entradas y salidas, es decir, se predicen dichas salidas, con base en datos históricos pasados cuya etiqueta se conoce, y donde se divide el conjunto de datos en dos subconjuntos, un conjunto de entrenamiento y otro de prueba. Se entrará más a detalle con este tipo de algoritmos cuando se empiece la creación de los modelos.

2.2.2. Mínimos cuadrados

Sea $\{(x_1, y_1), ..., (x_n, y_n)\}$ un conjunto de n puntos. La recta que minimiza la suma de los cuadrados de las desviaciones verticales de todos los puntos de la recta tiene la siguiente pendiente y el siguiente intercepto [11, cap 11.3]:

$$\hat{\beta_1} = \frac{\sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x})}{\sum_{i=1}^{n}(x_i - \bar{x})^2},$$

$$\hat{\beta_0} = \bar{y} - \hat{\beta_1}\bar{x},$$

donde $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ y $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$.

2.2.3. Regresión lineal

Las variables $X_1,...,X_k$ serán llamadas predictoras (o variables independientes) y la variable Y será llamada respuesta (o variable dependiente). El valor esperado de Y para unos valores $x_1,...,x_k$ de $X_1,...,X_k$ será llamada regresión de Y en $X_1,...,X_k$. En este caso, se asume que la regresión $E(Y|x_1,...,x_k)$ es una función lineal con la siguiente forma:

$$E(Y|x_1,...,x_k) = \beta_0 + \beta_1 x_1 + ... + \beta_k x_k$$

Los coeficientes $\beta_0,...,\beta_k$ serán llamados coeficientes de regresión. Dichos coeficientes tienen valores desconocidos, por tanto serán considerados como parámetros que deben ser estimados. Un conjunto de estimadores de los coeficientes de regresión $\beta_0 + ... + \beta_k$ que pueden ser calculados de este conjunto es $\hat{\beta}_0 + ... + \hat{\beta}_k$, los cuales son obtenidos usando el método de mínimos cuadrados [11, cap 11].

2.2.4. Máquinas de vectores de soporte (máquinas de vectores de regresión)

Las máquinas de vectores de soporte son un algoritmo que usa principios geométricos para encontrar un plano (separador, en el caso de problemas de clasificación, y regresor en el caso de problemas de regresión). La característica principal de dicho algoritmo es el uso de margenes, la cual es la distancia de cada punto al plano separador, el objetivo es minimizar dicha margen para tener el menor error posible en las predicciones. Sea $\{(x_1, y_1),...,(x_n, y_n)\} \subset X \times \mathbb{R}$ un conjunto de n puntos de entrenamiento, con $X \subset \mathbb{R}^d$. Se consideran las funciones lineales f de la forma:

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

con $w \in X$,$b \in \mathbb{R}$ y $\langle , \rangle$ que denota el producto interno (producto punto) en X. El objetivo es encontrar la función $f(x)$ que minimice el error ϵ, es decir, que tenga la menor distancia entre la predicción y el valor real.

Para reducir dicho error se necesita conseguir el w mas pequeño que sea posible. Esto se puede realizar con el siguiente problema de optimización convexa:

$$\min \frac{1}{2}\|w\|^2 \quad \text{sujeto a} \quad \begin{cases} y_i - \langle w, x_i \rangle - b \leq \epsilon \\ \langle w, x_i \rangle + b - y_i \leq \epsilon \end{cases}$$

Las condiciones a las cuales está sujeto el problema de optimización resultan en una suposición de que la función f existe y es posible encontrar una que ayude a aproximar todos nuestros puntos [3][10].

Para el desarrollo del código de este algoritmo se usan los siguientes hiperparametros:

- kernel: Especifica el kernel que usará el algoritmo (rbf, polinomial, lineal, sigmoide)
- Gamma: Coeficiente para el kernel escogido previamente
- Degree: Si el kernel es polinomial, será el grado de dicho polinomio, si se escoge otro kernel este hiperparametro se omite.
- C: Parámetro de regularización, penalizará la función de coste del algoritmo.

2.2.5. Bootstrap

El bootstrap es una herramienta para evaluar la precisión estadística. Esta puede ser usada para estimar el error en la predicción dependiendo del muestreo. Supóngase que se tiene el conjunto de entrenamiento $Z = \{(x_1, y_1), ..., (x_n, y_n)\}$, el objetivo es crear conjuntos de datos al azar y con reemplazo de los datos de entrenamiento, manteniendo el mismo tamaño que el conjunto de datos original. Este proceso se realiza B veces, lo cual producirá B conjuntos bootstrap, para luego reajustar el modelo a cada uno de los conjuntos bootstrap y así evaluar el comportamiento y rendimiento de los ajustes sobre las B repeticiones [5, cap 7.11].

2.2.6. Bagging

El bagging es una técnica o metaalgoritmo enfocado en disminuir la varianza y aumentar la precisión de los modelos (dado que se relaciona con el bootstrap). Supóngase que queremos ajustar un modelo de regresión al conjunto de datos $Z = \{(x_1, y_1), ..., (x_n, y_n)\}$ y se obtiene la predicción $\hat{f}(x)$ para el input x. El bagging promediará esta predicción sobre una colección de muestras de bootstrap, reduciendo su varianza [5, cap 8.7].

