Tipo de documento: Tesis de maestría

Escuela de Negocios. Master in Management + Analytics

Pronósticos de adquisición de nuevos partners en una aplicación de delivery de comida

Autoría: Cilotta, Franco

Año: *2024*

Pronósticos de adquisición de nuevos partners en una aplicación de delivery de comida

---

Tesis de maestría

Master in Management+Analytics

Universidad Torcuato Di Tella

Fecha: Mayo de 2024

Alumno: Franco Cilotta

Tutora: Magdalena Cornejo

1. Índice

2. Resumen

La presente tesis se propone construir una herramienta para elaborar un pronóstico de adquisición de nuevos partners (socios comerciales) en el contexto de una aplicación de delivery de comida. El objetivo en concreto es predecir el número de nuevos partners que se agregarán a la plataforma cada día, en cada mercado y a través de cada canal posible de adquisición. Esto resulta crucial tanto para la planificación y asignación de recursos dentro de la organización, como para la disposición de un control de seguimiento y alarma en caso de posibles desvíos de lo que efectivamente ocurre, versus lo esperado.

La metodología aplicada incluye un análisis profundo de datos disponibles en distintas fuentes internas de la compañía como tablas de Salesforce, planillas de cálculo y demás, así como el uso de técnicas estadísticas y de machine learning, incluyendo ARIMA, SARIMA, ETS, Prophet, XGBoost y Light GBM. Cada modelo se ajusta y se evalúa en cada mercado y canal, buscando el que mejor desempeño tiene, basado en métricas como el Error Cuadrático Medio.

Los resultados revelan patrones y tendencias en la adquisición de partners, permitiendo recomendaciones detalladas sobre la cantidad de personal necesario para cada etapa del proceso de adquisición. Además, la investigación aporta a la literatura ya existente, aplicando modelos de pronóstico de demanda tradicional al novedoso contexto de las plataformas digitales.

Las principales conclusiones refuerzan la importancia de un enfoque adaptado a las especificidades de cada mercado y canal, en contraposición a uno general. Se detectan, asimismo, patrones no lineales en los datos de demanda y un fuerte componente estacional a nivel tanto semanal como mensual. Se destaca, también, el impacto de variables como campañas de marketing y la estructura de personal, en la eficacia de los modelos predictivos. Es por esto, que un modelo como XGBoost termina siendo el que mejor amalgama las distintas particularidades de los datos para producir un pronóstico efectivo.

Esta tesis no solo aporta al campo académico, sino también a la práctica empresarial, proponiendo un marco para la toma de decisiones basado en datos y ofreciendo herramientas de uso simple para la gerencia, asistiendo estas decisiones.

3. Abstract

This thesis aims to build a tool for forecasting the acquisition of new commercial partners in the context of a food delivery app. The specific goal is to predict the number of new partners that will be added to the platform each day, in each market, and through each potential acquisition channel. This is crucial for planning and resource allocation within the organization, as well as for having a monitoring and alarm control in case of possible deviations of what actually occurs versus what is expected.

The applied methodology includes an in-depth analysis of available data, fed from various internal sources of the company such as Salesforce tables, spreadsheets, among others, as well as the use of statistical and machine learning techniques including ARIMA, SARIMA, ETS, Prophet, XGBoost, and Light GBM. Each model is adjusted and evaluated in each market and channel, seeking the one that performs best, based on metrics like the Mean Squared Error.

The results reveal patterns and trends in partner acquisition, leading to detailed recommendations on the amount of personnel needed for each stage of the acquisition process. Additionally, the research contributes to the existing literature by applying traditional demand forecasting models to the new context of digital platforms.

The main conclusions reinforce the importance of a local approach, focused on each market and channel, as opposed to a general one. Non-linear patterns in demand data and a strong seasonal component at both weekly and monthly levels are also detected. The impact of variables such as marketing campaigns and staff structure on the effectiveness of predictive models is also highlighted. For these reasons, a model like XGBoost ends up being the best at combining the various particularities of the data to produce an effective forecast.

This thesis not only contributes to the academic field but also to business practice, proposing a framework for data-based decision-making and offering simple tools for management, assisting these decisions.

4. Introducción

4.1. Contexto

En el mundo de las plataformas digitales, es común tener dos partes a las cuales atender al mismo tiempo: consumidores finales y socios. Estos últimos son quienes proveen un bien o servicio a los consumidores finales, actuando, las plataformas, como intermediarios y facilitadores. Ambos lados pueden ser tratados como "clientes". Enfocándonos en el lado de los socios (de aquí en más, "partners"), resulta interesante estudiar el proceso por el cual éstos se suman a la plataforma y, en especial, su cantidad.

En las plataformas de delivery de comida, el modelo de negocio es uno basado en *choice*. Esto quiere decir que la compañía se centra en brindar al consumidor final una gran variedad de opciones a la hora de elegir. Debido a esto, resulta fundamental lograr la mayor cantidad de adquisición de partners posible.

Hasta el momento, siendo este un mercado relativamente nuevo, existe muy poca literatura (desarrollada en la sección 4.4) acerca de pronósticos de adquisición de partners, probablemente debido a que el modelo de las plataformas es algo más bien novedoso y aún no hay mucho trabajo hecho sobre él. Sin embargo, si consideramos a los nuevos partners como nuevos clientes, entonces podríamos establecer ciertas analogías entre un modelo de pronóstico de ventas / de clientes y un modelo como el que precisamos.

De este modo, podemos decir que la presente tesis se propone como objetivo ser una primera aproximación en el desarrollo de pronósticos para plataformas digitales, en específico, de adquisición de partners, trayendo las enseñanzas de otros campos, como lo son la predicción de demanda común, pero adaptándose o aplicándose más específicamente a esta realidad particular del modelo de negocio de las plataformas.

4.2. Problema

Debido a que la cantidad de nuevos partners es una meta que se persigue con especial hincapié dentro del sector de Supply, resultaría de utilidad conocer cuál es la cantidad esperada de adquisiciones que se tendrá cada día a lo largo del mes y a lo largo del año. Hoy esto no se conoce, lo cual supone un problema por dos razones.

Una es no tener una cifra esperada contra la cual comparar luego la data real, viendo qué tan cerca/lejos se va evolucionando con respecto a lo esperado y siendo capaces de detectar a tiempo desfases para poder accionar, marcando el ritmo a los mercados

de cómo tienen que adquirir y dónde hay un déficit, así como brindando soporte en caso de que lo necesiten, acelerando procesos.

La otra es la incapacidad actual para planificar y asignar de la mejor manera los recursos de la organización. Los nuevos partners, en su proceso de iniciación en la app, pasan por distintas etapas en las cuales son acompañados por agentes de la empresa que los asisten en la puesta a punto para funcionar y comenzar a vender a través de la app. Es importante ser capaces de determinar, entonces, la cantidad de personal de cada tipo que se necesitará, en función del volumen que pueden manejar y de la cantidad de adquisiciones esperadas (pronóstico). De esta manera, se espera, a través de la presente tesis, arribar a una recomendación de personal en lo que respecta a:

- Agentes de *Quality Check*
- Agentes de *Menu Processing*
- Agentes de *Onboarding*

Todos ellos se encargan de tres etapas distintas del proceso y, en función de su capacidad y de la cantidad proyectada de adquisiciones, determinaremos el nivel óptimo de cada tipo de agente por mercado.

Es por esto, que el presente trabajo de tesis se propone elaborar un modelo de pronóstico de adquisición de partners, que nos permita estimar cuántos nuevos partners se sumarán por día a la aplicación. Particularmente, nos interesamos en predecir adquisiciones de *Food* (restaurantes y cafés, dejando fuera de la predicción a las *Local Stores*, otro tipo de tiendas).

4.3. Objetivo

En función del problema planteado, podemos establecer nuestro objetivo. El mismo es realizar un pronóstico de nuevos partners de *Food* por día, por país y por canal de adquisición (digital, físico o *Must have*). El *output* del modelo de pronóstico debería consistir en:

- Una planilla con estas cantidades especificadas por día, país y canal.
- Una planilla con las estimaciones de recursos (FTEs) necesarios de *Quality Check, Menu Processing* y *Onboarding*, calculada en función del pronóstico anterior.

Analizamos el desempeño del modelo en términos de pronóstico según medidas tales como el Error Cuadrático Medio (ECM), entre otras. Una vez elegida esta métrica, la usamos para comparar el rendimiento de los distintos modelos testeados con el

objetivo de determinar el que mejor explica el comportamiento de las adquisiciones a lo largo del tiempo y en función de los *features*. Consideramos un modelo exitoso aquel que muestra, entonces, un buen rendimiento, reflejándose en este tipo de indicadores.

4.4. Revisión de la literatura

Al realizar una investigación sobre el tema, se descubrió que no hay mucho escrito acerca del problema de pronóstico de adquisición de partners, probablemente debido a que el modelo de las plataformas es algo más bien novedoso y aún no hay vasta literatura que trabaja sobre él. Sin embargo, si consideramos a los nuevos partners como nuevos clientes, entonces podemos establecer ciertas analogías entre un modelo de pronóstico de ventas / de clientes y un modelo como el que precisamos. De este modo, podemos extraer algunos conocimientos o anotaciones útiles de distintos artículos o trabajos sobre el tema.

En general, la literatura encuentra una relación entre el éxito de las compañías de delivery de comida y la variedad de opciones ofrecidas. La tendencia del mercado es una de crecimiento, explicada no solo por la mayor disponibilidad de tecnología, sino también por ciertos cambios de hábitos en los consumidores. Los mismos, al acceder a tecnología que les permite visualizar distintas opciones, se han ido volviendo más exigentes y viraron hacia una mayor cultura de la comida (Surendhranatha Reddy y Aradhya, 2020).

Por otro lado, también hay más oferta. Para atender a esta mayor demanda, más y más restaurantes han ido abriendo. Las expectativas hacia futuro son las de un crecimiento continuado, siguiendo esta tendencia al alza (Surendhranatha Reddy y Aradhya, 2020). Esto implicaría un aún mayor crecimiento de restaurantes disponibles en las aplicaciones de delivery de comida.

Reddy y Aradhya destacan la estrecha relación entre la capacidad de ofrecer una variedad de opciones y las probabilidades de éxito del negocio. El usuario es más propenso a consumir si tiene más opciones entre las cuales elegir.

Otros autores, también coinciden con esta premisa. Bivona (2022), destaca la importancia del uso y seguimiento de indicadores clave como forma de velar por el crecimiento (y hasta supervivencia) de las plataformas digitales. Bivona (2022) prueba, a lo largo de su trabajo, cómo aquellas compañías que se apoyan en el seguimiento de estos indicadores terminan siendo más exitosas. Los indicadores incluyen: ratio de productos en el menú, usabilidad de la plataforma, cantidad de restaurantes por ciudad, atraso en delivery, órdenes por ciudad, inversión en marketing, entre otros. A

este enfoque, lo llama "*dynamic performance management framework*". Encontramos, entonces, cierta coincidencia y relación con el resto de la literatura explorada, pues se menciona la cantidad de restaurantes por ciudad, como un indicador importante del éxito de las plataformas.

En esa misma línea, encontramos trabajos que analizan los efectos de la experiencia del consumidor en el éxito de las aplicaciones de delivery de comida. La decisión del consumidor a la hora de comprar está muy basada en su experiencia al navegar por la aplicación o sitio web (Isa et al. 2021). Se encuentra una correlación entre la experiencia del consumidor en la web (destacando la estética) y la probabilidad de que se termine generando una orden de compra al restaurante. Dependiendo de qué tan placentera es esta experiencia, los consumidores terminan comprando o no. Podríamos agregar que, si bien no es el único factor, parte de lo que hace a una buena experiencia es la disponibilidad de distintas opciones entre las cuales elegir, para tener un abanico de alternativas y distintos menús para navegar. De nuevo: más opciones impacta en una mayor probabilidad de éxito, sobre todo en un mercado con consumidores exigentes y que buscan variedad (Surendhranatha Reddy y Aradhya, 2020).

Si nos corremos un poco del foco de las aplicaciones de delivery de comida y observamos un panorama más amplio, como lo es el de las plataformas digitales (sin restricción del producto o servicio que comercian), podemos entender un poco mejor por qué tanta necesidad de "cantidad" en relación a las opciones disponibles entre las cuales elegir.

Las plataformas digitales son casos de negocio en los cuales se tienen dos lados: generalmente, un lado es constituido por los usuarios finales, quienes acceden a esa plataforma para consumir cierto bien o servicio; y el otro lado son los proveedores de tal bien o servicio. De este modo, las plataformas digitales actúan como intermediarios, uniendo las necesidades con quienes tienen los recursos necesarios para satisfacerlas, de la manera más eficientemente posible. Sabido esto, la literatura considera que las plataformas digitales exitosas son aquellas que son capaces de atraer rápidamente a un gran número de usuarios, de un lado, y gran número de socios, del otro (Bivona, 2022). Para que una plataforma digital crezca, resulta clave la adquisición de una gran cantidad de socios.

En esta línea, Bivona (2022) destaca el fenómeno de los efectos indirectos en las redes, los cuales consisten en que la mayor cantidad de actores en un lado del negocio, impacte generando una mayor cantidad de actores, del otro lado. Aplicado a las plataformas de delivery de comida, sería: más restaurantes disponibles atraen a más usuarios a la aplicación, dado que tienen más opciones, al mismo tiempo que

más usuarios que utilicen la app, atraen a más restaurantes, ávidos de vender en un mercado en crecimiento. Por lo tanto, la decisión estratégica de la plataforma pasa por buscar llegar a una cantidad crítica de actores en un lado, para garantizar suficientes del otro.

Sin embargo, estos efectos de red, por más beneficiosos que puedan llegar a resultar, no se dan a nivel global. Que una aplicación cuente con muchos socios en un país determinado no garantiza una buena cantidad de clientes/usuarios en otro país. Por el contrario, los efectos de red se dan a nivel local: país e incluso, a veces, jurisdicciones más pequeñas como regiones o ciudades (Bivona, 2022). Esto tiene sentido: llevándolo a nuestro caso particular, que haya más restaurantes, por ejemplo, en Argentina, no influye en la cantidad de usuarios que buscarán entrar a la aplicación para comprar comida en Ecuador. De esto extraemos que el problema debe ser abordado a nivel local y no global, analizando el caso de cada país en particular.

Si la cantidad de restaurantes es tan importante para hacer el seguimiento del éxito de un negocio, podemos confirmar, entonces, la importancia de nuestro trabajo en tanto busca predecir esta cantidad. A la vez, la naturaleza local del negocio mencionada hace necesario que la unidad de análisis no sea simplemente una unidad de tiempo (como lo puede ser una fecha de adquisición) sino que (incluso en los casos en los que se intente predecir esta variable a nivel regional) el pronóstico se haga a nivel local, es decir, la unidad de análisis incluya el país y no solo la fecha. Por agregación, luego, se pueden agrupar los países en caso de solo necesitar el dato a nivel región.

De este modo, tenemos la variable tiempo y la variable país como dos variables independientes destacadas para utilizar en el trabajo, pero, ¿qué otras variables podemos utilizar para tratar de predecir la cantidad de adquisiciones? Ciertos artículos consideran que para predecir ventas, es clave tener en cuenta dos tipos de variables.

Uno de ellos es el de las variables tangibles, por ejemplo: histórico de ventas/adquisiciones, así como las ventas en proceso. El otro lo constituyen las variables intangibles, como lo pueden ser las campañas de marketing (Rock Content - ES, 2019). Esto coincide con la visión de Bivona de considerar el indicador de inversión de marketing como uno de importancia para el seguimiento del crecimiento del negocio (y su eventual éxito). Podemos inferir, entonces, que podría llegar a haber cierta relación entre la inversión en marketing y la cantidad de clientes que tiene, ya sea de modo causal (más marketing atrae a más clientes) o quizás también recursivo (más clientes implica que el mercado puede ser próspero, lo cual atrae más actores que buscarán captar dichos clientes a través del marketing). Como el objeto de la presente tesis no consiste en determinar la causa de las adquisiciones, sino simplemente realizar un pronóstico acerca de las mismas, mientras la inversión de

marketing tenga cierta capacidad predictiva de las adquisiciones o guarde cierta correlación con las mismas, entonces sirve incluir datos en relación a ella.

Volviendo, ahora, al primer grupo de variables, las tangibles, resulta interesante destacar uno de los ejemplos dados por la literatura para este tipo: el histórico de ventas/adquisiciones. Esto supone que las adquisiciones del futuro pueden ser explicadas no solo por variables externas, sino también por el mismo nivel de ventas (es decir, los valores de la misma variable) en el pasado. De este modo, y realizando la investigación pertinente, llegamos a conocer que existen dos tipos de modelos que sirven para la predicción de ventas.

El primero es el grupo de los modelos multivariados de aprendizaje supervisado. Estos modelos utilizan múltiples variables en sus predicciones (Weingertner, 2023). Entre estas variables, podemos encontrar las mencionadas anteriormente como intangibles, pero también algunas tangibles. Ejemplos de estas variables podrían ser la inversión en marketing, la cantidad de vendedores disponibles, entre otras. Algunos modelos que se podrían utilizar en este contexto son redes neuronales con perceptrones multicapa, Light GBM, árboles de decisión, y hasta XGBoost (Kerem Gülen, 2022, Optisol, 2023).

Ahondando en el estudio de aplicación de modelos como XGBoost para realizar pronósticos, es que encontramos un trabajo en el que se implementa esta estrategia para la previsión de ventas en una serie de tiempo de comercio electrónico (Pinzón Villanueva, 2023). En el mismo, se comienza ponderando la importancia del uso de pronósticos (o *forecasts*) para la planificación de recursos e inventario. El saber cómo va a ser la demanda permite anticipar la cantidad de recursos que uno va a necesitar para poder atenderla de forma correcta. En ese sentido, los pronósticos sirven a la hora de tomar decisiones de manera anticipada y previsora.

Tras una limpieza de datos, se procede a probar una primera instancia del modelo XGBoost, utilizando la métrica de R cuadrado para evaluar la performance del mismo. La siguiente etapa es la de la optimización de hiperparámetros. Los hiperparámetros del modelo que se deciden optimizar son los siguientes:

- *gamma*
- *learning rate*
- *max_depth*
- *min_child_weight*
- *n_estimators*
- *subsample*

(Pinzón Villanueva, 2023).

El trabajo de Pinzón Villanueva (2023) aborda dos estrategias de optimización, con el objeto de compararlas y decidir cuál mejor se adapta a su caso de pronóstico de demanda:

1. Optimización mediante grilla, la cual consiste en tener una lista de posibles valores para cada hiperparámetro y entrenar un modelo XGBoost para cada combinación posible de los mismos, quedándose con la combinación que mejor R cuadrado arroje en el conjunto de validación.
2. Algoritmos genéticos, los cuales incorporados distintos métodos de optimización que, mediante estrategias (selección de padres, cruces, mutación, etc), busca mínimos locales en funciones de pérdida, tratando de eficientizar el uso de recursos y sin recorrer la totalidad de las opciones posibles de combinaciones de hiperparámetros.

Finalmente, tras implementar ambas estrategias y comparar resultados, se observa cómo la grilla de hiperparámetros obtiene mejor R cuadrado que los algoritmos genéticos. Como contrapartida, los algoritmos genéticos demoran menos tiempo en ejecutarse, pero debido a la naturaleza del problema en cuestión, esto no resulta una limitante para inclinarse por la optimización por grilla, de modo que esta última es la estrategia elegida (Pinzón Villanueva, 2023).

En relación a la decisión de grilla vs. algoritmos genéticos, Mejía Tovar (2023) realiza un trabajo comparando ambos dentro del contexto de la elaboración de un modelo para predicción de demanda en una plataforma de *fast delivery* en Colombia. Se prueban distintos modelos, entre los que se destacan las redes neuronales. Posteriormente, se aborda el problema de la optimización de hiperparámetros. En torno a esto, se destaca la capacidad de los algoritmos genéticos de seleccionar respuestas de manera eficiente, sin necesidad de recorrer todas las posibilidades de manera exhaustiva (Mejía Tovar, 2023). De aquí podemos extraer, entonces, que este tipo de algoritmos son más útiles en casos de cantidad alta de opciones y data muy vasta y abundante. En palabras de Mejía Trovar, “Los algoritmos genéticos demuestran su eficacia cuando se enfrentan a espacios de solución complejos y de alta dimensión. En estos entornos, los algoritmos genéticos pueden explorar más posibilidades de manera más eficiente en comparación con un Grid Search, que puede volverse poco práctico o computacionalmente costoso en funciones grandes y complejas”.

En el trabajo, se proponen dos métricas como forma de medir el desempeño de los modelos:

- MSE (*Mean Squared Error* o Error Cuadrático Medio), definido como la medición de qué tan alejadas están las predicciones del modelo de la realidad. A mayor MSE, más dispersión de los puntos observados con respecto a la predicción. Se busca minimizarlo.
- R cuadrado, definido como una métrica estadística que mide qué parte de la variabilidad de los datos es explicada por el modelo. En este caso, a mayor R cuadrado ajustado, mejor es el desempeño del modelo, por lo que se busca maximizarlo.

(Mejía Tovar, 2023).

Utilizando estas métricas es que se comparan los resultados obtenidos de las distintas estrategias. Se lleva a cabo un experimento, que pone a competir a la optimización por grilla vs. los algoritmos genéticos, midiendo la calidad del ajuste, la cantidad de posibles soluciones exploradas por cada algoritmo y cuánto convergen los resultados en un tiempo específico. Al observar los resultados, se detecta que, aún habiendo recorrido menos opciones, el algoritmo genético arriba a un mejor resultado que el de búsqueda por grilla y, además, lo hace en menos tiempo, lo cual es altamente ponderado en el contexto de toma de decisiones rápidas, propias del *fast delivery*. (Mejía Tovar, 2023).

Finalmente, el trabajo concluye que la estrategia de optimización por algoritmos genéticos es la más adecuada y que el modelo de redes neuronales profundas es el que mejor se adapta a su caso de estudio, debido a que es capaz de captar la complejidad de ciertos patrones no lineales y prever picos de demanda (Mejía Tovar, 2023).

El otro gran grupo de modelos del que habla Weingertner (2023) es el de las series de tiempo univariadas. Estos modelos usan solamente el dato de la cantidad histórica de ventas (adquisiciones) por día, sin ninguna otra variable. Viene implícita la idea de que ventas pasadas ayudan a explicar ventas futuras, es decir, estamos ante una variable que se auto explica (Weingertner, 2023). Entre estos modelos, podemos encontrar Prophet, un modelo desarrollado por Facebook. Prophet incluye la variable del día de semana y días festivos para tener en cuenta a la hora del pronóstico. A estos últimos modelos se los llama autorregresivos.

En el contexto de la previsión de demanda en plataformas de delivery y consumo masivo, encontramos otro estudio que aborda la eficacia de diferentes modelos predictivos (ya no solo multivariados, sino también univariados, autorregresivos). Jaimes Campos y López Zúñiga (2021) analizan la aplicación de ARIMA, Prophet, Redes Neuronales Artificiales, Random Forest y Máquinas de vector soporte para

predecir la demanda semanal de productos de consumo masivo en Colombia. Sus hallazgos indican que la selección del modelo adecuado puede mejorar significativamente la precisión del pronóstico y optimizar los niveles de inventario, lo cual es crucial para la reducción de costos y la satisfacción del cliente. (Jaimes Campos y López Zúñiga, 2021).

El estudio hecho, revela que los modelos ARIMA y Random Forest son los algoritmos de mejor desempeño para el caso de Jaimes Campos y López Zúñiga (2021), especialmente para el pronóstico de productos individuales, proporcionando una mayor precisión y una gran reducción de costos de inventario. Por ejemplo, para el modelo de pronóstico de 12 productos, los modelos ARIMA y Random Forest reducen el costo de inventario en un 44.2% comparado con el modelo actual, ahorrando aproximadamente 67,516 USD anuales. De este modo, encontramos ejemplos de cómo no solo los modelos multivariados (Random Forest, XGBoost, etc) pueden realizar pronósticos útiles, sino también modelos como ARIMA o Prophet pueden arrojar buenos resultados. Se trata, entonces, de encontrar aquel que mejor se adapte a la data que se tiene disponible y la realidad del negocio en estudio.

Es en esta misma sintonía que el trabajo destaca la importancia de adaptar los enfoques según las particularidades del mercado y el contexto operacional y no usar un único modelo para todo negocio o mercado (Jaimes Campos y López Zúñiga, 2021).