Para cada muestra de bootstrap Z^{*b}, con $b = 1, 2, ..., B$. Se ajustará el modelo dada la predicción $\hat{f}^{*b}(x)$. El bagging estimado se define como:

$$\hat{f}_{\text{bag}}(x) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}^{*b}(x)$$

2.2.7. Bosques aleatorios

Los bosques aleatorios son un algoritmo basado en árboles, la idea principal detrás de este algoritmo es similar a la del bagging, usa bootstrap para lograr promediar la predicción de tal forma que la varianza en los datos sea lo menor posible. Los árboles pueden ser vistos como una buena opción de bagging ya que suelen trabajar bien con interacciones complejas entre las variables. El algoritmo de bosque aleatorio se compone de los siguientes pasos [5, cap 15]:

1. Para $b = 1$ a B:
    a. Cree una muestra bootstrap Z^* de tamaño N del conjunto de datos de entrenamiento.
    b. Cree un árbol T_b de un bosque aleatorio con los datos bootstrap, de forma recursiva repita los siguientes pasos para cada uno de los nodos terminales del árbol hasta alcanzar el tamaño mínimo de nodo (nmin).
        i. Seleccione m variables al azar de las p variables.
        ii. Escoja la mejor variable (punto de división) entre las m variables.
        iii. Divida el nodo en dos nodos hijo.
2. La salida de los árboles viene dada por $\{T_b\}_1^B$

Para predecir un nuevo valor para un dato x en un problema de regresión se hace:

$$\hat{f}_{\text{rf}}^B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$$

Para el desarrollo del código de este algoritmo se usan los siguientes hiperparametros:

- bootstrap: Especifica si se usa bootstrap para crear cada uno de los árboles.
- Max depth: Profundidad máxima de cada árbol en el bosque.
- Max features: El número de características a considerar al buscar la mejor división (auto, sqrt, log2). Cada árbol se podrá entrenar con diferentes características.
- Min samples leaf: El número mínimo de muestras requeridas para estar en un nodo hoja.
- Min samples split: El número mínimo de muestras requeridas para dividir un nodo interno.
- N estimators: Número de árboles en el bosque.

2.2.8. Variable Dummy

Una variable Dummy es aquella variable categórica cuyos únicos valores posibles son el 0 y el 1. Se usa para representar de forma numérica variables categóricas cuyos posibles valores suelen ser texto. Así, una variable categórica con dos posibles valores, Carro y Moto, tomaría la siguiente forma:

Vehículo	Dum_Carro	Dum_Moto
Carro	1	0
Moto	0	1
Moto	0	1
Carro	1	0
Carro	1	0
Carro	1	0
Moto	0	1

Figura 1: Variables dummy para la variable vehículo

Si se considera una variable dummy como variable predictora o independiente en un modelo de regresión

$$Y = \beta_0 + \beta_1 D + u$$

La regresión ahora se divide en dos posibles formas dependiendo el valor de la variable dummy, esto es:

$$Y = \begin{cases} \beta_0 + u, & \text{cuando } D = 0 \\ (\beta_0 + \beta_1) + u & \text{cuando } D = 1 \end{cases}$$

2.3. Medidas del error y/o precisión de modelos

En el proceso de creación de cualquier modelo estadístico o de aprendizaje automático es necesario contar con una forma de aproximar el error que tiene dicho modelo en sus predicciones. Por tal motivo en la presente sección se definirán algunas medidas que facilitan dicha tarea y que permiten tener una idea de qué tan preciso es el modelo.

2.3.1. R2 (Coeficiente de determinación R cuadrado)

Bajo la necesidad de estimar qué tan bien las variables $X_1, ..., X_k$ explican la variación en la variable Y, se usará el coeficiente de determinación R cuadrado. Dicho de otra forma, la proporción de la

variación entre los valores $y_1, ..., y_n$ que es explicada por la regresión (ajustada) es dada por el valor [4]:

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

con y_i los valores reales y $\hat{y}_i$ su predicción correspondiente.

2.3.2. MAPE (Error absoluto medio porcentual)

El MAPE es una medida que sirve para evaluar el error de los modelos de predicción, este se calcula de la siguiente forma [2]:

$$\text{MAPE} = \frac{100\,\%}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

con y_i los valores reales y $\hat{y}_i$ su predicción correspondiente.

2.3.3. RMSE (Error cuadrático medio)

Otro estadístico de error común es el RMSE. Este se calcula de la siguiente forma [11]:

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

con y_i los valores reales y $\hat{y}_i$ su predicción correspondiente.

2.4. Validación cruzada

Con el fin de evitar el sobreajuste (el modelo se ajusta bien a los datos de entrenamiento y prueba pero sus resultados generales o con datos nuevos son poco precisos), se usará una herramienta que ayuda a validar el rendimiento del modelo realizando pruebas de forma iterativa en el conjunto de datos, tal que al conjunto de datos lo dividirá en k partes (o pliegues) y entrenará el modelo con $k-1$ de dichas partes, mientras que con la parte restante se probará el modelo. De tal forma, se tendrá una validación más exacta de la precisión que se presenta. En rasgos generales, el error o precisión que se presenta en el conjunto de entrenamiento suele ser muy bueno, sin embargo en la mayoría de casos se trata de un valor muy optimista que se aleja de la realidad. Es por tal motivo que la implementación de la validación cruzada es primordial en la creación de los modelos [6].

3. Modelo para la predicción de KPI y/o métricas

En este proyecto se hará énfasis en la predicción de tres métricas diferentes, las cuales son clicks, impresiones y conversiones. Además, cada campaña tendrá un objetivo, es decir, la campaña se pauta para conseguir una métrica o KPI en específico. Para la predicción de las métricas se usarán los resultados obtenidos de algunos KPI en el histórico de datos.

3.1. Generalidades y aspectos clave

Las campañas son una parte fundamental del marketing digital y es por tal razón que siempre se buscan optimizar los KPI y métricas en ellas, es decir, se quiere obtener los mejores resultados para el KPI o métrica según el objetivo que se tenga. Si en la campaña se implementa un vídeo, es posible que el objetivo sea conseguir la mayor cantidad de reproducciones; por otro lado, si en una página web se tiene un formulario y la intención es conseguir la mayor cantidad de personas registradas, se podría pensar en conseguir la mayor cantidad de leads posibles o incluso optimizar el CPL (costo por lead). Es bajo estas premisas que surge la pregunta, ¿cómo se puede saber la cantidad de KPI o métricas que puede obtener una campaña y así mismo cómo podemos optimizar estos resultados y la inversión que debemos hacer?. La respuesta, aunque depende mucho de la calidad de los datos, es usar métodos estadísticos y de aprendizaje automático (*machine learning*).

3.2. Herramientas

Algunas de las herramientas usadas en este proyecto son:

- SQL (*Structured Query Language*): Es un tipo de lenguaje de programación estructurado enfocado en administrar, consultar y modificar sistemas de bases de datos relacionales.
- GCP (*Google Cloud Platform*): Plataforma de servicios de computación en la nube de Google.
- BigQuery: Servicio de almacenamiento de datos en la nube, proporcianada por GCP. Se permiten realizar diferentes procesos usando lenguaje SQL, además cuenta con herramientas para realizar modelos de aprendizaje automático.

Inicialmente se hace uso del lenguaje SQL por medio de GCP, más en específico BigQuery. Se usa esta herramienta para hacer la limpieza inicial de los datos, ya que todas las bases de datos están almacenadas en GCP, es importante realizar una primera limpieza en dicha plataforma para luego extraer los datos y realizar el desarrollo respectivo en el lenguaje de programación escogido.

Todos los modelos se desarrollaron usando el lenguaje de programación Python el cual es un lenguaje de programación multiparadigma que cuenta con bastantes aplicaciones y popularidad, además posee facilidades a la hora de realizar código y programas de ciencia de datos. En principio, los modelos fueron desarrollados y ejecutados de forma local en notebooks de Jupyter, sin embargo se está en proceso de migración de los modelos al servicio en la nube de Google, GCP.

Se usan algunas librerías diferentes en el desarrollo de los modelos en función de los requerimientos que se iban presentando, entre ellas están:

- Numpy: Librería especializada en el tratamiento de vectores y matrices.
- Pandas: Extensión de Numpy, diseñada principalmente para tareas de ciencia de datos.
- Statsmodel: Modulo para la creación y análisis de modelos estadísticos.
- Scikit-learn: Biblioteca especializada en la creación de modelos de Machine Learning.
- Matplotlib: Librería enfocada en la creación de gráficos haciendo uso de vectores y matrices.

3.3. Limitaciones

En el desarrollo del modelo se presentaron ciertas limitaciones relacionadas con los datos y cómo los modelos no se ajustaban correctamente a ellos. En el mundo del marketing digital existen bastantes efectos externos que afectan el rendimiento de las campañas y en la mayoría de casos, no se tiene control suficiente sobre estos efectos. De igual manera, la forma en la que se implementa cada una de estas campañas y la estrategia previa a dicha implementación afecta el rendimiento de estas campañas, por lo tanto, si se cambia constantemente de estrategia se podrían experimentar cambios abruptos en sus resultados, incluso en campañas que cuentan con una misma inversión, objetivo y canal. Es por estas dos razones principales que la predicción se dificulta para algunos KPI, en específico, las conversiones son el KPI con el comportamiento menos lineal de todos, teniendo variaciones que generan problemas al intentar ajustar los modelos, lo cual se ve reflejado al momento de probar con datos nuevos para el modelo.

3.4. Datos

Los datos usados para la predicción son una lista de registros o etiquetas de las campañas ejecutadas durante el año 2021 así como las etiquetas conseguidas para enero y febrero del 2022. Cada una de las etiquetas hace referencia a una campaña y los datos se encuentran segmentados por día. En total se cuenta con 4226 etiquetas, y con 4 características (variables predictoras o independientes) iniciales (durante el desarrollo de los modelos las variables que se usan cambian dependiendo del modelo escogido) y 8 características luego de las transformaciones y limpieza de datos.

4. Metodología

La metodología empleada en el desarrollo del proyecto se fundamentó en proceder en 4 etapas diferentes, una etapa inicial de entendimiento del problema, la cual se ha desarrollado desde el inicio del documento, siguiendo con una etapa de limpieza, transformación y análisis descriptivo de los datos, luego, una etapa de modelamiento, y finalmente, una etapa de conclusiones y recomendaciones.

Etapa	Descripción
1	Entendimiento del problema
2	Limpieza, transformación y análisis descriptivo de los datos
3	Modelamiento
4	Conclusiones y recomendaciones

4.1. Limpieza, transformación y análisis descriptivo de los datos

El conjunto de datos sin modificaciones se ve de la siguiente forma:

mes	canal	objetivo	inversion_ejecutada	impresiones	clicks	conversiones
1	VIDEO	AWR	1.455658e+05	53384.0	60.0	3.00
1	SH	LEAD	1.915885e+06	19662.0	4490.0	660.92
1	SH	LEAD	1.777105e+06	22631.0	3628.0	555.92
1	SH	LEAD	1.493301e+06	13089.0	4059.0	616.42
1	SH	LEAD	1.587242e+06	13306.0	4272.0	615.42
...	...	...	...	...	...	...
12	SH	ACQ	7.731000e+04	1900.0	155.0	337.84
12	SH	ACQ	7.272500e+04	1810.0	165.0	325.34
12	SH	ACQ	7.453500e+04	2865.0	155.0	319.34
12	SH	ACQ	5.991500e+04	2350.0	125.0	257.84
12	SH	ACQ	6.337500e+04	1505.0	125.0	281.84

Figura 2: Algunos registros del conjunto de datos inicial

Inicialmente, se eliminan aquellas etiquetas cuya inversión ejecutada (es decir la inversión que se realizó para la campaña) es igual a cero, claramente no es de interés tener estas etiquetas ya que el supuesto es que se va a invertir dinero en las campañas. De igual forma se eliminan aquellas etiquetas con valores NaN o null (NaN, del acrónimo *Not a number*, es un valor indeterminado. null, hace referencia a la nada o ausencia de un dato) para la inversión. Para aquellas variables categóricas

(canal y objetivo), se aplica una transformación en variables dummys ya que muchos de los modelos realizados necesitan exclusivamente variables de tipo numérico.