En su trabajo, Jia et al. (2022) se propone elaborar un modelo de decisión que pueda ser utilizado por los restaurantes para decidir si asociarse con un servicio de delivery online o no. Como parte de este modelo, se confecciona un pronóstico de ventas esperadas, que luego serán utilizadas como *input* y, en función de su cantidad (y otras variables), se determinará si conviene o no asociarse. Lo interesante de este modelo es la metodología utilizada: se trata de un modelo ARMA, muy común en series de tiempo. ARMA (por sus siglas en inglés, Autoregressive Moving Average) es un modelo autorregresivo que utiliza valores rezagados de la variable a predecir como *feature* para alimentar al modelo de pronóstico. En particular, este trabajo busca predecir cantidad de órdenes recibidas en un período t, y lo hace mediante tres variables independientes:

a. Día de la semana. Para representar esta variable, se crean 6 variables dummies (de lunes a sábado) que toman el valor 1 si el día de la semana es el de esa variable en cuestión y 0 en caso contrario, siendo la combinación de los 6 ceros el escenario domingo.
b. Casos de infección por COVID-19 diarios (el trabajo se realizó en plena pandemia). Estos casos se usan como *input* con un desfase de un

período. Es decir, los casos de infección en el momento t se usan para predecir demanda en el momento t+1.

c. El tercer tipo de variable es la misma a predecir, pero desfasada. Se considera que el nivel de órdenes en el momento t es influenciado por el nivel de órdenes en momentos anteriores a t (J2 días).

(Jia et al., 2022)

Lo curioso de este trabajo es cómo combina las distintas técnicas comentadas: uso de una variable autorregresiva como *feature*, adición de una variable externa como la cantidad de casos COVID-19, y hasta el día de la semana. La inclusión del día de la semana resulta un aspecto a tener en cuenta. ¿Por qué se incluye? Debido a la existencia de la estacionalidad, que tiene que ver con la repetición de ciertos ciclos de comportamiento de una variable a lo largo del tiempo. De este modo, si una variable tiene estacionalidad semanal, significa que el día de la semana influye en su valor. Es por esto, que la estacionalidad es, para algunos, un elemento importante a ser tenido en cuenta a la hora de realizar pronósticos (Asana, 2023).

Otro trabajo estudiado en la presente tesis es el de Capote Pérez (2022), en el que se realiza una previsión de ventas de frutos secos. Para ello, utiliza modelos como ARIMA, SARIMA, Prophet y Random Forest. Gracias a la exploración y selección del modelo adecuado, se logra mejorar la precisión del pronóstico y optimizar los niveles de inventario, lo cual es crucial para la reducción de costos.

En el estudio se comparan los resultados de desempeño de los distintos modelos, destacando que SARIMA y Autoregresión ofrecen los mejores resultados. La particularidad del estudio consiste en que, en él, se realiza un análisis detallado de diferentes ventanas de predicción. Al hacer esto, se observa cómo los modelos de series temporales como SARIMA y Autoregresión tienen mejor desempeño en ventanas más cortas de tiempo, mientras que la media móvil es más adecuada para predicciones a largo plazo. Por lo tanto, una conclusión a extraer de ello es la importancia de considerar el horizonte temporal en la selección de modelos predictivos (Capote Pérez, 2022).

Como detalle al margen, se nota el impacto de la pandemia de COVID-19 en los datos de ventas, el cual es abordado mediante la eliminación de datos posteriores a diciembre de 2019 para evitar sesgos. Además, destacamos el enriquecimiento de modelos con atributos adicionales, como la media de los últimos meses de ventas y características específicas del año y mes de la venta. Esta inclusión de variables adicionales se ve reflejada en una mejora de precisión en las predicciones (Capote Pérez, 2022).

En conclusión, la revisión de la literatura sugiere que la diversidad de opciones en plataformas digitales es clave para el éxito empresarial, resaltando la importancia de adaptar estrategias de adquisición basadas en indicadores clave. Los modelos predictivos pueden ser tanto univariados (autorregresivos) como multivariados y se comprueba la eficacia de la optimización de hiperparámetros, ya sea mediante búsqueda en grilla o algoritmos genéticos. Se subraya la necesidad de un enfoque local (no global). Estos modelos integran variables tangibles e intangibles, permitiendo una comprensión más profunda del mercado, sus tendencias y estacionalidades.

5. Datos

5.1. Introducción al negocio y descripción de los datos existentes

El negocio en cuestión funciona como una plataforma digital intermediaria entre los consumidores finales (clientes que entran a la aplicación buscando comprar a través de ella) y los restaurantes y tiendas que buscan un marketplace donde exhibir sus productos. El negocio se subdivide en dos verticales: *Food* (integrado por restaurantes y cafés) y *Local Stores* (tiendas locales como lo pueden ser supermercados, farmacias, almacenes, etc). El foco del presente trabajo es la vertical de *Food*, dado que se dispone de más y mejor calidad de data y conocimiento sobre esta vertical (en comparación con *Local Stores*) y debido a que el mayor foco del negocio está actualmente puesto sobre *Food*. Asimismo, dentro de una misma vertical, encontramos distintas subcategorías. Por ejemplo, dentro de *Food* están las distintas "cocinas": empanadas, pizzas, hamburguesas, helados, etc. El negocio de esta plataforma opera en 15 países distintos de Latinoamérica: Argentina, Bolivia, Chile, Costa Rica, Ecuador, El Salvador, Guatemala, Honduras, Nicaragua, Panamá, Paraguay, Perú, República Dominicana, Uruguay y Venezuela.

El sector de *Acquisitions* es responsable por sumar nuevos negocios a la app, velando por la cantidad y calidad de los mismos. Estos negocios que se suman son los llamados "partners", ya que son socios de la plataforma. Se busca, desde una administración regional, pero contando con vendedores en cada mercado local (cada país) sumar la mayor cantidad de partners por día.

El proceso por el cual un candidato se termina convirtiendo en partner y se suma a vender a través de la aplicación tiene distintas aristas y varía dependiendo del canal de adquisición. Existe el canal físico o presencial, el llamado *Field,* en el cual los vendedores van presencialmente al negocio para buscar sumarlos a la plataforma. Por otro lado, está el canal digital (*SSU*), el cual implica que el candidato completa su registración y varias etapas del proceso de adquisición de forma autónoma y por la

página web, sin necesidad de interacción humana, aunque también hay vendedores de *SSU* para asistir en cualquier traba del proceso.

El proceso por el cual dichos candidatos (o *Leads*) se pueden sumar a la plataforma se llama “oportunidad”. La plataforma registra cada oportunidad, con sus fechas de inicio y fin. Una fecha de fin con estado exitoso implica que el candidato se ha convertido en partner y ahora puede comenzar a vender a través de la app. Esto es lo que se considera una “adquisición”.

Al mismo tiempo, se registran datos tales como los partners activos en la aplicación, sus datos geográficos, la inversión en marketing hecha en cada canal digital posible (distintas redes sociales), la cantidad de vendedores disponibles en cada mercado y canal durante cada mes, entre otros datos. La mayoría de esta data está guardada en tablas de Salesforce, aunque otra parte de ella viene de planillas de *Google Sheet* mantenidas por los analistas de cada equipo pertinente. Es habitual utilizar BigQuery para realizar consultas a todas estas tablas y obtener información valiosa acerca de las adquisiciones.

En relación a los datos en sí, podemos decir que son bien variados y cuantiosos. Se cuenta con información histórica de cada partner adquirido desde 2023 con una frecuencia diaria. Para cada partner, se cuenta con la fecha exacta de adquisición, el nombre del vendedor que logró concretar la operación, la vertical del partner (*Food / Local Stores*) así como su categoría de negocio (hamburguesas, pizzas, empanadas, etc). También se dispone de datos de procedencia del partner en tanto país, ciudad y área geográfica (regiones/barrios). Los datos se obtienen de distintas tablas de Salesforce, las cuales son combinadas de tal modo de obtener, en una sola base de datos, toda la información necesaria.

En forma sintética y simplificada, explicamos la procedencia de cada una de las columnas de nuestra base de datos, obtenido a través de BigQuery:

- Se parte de la tabla `fact_partners_monthly`, la cual es un registro de todos los partners de la aplicación, mes a mes.
- Acquisition_Date: es el campo closedate de la tabla `pro.cl_salesforce.opportunity`, que registra todas las oportunidades de candidatos a nuevos partners. Que una oportunidad se cierre exitosamente implica que el partner finalmente se sumó a la app, es decir, es una adquisición. Esta tabla se une por ID de Cuenta y mes de closedate a la partners_monthly.
- Country_Name: es el campo Country_Name de la tabla `pro.il_core.dim_country`, la cual contiene los nombres oficiales de los países en los cuales tiene

operaciones la compañía de delivery. Esta tabla se une a partners_monthly por country_id.

- AccountSource: obtenido de la tabla `pro.cl_salesforce.account`, modificado con un condicional CASE WHEN y finalmente unido a partners_monthly via salesforce_id y mes de closedate de la Oportunidad.
- Seller_Head_Count: el conteo es manual y registrado mes a mes por el personal del equipo de *Planning* en un *Google Sheet*. Se registra a nivel mes-Country_Name-AccountSource-Vertical (*Food*/*Local Stores*/Híbridos) la cantidad de vendedores. Además, se incluye el % de tiempo trabajado del mes (si de las 4 semanas, en una de ellas el vendedor estuvo de vacaciones, entonces es 75%). Para nuestra tesis centrada en *Food*, entonces, sumamos el tiempo trabajado de los vendedores de *Food* e Híbridos, considerando a los últimos no como un vendedor entero sino como la proporción de adquisiciones de *Food* que tuvo en el mes (en caso de no tener este dato disponible, se toma por default el número de 0.95, es decir, un 95%, ya que desde la gerencia se estima que la proporción del tiempo de trabajo que su personal dedica a *Local Stores* es del 5%). Como resultado, obtenemos un número con decimales, que unimos a nuestra *query* principal por mes-Country_Name-AccountSource.
- is_holiday: se arma manualmente un calendario de feriados para cada Country_Name de la región de Latinoamérica. El mismo, cuenta con una columna de fecha, otra de Country_Name y una tercera que es un booleano: is_holiday. Esta puede tomar 1 o 0 en caso afirmativo/negativo y se une a la *query* principal por fecha y Country_Name.
- is_holiday_uruguay: ídem is_holiday, pero solo centrado en Uruguay.
- Acquisitions: es un conteo de partner_ids de la tabla `pro.il_core.dim_partner`, unida por partner_id a partners_monthly.
- Marketing_Cost_This_Month,Marketing_Cost_Last_Month,Marketing_Cost_Last_30_days: datos obtenidos de un *Google Sheet* en el que se registran, para cada día, Country_Name y red social, el gasto en marketing. Esto luego se agrupa a nivel día y Country_Name y se une por estas dos variables a nuestra *query* principal, realizando el cálculo y sumando los gastos correspondientes para cada campo (mes corriente, mes pasado o últimos 30 días).

5.2. Datos hacia futuro

Dos tipos de *features*, los relacionados con la cantidad de vendedores y con marketing, cuentan con una casuística: conocemos sus datos históricos, pero no tenemos certeza de cuáles serán sus valores futuros, para poder incluirlos como variables predictoras a la hora de realizar el pronóstico. Debido a esto, se siguen los siguientes criterios para cada uno:

- Seller_Head_Count: hay 3 posibilidades:
    a. Tenemos toda la data sobre su cantidad: simplemente se usa como está.
    b. Tenemos la data del conteo de vendedores del mes en cuestión, pero no tenemos el tiempo trabajado: se cuenta cada vendedor considerando que ese mes trabaja al 100%.
    c. No tenemos el conteo de vendedores para ese mes:
        - Por un lado, se toma la data de *Best Estimates* (BE), que contiene la cantidad de vendedores que se espera tener cada mes, según el presupuesto. Este dato se encuentra en un Google Sheet que pertenece al equipo de *Planning*. Conectamos dicho Google Sheet a BigQuery para obtener el dato de forma automática.
        - Por otro lado, se recolecta la data del conteo de vendedores del mes anterior al mes de análisis, discriminando por:
            - Seller_Head_Count_exp
            - Seller_Head_Count_new a 1 mes de pasar a experimentado
            - Seller_Head_Count_new a más de 1 mes de pasar a experimentado
        - Se compara el BE vs el Real (total de vendedores del mes anterior: -1m) y se sigue la siguiente regla decisoria para determinar el número de vendedores en el mes en cuestión:
            - Vendedores experimentados:
                - Si BE >= Real (-1m), entonces los experimentados son igual a los Experimentados Reales (-1m), más los Nuevos a 1 mes de ser experimentados (-1m).
                - Si BE < Real (-1m), entonces los experimentados son igual a los Experimentados Reales (-1m), más los Nuevos a 1 mes de ser experimentados (-1m), menos la diferencia entre Real (-1m) y BE.
            - Vendedores nuevos:
                - Si BE <= Real (-1m), entonces los nuevos son igual a los Nuevos Reales (-1m) que no están a 1 mes de ser experimentados.
                - Si BE > Real (-1m), entonces los nuevos son igual a los Nuevos Reales (-1m) que no están a 1 mes de ser experimentados, más la diferencia entre BE y Real (-1m).
            - La lógica de esto es: si no hay diferencias entre lo planificado y lo real del mes anterior, todo sigue igual, y solo paso a experimentados a los vendedores que estaban a 1 mes de serlo. En caso de haber sobrante de

vendedores, se desvincula a los experimentados y en caso de haber faltante, se contrata, por lo que suben los nuevos.

- *Features* relacionados a costos de marketing: para estos *features*, siempre conocemos el dato del gasto en marketing hasta el día anterior a la fecha en que se realiza el análisis. Para proyectar hacia futuro, lo que se hace, entonces, es:

    1. Se toma el "*Budget*" (presupuesto) asignado para gastos de marketing en el mes que se quiere pronosticar.
    2. Se calcula la estacionalidad de los gastos de marketing a lo largo de un mes, como el promedio de las estacionalidades de los últimos dos meses. Para ello:
        a. Se obtiene la data del gasto en marketing diario en los últimos dos meses.
        b. Se divide el gasto de cada día por el total del mes para obtener un porcentaje.
        c. Se promedian los porcentajes de cada día de los últimos dos meses
    3. Aplicamos esta estacionalidad obtenida al total de los gastos de marketing presupuestados para el mes que se quiere predecir. El resultado es el *feature* que incluiremos en nuestro modelo.

5.3. Sobre la base de datos final

La base de datos cuenta con 54.218 adquisiciones de partners nuevos, las cuales van desde enero de 2023 hasta el 14/12/2023. Tomando esta base como punto de partida, se pueden obtener reportes con distintos volúmenes, dependiendo de su agrupación.

La fuente de los datos es una tabla interna del equipo de Acquisitions de la empresa, armada en base a una *query* de BigQuery. La misma, consulta distintas tablas de Salesforce, en las cuales se registran datos de Cuentas, Oportunidades, Candidatos, etc. Todas estas tablas, combinadas, permiten obtener la base de datos final, a partir del cual tenemos una línea por cada adquisición, es decir, una línea por cada nuevo partner.

La unidad de análisis del presente trabajo de investigación, depende de la precisión con la cual se puedan predecir las adquisiciones. En un escenario ideal, sería interesante obtener la predicción de cantidad de adquisiciones por:

Acquisition_Date - Country_Name - AccountSource

Es decir, esta sería nuestra unidad de análisis. Sin embargo, si en el proceso de la elaboración y testeo del modelo, no se logran buenos resultados, o no con una suficiente precisión, entonces se consideraría la posibilidad de restringir el análisis a una unidad más general, que tenga un menor margen de error, quitando Account Source y conservando el resto de las variables.

En cuanto a las variables, expliquemos de qué se trata cada una:

- Acquisitions: es la variable a predecir: la cantidad de adquisiciones. En el presente trabajo, por cuestiones de redacción, se usará Acquisitions o adquisiciones indistintamente para referirse a esta variable.
- Acquisition_Date: se trata del día en que fue adquirido el partner. Consideramos como fecha de adquisición al día en el que se cerró satisfactoriamente la oportunidad del candidato y, consecuentemente, la cuenta pasó a estar activa.
- Country_Name: el país en el cual el partner va a operar. Tenemos 15 países distintos, dentro de la región de Latinoamérica, la cual es nuestro área de análisis: Argentina, Bolivia, Chile, Costa Rica, Ecuador, El Salvador, Guatemala, Honduras, Nicaragua, Panamá, Paraguay, Perú, República Dominicana, Uruguay y Venezuela. En el presente trabajo, nos referimos a Country_Name también como “país” o “mercado”.
- AccountSource: consiste en el canal o la fuente de donde viene la cuenta. ¿Fue adquirida por un vendedor de campo, presencial (*Field*), por medios digitales (*Self Sign Up)?* En relación al Account Source, se menciona que se agregó un tercer grupo de vendedores, los “*Must Have*” (estos vendedores están dedicados a incorporar partners *Must Have*, que suelen ser grandes restaurantes que traen mucho volumen y tráfico a la app. Se caracterizan por tener una baja productividad, ya que se centran en un reducido grupo de candidatos bien importantes). En este trabajo, usamos AccountSource y “canal” como sinónimos. Otras equivalencias son *Field* con físico/campo y *SSU* con digital.

En cuanto a *features*, además de las variables recién mencionadas, durante el análisis, podemos crear *features* adicionales relativos a la fecha como:

- Día de la semana
- Semana del mes (1,2,3,4)
- is_holiday: booleano sí/no en función de un calendario de feriados del país correspondiente
- is_holiday_uruguay: ídem is_holiday pero refiriendo específicamente a Uruguay, donde se manejan muchas operaciones que afectan a nivel regional en las adquisiciones. Esto quiere decir que si un día particular es feriado en Uruguay,

no solo las adquisiciones de este país se pueden ver afectadas negativamente, sino las de toda la región, dado que no habrá, en Uruguay, personal trabajando para sumar socios para ninguno de los quince países.
- Cualquier otro *feature engineering* en función de la fecha de adquisición

Obtención de *features* adicionales: se obtienen los siguientes *features* adicionales para agregar al modelo:

- Seller_Head_Count: consiste en la cantidad de vendedores que tiene cada país en un mes determinado, diferenciado por si es un vendedor de campo (*Field*), por medios digitales (*SSU*) o *Must Have*. Este conteo incluye decimales, dado que se tiene en cuenta el porcentaje de tiempo trabajado por el vendedor en el mes (en caso de vacaciones, licencias, etc, puede que no sea 100%, por lo que contaría como menos de 1 vendedor). Se hace, además, una apertura de este campo en 2:
    - Seller_Head_Count_exp: los vendedores con más de 2 meses completos en la empresa.
    - Seller_Head_Count_new: los vendedores con menos de 2 meses completos en la empresa.

Esta subdivisión se hace porque se entiende que la productividad de un vendedor experimentado suele ser bien superior a la de uno nuevo, por lo que contarlos como el mismo recurso, llevaría a sobreestimar las adquisiciones proyectadas en algunos casos.

- Costos de marketing: la inversión hecha en marketing. La tenemos a nivel Country_Name-día, pero en función de ello, podemos calcular los costos:
    - Marketing_Cost_This_Month: en el mes en que ocurre la adquisición
    - Marketing_Cost_Last_Month: el mes anterior a la adquisición
    - Marketing_Cost_Last_30_days: los 30 días inmediatamente anteriores a la adquisición

Por lo tanto, bajo la unidad de análisis mencionada y con las variables especificadas, la dimensión de la base de datos en forma agrupada es la siguiente:

- 15660 filas
- 4 columnas: las 3 que componen la unidad de análisis, más la cuarta: Acquisitions, la variable a predecir. En caso de incorporar todos los *features* adicionales comentados, la cantidad de columnas pasa a ser de al menos 11 (dependiendo de cuantos features se necesiten crear para cada modelo distinto en el *feature engineering*).

En cuanto a la posibilidad de obtención de datos adicionales, podemos comentar los siguientes puntos:

- Como punto de partida, a medida que va pasando el tiempo, se va generando nueva data. Cada día supone un nuevo registro que podemos tomar para contabilizar adquisiciones y que luego también podemos sumar a nuestra base de datos.
- Se está investigando la posibilidad de conseguir data previa a 2023 y si la misma es confiable. Debido a migraciones de data y fusiones con otras apps, no está claro aún qué tan precisa es esta data.
- Lamentablemente, aunque se cuenta con el *pipeline* de leads / cuentas aún no activas al día de hoy, el mismo no está disponible en el histórico. Esto es una cantidad de candidatos a nuevos partners que están en el proceso de adquisición (ventas en proceso). Hoy podemos ver la data actual, la cual quizás nos podría dar algo de información acerca del futuro, pero no contamos con la misma información para el pasado, solo la presente, al día de hoy.

5.4. Análisis descriptivo

La base de datos agrupada, que es utilizada como insumo para el modelo, entonces, cuenta con 15660 filas (observaciones) y 11 columnas (atributos/*features*). Sin embargo, debido a la gran cantidad de valores 0 en el caso del canal *Must Have*, se considera más valioso hacer un análisis exploratorio de las variables excluyendo estas observaciones de la base de datos, para ver cómo se comportan los otros dos canales. A continuación, una tabla resumen de la base de datos (con datos *Must Have* incluidos, para luego realizar el resto del análisis sin ellos):

	Variable class	# unique values	Missing observations
Acquisition_Date	Date	348	0.00 %
Country_Name	character	15	0.00 %
AccountSource	character	3	0.00 %
Seller_Head_Count_exp	numeric	87	0.00 %
Seller_Head_Count_new	numeric	33	0.00 %
is_holiday	integer	2	0.00 %
is_holiday_uruguay	integer	2	0.00 %
Acquisitions	integer	86	0.00 %
Marketing_Cost_This_Month	numeric	153	0.00 %
Marketing_Cost_Last_Month	numeric	138	0.00 %
Marketing_Cost_Last_30_days	numeric	4270	0.00 %

Figura 1 - Resumen exploratorio de variables

Podemos mencionar que contamos con:

- Variable tipo fecha: Acquisition_Date
- Variables tipo string: Country_Name y AccountSource
- Variables tipo numérico: Seller_Head_Count_exp, Seller_Head_Count_new, Marketing_Cost_This_Month, Marketing_Cost_Last_Month, Marketing_Cost_Last_30_days y Acquisitions, la variable a predecir
- Variables también numéricas pero del estilo booleano (1/0): is_holiday, is_holiday_uruguay

Además, observar que no hay ningún valor faltante en ninguna de las variables. Esto se debe a que el trabajo de obtención de la data completa se hace con anterioridad a la importación de la base de datos en R. Todos los datos son obtenidos y limpiados en BigQuery para luego poder importar la base de datos completa al modelo. Cabe mencionar:

- Se podría conseguir data de adquisiciones previa a 2023, pero ello implicaría datos faltantes en Seller_Head_Count.
- En caso de querer reconstruir la data de vendedores, habría que hacer un trabajo adicional para obtenerla, lo cual no garantiza el nivel de precisión que hoy se tiene sobre la misma.

Hacemos foco, ahora, en algunas variables en particular.

Notamos que la cantidad de observaciones en cuanto a Acquisition_Date es pareja a lo largo del tiempo (Figura 2). Esto se debe a que se toma la foto de la cantidad de adquisiciones cada día (independientemente de que se registren adquisiciones o las mismas sean cero), por lo que no tenemos significativamente más observaciones en un mes que en otro.

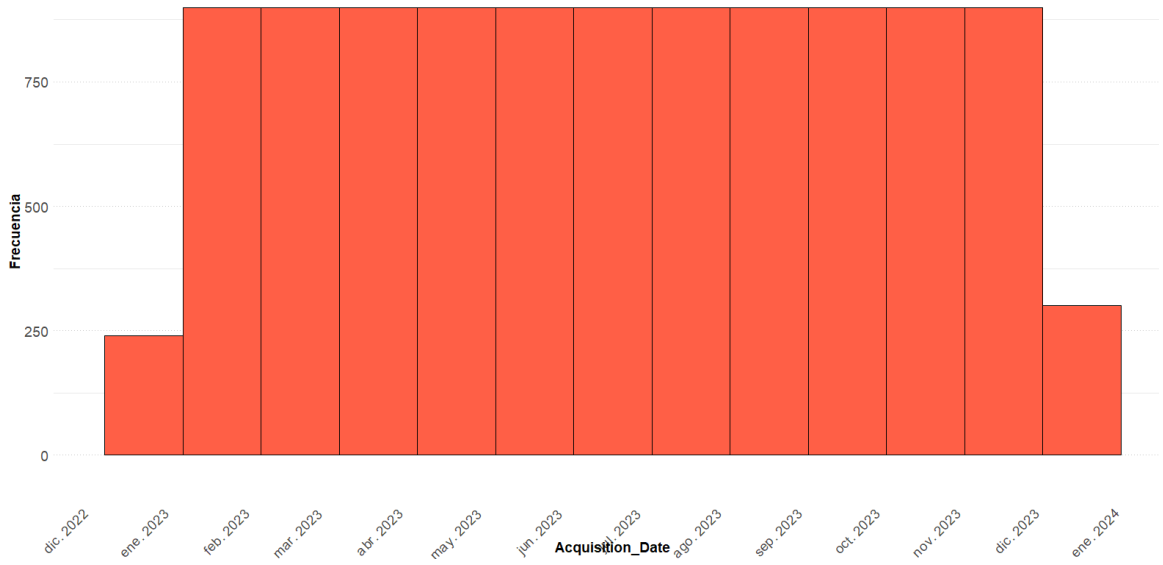

Figura 2 - Frecuencia de observaciones por fecha de adquisición

En cuanto a Seller_Head_Count_exp, vemos que la mayoría de las observaciones se concentran hacia la izquierda, esto es, en bajos valores (Figura 3). Quiere decir que en la mayor parte de los casos contamos con un número de vendedores por debajo de 15, con especial cantidad en casos de solo 1 vendedor para un Country_Name-mes-AccountSource.