Se realiza una prueba para conocer el coeficiente de correlación de Pearson de las variables, con el fin de verificar si se puede tener multicolinealidad (es decir, se presenta un coeficiente muy alto de correlación entre las variables predictoras o independientes), ya que esto podría afectar negativamente los modelos:

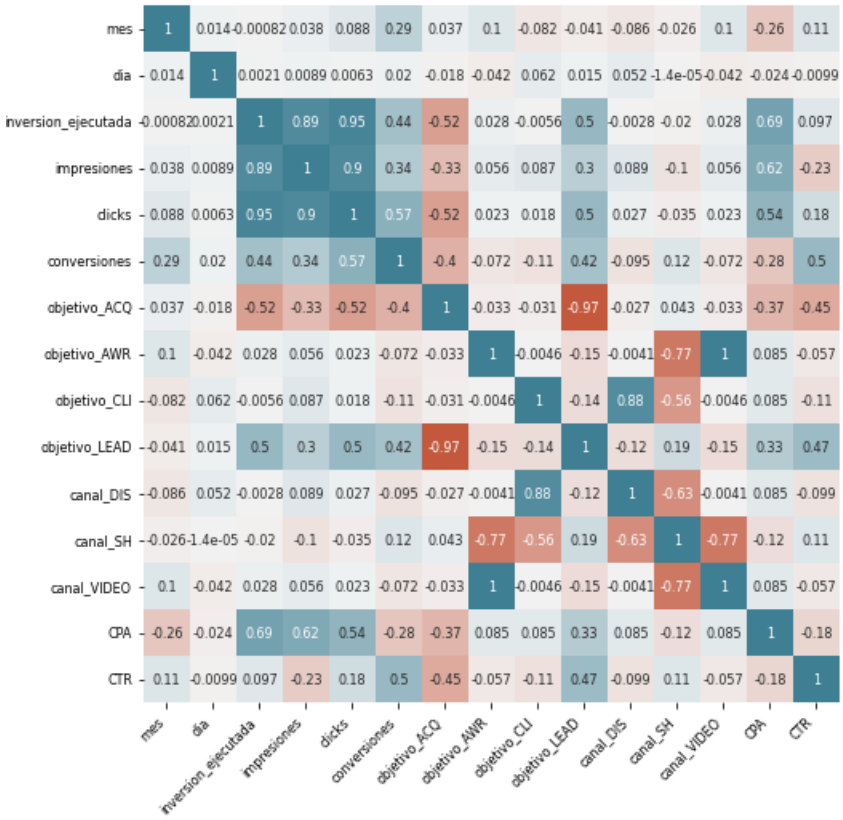


Figura 3: Matriz de calor con coeficientes de correlación

En primer lugar, se observa que no existe correlación significativa entre las variables dependientes, por lo tanto no se tendrían problemas de multicolinealidad, lo cual permite seguir considerando las mismas variables.

Además, se evidencia que el coeficiente de correlación de la inversión ejecutada con las métricas es positivo, esto da indicios de cuál de ellos puede ser más fácil de predecir. Se consideran los siguientes gráficos de dispersión de la inversión ejecutada frente a cada métrica:

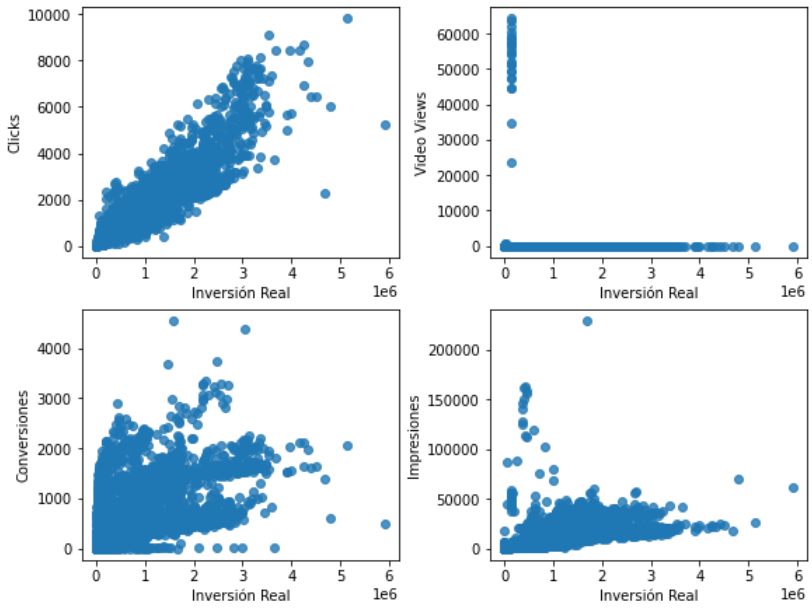


Figura 4: Gráfico de dispersión de los KPI's frente a la inversión ejecutada

Visualizando los gráficos, parecen haber indicios de que la métrica de Video Views tiene gran parte de sus datos iguales a cero, lo que imposibilita la predicción de dichos valores por medio de un modelo (los datos no tienen la calidad suficiente para entrenar un modelo). Por otro lado, los clicks e impresiones tienen un coeficiente de correlación cercano a uno, comparando con la inversión ejecutada, esto indica que dichas variables podrían ser predichas de forma más sencilla únicamente usando la variable inversión ejecutada, ya que al parecer esta explica de buena forma el comportamiento de dichas métricas. Finalmente, las conversiones obtienen un coeficiente de correlación positivo bajo, además de tener bastante dispersión en sus datos, lo cual genera indicios de que será una métrica difícil de predecir.

Inicialmente se planteó la predicción de los Video Views, sin embargo más del 90 % de sus registros eran cero, lo cual imposibilita su predicción, por tanto se decide no tomar en cuenta dicha métrica.

En el siguiente paso se realizaron distintas pruebas para identificar la distribución a la que se ajustan los datos. Sin embargo, los datos no se ajustan de manera correcta y/o precisa a alguna distribución conocida, como lo puede ser la distribución Normal, Binomial, Binomial negativa o Poisson.

Se realiza un análisis con gráficos de tipo boxplot donde se evidencia una gran cantidad de datos atípicos para la inversión ejecutada y las métricas, no obstante, se trata de valores que en situaciones especificas pueden suceder y por tanto se deben considerar en el modelo. Sin embargo, en el caso de las impresiones se encuentra un registro en especifico que provoca que el modelo no se ajuste bien a los datos (es decir, al probar el modelo con dicho registro, se nota un R2 mucho menor que sin

el). En ese caso en particular, se contaba con una inversión dentro de los rangos normales y con una cantidad de 229680 impresiones:

```
count      4389.000000
mean       8700.414673
std       13579.782298
min           4.000000
25%         979.000000
50%        3140.000000
75%       11280.000000
max      229680.000000
Name: impresiones, dtype: float64
```

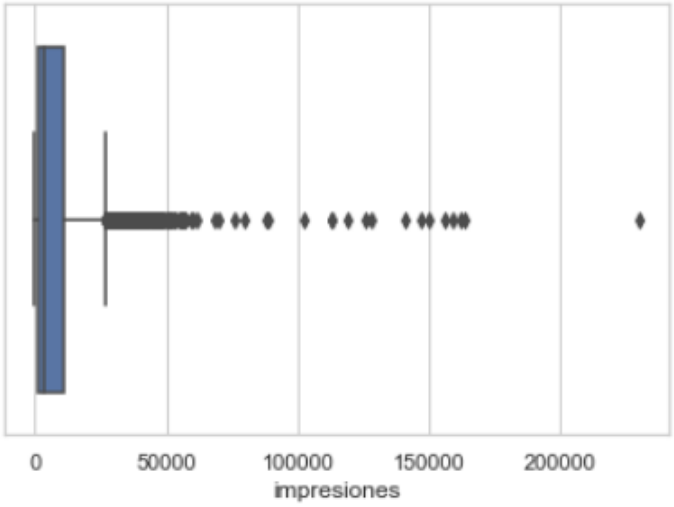


Figura 5: Boxplot para las impresiones

Este representa un evento particular que se aleja de los rangos normales del conjunto de datos, por tal razón se elimina dicho punto.

Por otro lado, en el marketing se suele presentar estacionalidad, esto significa que en el transcurso del tiempo se presentan alteraciones y cambios en el rendimiento de las campañas. Un ejemplo de esto son las campañas realizadas en diciembre, por la cantidad de personas buscando regalos para navidad se podría esperar que ciertas campañas generen un impacto más significativo que en meses como febrero o julio, donde la actividad económica baja considerablemente. Además, continuando con el ejemplo de diciembre, es claro que meses como noviembre y enero son muy cercanos a diciembre, por lo tanto seria ideal enseñarle al modelo que dichos meses son cercanos entre ellos y por tanto podrían heredar o tener resultados similares, esto ayudará a conservar la propiedad natural cíclica de las fechas. La

transformación se realiza únicamente a los meses, ya que el día en especifico no es relevante en este contexto. La transformación se hace en dos dimensiones, es decir, la variable mes se divide en dos nuevas variables, esto se hace usando dos funciones cíclicas muy conocidas, como lo son la función seno y coseno [1]. Sin entrar mucho en detalle, el código es:

```
def encode(data, col, max_val):
    data[col + '_sin'] = np.sin(2 * np.pi * data[col]/max_val)
    data[col + '_cos'] = np.cos(2 * np.pi * data[col]/max_val)
    return data
```

Figura 6: Función de transformación de los datos

Para recordar: el objetivo de esta transformación es que el modelo tenga en cuenta que algunos meses son cercanos/lejanos a otros.

Se cuenta con los siguientes parámetros:

- data: hace referencia al conjunto o marco de datos que usamos.
- col: la columna o variable a la cual se aplica la función.
- max_val: hace referencia al máximo valor que toma la columna, si se trabaja con días, el máximo será 31, en este caso al trabajar con meses, el máximo será 12.

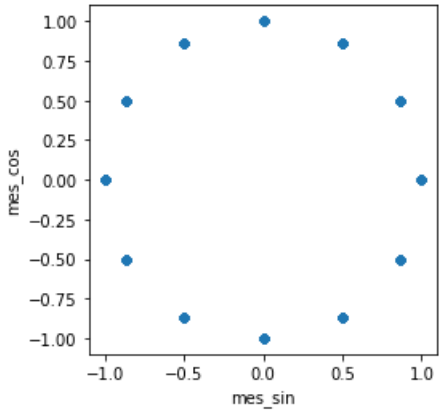


Figura 7: Transformación de la variable mes

Finalmente, el modelo contará con las variables predictoras mes (con su respectiva transformación en dos componentes), canal (en variables dummys), objetivo (en variables dummys) e inversión. Por el lado de las variables de respuesta se tienen conversiones, clicks e impresiones.