La mediana de Seller_Head_Count_exp es 3 y el 1er cuartil es el 1. Esto refuerza lo que venimos exponiendo: muchas observaciones con un nivel bajo de vendedores, y solo unas pocas con alta cantidad: observamos que el 3er cuartil comienza en 5. Esto lleva a encontrar posibles valores atípicos en el caso de aquellas observaciones con un conteo mayor a 20. Corresponden todas a vendedores *Field* de Argentina y Perú.

En cuanto a Seller_Head_Count_new, observamos un panorama similar, aunque con cifras en general menores (Figura 4). La mediana es de 0 y la mayoría de los casos se concentran en este mismo valor, con el tercer cuartil en 1.

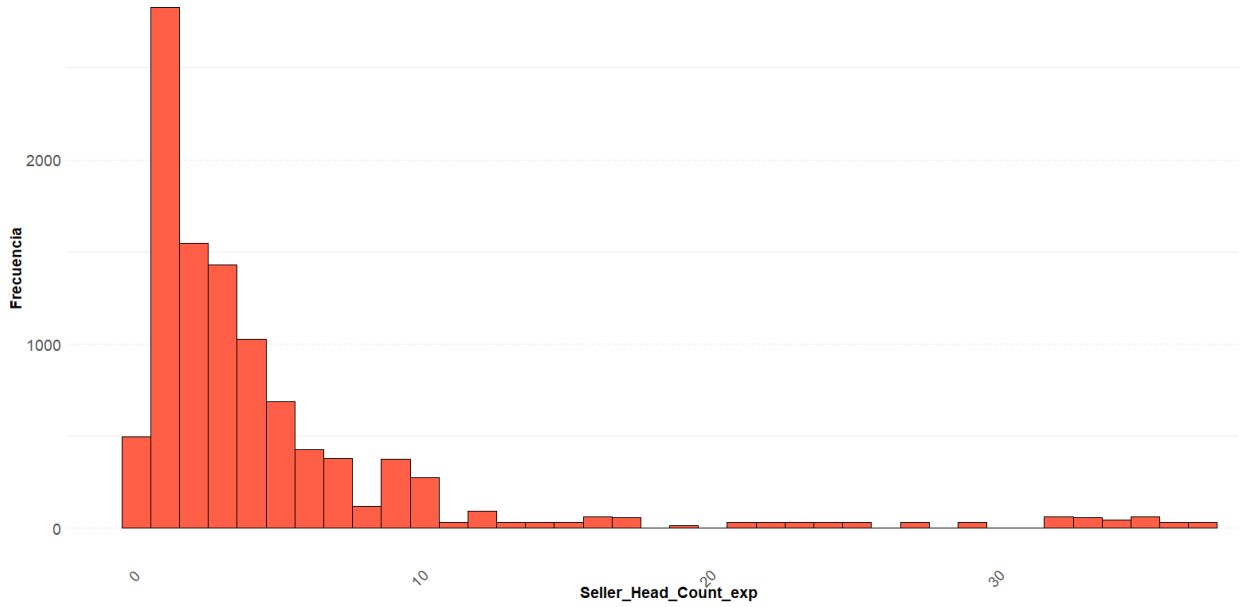

Figura 3 - Frecuencia de observaciones por cada valor del conteo de vendedores experimentados

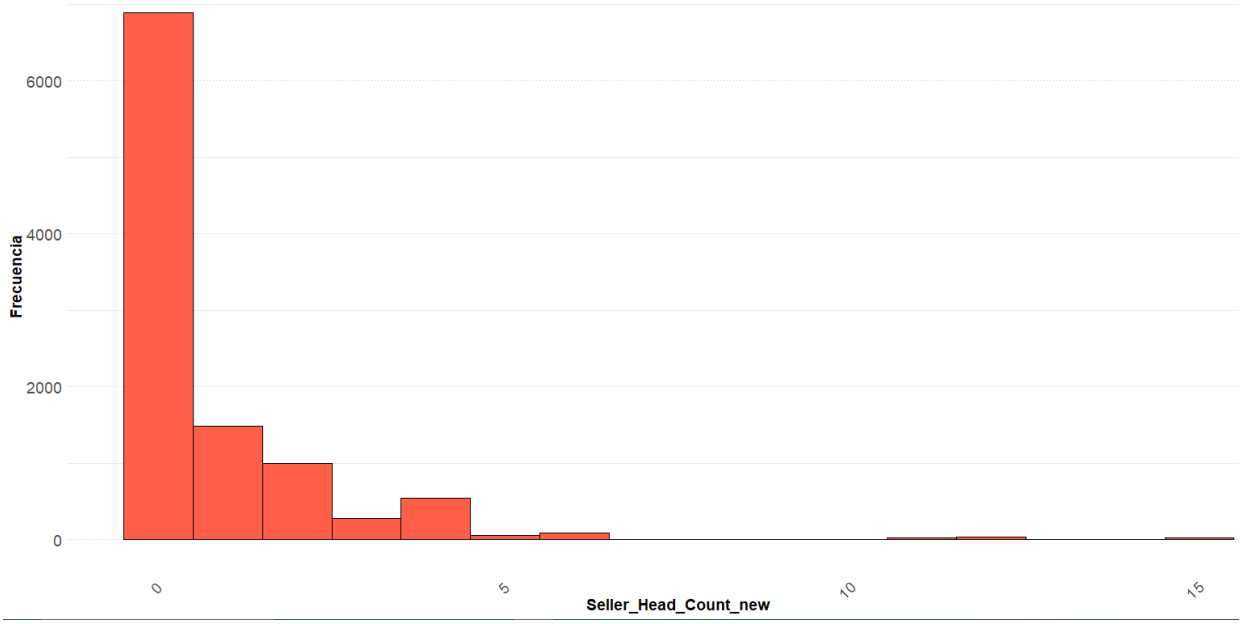

Figura 4 - Frecuencia de observaciones por cada valor del conteo de vendedores nuevos

Un caso similar es el de la variable a predecir: Acquisitions. La mayoría de las observaciones están en valores bajos, siendo 2 la mediana, con el 1er cuartil en 0 y el 3er cuartil comenzando en 6 (Figura 5).

Con estos datos, observamos la posibilidad de valores atípicos en 177, 145, 121, 113, etc. Ahondando en qué observaciones presentan estos valores altos, corroboramos que se trata de, principalmente, observaciones de Perú y Argentina *Field*, así como

algunas de Argentina *SSU*. Estos dos países presentan un nivel de adquisiciones claramente superior al resto.

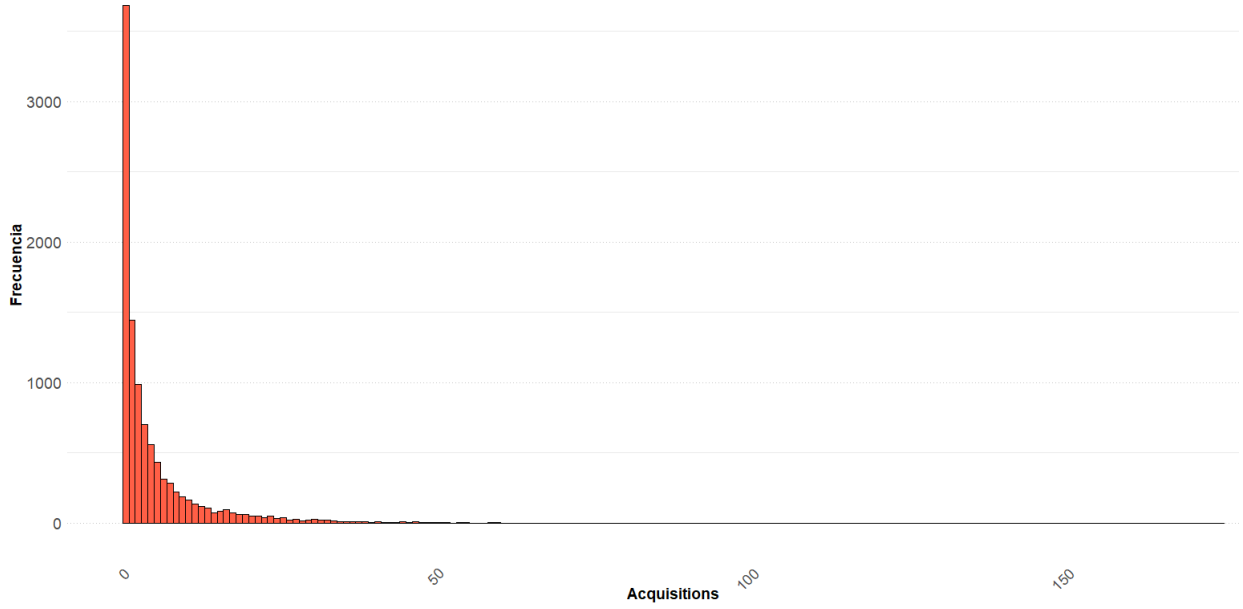

Figura 5 - Frecuencia de observaciones por adquisiciones

En cuanto a los costos de marketing, observamos que todos los datos están cerca del 0, con 0 como mediana (Figuras 6, 7 y 8). Esto es esperable, teniendo en cuenta que cualquier observación de adquisiciones de *Field* tendrá 0 de inversión de marketing, dado que las acciones de marketing son todas por vías digitales (*SSU*).

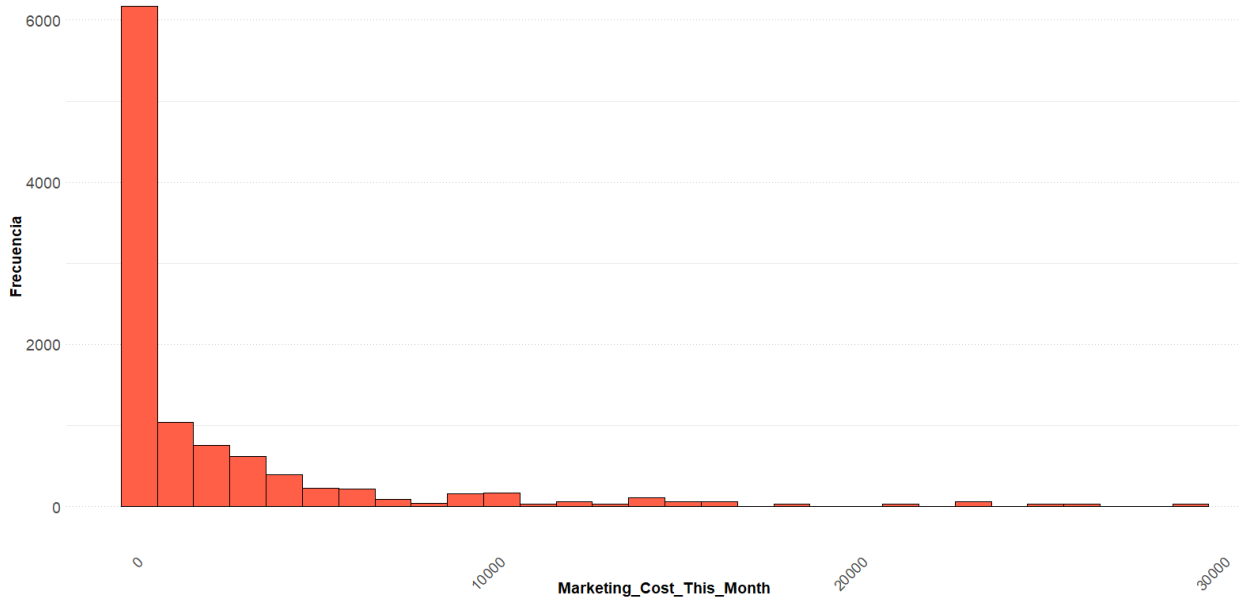

Figura 6 - Frecuencia de observaciones por cada valor del costo de marketing del mes de adquisición

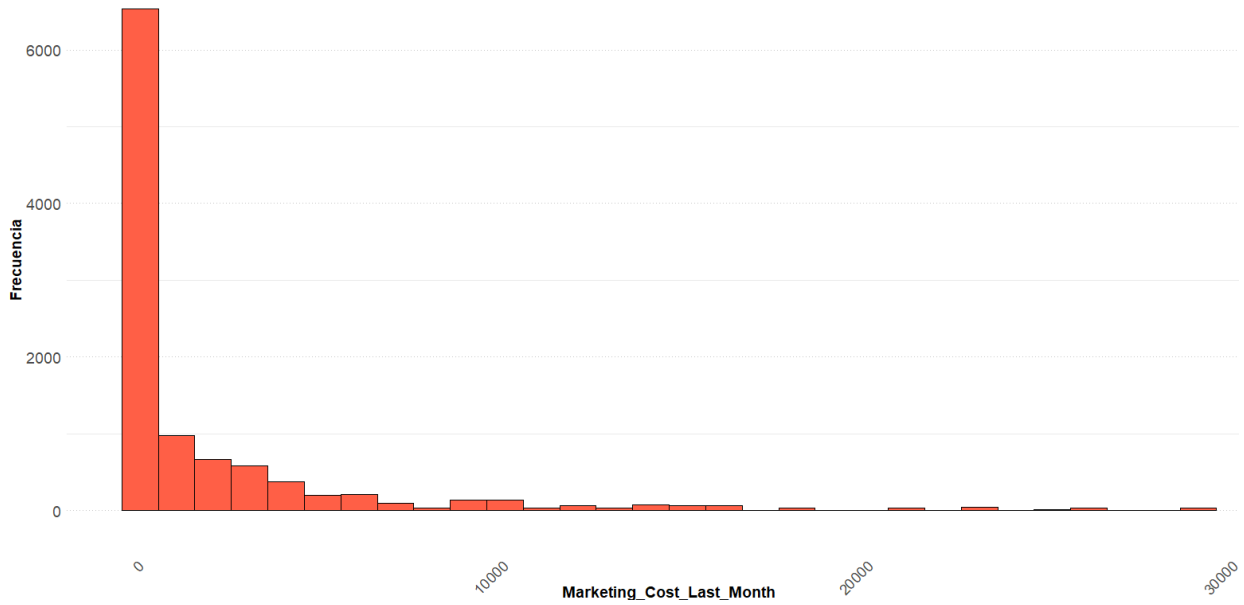

Figura 7 - Frecuencia de observaciones por cada valor del costo de marketing del mes anterior

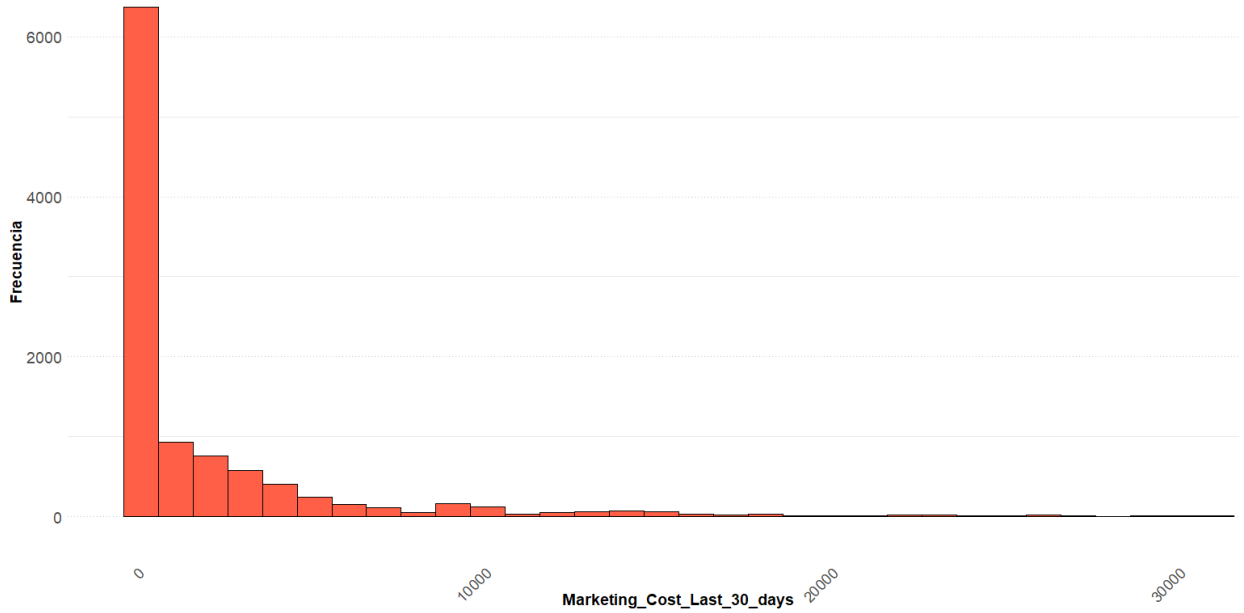

Figura 8 - Frecuencia de observaciones por cada valor del costo de marketing de los 30 días anteriores

Entonces, si filtramos la base de datos únicamente por las observaciones de *SSU*, y volvemos a fijarnos en las variables relacionadas a marketing, vemos una mayor distribución de los valores, los cuales aún se concentran en torno al 0, dado que no todos los países invierten en marketing (o al menos no en todos los meses), pero ya presentan otros valores de Mediana: 2145.61, 1835.55 y 2080.24 para Marketing_Cost_This_Month, Marketing_Cost_Last_Month y Marketing_Cost_Last_30_days, respectivamente (Figuras 9, 10 y 11). Se observa mayor cantidad de inversión en marketing en Chile y en Argentina, con respecto al

resto de los mercados, siendo Bolivia, con una inversión de 10174 en Septiembre 2023, el mayor registro en cuanto a costos por fuera de estos dos países (que alcanzan un máximo de 22759, es decir, más del doble).

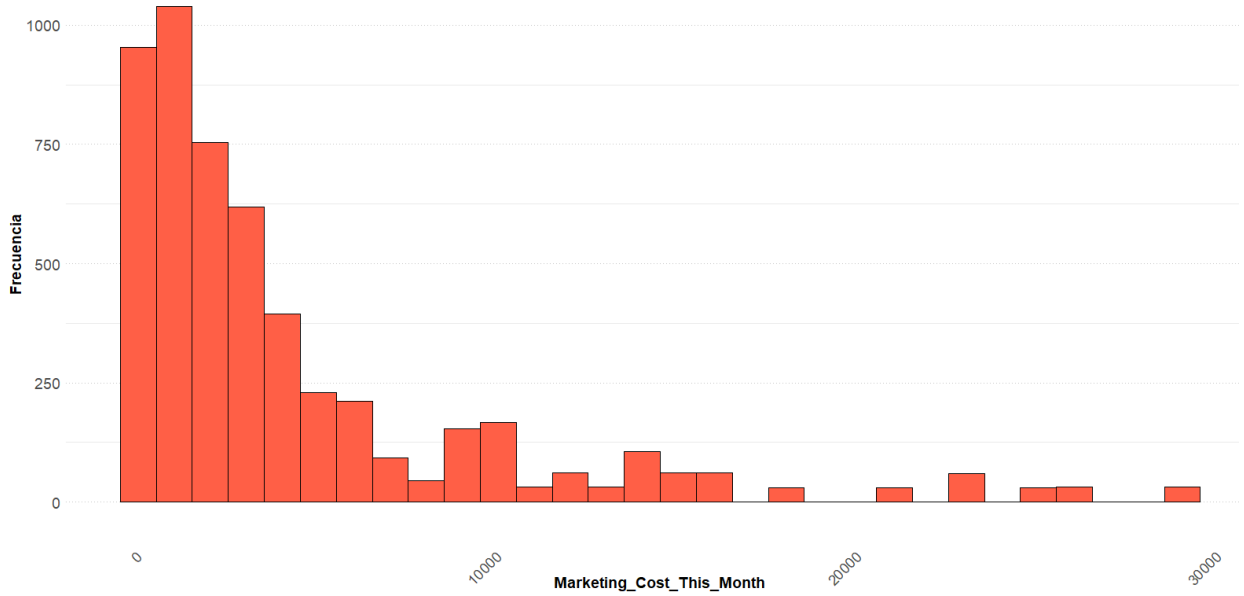

Figura 9 - Frecuencia de observaciones por cada valor del costo de marketing del mes de adquisición (filtrado por SSU)

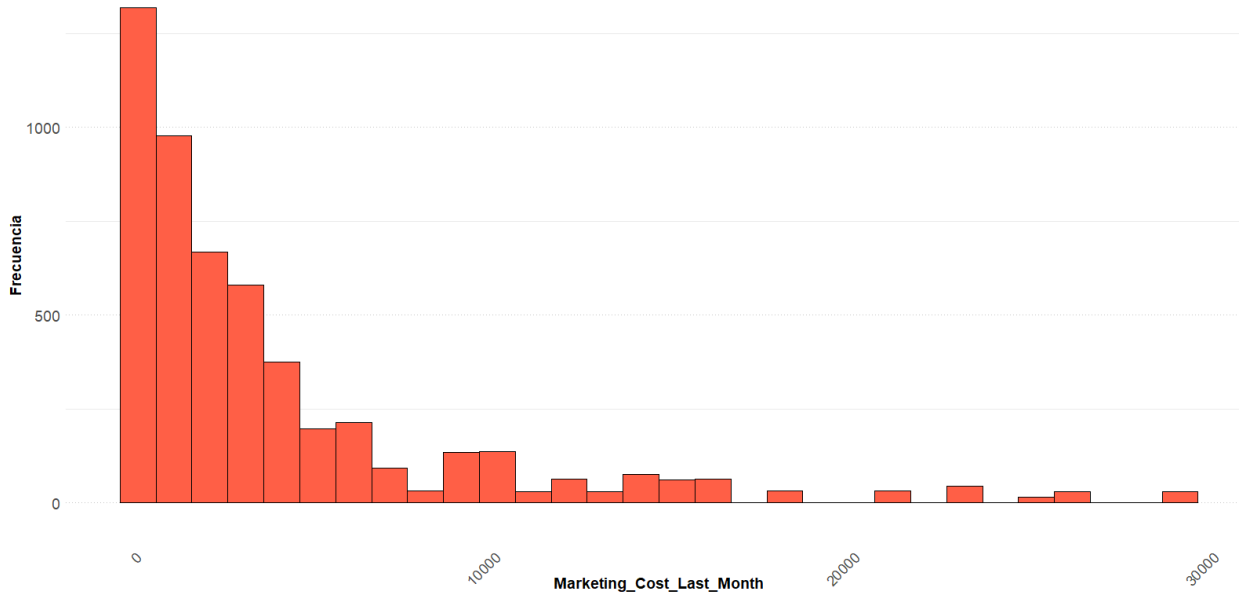

Figura 10 - Frecuencia de observaciones por cada valor del costo de marketing del último mes (filtrado por SSU)