Después de cada una de las transformaciones realizadas, el conjunto de datos toma la siguiente forma:

inversion_ejecutada	impresiones	clicks	conversiones	objetivo_ACQ	objetivo_CLI	objetivo_LEAD	canal_DIS	canal_SH	mes_sin	mes_cos
1.455658e+05	53384.0	60.0	3.00	0	0	0	0	0	5.000000e-01	0.866025
1.915885e+06	19662.0	4490.0	660.92	0	0	1	0	1	5.000000e-01	0.866025
1.777105e+06	22631.0	3628.0	555.92	0	0	1	0	1	5.000000e-01	0.866025
1.493301e+06	13089.0	4059.0	616.42	0	0	1	0	1	5.000000e-01	0.866025
1.587242e+06	13306.0	4272.0	615.42	0	0	1	0	1	5.000000e-01	0.866025
...	...	...	...	...	...	...	...	...	...	...
7.731000e+04	1900.0	155.0	337.84	1	0	0	0	1	-2.449294e-16	1.000000
7.272500e+04	1810.0	165.0	325.34	1	0	0	0	1	-2.449294e-16	1.000000
7.453500e+04	2865.0	155.0	319.34	1	0	0	0	1	-2.449294e-16	1.000000
5.991500e+04	2350.0	125.0	257.84	1	0	0	0	1	-2.449294e-16	1.000000
6.337500e+04	1505.0	125.0	281.84	1	0	0	0	1	-2.449294e-16	1.000000

Figura 8: Algunos registros del conjunto de datos final

4.2. Modelamiento

Se probaron 6 modelos de regresión diferentes: regresión lineal múltiple, regresión de Poisson, árboles de decisión, bosques aleatorios, máquinas de vectores de soporte, y redes neuronales. Cada modelo presenta particularidades diferentes, lo cual genera que se deban hacer ajustes en el conjunto de datos dependiendo del modelo (por ejemplo, normalización de los datos cuando se trabaja con redes neuronales o máquinas de vectores de soporte, en este último se usará solo la variable inversión como variable predictora). Sin embargo los modelos que mejor rendimiento presentan son los bosques aleatorios, la regresión lineal múltiple y las máquinas de vectores de soporte, por tanto dichos modelos son los que se exponen en el presente documento.

Dado que el objetivo es darle más peso a las variables cuyo costo (CPC, CPM, CPC) es menor, se hace uso de una función peso para penalizar aquellos registros cuyo costo es muy alto. La variable peso se define como W^{-1}, con $W \neq 0$, dicha variable W es el costo correspondiente a cada registro en los datos, por ejemplo, en el caso de trabajar en la predicción de clicks, W será el CPC que corresponde a cada registro en particular. De esta forma, cuando el peso W es una cantidad grande (es decir un costo grande), la cantidad W^{-1} será un número pequeño, convirtiéndolo en un número con menos impacto (su coeficiente es menor que los que no tienen un costo tan alto).

Es importante recordar que se realizan modelos diferentes para predecir cada uno de las métricas, es decir, modelos diferentes para la predicción de los clics, las impresiones y las conversiones.

Inicialmente se realiza una división en conjunto de entrenamiento y prueba, con el 80 % de entrenamiento y 20 % de prueba, para luego desarrollar una validación cruzada de tres iteraciones.

Se presentan los resultados de algunos de los modelos para cada una de las métricas, así, se logran encontrar los mejores hiperparametros para el bosque aleatorio y máquina de vectores de soporte haciendo uso de un random grid (conjunto de funciones para iterar sobre todos los posibles hiperparametros elegidos y seleccionar los que dan mejores y mas precisos resultados), de la librería Scikit-learn. Para las regresiones se usó la librería Statsmodels. Además, se usan 3 métricas para medir error de los modelos, estos son el MAPE, RMSE y R2.

En los siguientes resultados se exponen solo los resultados de la regresión lineal, bosques aleatorios y máquinas de vectores de soporte, a nivel general fueron los que presentaron el mayor nivel de precisión.

Es importante aclarar que para las máquinas de vectores de soporte se toma como variable predictora únicamente la inversión, ya que luego del análisis de residuales se notó que la varianza en las predicciones era no lineal (dichos resultados se presentan más adelante), por tal motivo, con el fin de diversificar e intentar arreglar este error se prueba una combinación distinta de variables para dicho modelo.

En la siguiente tabla se muestran los resultados obtenidos para cada uno de los modelos desarrollados [9].

Modelo	Error	Hiperparametros	Clics	Hiperparametros	Impresiones	Hiperparametros	Conversiones
Regresión Lineal	R2	Constant: False	0.86	Constant: False	0.33	Constant: False	0.01
	RMSE		587.83		8754.92		588.61
	MAPE		44%		571%		129%
	ERR.PROMEDIO		360.85		5871.56		478.02
Bosque aleatorio	R2	bootstrap: True, max depth: 90, max features: sqrt, min samples leaf: 1, min samples split: 7, n estimators: 1900	0.89	bootstrap: True, max depth: 90, max features: sqrt, min samples leaf: 1, min samples split: 7, n estimators: 1900	0.72	bootstrap: True, max depth: 90, max features: sqrt, min samples leaf: 1, min samples split: 7, n estimators: 1900	0.4
	RMSE		513.5		5600.81		451.78
	MAPE		25%		39%		54%
	ERR.PROMEDIO		266.73		3089.18		278.73
Máquina de vector de soporte	R2	kernel: rbf, gamma: 1, degree: 3, C:100	0.86	kernel: rbf, gamma: 1, degree: 1, C:1500	0.62	kernel: rbf, gamma: 1, degree: 1, C:1500	0.21
	RMSE		505.05		6319.93		502.01
	MAPE		42%		55%		75%
	ERR.PROMEDIO		313.1		3251.47		337.63

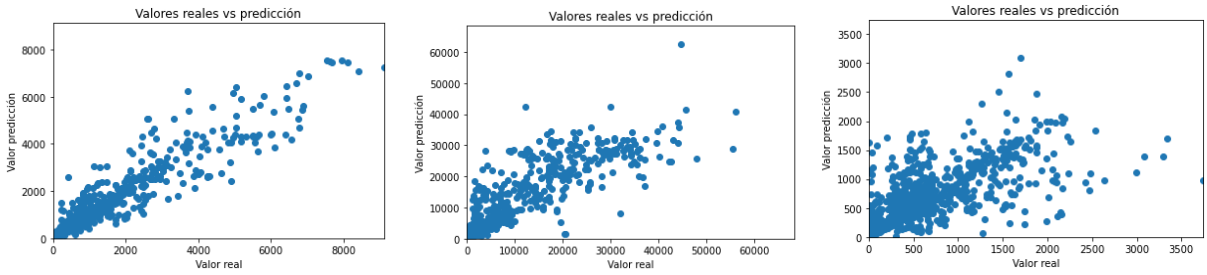
***En azul el modelo con menor error*

Figura 9: Resumen de los resultados de los modelos junto con los mejores hiperparametros

Para las tres variables respuesta se encuentra que el modelo que mejores resultados posee es el bosque aleatorio, se puede observar que en las impresiones y conversiones es con diferencia el que menor error presenta, dado que obtiene el menor RMSE, error promedio y MAPE, además el mayor R2. Por otro lado, en la variable clics se presentan resultados similares con la máquina de vectores de soporte, y aunque el R2 es mayor con la máquina de vectores de soporte, las demás métricas presentan mejores resultados.

Dado que con estos modelos se obtienen mejores resultados que con el modelo base (es decir, la regresión lineal), se puede afirmar que hay una mejora considerable cuando se usan los modelos de aprendizaje automático.

Las siguientes gráficas comparan las predicciones del mejor modelo (bosque aleatorio) con los valores reales:

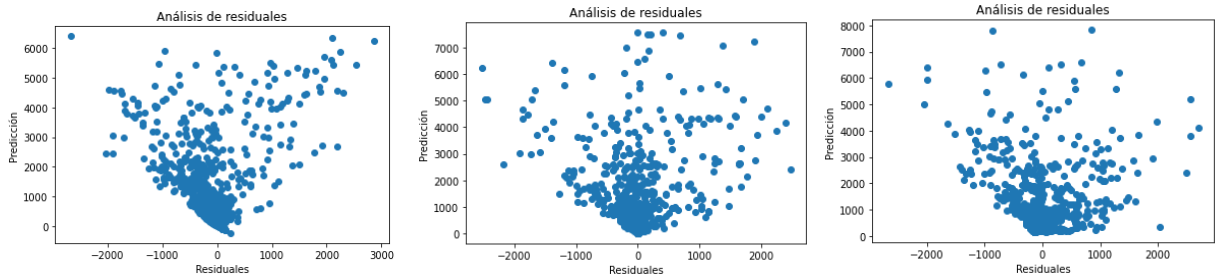


(a) Clics (b) Impresiones (c) Conversiones

Figura 10: Datos reales vs predicción

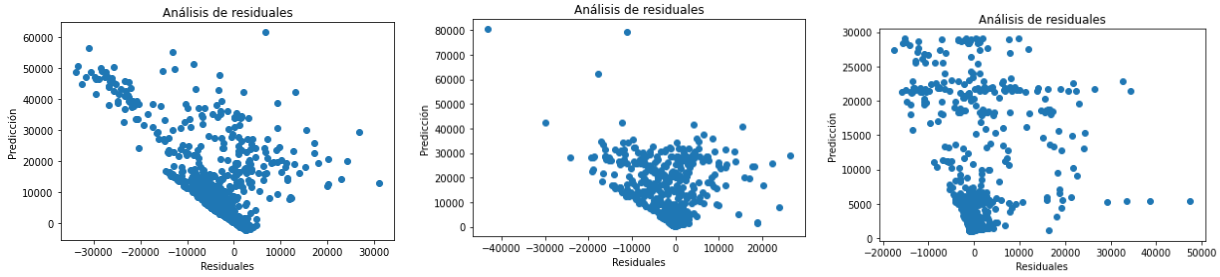
Para clicks e impresiones se nota un comportamiento aceptable en las predicciones y se evidencia cierto grado de dispersión, así mismo cierto grado de error. En cuanto a conversiones la dispersión de los datos da indicios de que la predicción obtuvo un grado de error alto.

Adicional a esto, se decide realizar un análisis de residuales con cada modelo, se crean las siguientes gráficas:



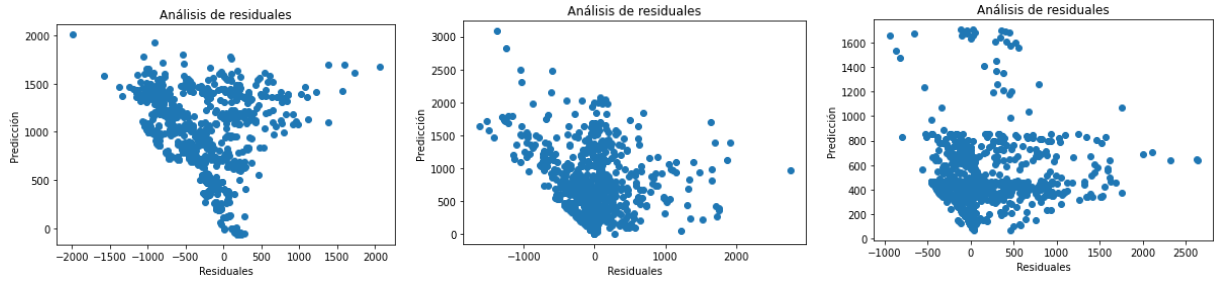
(a) Regresión lineal (b) Bosque aleatorio (c) Máquina de vectores de soporte