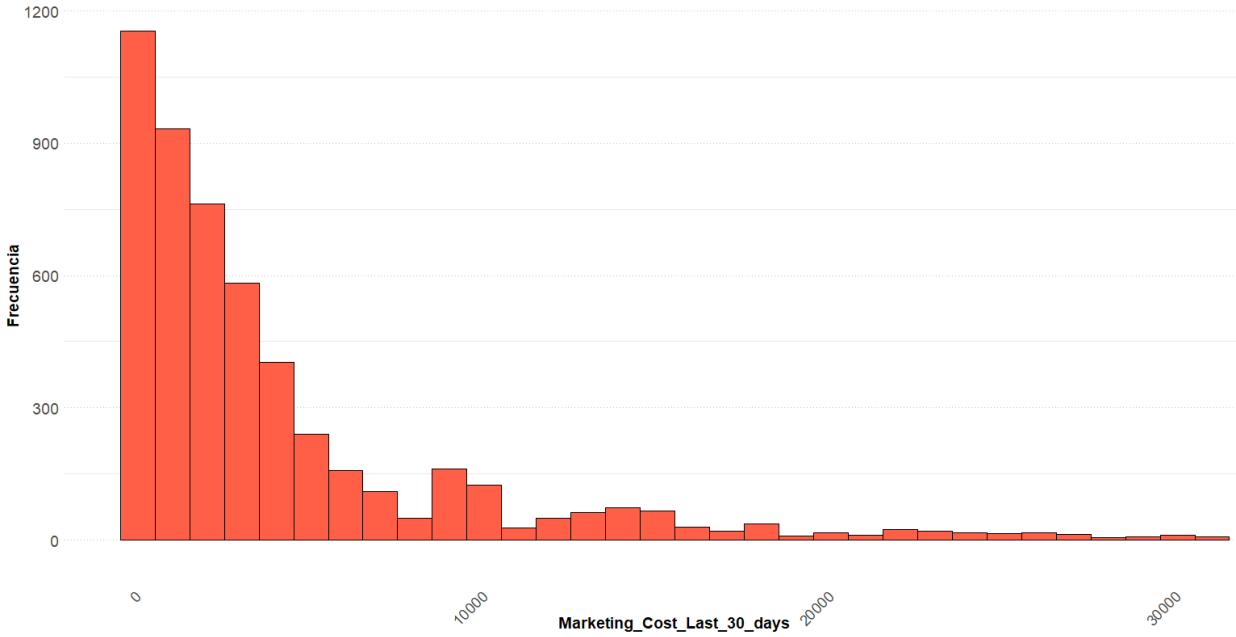

Figura 11 - Frecuencia de observaciones por cada valor del costo de marketing de los últimos 30 días (filtrado por SSU)

Ahora, armando gráficos que relacionan las variables predictoras con la variable a predecir (Acquisitions), podemos observar:

- Acquisition_Date: la relación entre Acquisition_Date y Acquisitions muestra una curva más o menos constante con un leve crecimiento a lo largo del tiempo, sin grandes saltos (Figura 12). Además, observamos cierta estacionalidad o ciclos que se forman, con una campana de adquisiciones que toma una curva ascendente para luego descender y reiniciar el ciclo

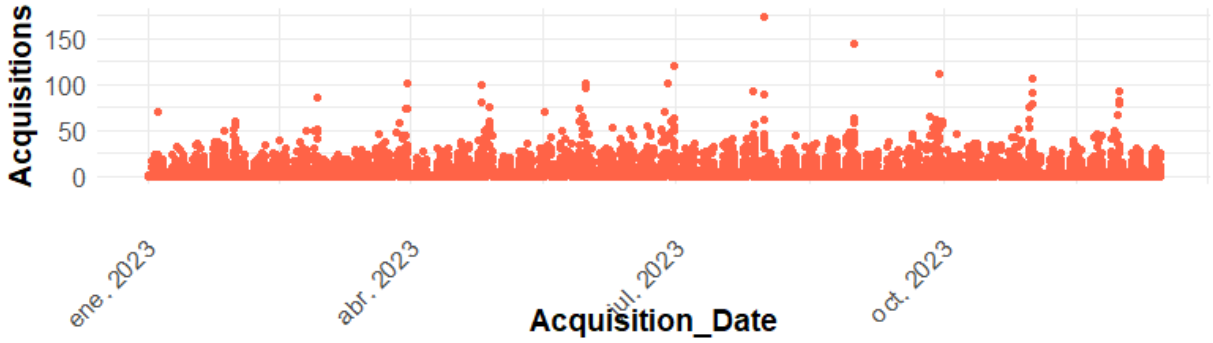

Figura 12 - Observaciones con adquisiciones por fecha de adquisición

Investigando un poco más a fondo, y readaptando las escalas del gráfico a nivel mes, día y semana, observamos que este ciclo se da de forma semanal: una subida fuerte de adquisiciones en los primeros días de la semana, cayendo luego hacia el fin de la misma, para volver a comenzar en la siguiente. En la Figura 13, vemos cómo cada

marca en el eje horizontal indica el inicio de una nueva semana, separando de forma bien clara los ciclos notados.

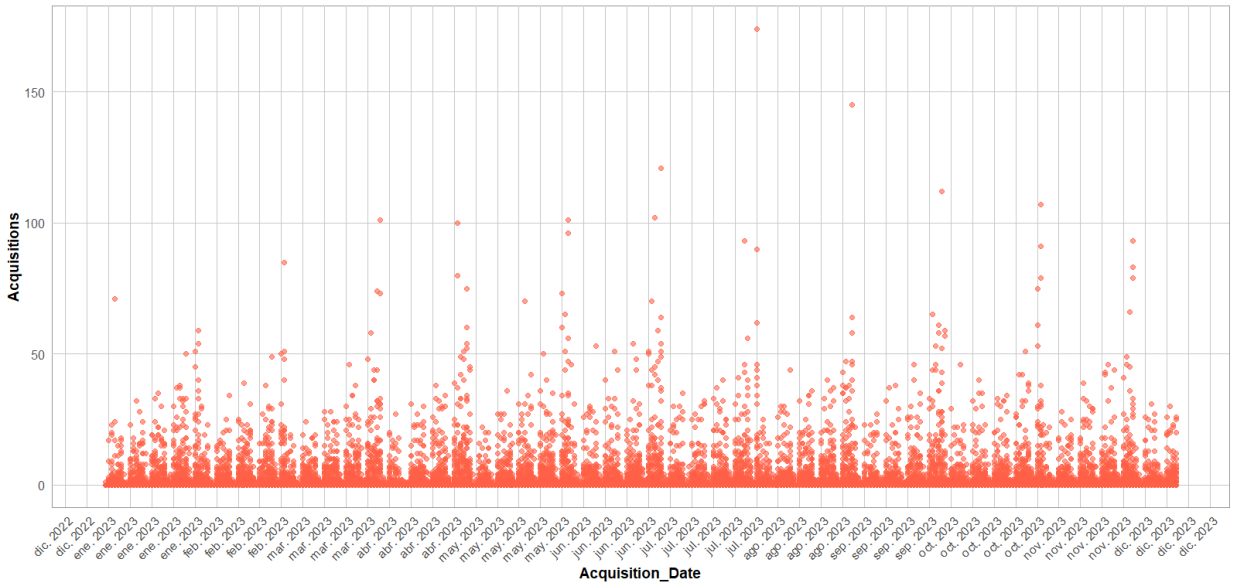

Figura 13 - Observaciones con adquisiciones por fecha de adquisición (escala semanal)

Observando, ahora, la serie a nivel semanal, obtenemos la Figura 14. Aquí vemos cómo en las semanas en torno a la mitad del año se alcanzan los valores más altos de adquisiciones en algunos mercados-canales, pero luego la tendencia vuelve a caer hacia fin de año.

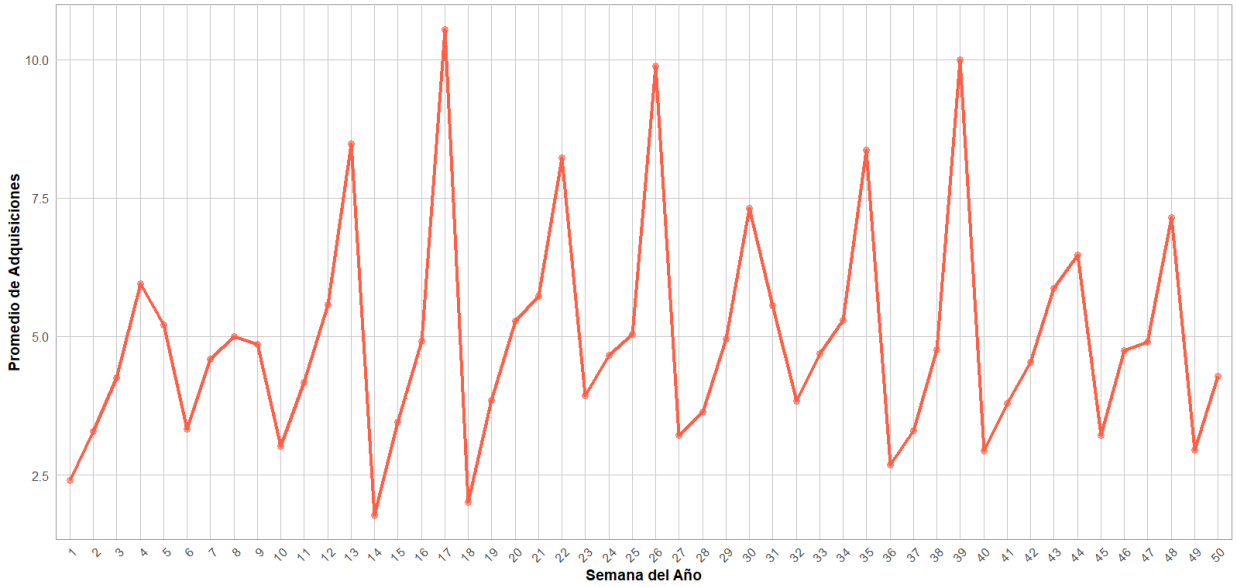

Figura 14 - Promedio de adquisiciones por semana del año

Para observar tendencias hacia dentro de la semana, realizamos un *boxplot* de las adquisiciones por día de semana. Para una mejor observación, sin pérdida de datos,

se muestra la Figura 15: el *boxplot* original, con todos los datos (incluyendo posibles *outliers* o valores muy altos en la distribución); la Figura 16: el mismo *boxplot*, con un límite en su escala vertical, para poder observar mejor las cajas correspondientes a cada día. Notamos:

- Baja dispersión en todos los días: las cajas son muy cortas y no tienen una dispersión de datos grande.
- La mayoría de los datos de días de semana se concentran en niveles bajos, terminando en general el 3er cuartil en torno a 10 adquisiciones, y todo lo superior corresponde al 4to cuartil.
- La dispersión de datos de fines de semana es casi nula y se concentra más que nada en 0, con una mayor cantidad de datos con niveles más altos el sábado vs el domingo. De todos modos, el inicio del 4 cuartil ni siquiera llega a alcanzar la mediana de los días de semana.
- Los datos a lo largo de los días de semana parecen bastante similares y presentan casi la misma mediana.
- La mediana, en lunes, martes y miércoles es más baja que en jueves y viernes, lo que significa que el 3er cuartil tiene más densidad en estos últimos dos días vs los 3 primeros (mayor cantidad de observaciones con más adquisiciones).
- Muchos valores aparentemente atípicos, que se extienden más allá de las líneas o “bigotes” del *boxplot*.
- Poca simetría: como se mencionó, las cajas están en torno a valores bajos, bien lejos de los valores atípicos.

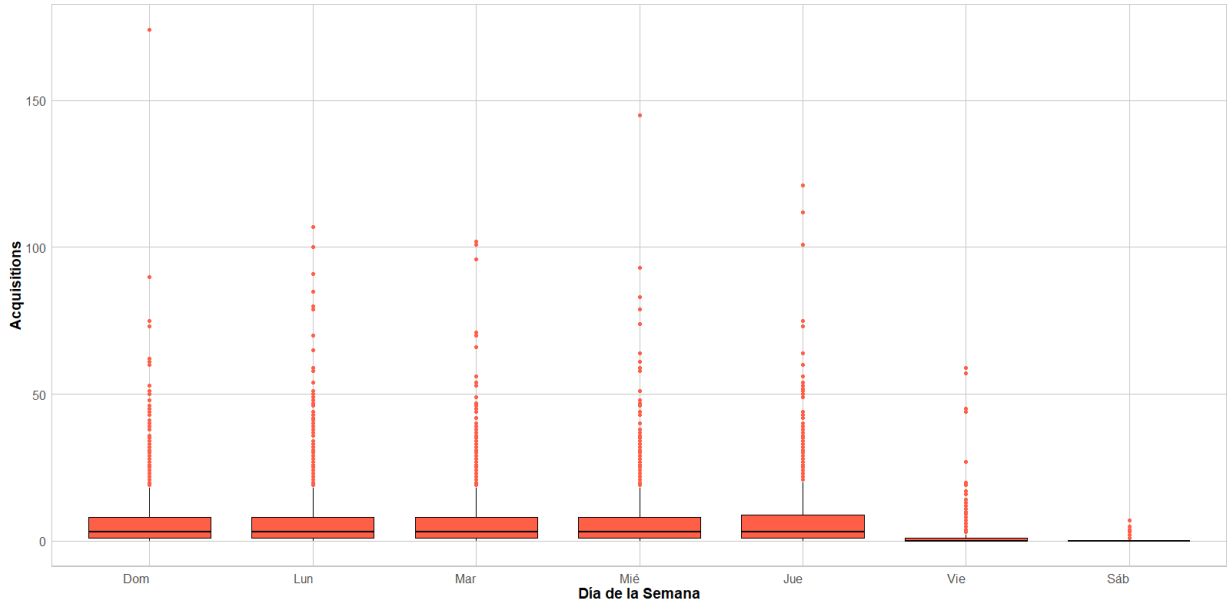

Figura 15 - Boxplot de adquisiciones por día de la semana

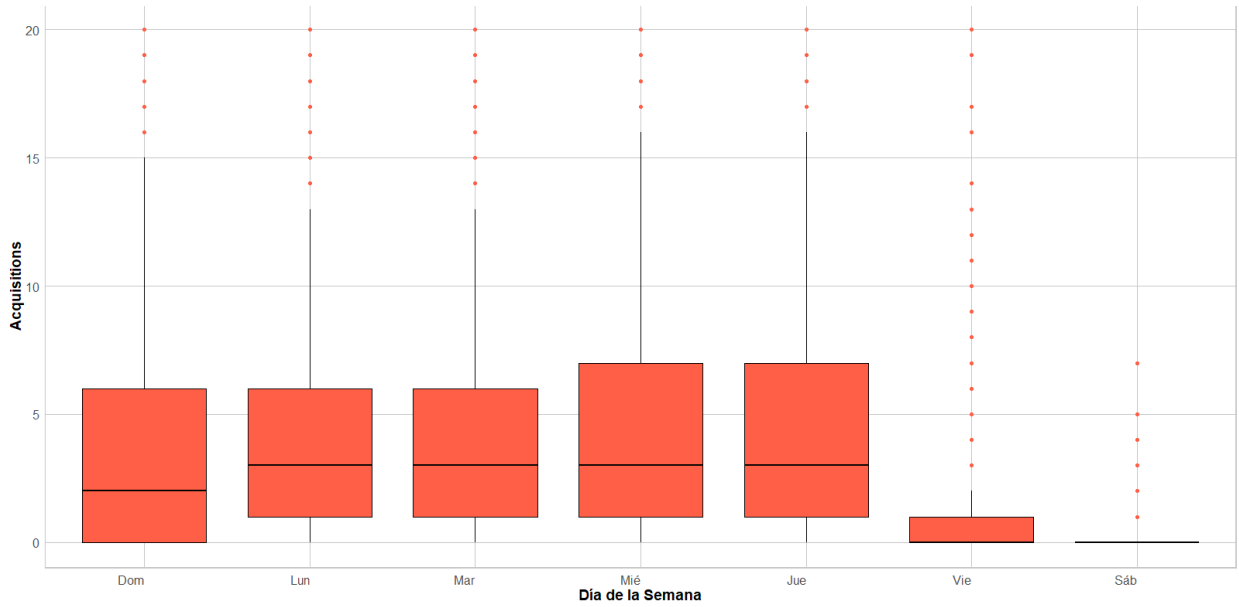

Figura 16 - Boxplot de adquisiciones por día de la semana (escala vertical limitada)

- Feriados: como era de esperar, los días feriados en los respectivos países (is_holiday) y en Uruguay (is_holiday_uruguay) derivan en menor cantidad de adquisiciones. Se observa un nivel claramente superior en el promedio de adquisiciones los días no feriados (5.1 adquisiciones), en oposición a los feriados (1 adquisición promedio). Lo mismo ocurre en el caso de los días festivos de Uruguay (donde la relación de los promedios es de 5.01 vs 3.78).

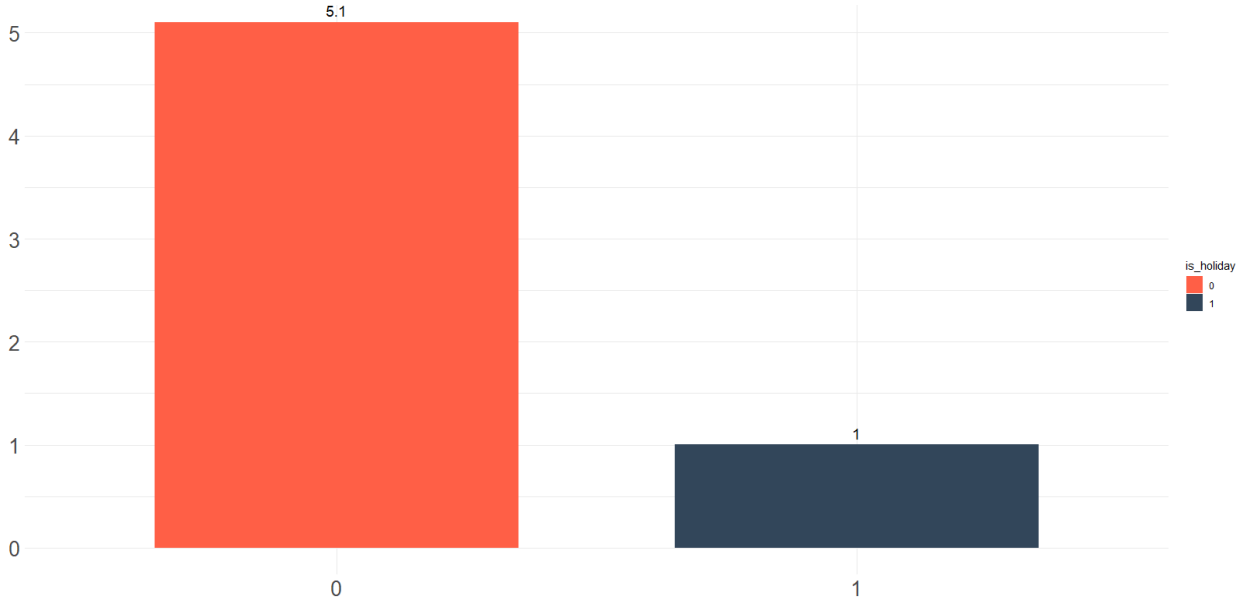

Figura 17 - Promedio de adquisiciones por día feriado/no feriado

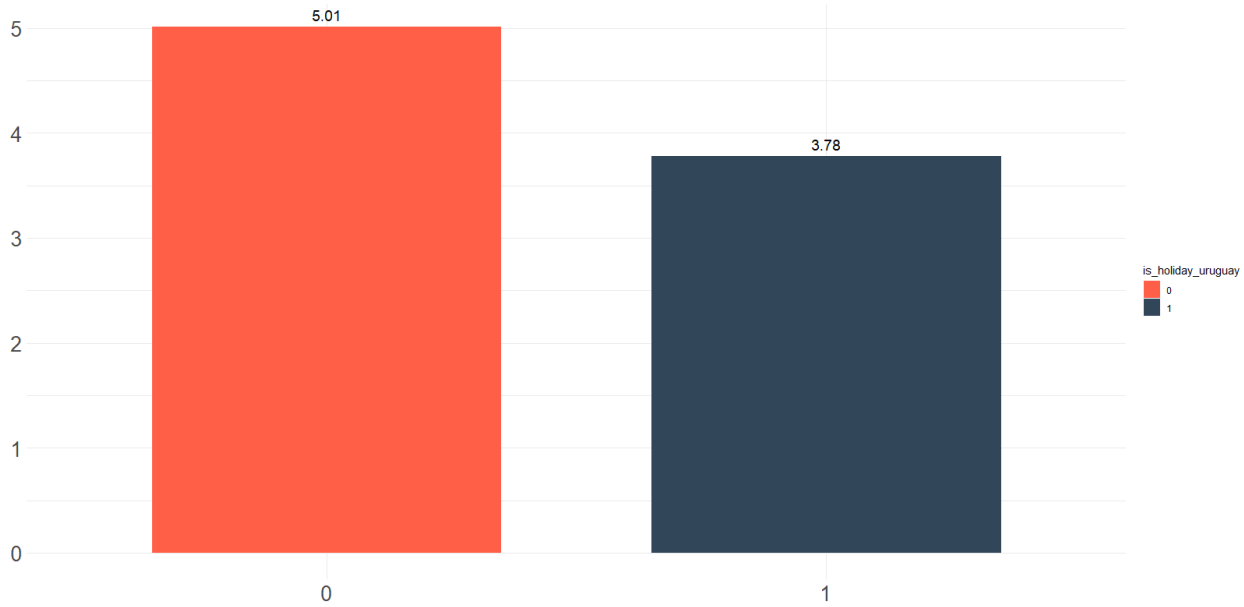

Figura 18 - Promedio de adquisiciones por día feriado/no feriado en Uruguay

- AccountSource: se nota un mayor nivel de adquisiciones en *Field* (58.4% del total) vs *SSU* (41.6%). La media de *Field* es de 5.80, vs 4.13 de *SSU* y las medianas son 2 vs 1 respectivamente (Figuras 19 y 20).

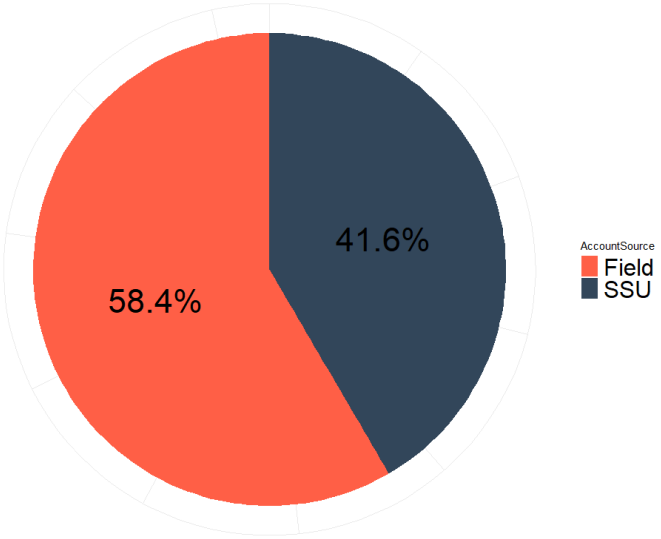

Figura 19 - Proporción de adquisiciones por canal

SSU		*Field*	
media	mediana	media	mediana
4.12682	1	5.798851	2

Figura 20 - Medidas de tendencia central de Acquisitions por AccountSource

- Seller_Head_Count_exp: un avistamiento del gráfico podría dar una idea en contra de la intuición, no registrando un nivel altamente superior de

adquisiciones en mercados-canales con mayor cantidad de vendedores, vs los que tienen menos (Figura 21). Sin embargo, en promedio, tienen más adquisiciones los mercados-canales con alto nivel de Seller_Head_Count_exp que los de un nivel menor. Para corroborar esto, separamos a las observaciones en 2: aquellas por encima de la media de Seller_Head_Count_exp y aquellas por debajo y comparamos su media y mediana de adquisiciones, corroborando finalmente la gran diferencia entre ellas en favor de un nivel mayor de Seller_Head_Count_exp (Figura 22).

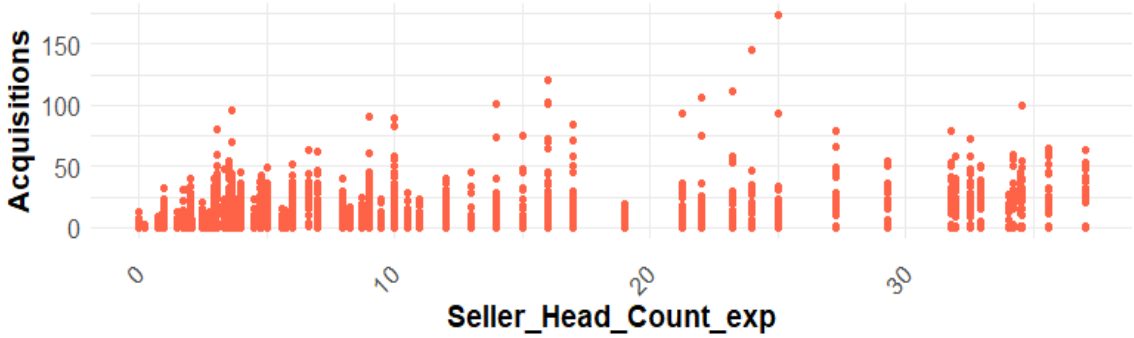

Figura 21 - Observaciones con adquisiciones por cada valor del conteo de vendedores experimentados

HC >= media		HC < media	
media	mediana	media	mediana
8.562783	4	2.542688	1

Figura 22 - Medidas de tendencia central de Acquisitions por Seller_Head_Count_exp

- Lo mismo ocurre con Seller_Head_Count_new: media de Adquisiciones de 8.22 vs 3.29 y mediana de 3 vs 1 para observaciones con un Seller_Head_Count_new superior a la media e inferior a la media respectivamente. Conclusión: a más vendedores (nuevos o experimentados), más adquisiciones.