Figura 11: Residuales para clics



(a) Regresión lineal (b) Bosque aleatorio (c) Máquina de vectores de soporte

Figura 12: Residuales para impresiones



(a) Regresión lineal (b) Bosque aleatorio (c) Máquina de vectores de soporte

Figura 13: Residuales para conversiones

Sin embargo, en las gráficas se observa como la varianza de las predicciones no es constante, por el contrario, en todos los casos es no lineal, incluso con las máquinas de vectores de soporte, donde se

hicieron modificaciones (se toma solo inversión como variable predictora y se normalizan los datos). Con esto se puede concluir que realmente los modelos no se ajustan bien a los datos, y aunque en clics se obtienen unas métricas de error buenas, los resultados no serían 100 % confiables al momento de hacer predicciones con datos nuevos [7].

4.3. Conclusiones proyecto

- Aunque se logran buenos resultados y métricas con los bosques aleatorios para las impresiones y clics, los resultados del análisis de residuales indican que los resultados podrían estar sesgados o no ser confiables al aplicar los modelos con nuevos datos.
- Aunque las impresiones presentaban un coeficiente de correlación positivo alto, se presenta un error promedio y RSME muy alto.
- Las conversiones presentaban mucha dispersión lo cual dificultó la predicción de dicho KPI, no se evidencia un comportamiento lineal o que se pueda explicar de alguna forma con los datos que se tienen.
- La dispersión de los datos, así como algunos resultados no satisfactorios de las campañas (no se evidencia que con cierta inversión se logren adquirir mejores resultados en cuanto a impresiones y conversiones) indican que la correcta implementación y ejecución de las campañas de marketing digital influyen en sus resultados, por tanto, si no se hace este proceso de forma correcta, los resultados van a variar demasiado y así mismo no será posible encontrar patrones y predecir dichas métricas y KPI.
- El cambio constante o abrupto en la estrategia de implementación de las campañas influye en los resultados del modelo. Si se obtienen resultados muy diferentes cada vez que se cambia la estrategia y esto se hace de forma muy concurrente, los modelos no podrán aprender dicho comportamiento e imposibilitará la predicción de futuros resultados.

5. Recomendaciones

- Se recomienda a los equipos que implementan y planean las campañas que intenten mantener estrategias similares y poner especial atención a la inversión de las campañas. Si se cambia de estrategia constantemente los datos tendrán un comportamiento muy aleatorio, lo que dificultaría la creación de modelos.
- Dados los problemas con los residuales, se recomienda realizar nuevas transformaciones a los datos, uso de nuevas variables predictoras y uso de modelos no lineales.
- Se recomienda realizar los modelos con datos de un cliente diferente, donde la estrategia implementada en las campañas sea diferente y así poder evaluar los modelos en un contexto diferente.

6. Conclusiones pasantía

Aunque el campo del marketing digital podría parecer ajeno a las matemáticas, en el transcurso de la pasantía se logró ver como los conocimientos en matemáticas pueden ser aplicados a problemas de este campo, se lograr abrir la mente a una nueva forma de aplicar los conocimientos en contextos donde antes no se habría imaginado. Aunque en el proyecto principal de la pasantía no se pudieron predecir todos las métricas y KPI satisfactoriamente, se lograron implementar los conocimientos matemáticos de modelamiento tanto estadístico como de aprendizaje automático y se hallaron nuevas técnicas de optimización, así como de tratamiento de los datos para lograr una mejoría en las predicciones tanto en la parte de programación, como la parte teórica. Además, se logró fortalecer el pensamiento lógico y analítico para la toma de decisiones y conclusiones, no solo respecto a la parte técnica del análisis de datos o modelamiento, sino de un lado mas profesional y de desarrollo personal. Sin duda ha sido una experiencia enriquecedora que ha demostrado que hasta en las áreas del conocimiento con más diferencias, se puede aportar como matemático.

Referencias

[1] Bescond, P. Cyclical features encoding, it's about time! https://towardsdatascience.com/cyclical-features-encoding-its-about-time-ce23581845ca, 2021.

[2] de operaciones, G. Error porcentual absoluto medio (mape) en un pronóstico de demanda. https://www.gestiondeoperaciones.net/proyeccion-de-demanda/error-porcentual-absoluto-medio-mape-en-un-pronostico-de-demanda/, 2019.

[3] Deisenroth, M., Faisal, A., and Ong, C. *Mathematics for Machine Learning.* Cambridge University Press, 2020.

[4] Fernando, J. R-squared. https://www.investopedia.com/terms/r/r-squared.asp#:~:text=our%20editorial%20policies-,What%20Is%20R%2DSquared%3F,variables%20in%20a%20regression%20model., 2022.

[5] Hastie, T. *The Elements of Statistical Learning: Data Mining, Inference and Prediction.* Springer Series in Statistics, 2009.

[6] Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of Machine Learning.* Mit Press, 2012.

[7] qualtrics. Interpreting residual plots to improve your regression. https://www.qualtrics.com/support/stats-iq/analyses/regression-guides/interpreting-residual-plots-improve-regression/, 2022.

[8] rockcontent. Métricas de marketing digital. https://rockcontent.com/es/blog/metricas-de-marketing-digital/, 2019.

[9] Scikit-learn. Scikit-learn. https://scikit-learn.org/, 2022.

[10] Smola, J., and Scholkopf, B. A tutorial on support vector regression.

[11] Walpole, R. E. *Probabilidad Y Estadística Para Ingeniería Y Ciencias (9.a ed.).* Pearson eduación, 2012.