- Marketing: si observamos la totalidad de la base de datos, no se ve una clara relación, pero al filtrar solo por *SSU* (Figuras 23, 24 y 25), observamos que, a más inversión en marketing, más adquisiciones (los valores más altos en adquisiciones se alcanzan en los valores más altos de estas tres variables, no pudiendo perforar la barrera de las 50 adquisiciones diarias con una baja inversión).

Figura 23 - Observaciones con adquisiciones por cada valor del costo de marketing del mes de adquisición

Figura 24 - Observaciones con adquisiciones por cada valor del costo de marketing del último mes

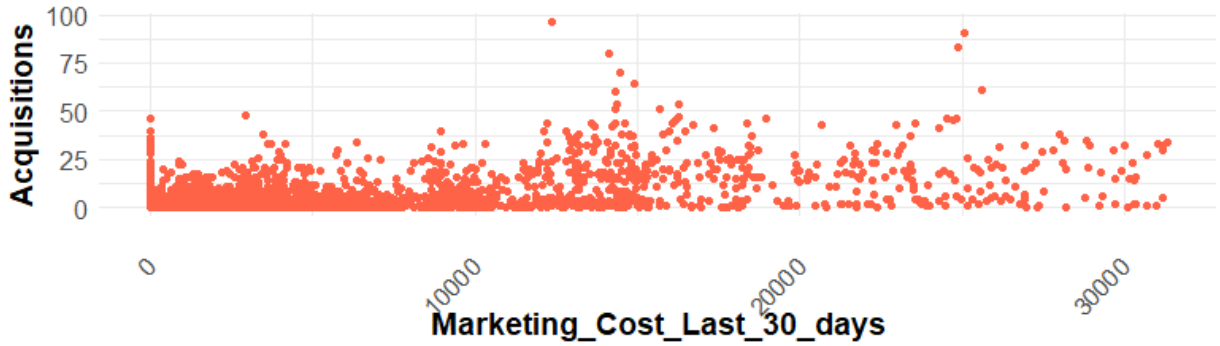

Figura 25 - Observaciones con adquisiciones por cada valor del costo de marketing de los últimos 30 días

En cuanto a los datos Must Have, como se mencionó, encontramos valores en 0 en la mayoría de los campos, por lo que no se muestran mayores detalles aquí. Sí que incluimos, más adelante, un gráfico comparando adquisiciones de los 3 canales, para dar cuenta de la diferencia entre ellos.

Dado que notamos ciertos datos atípicamente más altos que otros en el caso de Acquisitions, vamos a tomar una mirada más cercana a esta variables.

En el caso de Acquisitions, graficamos el total de adquisiciones por día, sumando las adquisiciones para todos los registros que coincidan en su valor de Acquisition_Date

(Figura 26). Al hacerlo, seguimos corroborando lo que habíamos notado anteriormente: ciclos semanales que se van repitiendo, pero también una leve tendencia al alza a medida que pasa el tiempo, con una aparente caída hacia el final del 2023.

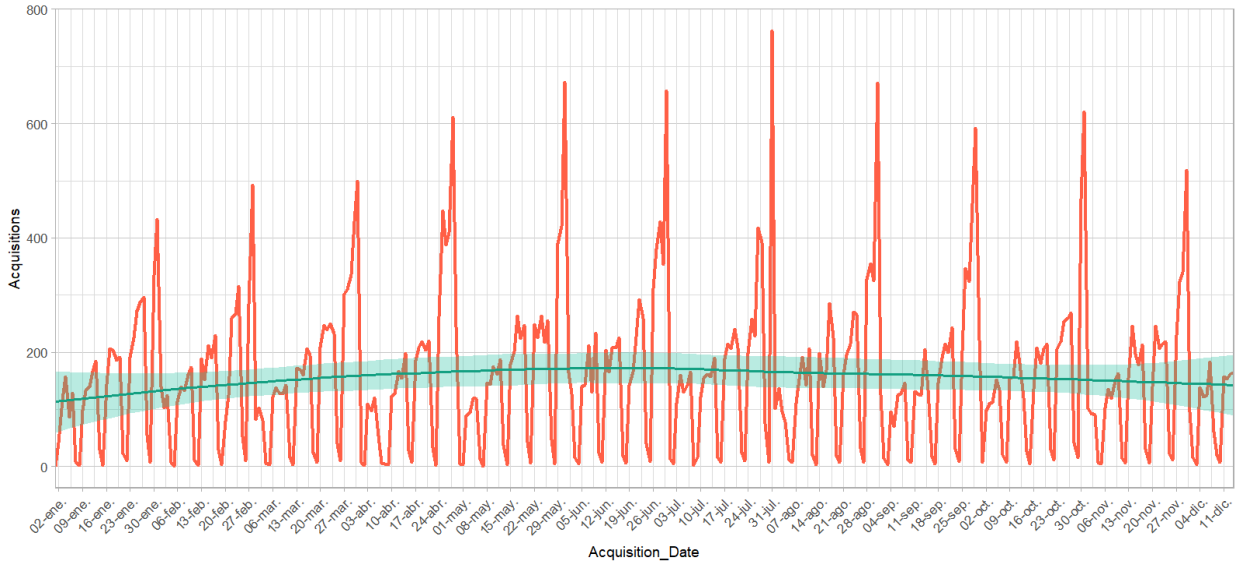

Figura 26 - Adquisiciones por fecha

Al observar el mismo gráfico a nivel semanal (Figura 27), podemos notar cómo la tendencia de adquisiciones va en aumento a medida que avanza el año hacia semanas pertenecientes a meses centrales, y luego vuelve a disminuir a fin de año. También, vemos cómo el nivel de adquisiciones logra generalmente su pico mensual en la última semana de cada mes, volviendo a descender al cambiar al mes siguiente.

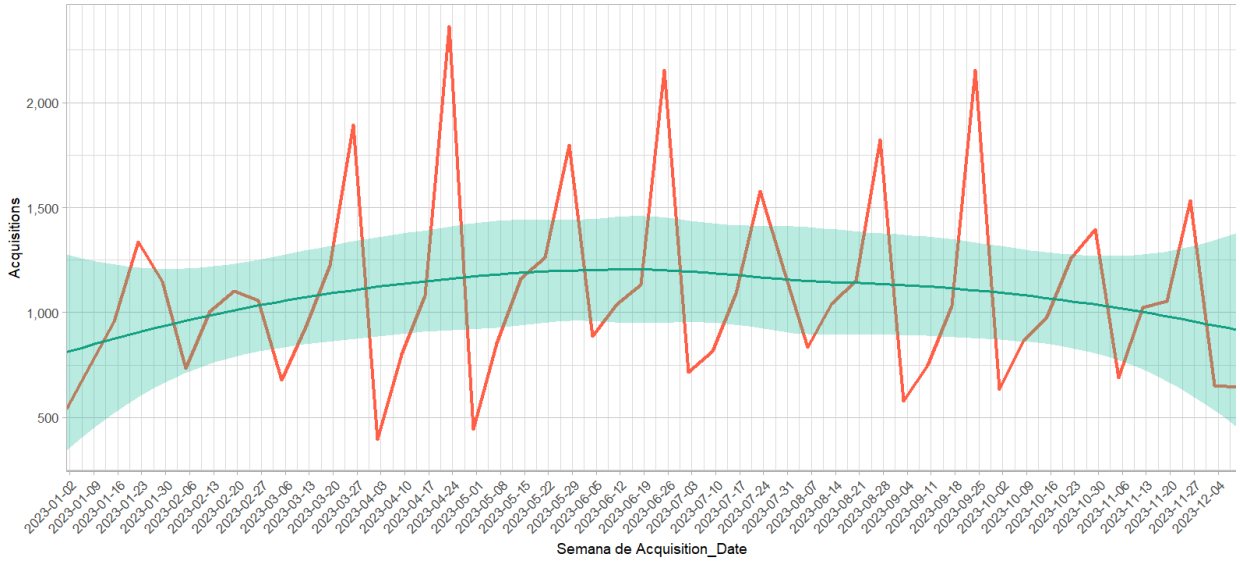

Figura 27 - Suma de adquisiciones por semana

Si hacemos el mismo gráfico, pero abriendo por AccountSource (Figura 28), variable que compone parte de nuestra unidad de análisis, observamos 3 cuestiones:

- El ciclo estacional de una semana coincide en las 3 fuentes de nuevas cuentas, aunque haya diferencias en los niveles de adquisiciones de cada fuente.
- Las adquisiciones de *Must Have* se mantienen más o menos estables a lo largo del tiempo, mientras que las de los otros dos canales tienen mayor variación y ciclos paralelos (suben juntas y bajan juntas).
- Se nota la gran diferencia entre *Field* (mayor nivel de adquisiciones) y *SSU* a lo largo de todo el tiempo, salvo al final: momento en el que *SSU* alcanza el nivel de *Field* y hasta quiebra la línea en algunos días. Las adquisiciones de *Must Have*, siempre bajas, en niveles de 0 y ningún pico supera ampliamente las 50 diarias.
- Se nota una segunda tendencia, ya no semanal sino mensual: los ciclos semanales se mantienen, pero hacia final del mes, el nivel de adquisiciones en general de *Field* y *SSU* aumenta mucho, teniendo picos mucho más altos en la última semana del mes, si se la compara con el resto. Esta tendencia no se observa en *Must Have*.

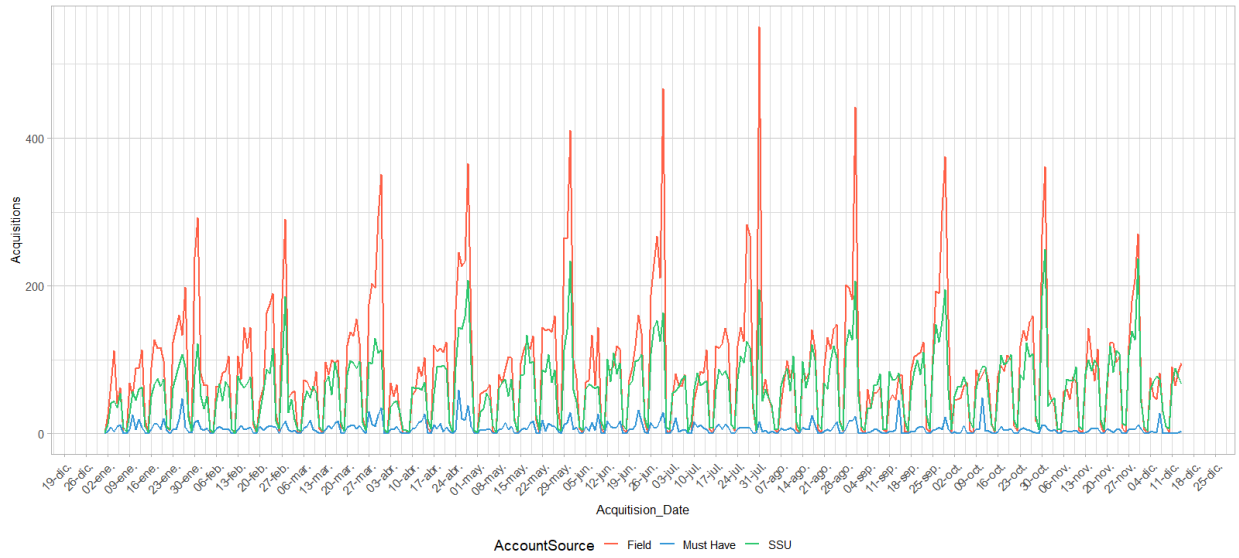

Figura 28 - Adquisiciones por fecha, abiertas por canal

Estas mismas tendencias y observaciones se corroboran al ver la data a nivel semanal (Figura 29). El quiebre que hace *SSU* vs *Field* sobre fin de año aquí es menos notorio, debido a la unidad de tiempo más general, pero sigue siendo perceptible:

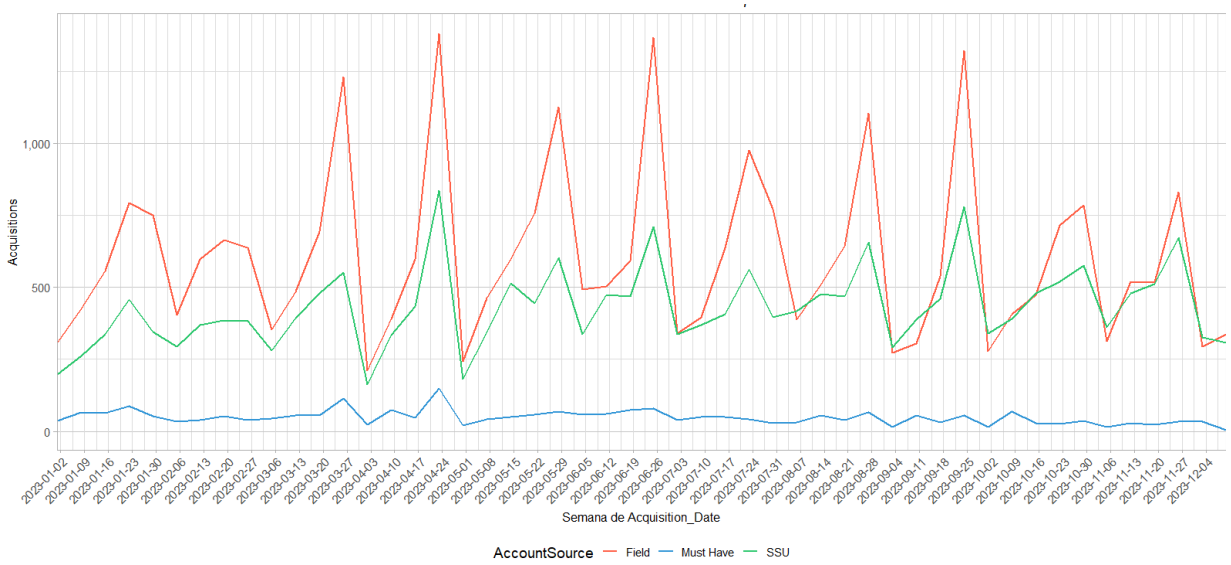

Figura 29 - Suma de adquisiciones por semana, abiertas por canal

Del mismo modo, abrimos por Country_Name y notamos que estos ciclos se repiten en cada mercado (Figura 30). A su vez, vemos cómo el nivel de adquisiciones en Argentina y Perú es altamente superior al resto, con picos que superan las 200 adquisiciones diarias en Argentina y Perú, mientras que en el resto, como máximo se llega a 100 (Guatemala), siendo más común un pico máximo de alrededor de 60. En conclusión, los posibles valores atípicos en Acquisitions en Argentina y Perú se deben a que, en realidad, estos valores no son valores atípicos en sus respectivos mercados, sino que todo el mercado muestra un nivel superior al resto de la región.

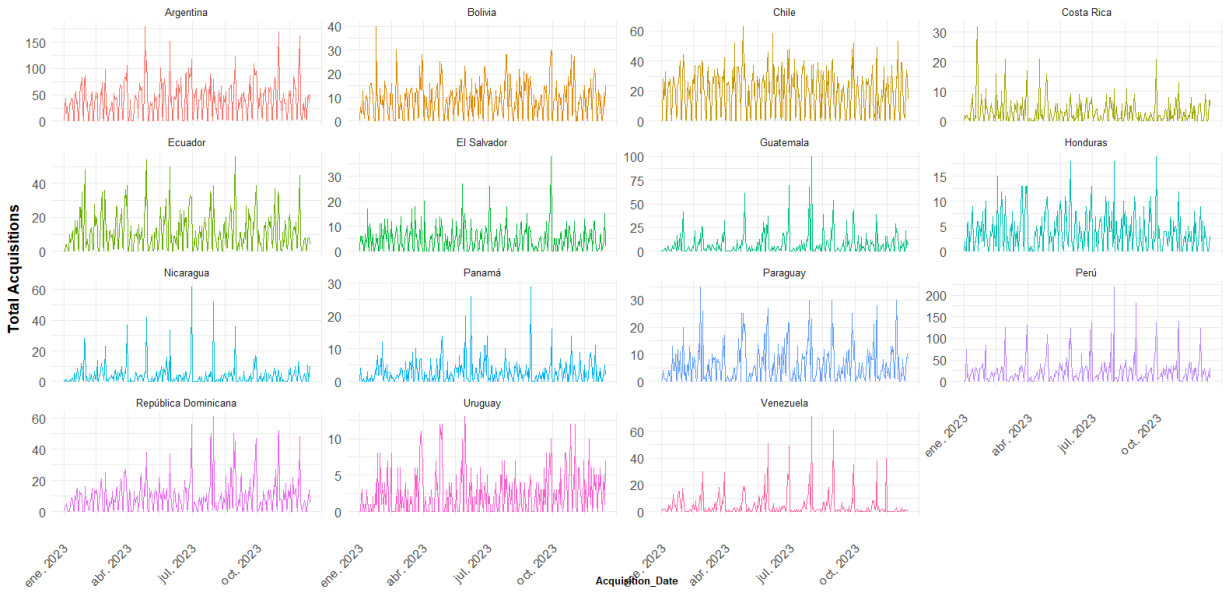

Figura 30 - Adquisiciones por fecha, abiertas por país

Si observamos estos datos, en frecuencia semanal (Figura 31), notamos algo similar:

Argentina y Perú con picos de +400 adquisiciones semanales, mientras el resto de los países se mantienen por debajo de 200.

Figura 31 - Suma de adquisiciones por semana, abiertas por país

Por último, realizamos un mapa de calor (Figuras 32, 33 y 34) que muestra las correlaciones entre cada uno de los *features* (variables explicativas) y la variable explicada: Acquisitions. Debido a que nuestro trabajo se centra en predecir Acquisitions para cada Country_Name - AccountSource, decidimos elaborar este mapa de calor con la siguiente estructura, que permita establecer correlaciones más precisas: Se realizan tres iteraciones, filtrando en cada una por uno de los tres valores posibles de AccountSource ('Field', 'SSU', 'Must Have'). En cada iteración se tiene:

- Filas: cada uno de los *features* tenidos en cuenta como variables explicativas de Acqusitions.
- Columnas: en este caso, la variable tiempo es parte de las variables que explican Acquisitions, no la usamos en el eje horizontal. Por el contrario, usamos dicho eje para representar cada uno de los países (Country_Name) de nuestra base de datos.
- Color y etiqueta: representando la intensidad de la correlación entre el *feature* en cuestión y Acquisitions.

De este modo, se logra representar la correlación entre Acquisitions y cada uno de sus *features*, incluyendo la variable tiempo como una predictora, y siendo capaces de separar cada Country_Name - AccountSource de modo de obtener la mayor precisión posible en la correlación. Para el caso de Acquisition_Date, la variable fecha, se realiza una transformación de la misma para poder ser incluida en el análisis de

correlación, obteniendo el mes (Month), día del mes (Day) y día de la semana (Dayofweek) de cada Acquisition_Date. El año no es analizado, dado que toda nuestra base está construida sobre datos de 2023.

Cabe mencionar que, en ciertos casos, las variables relacionadas a costos de marketing y Seller_Head_Count_new / Seller_Head_Count_exp no encuentran una correlación (N/A), debido a que sus valores no presentan variabilidad día a día (sino más bien mes a mes). Esto, especialmente en el caso de las variables relacionadas a costos de marketing, que obviamente solo presentan valores (distintos de cero) para el caso de *SSU*. Por lo tanto, si bien omitidas del análisis correlativo en algunos casos, seguiremos incluyéndolas como *features* en nuestros modelos.

Al analizar los resultados notamos que:

- El día del mes (Day) es de las variables que mejor explican Acquisitions, teniendo correlación positiva, esto quiere decir, a más tarde en el mes, más Acquisitions esperadas.
- is_holiday es otro gran predictor, con una correlación negativa. Esto quiere decir que si es feriado, las Acquisitions tienden a ser menores.
- is_holiday_uruguay no parece tener demasiada incidencia en las Acquisitions, o al menos no se correlacionan con esta variable de forma significativa.
- El resto de las features oscilan en términos de sus valores de correlación, pero todos con suficiente significación como para considerarse buenos candidatos a predictores.

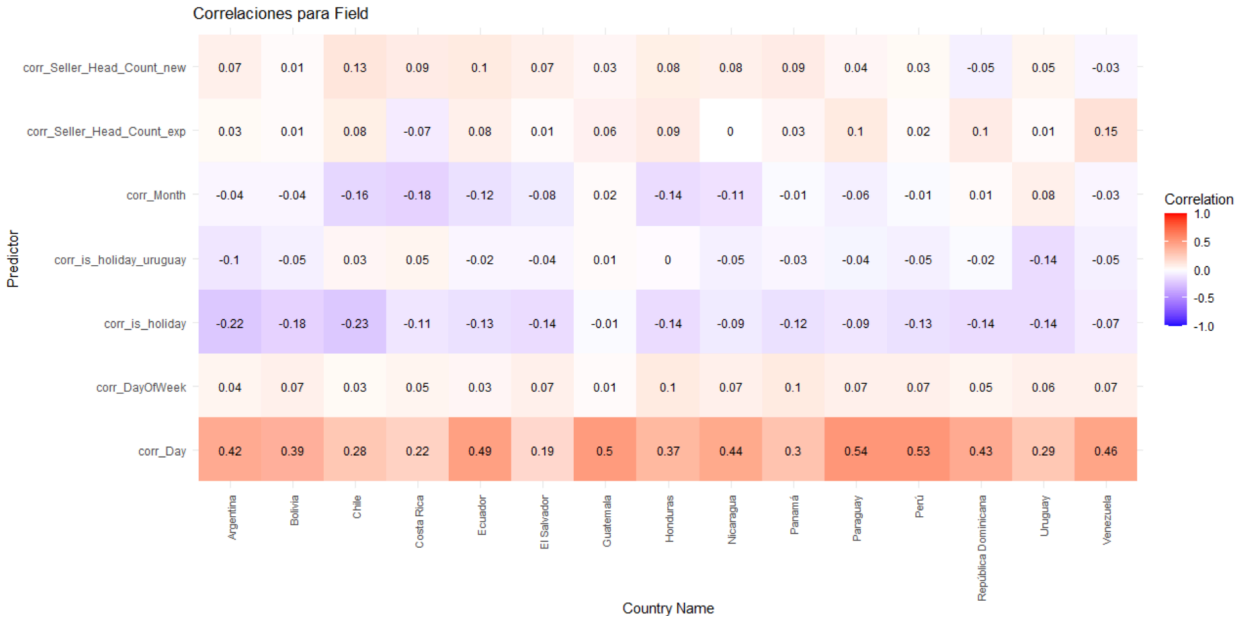

Figura 32 - Mapa de calor de correlación de adquisiciones con features, por país (Field)

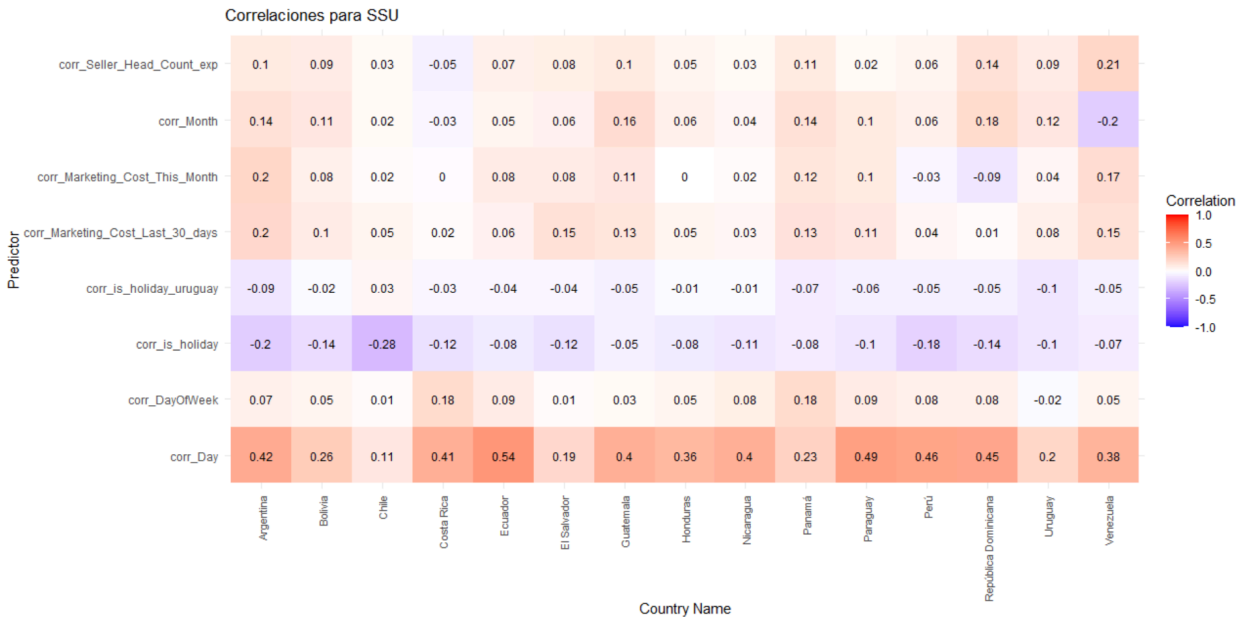

Figura 33 - Mapa de calor de correlación de adquisiciones con features, por país (SSU)

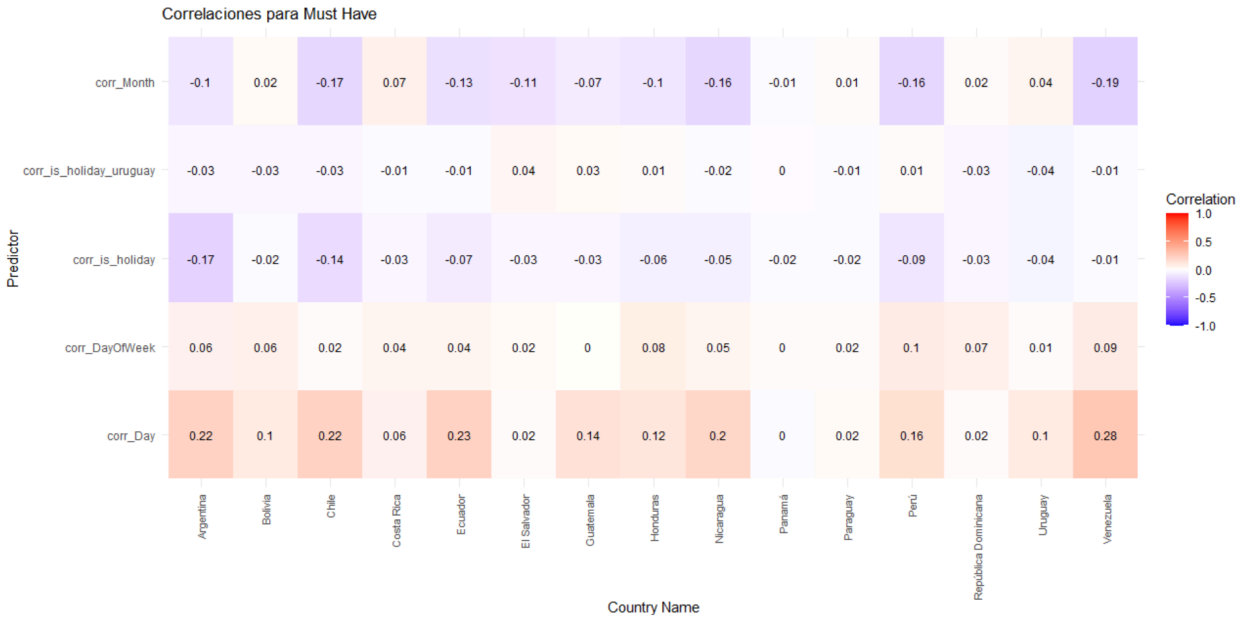

Figura 34 - Mapa de calor de correlación de adquisiciones con features, por país (Must Have)

Para concluir el análisis descriptivo, incluimos, a modo de resumen, una tabla (Figura 35) con distintas medidas estadísticas aplicadas sobre la variable a predecir y las predictoras, con excepción de is_holiday e is_holiday_uruguay. Esta tabla y sus indicadores estadísticos sintetizan, en parte, lo que en el presente análisis descriptivo se ha hallado.

Country_Name	AccountSource	Acquisitions				Seller_Head_Count_exp				Seller_Head_Count_new				Marketing_Cost_This_Month				Marketing_Cost_Last_Month				Marketing_Cost_Last_30_days			
		Min	Max	Media	DE	Min	Max	Media	DE	Min	Max	Media	DE	Min	Max	Media	DE	Min	Max	Media	DE	Min	Max	Media	DE
Argentina	SSU	0	96	19.30	16.57	2.97	10.00	5.63	2.54	0.00	2.82	1.03	0.97	2,683	26,133	14,774	7,040	0	26,133	13,185	7,801	64	31,322	14,076	7,743
Bolivia	SSU	0	13	3.54	2.92	1.00	3.00	1.95	0.93	0.00	2.00	0.52	0.88	1,483	14,361	6,574	3,586	0	14,361	5,611	3,572	0	14,807	6,172	3,721
Chile	SSU	0	33	12.04	8.55	2.00	5.00	3.83	1.01	0.00	1.00	0.35	0.48	4,038	28,684	15,075	7,022	0	28,684	13,431	7,950	33	30,343	14,247	7,493
Costa Rica	SSU	0	11	1.23	1.67	0.88	1.00	0.99	0.03	0.00	0.00	0.00	0.00	439	7,146	3,855	2,182	0	7,146	3,697	2,347	12	7,881	3,770	2,285
Ecuador	SSU	0	29	4.24	4.77	1.00	3.00	1.65	0.85	0.00	0.00	0.00	0.00	0	8,933	2,553	2,303	0	8,933	2,001	1,904	0	8,991	2,325	2,185
El Salvador	SSU	0	13	1.32	1.83	0.00	1.00	0.63	0.47	0.00	0.00	0.00	0.00	681	2,146	1,202	398	0	2,146	1,064	472	0	2,158	1,146	447
Guatemala	SSU	0	12	1.20	1.92	0.86	1.00	0.96	0.04	0.00	0.00	0.00	0.00	1,218	5,382	3,080	1,277	0	5,382	2,736	1,436	41	5,912	2,936	1,362
Honduras	SSU	0	5	0.70	1.05	0.83	1.00	0.95	0.07	0.00	0.00	0.00	0.00	0	3,824	1,870	1,313	0	3,824	1,735	1,406	0	4,235	1,803	1,392
Nicaragua	SSU	0	18	1.24	2.07	0.75	1.00	0.98	0.07	0.00	0.00	0.00	0.00	0	3,884	2,652	1,221	0	3,884	2,408	1,395	0	4,519	2,534	1,333
Panamá	SSU	0	7	0.90	1.18	0.00	1.00	0.57	0.50	0.00	1.00	0.26	0.44	0	3,441	1,379	866	0	3,441	1,257	944	0	3,598	1,311	801
Paraguay	SSU	0	18	2.95	3.13	1.96	2.00	2.00	0.01	0.00	0.00	0.00	0.00	2,308	10,235	7,426	2,585	0	10,235	6,744	3,233	20	10,526	7,110	2,953
Perú	SSU	0	46	7.03	7.12	2.00	5.00	3.03	1.03	0.00	2.00	0.61	0.88	0	2,111	85	415	0	0	0	0	0	1,061	3	57
República Dominicana	SSU	0	33	4.14	4.91	1.00	3.00	1.76	0.65	0.00	1.00	0.09	0.28	0	412	17	81	0	0	0	0	0	207	1	11
Uruguay	SSU	0	8	0.68	1.06	0.00	1.00	0.50	0.45	0.00	0.00	0.00	0.00	720	4,167	2,171	1,045	0	4,167	1,968	1,193	2	4,581	2,078	1,170
Venezuela	SSU	0	21	1.38	2.70	0.00	2.00	1.33	0.77	0.00	0.00	0.00	0.00	0	6,024	1,712	1,690	0	6,024	1,715	1,692	0	7,334	1,698	1,793
Argentina	Field	0	100	19.97	18.03	27.25	37.05	32.91	2.62	0.00	6.25	3.65	1.36	0	0	0	0	0	0	0	0	0	0	0	0
Bolivia	Field	0	33	5.47	5.18	7.00	10.00	8.05	1.09	0.00	2.00	1.06	0.89	0	0	0	0	0	0	0	0	0	0	0	0
Chile	Field	0	36	7.08	6.90	4.00	9.00	5.24	1.80	0.00	3.00	0.87	0.99	0	0	0	0	0	0	0	0	0	0	0	0
Costa Rica	Field	0	30	1.81	3.07	1.00	3.00	2.44	0.75	0.00	2.00	0.52	0.65	0	0	0	0	0	0	0	0	0	0	0	0
Ecuador	Field	0	40	7.07	7.55	8.00	12.00	9.80	1.34	0.00	4.00	2.05	1.45	0	0	0	0	0	0	0	0	0	0	0	0
El Salvador	Field	0	36	3.36	4.09	3.00	5.00	3.97	0.96	0.00	2.00	0.52	0.78	0	0	0	0	0	0	0	0	0	0	0	0
Guatemala	Field	0	90	6.01	10.37	6.00	13.00	9.06	2.32	0.00	4.00	2.17	1.78	0	0	0	0	0	0	0	0	0	0	0	0
Honduras	Field	0	17	3.03	3.21	2.11	5.79	4.46	1.19	0.00	3.00	0.94	1.07	0	0	0	0	0	0	0	0	0	0	0	0
Nicaragua	Field	0	49	2.40	5.10	1.00	5.00	2.52	1.38	0.00	5.00	1.57	1.49	0	0	0	0	0	0	0	0	0	0	0	0
Panamá	Field	0	26	1.82	2.95	2.00	3.00	2.35	0.48	0.00	2.00	0.44	0.65	0	0	0	0	0	0	0	0	0	0	0	0
Paraguay	Field	0	29	3.03	4.26	3.00	5.00	3.74	0.59	0.00	1.00	0.09	0.29	0	0	0	0	0	0	0	0	0	0	0	0
Perú	Field	0	174	16.09	23.09	14.00	25.00	19.15	3.76	0.00	15.00	5.21	4.84	0	0	0	0	0	0	0	0	0	0	0	0
República Dominicana	Field	0	44	5.18	6.39	4.00	7.00	5.07	0.98	0.00	4.00	1.64	1.81	0	0	0	0	0	0	0	0	0	0	0	0
Uruguay	Field	0	12	1.62	2.29	1.00	3.00	2.35	0.63	0.00	1.00	0.26	0.44	0	0	0	0	0	0	0	0	0	0	0	0
Venezuela	Field	0	62	3.04	7.02	2.00	7.00	5.42	1.57	0.00	2.00	0.69	0.85	0	0	0	0	0	0	0	0	0	0	0	0
Argentina	Must Have	0	8	1.49	1.71	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0	0	0	0	0	0
Bolivia	Must Have	0	3	0.13	0.41	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0	0	0	0	0	0
Chile	Must Have	0	10	0.84	1.39	0.00	5.00	3.72	1.82	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0	0	0	0	0	0
Costa Rica	Must Have	0	38	0.42	2.21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0	0	0	0	0	0
Ecuador	Must Have	0	21	0.82	1.88	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0	0	0	0	0	0
El Salvador	Must Have	0	40	0.57	3.13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0	0	0	0	0	0
Guatemala	Must Have	0	37	0.69	2.53	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0	0	0	0	0	0
Honduras	Must Have	0	3	0.13	0.37	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0	0	0	0	0	0
Nicaragua	Must Have	0	4	0.11	0.45	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0	0	0	0	0	0
Panamá	Must Have	0	13	0.09	0.81	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0	0	0	0	0	0
Paraguay	Must Have	0	24	0.24	1.60	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0	0	0	0	0	0
Perú	Must Have	0	19	1.07	2.25	0.00	2.00	1.66	0.75	0.00	1.00	0.18	0.38	0	0	0	0	0	0	0	0	0	0	0	0
República Dominicana	Must Have	0	3	0.03	0.22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0	0	0	0	0	0
Uruguay	Must Have	0	2	0.05	0.22	0.00	1.00	0.57	0.50	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0	0	0	0	0	0
Venezuela	Must Have	0	10	0.25	0.89	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0	0	0	0	0	0

Figura 35 - Resumen estadístico por variable, mercado y canal

6. Metodología

La metodología consiste en iniciar realizando un *feature engineering* para ajustar los distintos predictores y adecuarlos mejor al modelo. Sobre todo, los *features* de tipo fecha pueden ser usados para obtener varios *features* nuevos (como día de la semana, mes del año, etc). También el *one-hot-encoding* es necesario para las variables categóricas.

Luego del *feature engineering*, se pasa a la prueba de distintos modelos. Como tenemos 15 Country_Names 3 AccountSources posibles, la combinación entre estos es de 45 valores únicos. Como se recoge, de la literatura, que cada país/mercado puede tener su propia particularidad, se decide, en un principio, entrenar, un modelo para cada uno de estas 45 combinaciones de Country_Names-AccountSources posibles de modo que el resultado final sea el que mejor se ajusta a cada realidad y no un modelo tan general que no se adapte a cada mercado.

Los modelos probados son los que siguen:

ARIMA: Este es un modelo univariado tradicional en el análisis de series de tiempo. Fue desarrollado por Box y Jenkins, con el objeto de estimar modelos de series

temporales, usando exclusivamente el dato de los valores pasados de la variable a predecir. Esta metodología es especialmente útil para economistas que desean predecir el comportamiento de variables económicas sin tener que especificar modelos basados en teorías subyacentes (De Arce y Mahía, 2003).

Este modelo trabaja sobre el supuesto de que los datos son estacionarios. La estacionariedad para una variable se cumple si:

- Su esperanza matemática es invariante en el tiempo.
- Su varianza tampoco depende del tiempo.
- La covarianza entre dos variables aleatorias en distintos momentos del tiempo solo depende del lapso de tiempo que entre ellas ha transcurrido

(De Arce y Mahía, 2003).

Una vez cumplido el supuesto de estacionariedad, se puede implementar el modelo ARIMA, el cual tiene en su nombre iniciales en inglés para:

- AR = *Autoregressive*: Los modelos AR explican la variable endógena de un período utilizando sus valores pasados y un término de error (De Arce y Mahía, 2003). Esto quiere decir que la misma variable que queremos predecir, (en nuestro caso, Acquisitions), se explica a sí misma, con cierto desfase en el tiempo, es decir, la variable "regresa" sobre sus propios rezagos. En resumen, se usan valores pasados de Acquisitions para predecir sus valores futuros. El hiperparámetro p significa qué orden de componente autorregresiva tiene el modelo (cuál es la longitud de rezago que se utiliza para predecir una observación nueva).
- I = *Integrated*: como se ha comentado, para poder realizar un pronóstico, la serie de tiempo que se toma como *input* debe ser estacionaria, es decir, con propiedades estadísticas (media, varianza, autocorrelación) constantes a lo largo del tiempo (De Arce y Mahía, 2003). Si la serie no lo es, entonces se debe realizar una diferenciación, que consiste en restar, a cada valor de la variable a predecir (Acquisitions, en nuestro caso), el valor de la misma variable en una observación anterior. Esto implicaría una diferenciación de orden 1, aunque las hay de mayor orden (cada uno implicando restar valores de observaciones más desfasadas en el tiempo). El hiperparámetro d corresponde a la diferenciación: será 0 si no hay, 1 si se diferencia una vez, 2 en caso de dos veces, y así sucesivamente.
- MA = *Moving Average*: Los modelos MA explican la variable endógena en función de errores pasados. Este componente le da aleatoriedad a la predicción (De Arce y Mahía, 2003). La medida móvil implica regresar a nuestra variable

de interés en función de shocks pasados (errores del pasado). A la vez, por ser móvil, implica que los eventos más alejados en el pasado dejan de tener influencia en el presente, pues se toman las "x" últimas observaciones, evitando así que un suceso demasiado alejado siga influyendo en nuestro pronóstico. El hiperparámetro q corresponde al orden de la media móvil: refleja la longitud de rezago de errores de pronósticos anteriores que se incluyen para calcular el valor actual.

SARIMA:

El modelo ARIMA está diseñado para datos no estacionales. Para lidiar con datos de este tipo, Boy y Jenkins generalizaron el modelo original, incluyendo el concepto de estacionalidad (Adhikari y Agrawal, 2013). De este modo, el modelo es similar al ARIMA mencionado, pero agrega un componente estacional (S = *Seasonal*). Que haya estacionalidad quiere decir que hay ciertos patrones de comportamiento de la variable a predecir que se repiten en el tiempo. Por ejemplo, dado que se ha observado en el análisis exploratorio como hay cierto ciclo en el comportamiento de Acquisitions a lo largo de cada semana, se podría inferir que hay una estacionalidad semanal. SARIMA hace uso de ese dato y lo incorpora en sus pronósticos. Además de los hiperparámetros mencionados en ARIMA, se le añaden P,D,Q (que son los análogos a p,d,q pero estacionales) y s, que representa la periodicidad de la estacionalidad (mensual, semanal, anual, etc.)

ETS:

El suavizamiento exponencial es un modelo que se originó en el trabajo de Robert G. Brown durante la Segunda Guerra Mundial. Brown desarrolló modelos de rastreo para el control de fuego en la Marina de los EE. UU., y más tarde extendió sus métodos para incorporar tendencias y estacionalidades (Gardner, 1985).

Este enfoque estadístico univariado permite pronosticar una serie temporal, descomponiendo dicha serie en función de su error (E), tendencia (T) y estacionalidad (S). Por lo tanto, ETS es un algoritmo que describe una serie a partir de su:

- E = Error: este error puede ser aditivo (los errores se suman de forma constante en el tiempo) o multiplicativo (se multiplican los errores por un factor, penalizando aún más).
- T = *Trend*: se considera la tendencia de la serie de tiempo, es decir, su comportamiento a lo largo del tiempo (sin considerar estacionalidad). Esta tendencia puede ser modelada de forma aditiva o multiplicativa también (para los casos con cada vez más variaciones a lo largo del tiempo).
- S = *Seasonality*: este componente considera la estacionalidad: patrones en cierto intervalo de tiempo regular. La estacionalidad también se puede modelar de forma aditiva o multiplicativa.

La taxonomía de estos modelos es útil para describir el método que se usa en casa instancia. Cada método se indica con una letra para el error, otra para la tendencia y una tercera para la estacionalidad. De este modo, a partir de las distintas combinaciones de error, tendencia y estacionalidad, podemos obtener diferentes modelos, por ejemplo: N-N-N: que no tiene ni error, ni tendencia ni estacionalidad, M-A-N, con error multiplicativo, tendencia aditiva, sin estacionalidad, M-N-A, con error multiplicativo, sin tendencia, estacionalidad aditiva, entre tantas otras posibles combinaciones. Dependiendo de cada caso particular, un método u otro será el que mejor se adapte a la realidad de los datos de estudio. (Gardner, 1985).

A lo largo del paper, se menciona cómo los métodos como ETS son útiles para pronosticar demanda, lo cual es determinante para costos de inventario, planificación de recursos humanos y económicos, así como de nivel de servicio (Gardner, 1985).

Prophet:

Prophet es un modelo desarrollado por Facebook y es muy utilizado en series temporales con tendencias y estacionalidades. Es muy popular en el ámbito empresarial (Taylor y Letham, 2017).

En el paper oficial de Prophet, Taylor y Letham (2017) plantean el problema del pronóstico a gran escala, fundamental para la planificación de capacidad, establecimiento de metas y detección de anomalías en las organizaciones. Destacan, asimismo, los desafíos de producir pronósticos fiables y de alta calidad, especialmente cuando se manejan múltiples series temporales y hay escasez de expertos en modelado de series temporales.

En especial, se hace hincapié en la particularidad de las series temporales empresariales, las cuales suelen tener múltiples estacionalidades fuertes, cambios de tendencia, valores atípicos y efectos de días festivos. En este contexto y para atender a este tipo de problema, es que se elabora el modelo Prophet (Taylor y Letham, 2017).

Este modelo tiene un especial foco en tres componentes:

- Tendencia: se capta la tendencia de la variable en cuestión, con la capacidad de detectar crecimientos no lineales. También el modelo atiende a los puntos de cambio, que son aquellos momentos donde la tendencia deja de ser una y pasa a ser otra. Esta selección de los puntos de cambios es automática y también incluye una simulación de futuros cambios de tasa de tendencia, para estimar este tipo de incertidumbre de la variable a predecir en el futuro.
- Estacionalidad: Se modelan los efectos estacionales mediante series de Fourier, lo que permite capturar estacionalidades semanales y anuales. Los

parámetros estacionales se ajustan con un enfoque de suavizamiento para evitar el sobreajuste.
- Días festivos: se agrega la posibilidad de proveer una base de datos con los días feriados de modo que funcionen como un *feature* más en el modelo. Es esperable, por ejemplo, en ventas, que un día feriado se venda menos que un día laborable común de semana. Prophet capta muy bien este tipo de comportamientos.

(Taylor y Letham, 2017).

Prophet tiene varios hiperparámetros que se pueden utilizar y ajustar para captar distintos aspectos de la serie de tiempo, como por ejemplo:

- *changepoints*: cambios de tendencia
- *seasonality mode*: ¿qué tipo de estacionalidad hay: semanal, mensual, anual…?
- *holidays*: para incluir el componente de días festivos.

XGBoost (*Extreme Gradient Boosting*): este modelo es ampliamente usado en distintos ámbitos y casos, por lo que no se restringe a su uso para series de tiempo, sino que también es usado en muchos otros ámbitos, como por ejemplo la detección de fraudes (Chen y Guestrin, 2016).

El algoritmo consiste en la realización de un ensamble de árboles de decisión: crea un árbol de decisión para realizar la predicción, luego crea un árbol para mejorar el rendimiento del primer árbol allí donde la predicción no fue satisfactoria y así, sucesivamente, ensamblando árboles hasta llegar a un modelo general mucho más potente que lo que hubiese sido el árbol original. Lo que el algoritmo busca hacer al "mejorar el rendimiento" de cada árbol ensamblado es minimizar una función de pérdida, como por ejemplo, el MSE (Chen y Guestrin, 2016).

Otra de las características de XGBoost es que tiene un algoritmo *sparsity-aware*, el cual permite trabajar sin problemas con datos con una gran cantidad de valores nulos o en cero (Chen y Guestrin, 2016). Esto lo hace ideal para nuestro caso, dado que existen canales de adquisición, como el *Must Have*, que a menudo presenta valores de 0 en sus adquisiciones.

Asimismo, XGBoost utiliza una regularización que ayuda a evitar el overfitting, haciendo que los modelos sean más generalizables y robustos. Este punto es esencial para resaltar cómo XGBoost no solo busca precisión en el entrenamiento, sino también un buen rendimiento en datos no vistos, como los datos del conjunto de prueba (Chen y Guestrin, 2016).

El éxito de XGBoost en competiciones como Kaggle y KDDCup 2015 confirma la capacidad de este modelo para proporcionar soluciones eficientes y eficaces en diversos ámbitos de aplicación (Chen & Guestrin, 2016).

Este modelo requiere que se le pasen como entrada variables numéricas, por lo que variables del tipo texto deben pasar por un proceso de *one-hot-encoding*.

XGBoost tiene distintos hiperparámetros que se pueden ajustar para mejorar el rendimiento del modelo. Entre ellos, destacamos:

- eta: se trata de la tasa de aprendizaje. A menor tasa de aprendizaje, cada árbol aprende menos y lleva a necesitar más rondas de ensamble para poder llegar al objetivo. Esto puede mejorar el rendimiento, bajo riesgo de sobreajuste.
- max_depth: es la máxima profundidad de los árboles. Al establecer un número más bajo, se limita qué tan profundos pueden ser y se evita, así, el sobreajuste.
- nrounds: es el número de árboles que se construirán para ser ensamblados de forma secuencial. A más árboles, mayor ajuste a los datos, debido a que cada árbol tratará de corregir el error de los anteriores ensamblados.
- gamma: de este hiperparámetro depende si un nodo del árbol se divide basado en la reducción esperada de la función de pérdida que el modelo intenta minimizar.

Light GBM (*Light Gradient Boosting Machine*): es otro modelo de ensamble, caracterizado por ser muy eficiente y veloz en términos de cómputo, aún cuando se manejan gran cantidad de datos. Al igual que XGBoost, intenta minimizar una función de pérdida ensamblando varios modelos para corregir errores de manera secuencial. Este modelo se destaca por su capacidad de manejar muchas dimensiones, es decir, una gran cantidad de variables explicativas de una variable a predecir (Ke et al., 2017).

Se trata de un modelo de conjunto de árboles de decisión entrenados secuencialmente, donde cada árbol se ajusta a los errores residuales del modelo anterior (similar a XGBoost). Este modelo selecciona todas las instancias con gradientes grandes y realiza un muestreo aleatorio de aquellas con gradientes pequeños. Esto sirve para lograr un modelo que siga siendo preciso, pero eficiente en términos de cómputo, ya que mediante el muestreo aleatorio, reduce la cantidad de datos a procesar (Ke et al., 2017).

Por otro lado, el algoritmo agrupa características mutuamente exclusivas para reducir el número de características efectivas. Esto sirve para poder utilizar de manera eficiente variables categóricas (no numéricas) sin la necesidad de realizar un *one-hot-encoding* previo, como sí ocurre con XGBoost (Ke et al., 2017).

Algunos hiperparámetros a destacar de este modelo son:

- num_iterations: es el número máximo de itrraciones o árboles que se construirán para entrenar el modelo. Al construir más árboles, se logra un mayor ajuste a los datos.
- max_depth: regula la máxima profundidad de cada árbol (como en XGBoost).
- learning_rate: funciona de forma similar a la tasa de aprendizaje de XGBoost.
- num_leaves: implica el número de hojas de cada árbol. A más hojas, hay mayor división de cada nodo, lo cual puede mejorar la precisión, bajo riesgo de sobreajuste.
- feature_fraction: el porcentaje de *features* tomados aleatoriamente para ser usados en cada árbol entrenado.
- bagging_fraction: el porcentaje de observaciones tomadas en cada árbol para ser entrenado.

En el caso de XGBoost y Light GBM, además de entrenar un modelo para cada una de las 45 combinaciones de mercado-canal, también se prueban modelos pool, es decir, un modelo único para explicar los 45 mercados-canales a la vez, utilizando Country_Name y AccountSource como una variable explicativa más que se le pasa al modelo.

Los modelos ARIMA, SARIMA, ETS y Prophet asumen normalidad en los residuos, mientras que XGBoost y ETS no asumen ninguna distribución. Si prestamos atención a la distribución de los datos, en el histograma de Adquisiciones presentado en la Figura 5, notamos cómo la cola larga en este gráfico evidencia una distribución claramente similar a una exponencial. Esto, a priori, nos anticipa la posibilidad de un mejor ajuste de parte de los modelos de boosting, en comparación con el resto.

En términos analíticos, podemos considerar una ecuación que representa la estimación que se busca hacer con la presente tesis, por medio de los distintos modelos, donde nuestra variable “y” a predecir es Acquisitions y tenemos varios “x” o variables explicativas. A continuación, la ecuación pertinente a cada modelo probado:

ARIMA:

$$y'_t = c + \phi_1 y'_{t-1} + \dots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

Donde:

- y'_t es el valor que toma la variable Acquisitions en un momento t, luego de ser diferenciado (la cantidad de veces que corresponda según el hiperparámetro d).
- $\phi_1 \ , \dots , \phi_p$ son los p parámetros del modelo, relativos al componente autorregresivo, aprendidos durante el entrenamiento.

- ε_t es el término de error, el componente aleatorio que el modelo no es capaz de explicar o captar para el momento t.
- θ_1 , … , θ_q son los q parámetros del modelo, relativos al componente de media móvil, aprendidos durante el entrenamiento.
- c es el valor que toma la variable Acquisitions en el momento t, cuando todos sus rezagos usados para el pronóstico son 0, los componentes de media móvil del error son 0 y el error del modelo también es 0.

SARIMA:

$y'_t = c + \phi_1 y'_{t-1} + \dots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q} + \Phi_1 y'_{t-P} + \dots + \Phi_P y'_{t-Ps} + \Theta_1 \varepsilon_{t-Q} + \dots + \Theta_Q \varepsilon_{t-Qs} + \varepsilon_t$

Donde:

- y'_t es el valor que toma la variable Acquisitions en un momento t, luego de ser diferenciado (la cantidad de veces que corresponda según el hiperparámetro d) y también tras la diferenciación estacional (según el hiperparámetro D).
- ϕ_1 , … , ϕ_p son los p parámetros del modelo, relativos al componente autorregresivo no estacional, aprendidos durante el entrenamiento.
- ε_t es el término de error, el componente aleatorio que el modelo no es capaz de explicar o captar para el momento t.
- θ_1 , … , θ_q son los q parámetros del modelo, relativos al componente de media móvil no estacional, aprendidos durante el entrenamiento.
- Φ_1 , … , Φ_P son los P parámetros del modelo, relativos al componente autorregresivo estacional, aprendidos durante el entrenamiento.
- Θ_1 , … , Θ_Q son los Q parámetros del modelo, relativos al componente de media móvil estacional, aprendidos durante el entrenamiento.
- c es el valor que toma la variable Acquisitions en el momento t, cuando todos sus rezagos usados para el pronóstico son 0, los componentes de media móvil del error son 0 y el error del modelo también es 0.

ETS:

$y_t = l_t + b_t + s_t + \varepsilon_t$

Donde:

- y_t es el valor que toma la variable Acquisitions en un momento t.

- l_t es el nivel del componente correspondiente a la tendencia de la serie, en el momento t.
- b_t es el componente de tendencia, la cual puede ser aditiva / multiplicativa, en el momento t.
- s_t es el componente estacional en el momento t.
- ε_t es el término de error, el componente aleatorio que el modelo no es capaz de explicar o captar para el momento t.

Prophet:

$y(t) = g(t) + s(t) + h(t) + \varepsilon_t$

Donde:

- y(t) es el valor que toma la variable Acquisitions predicha.
- g(t) es el componente de tendencia del modelo, que capta los cambios no periódicos de la serie.
- s(t) es el componente de estacionalidad del modelo, que capta los cambios periódicos (diarios, semanales, mensuales, anuales).
- h(t) es el componente que capta los efectos de los días feriados.
- ε_t es el término de error, el componente aleatorio que el modelo no es capaz de explicar o captar para el momento t.

XGBoost y Light GBM:

Acquisitions = f (day_of_week*, day_of_month, is_last_day, week_of_year, last_week_of_month, first_week_of_month, Seller_Head_Count_exp, Seller_Head_Count_new, is_holiday, is_holiday_uruguay, Marketing_Cost_This_Month, Marketing_Cost_Last_Month, Marketing_Cost_Last_30_days) + ε

* sometida a un *one-hot-encoding* en el caso de XGBoost (no en el caso de Light GBM).

Donde:

- Acquisitions es la variable “y” a predecir.
- f es la función aprendida por el modelo correspondiente, la cual representa cómo se comporta la variable en función de sus *features.*
- ε es el término de error, el componente aleatorio que el modelo no es capaz de explicar o captar.

En cada modelo, podemos probar grillas de hiperparámetros para obtener la mejor combinación de los mismos. Finalmente, evaluamos en un conjunto de prueba la

perfomance de los distintos modelos para quedarnos con el que mejor explica el comportamiento de las adquisiciones.

Por último, con el *output* de las predicciones del modelo elegido ya en mano, podemos armar una regla que determina, dependiendo de la cantidad de adquisiciones esperadas en cada canal y Country_Name, para cada mes, la cantidad de personal de distinto tipo que se necesita.

Tratándose de una serie de tiempo, los datos deben ser tratados respetando su orden temporal. Decidimos trabajar con las siguientes ventanas temporales:

- Conjunto de entrenamiento: 80% de los registros (del 1 de enero de 2023 al 5 de octubre de 2023, inclusive)
- Conjunto de prueba: 20% de los registros (del 6 de octubre de 2023 al 14 de diciembre de 2023, inclusive)
- Ventana de predicción: Mes corriente + mes subsiguiente (o solo mes corriente, dependiendo del día en que uno se para para realizar la predicción. En el caso del ejemplo proporcionado en la sección 10, se trata de una ventana que va desde el 1 de febrero de 2024 al 29 de febrero de 2024).

La medida de rendimiento que utilizamos para evaluar y comparar modelos es el Error Cuadrático Medio (MSE, por sus siglas en inglés), métrica que permite registrar errores por arriba y por debajo de lo predicho y además le da un mayor peso a los errores grandes, debido a elevar al cuadrado las diferencias.

7. Desarrollo

7.1. Modelos por mercado-canal

En la presente sección, todas las referencias a MSE y comparativas entre datos reales y pronosticados, son relativos a la ventana de tiempo correspondiente al conjunto de validación.

7.1.1. ARIMA

Comenzamos probando con un modelo ARIMA, como *benchmark*. Debemos determinar los valores de los hiperparámetros p,d,q. La combinación ideal de estos hiperparámetros, será encontrada con el comando auto.arima() de R.

Este comando es un algoritmo que busca obtener el modelo ARIMA de mejor ajuste dentro de la muestra siguiendo un *step-wise search*.

Primero, se asegura de que la serie de tiempo sea estacionaria. Como se comentó anteriormente, para poder realizar pronósticos con ARIMA es necesario que la serie de tiempo pasada como *input* sea estacionaria. De lo contrario, es necesario una

diferenciación. Entonces, mediante un test de estacionariedad (KPSS) el algoritmo determina el número de diferenciaciones necesarias (entre 0 y 2) para poder convertir la serie en estacionaria. Esto, entonces, determina el hiperparámetro d.

Una vez determinado d, se busca la mejor combinación de p y q. En lugar de probar todas las combinaciones posibles, lo cual sería altamente ineficiente (más bien un proceso infinito), se parte de cuatro modelos básicos, evaluando cuál es el de menor AICc (mejor ajuste). Una vez elegido, se realizan ciertas iteraciones variando los valores de p y q de este modelo en +/-1 y comprobando si mejora el ajuste, hasta que se halle que no hay mejora, en cuyo caso el modelo iterado es el elegido. Para más detalles sobre el algoritmo de selección automática de modelos ARIMA en R ver el trabajo de Hyndman y Khandakar (2008).

Al correr el modelo, llegamos a los siguientes resultados:

Country_Name	AccountSource	p	d	q	MSE
Argentina	SSU	1	1	4	338.01
Bolivia	SSU	1	1	1	10.51
Chile	SSU	1	0	5	82.84
Costa Rica	SSU	0	0	1	2.33
Ecuador	SSU	1	0	0	27.44
El Salvador	SSU	0	0	1	3.57
Guatemala	SSU	0	1	2	4.43
Honduras	SSU	0	0	0	1.59
Nicaragua	SSU	0	0	1	3.49
Panamá	SSU	0	1	1	1.31
Paraguay	SSU	0	1	3	9.31
Perú	SSU	1	0	2	56.16
República Dominicana	SSU	0	1	2	40.50
Uruguay	SSU	0	0	1	0.90
Venezuela	SSU	0	0	2	3.60
Argentina	Field	2	0	1	365.84
Bolivia	Field	0	0	1	22.41
Chile	Field	0	0	2	37.49
Costa Rica	Field	1	1	1	3.69
Ecuador	Field	0	0	1	32.21
El Salvador	Field	1	0	3	9.44
Guatemala	Field	2	0	1	45.50
Honduras	Field	0	0	1	4.97
Nicaragua	Field	1	0	2	6.31
Panamá	Field	2	0	2	5.07
Paraguay	Field	1	0	0	15.39
Perú	Field	1	0	0	365.66
República Dominicana	Field	0	0	1	22.74
Uruguay	Field	0	0	1	7.57
Venezuela	Field	1	0	1	43.61
Argentina	Must Have	3	0	1	2.79
Bolivia	Must Have	0	0	1	0.20
Chile	Must Have	1	0	2	1.14
Costa Rica	Must Have	0	0	0	1.40
Ecuador	Must Have	1	0	2	0.86
El Salvador	Must Have	0	1	2	22.89
Guatemala	Must Have	0	0	0	0.90
Honduras	Must Have	0	0	0	0.06
Nicaragua	Must Have	0	1	2	0.06
Panamá	Must Have	0	0	0	0.00
Paraguay	Must Have	0	0	0	8.30
Perú	Must Have	1	1	3	1.14
República Dominicana	Must Have	0	0	1	0.01
Uruguay	Must Have	0	0	0	0.04
Venezuela	Must Have	3	0	0	0.03

Figura 36 - ARIMA: MSE por país y canal (en conjunto de validación)

Al mismo tiempo, se menciona que el MSE a nivel global, es decir, tomando de la base de datos como un todo, es de 35.86. Se nota un alto MSE en Argentina *Field* y *SSU* y en Perú *Field*, en contraposición al resto de los mercados/canales.

Aclaración de criterio: tanto en este modelo como en cualquier otro probado en la presente tesis, toda predicción negativa de adquisiciones es truncada con un 0 y cualquier número con decimales será redondeado, para solo tener valores enteros y mayores o iguales a 0 en la variable predicha.

Al graficar las series temporales reales (azul) junto con el pronóstico (rojo), para el subconjunto de prueba, observamos un patrón repetido en prácticamente todos los modelos: la curva del pronóstico es casi horizontal, es decir, se predice siempre la misma cantidad para todos los días. Nos damos cuenta, entonces, que los datos predichos por el modelo no tienen casi variabilidad (el modelo no se ajusta mucho a los datos, quedando lejos de cada observación).

A modo de ejemplo, mostramos el caso de Argentina-*Field* (Figura 37), aunque esto se repite en todos los modelos:

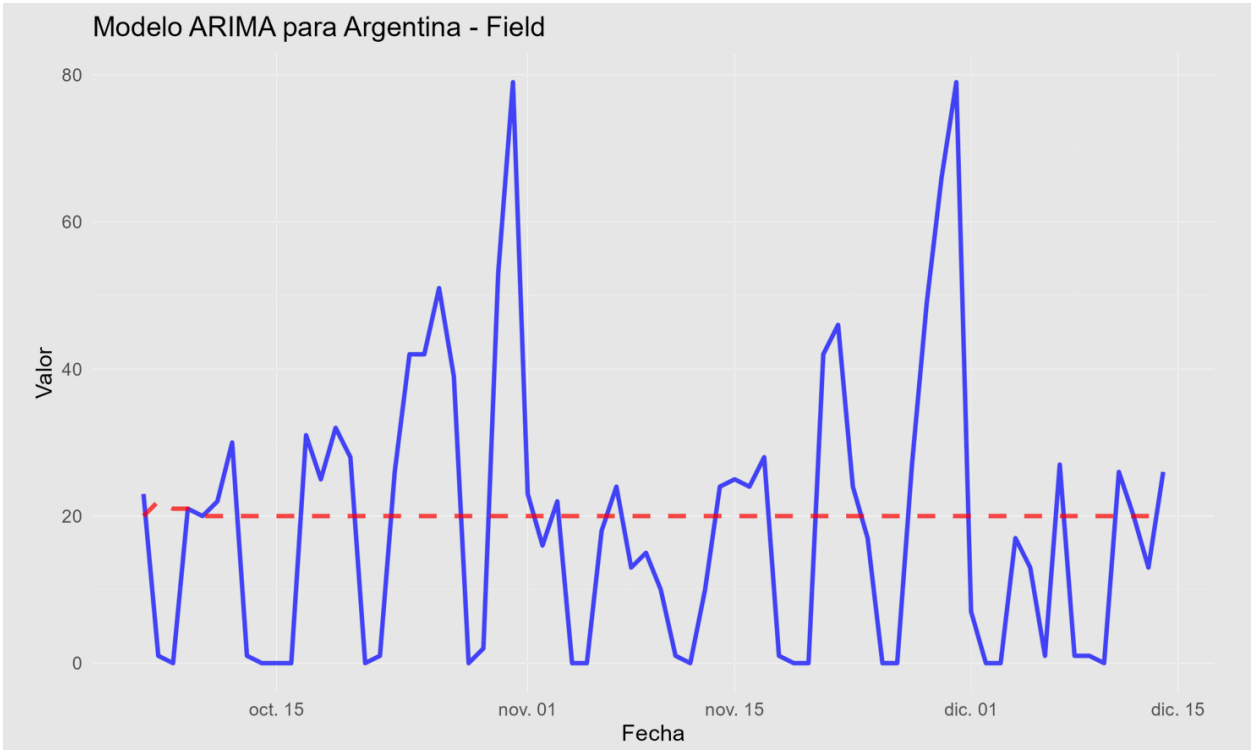

Figura 37 - ARIMA: Ajuste del modelo (rojo) a los datos reales (azul) en Argentina Field (en conjunto de validación)

Si regresamos al análisis exploratorio, recordamos que, en general, Argentina y Perú-*Field* tienen un nivel de adquisiciones superior al resto. Intuimos, entonces, que el mayor MSE en los mercados/canales mencionados no es por un peor rendimiento

comparado con los demás casos, sino simplemente por la naturaleza de estos mercados y canales. Al trabajar con cifras mayores, el error en cada observación también es mayor. Más aún si nuestra medida de rendimiento es el MSE, el cual penaliza más los errores en números grandes, elevándolos al cuadrado.

Por lo tanto, llegamos a la conclusión de que el modelo en sí ajusta igual de deficientemente en todos los casos y no es un problema de un mercado o un canal en particular.

7.1.2. SARIMA

Debido al poco ajuste del modelo y la poca variabilidad en las predicciones, decidimos probar un modelo que trabaja con series de tiempo autorregresivas, pero agregue un componente de estacionalidad: SARIMA. Este tipo de modelo es similar a ARIMA, pero permite agregar una componente de estacionalidad, en este caso, semanal. Al hacer la prueba, comprobamos que los resultados son mucho mejores:

Country_Name	AccountSource	MSE
Argentina	SSU	223.19
Bolivia	SSU	6.69
Chile	SSU	36.74
Costa Rica	SSU	2.23
Ecuador	SSU	23.36
El Salvador	SSU	2.46
Guatemala	SSU	3.90
Honduras	SSU	1.30
Nicaragua	SSU	2.83
Panamá	SSU	1.14
Paraguay	SSU	7.97
Perú	SSU	37.89
República Dominicana	SSU	33.24
Uruguay	SSU	0.70
Venezuela	SSU	2.71
Argentina	Field	224.87
Bolivia	Field	16.10
Chile	Field	24.49
Costa Rica	Field	3.50
Ecuador	Field	23.90
El Salvador	Field	7.77
Guatemala	Field	48.37
Honduras	Field	5.09
Nicaragua	Field	5.03
Panamá	Field	4.66
Paraguay	Field	13.81
Perú	Field	348.04
República Dominicana	Field	26.20
Uruguay	Field	5.01
Venezuela	Field	54.37
Argentina	Must Have	1.60
Bolivia	Must Have	0.20
Chile	Must Have	1.11
Costa Rica	Must Have	1.46
Ecuador	Must Have	0.36
El Salvador	Must Have	22.90
Guatemala	Must Have	0.99
Honduras	Must Have	0.06
Nicaragua	Must Have	0.06
Panamá	Must Have	0.00
Paraguay	Must Have	8.30
Perú	Must Have	1.06
República Dominicana	Must Have	0.01
Uruguay	Must Have	0.04
Venezuela	Must Have	0.03

Figura 38 - SARIMA: MSE por país y canal (en conjunto de validación)

Se observa cómo, en general, los MSE son mejores que el ARIMA básico. Por otro lado, el MSE a nivel general también es superador: 27.46.

Además, comprobando los gráficos de las series de tiempo, vemos cómo ahora sí las predicciones tienen variabilidad y se logra percibir la estacionalidad semanal en las mismas:

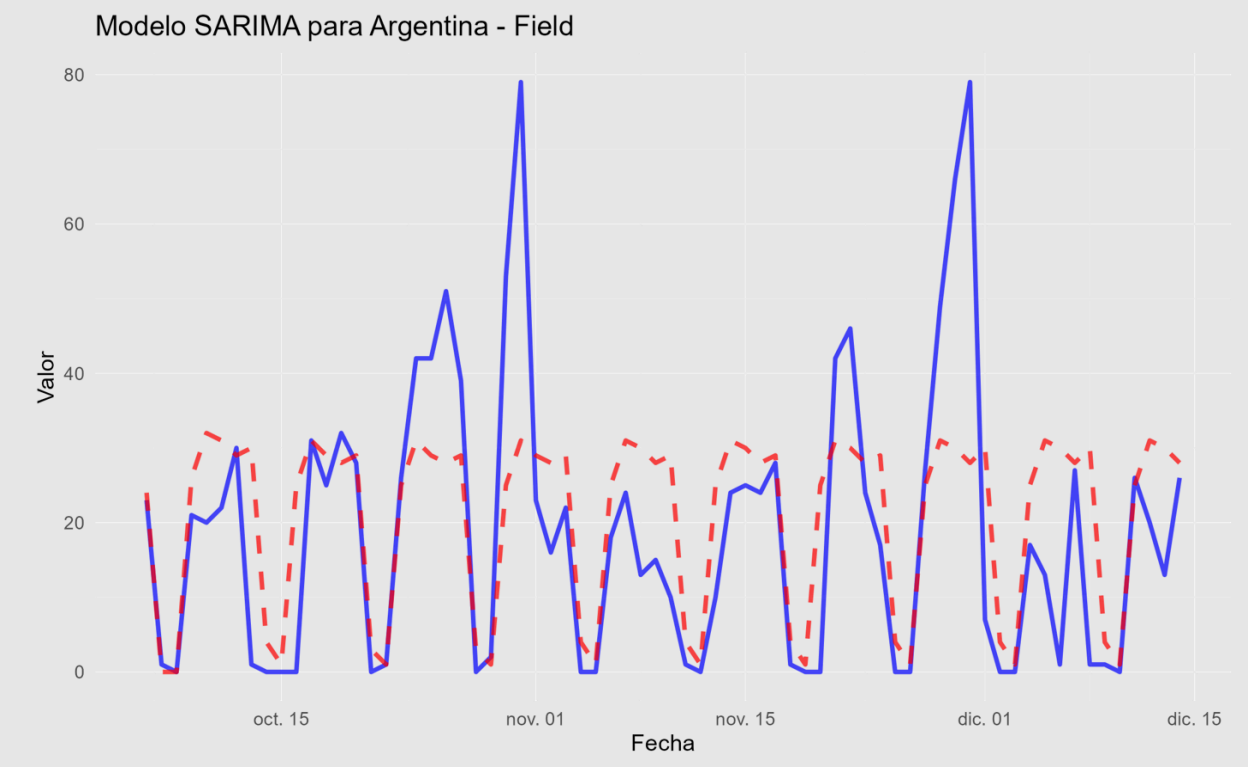

Figura 39 - SARIMA: Ajuste del modelo (rojo) a los datos reales (azul) en Argentina Field (en conjunto de validación)

7.1.3. ETS

Se prueba un modelo ETS (algoritmo de suavizado exponencial), con el objeto de testear otra estrategia univariada. Para el entrenamiento del mismo, se le pasa al algoritmo una serie de tiempo diaria, buscando que el modelo detecte la estacionalidad semanal de los datos.

Al observar los resultados en el conjunto de validación, percibimos cómo la estacionalidad semanal es detectada, aunque, igual que SARIMA, los picos de Acquisitions a final de mes no son correctamente captados.

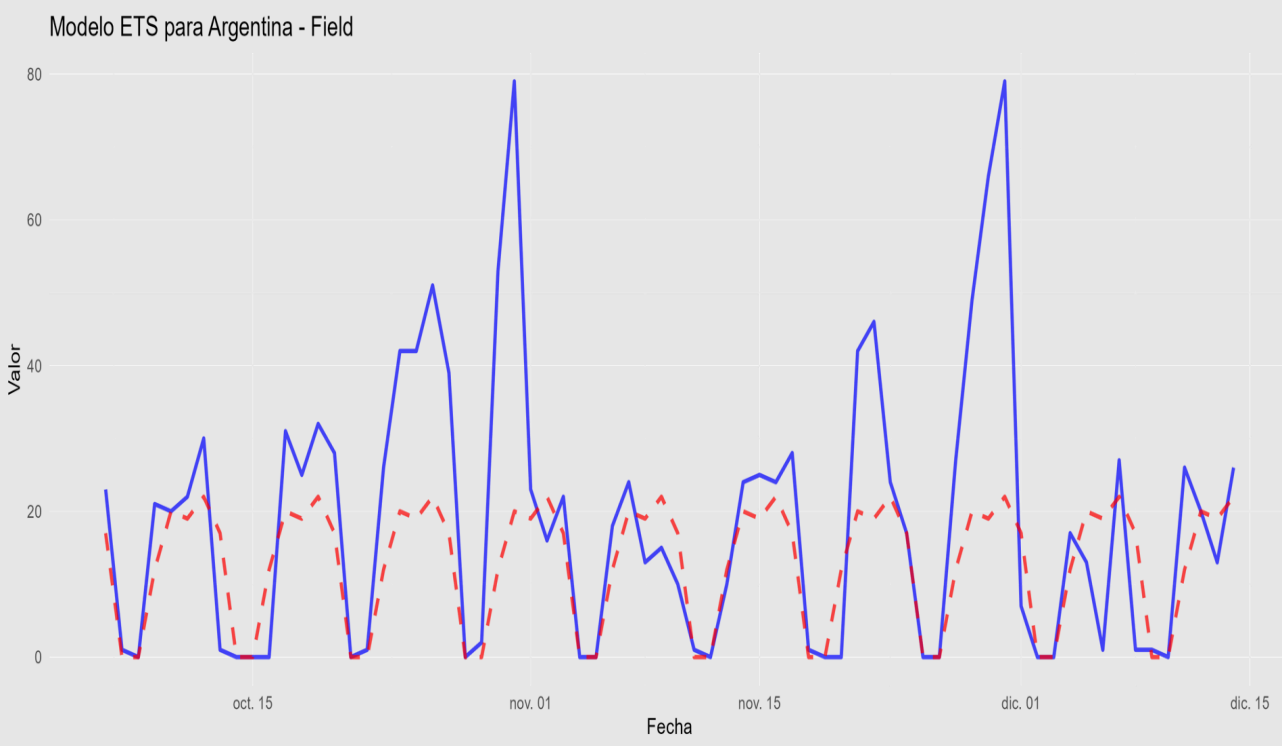

Figura 40 - ETS: Ajuste del modelo (rojo) a los datos reales (azul) en Argentina Field (en conjunto de validación)

Los resultados en cada mercado-canal, si bien aceptables, no son mejores que los obtenidos con SARIMA, y, a nivel global, el MSE también es levemente peor: 29.68 (Figura 41).

Country_Name	AccountSource	MSE
Argentina	SSU	208.14
Bolivia	SSU	5.71
Chile	SSU	42.74
Costa Rica	SSU	2.27
Ecuador	SSU	30.86
El Salvador	SSU	2.83
Guatemala	SSU	3.79
Honduras	SSU	1.39
Nicaragua	SSU	2.97
Panamá	SSU	1.17
Paraguay	SSU	14.36
Perú	SSU	42.93
República Dominicana	SSU	49.53
Uruguay	SSU	1.04
Venezuela	SSU	1.47
Argentina	Field	257.56
Bolivia	Field	19.49
Chile	Field	25.30
Costa Rica	Field	3.74
Ecuador	Field	26.29
El Salvador	Field	9.76
Guatemala	Field	59.99
Honduras	Field	3.74
Nicaragua	Field	4.91
Panamá	Field	4.83
Paraguay	Field	17.33
Perú	Field	369.31
República Dominicana	Field	29.23
Uruguay	Field	5.67
Venezuela	Field	46.09
Argentina	Must Have	2.09
Bolivia	Must Have	0.20
Chile	Must Have	1.57
Costa Rica	Must Have	2.97
Ecuador	Must Have	0.19
El Salvador	Must Have	23.17
Guatemala	Must Have	1.56
Honduras	Must Have	0.14
Nicaragua	Must Have	0.06
Panamá	Must Have	0.00
Paraguay	Must Have	8.30
Perú	Must Have	1.29
República Dominicana	Must Have	0.01
Uruguay	Must Have	0.04
Venezuela	Must Have	0.03

Figura 41 - ETS: MSE por país y canal (en conjunto de validación)

7.1.4. Prophet

7.1.4.1. *Prophet Inicial*

Siguiendo el mismo enfoque de un modelo por cada mercado/canal, procedemos a probar ahora con una estrategia Prophet. Este tipo de modelo es conocido por su capacidad de pronóstico de series de tiempo, la cual detecta componentes de estacionalidad.

Al hacer una primera aproximación al modelo, obtenemos los siguientes resultados:

Country_Name	AccountSource	MSE
Argentina	SSU	218.06
Bolivia	SSU	6.51
Chile	SSU	37.31
Costa Rica	SSU	2.44
Ecuador	SSU	24.30
El Salvador	SSU	2.46
Guatemala	SSU	3.91
Honduras	SSU	1.30
Nicaragua	SSU	2.83
Panamá	SSU	1.17
Paraguay	SSU	7.80
Perú	SSU	38.23
República Dominicana	SSU	33.17
Uruguay	SSU	0.70
Venezuela	SSU	2.81
Argentina	Field	225.03
Bolivia	Field	16.74
Chile	Field	26.13
Costa Rica	Field	3.44
Ecuador	Field	25.59
El Salvador	Field	7.24
Guatemala	Field	48.07
Honduras	Field	6.00
Nicaragua	Field	5.03
Panamá	Field	4.46
Paraguay	Field	12.40
Perú	Field	366.71
República Dominicana	Field	24.59
Uruguay	Field	5.33
Venezuela	Field	53.17
Argentina	Must Have	1.60
Bolivia	Must Have	0.20
Chile	Must Have	1.69
Costa Rica	Must Have	1.34
Ecuador	Must Have	0.17
El Salvador	Must Have	22.97
Guatemala	Must Have	0.90
Honduras	Must Have	0.06
Nicaragua	Must Have	0.06
Panamá	Must Have	0.00
Paraguay	Must Have	8.30
Perú	Must Have	1.10
República Dominicana	Must Have	0.01
Uruguay	Must Have	0.04
Venezuela	Must Have	0.03

Figura 42 - Prophet: MSE por país y canal (en conjunto de validación)

En niveles generales, observamos un MSE inferior al obtenido con la estrategia ARIMA. A su vez, el MSE global del conjunto de prueba es menor que el ARIMA (pero mayor que el SARIMA: 27.81). Se repite el mayor MSE en casos en los que la cantidad de adquisiciones es superior al resto.

Ahora bien, al prestar atención a los gráficos, se puede percibir una gran diferencia con respecto a ARIMA: el modelo ajusta mucho más: la variabilidad de los datos predichos es mucho mayor, ya no mostrándose como una línea horizontal estable, sino como una curva con sus altos y bajos. Al mismo tiempo, observamos que dicha curva en muchas ocasiones se adapta bastante bien a la curva real (azul), por más que en algunos momentos se separen. A modo de ejemplo, se incluye el gráfico de Argentina - *Field* (Figura 43), pero este patrón se repite en general en todos los mercados - canales.

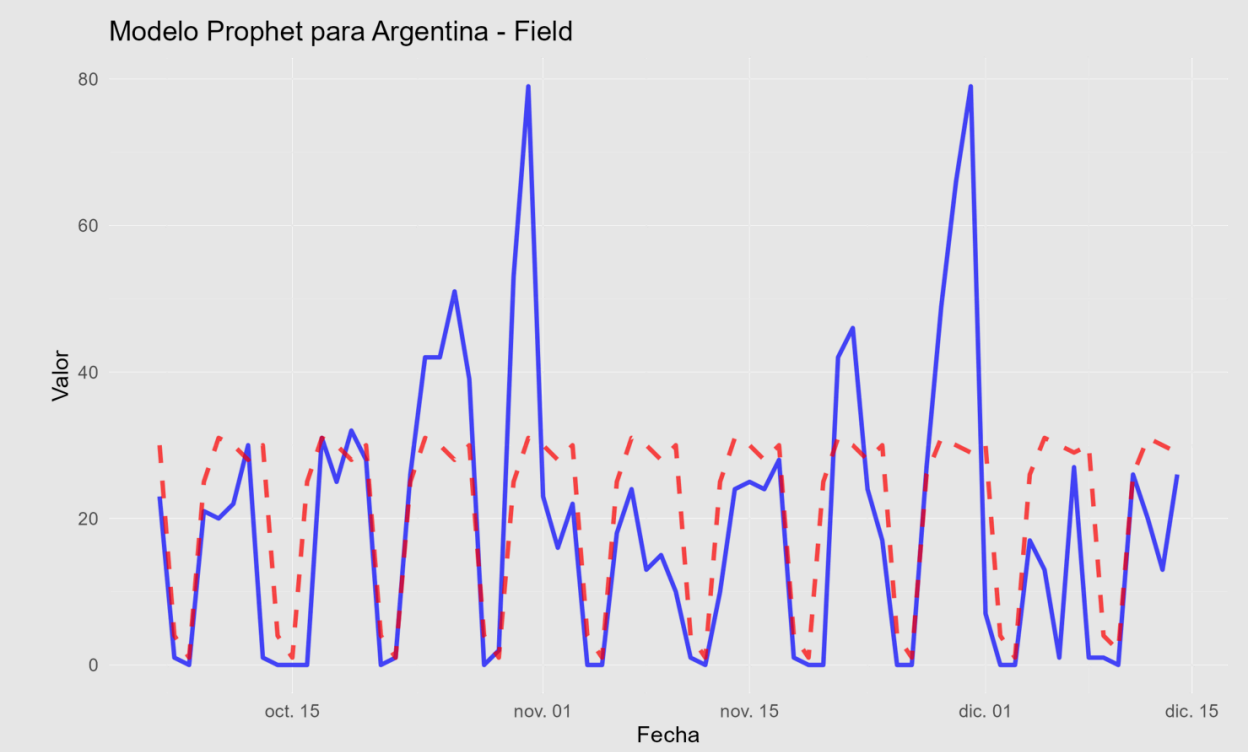

Figura 43 - Prophet: Ajuste del modelo (rojo) a los datos reales (azul) en Argentina Field (en conjunto de validación)

Con respecto al ajuste, apuntamos dos cuestiones:

- La estacionalidad semanal es muy bien captada por el modelo, comenzando el Lunes con una subida en las adquisiciones vs el fin de semana, alcanzando dos picos entre mitad de semana y viernes, y volviendo a caer hacia el fin de semana.
- La estacionalidad mensual no está siendo captada a la perfección: observamos cómo la curva roja (modelo) alcanza máximos más o menos estables a lo largo

del mes, mientras que la azul (real) muestra unos picos mucho más altos en las últimas semanas del mes vs el resto.

Repetimos este abordaje, pero incluyendo ahora la variable is_holiday para que Prophet haga uso de ella. A la vez, especificamos 2 estacionalidades para que estén activas: *daily* y *weekly*. La mejora es mínima: el MSE global pasa de 27.81 a 27.78. Sigue faltando ese elemento que capte el salto en las adquisiciones propio de la última semana (o últimos días) del mes.

7.1.4.2. *Prophet con estacionalidad mensual*

Debido a lo notado en los modelos iniciales de Prophet, se investigó la posibilidad de aplicar este mismo modelo que tan bien ajustaba a estacionalidades semanales, pero agregando una estacionalidad mensual, que permita observar ese salto en las adquisiciones hacia final de mes. Luego de una búsqueda, se llegó a la forma de poder agregar esta estacionalidad, sin perder el resto de la configuración del modelo que se estaba usando. Al hacerlo, obtenemos los siguientes resultados:

Country_Name	AccountSource	MSE
Argentina	SSU	165.17
Bolivia	SSU	5.86
Chile	SSU	39.59
Costa Rica	SSU	1.83
Ecuador	SSU	19.63
El Salvador	SSU	2.34
Guatemala	SSU	3.29
Honduras	SSU	1.34
Nicaragua	SSU	2.53
Panamá	SSU	1.20
Paraguay	SSU	6.53
Perú	SSU	29.01
República Dominicana	SSU	26.93
Uruguay	SSU	0.83
Venezuela	SSU	3.64
Argentina	Field	154.84
Bolivia	Field	12.76
Chile	Field	24.86
Costa Rica	Field	3.71
Ecuador	Field	17.39
El Salvador	Field	6.59
Guatemala	Field	40.06
Honduras	Field	5.84
Nicaragua	Field	7.74
Panamá	Field	3.90
Paraguay	Field	7.57
Perú	Field	262.40
República Dominicana	Field	17.80
Uruguay	Field	4.99
Venezuela	Field	52.50
Argentina	Must Have	1.76
Bolivia	Must Have	0.20
Chile	Must Have	1.39
Costa Rica	Must Have	1.51
Ecuador	Must Have	0.23
El Salvador	Must Have	23.13
Guatemala	Must Have	0.90
Honduras	Must Have	0.06
Nicaragua	Must Have	0.06
Panamá	Must Have	0.00
Paraguay	Must Have	8.31
Perú	Must Have	1.00
República Dominicana	Must Have	0.01
Uruguay	Must Have	0.04
Venezuela	Must Have	0.10

Figura 44 - Prophet con estacionalidad mensual: MSE por país y canal (en conjunto de validación)

Se observa cómo en general, baja el MSE en casi todos los casos. Sobre todo se nota la mejoría en Argentina, que de 218-225 pasa a 165-154. Perú *Field* 262 ahora, vs 366 antes. A nivel general, el MSE también mejora, pasando de 27.78 en la primera estrategia Prophet a 21.59 con el agregado de la estacionalidad mensual.

Al prestar atención a los gráficos (Figuras 45, 46 y 47), se percibe cómo, si bien en ocasiones el salto real de adquisiciones en fin de mes sigue siendo superior al estimado, la diferencia es ahora mucho menor:

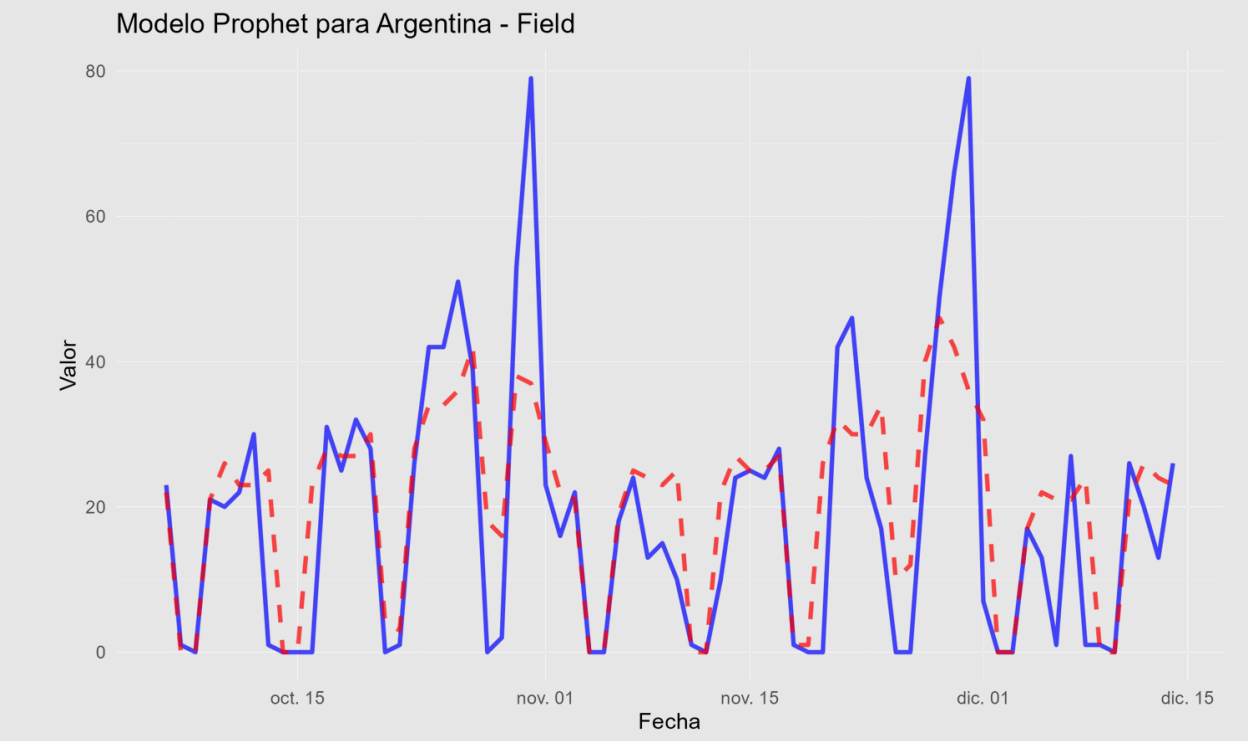

Figura 45 - Prophet con estacionalidad mensual: Ajuste del modelo (rojo) a los datos reales (azul) en Argentina Field (en conjunto de validación)

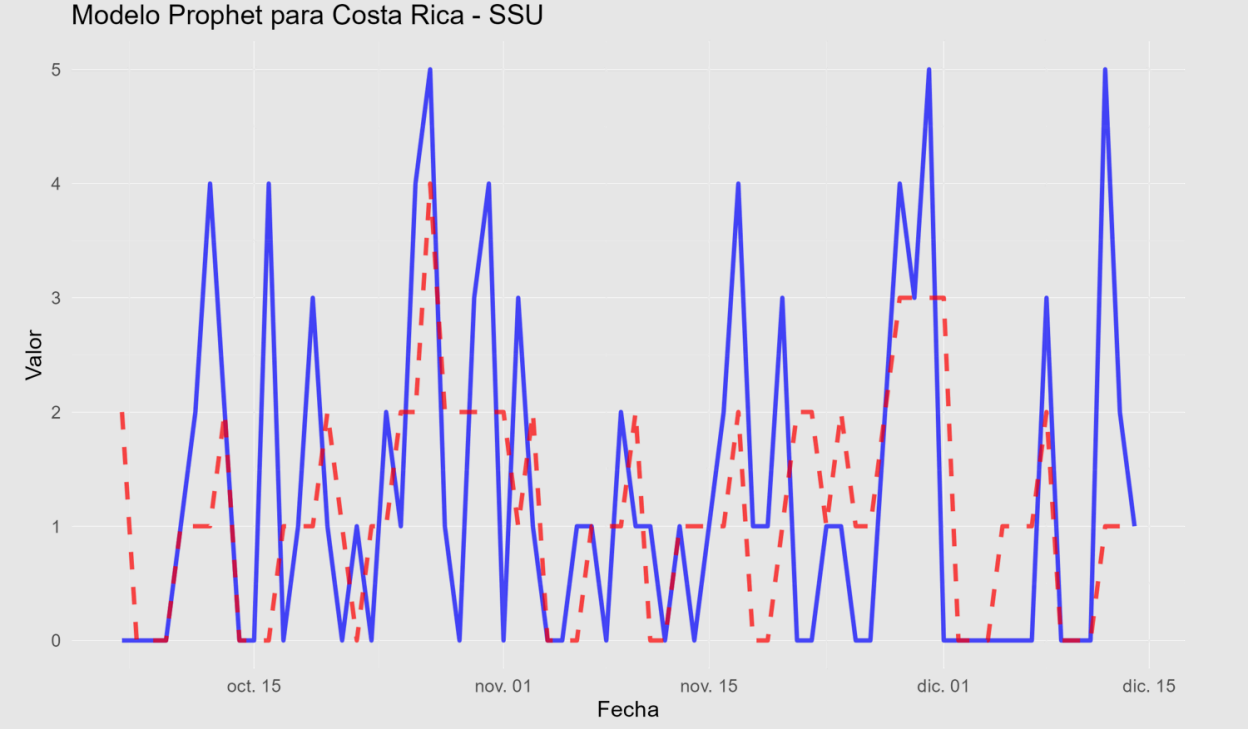

Figura 46 - Prophet con estacionalidad mensual: Ajuste del modelo (rojo) a los datos reales (azul) en Costa Rica SSU (en conjunto de validación)

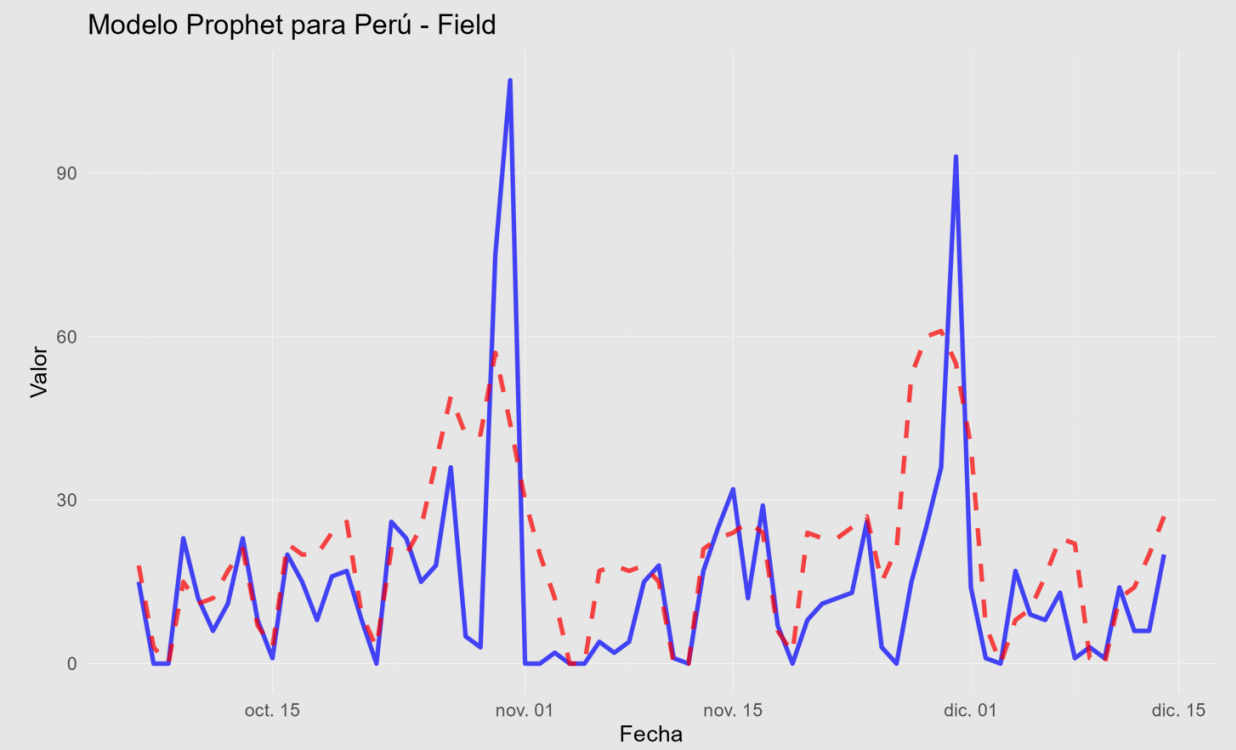

Figura 47 - Prophet con estacionalidad mensual: Ajuste del modelo (rojo) a los datos reales (azul) en Perú Field (en conjunto de validación)

7.1.4.3. *Prophet con grilla de hiperparámetros*

Por otro lado, se prueba una grilla de hiperparámetros, en la cual tuneamos los siguientes:

- Holidays.prior.scale
- Changepoint.prior.scale
- Seasonality.prior.scale
- Seasonality.mode

Para ello, conservamos el mismo conjunto de prueba correspondiente al último 20% de los datos (del 6 de octubre de 2023 al 14 de diciembre de 2023). Sin embargo, del otro 80%, dividimos los datos de modo tal que obtenemos un conjunto de entrenamiento de 70% de la base total (del 1 de enero de 2023 al 31 de agosto de 2023) y un conjunto de validación de 10% de la misma (del 1 de septiembre de 2023 al 14 de diciembre de 2023). Los entrenamientos de cada combinación de la grilla son realizados sobre el conjunto de entrenamiento, probando el MSE en el conjunto de validación y quedándonos con el set de hiperparámetros de mejor rendimiento. Posteriormente, se utiliza dicho set de hiperparámetros para entrenar el conjunto de entrenamiento + validación y así estimar los parámetros del modelo. Finalmente, se calcula el MSE en el conjunto de prueba. Todo este análisis se hace a nivel mercado-canal, esto quiere decir que se prueban distintas combinaciones de hiperparámetros y se elige la de mejor rendimiento para cada mercado-canal, entrenando luego los 45 modelos con los hiperparámetros óptimos de cada combinación y obteniendo 45 modelos distintos, con diferentes parámetros estimados, según el país y el canal de adquisición. Este mismo procedimiento, con los mismos puntos de corte, ventanas temporales y lógica de granularidad en el entrenamiento de modelos, son los usados en la estrategia de XGBoost con grilla de hiperparámetros (sección 7.1.5.3). El MSE, en el caso de Prophet con grilla, es de 24.46, por lo tanto, sigue siendo mejor el resultado obtenido en el modelo Prophet simple con estacionalidad mensual.

7.1.5. XGBoost

Probamos, ahora, con una estrategia multivariada de boosting: XGBoost.

Para ello, hacemos un *feature engineering* previo, con el objeto de pasarle al modelo *features* más significativas, sobre todo en el caso de la fecha.

7.1.5.1. *Feature Engineering*

Probamos añadiendo/ajustando los siguientes *features* para enriquecer el modelo con información valiosa:

- day_of_week: se toma, a partir de Acquisition_Date, el día de la semana que corresponde (Lunes, Martes, Miércoles, etc). Esto sirve para aportar estacionalidad semanal.
- day_of_month: se realiza el mismo procedimiento para el día del mes (del 1 al 31), con el objeto de aportar a la estacionalidad mensual.
- is_last_day: se incluye este *feature* binario que vale 1 si se trata del último día del mes y 0 en caso contrario (estacionalidad mensual).
- week_of_year: para añadir una componente de estacionalidad anual, se le pasa al modelo el número de semana del año (del 1 al 52).
- last_week_of_month y first_week_of_month: son variables binarias que toman el valor 1 en caso de tratarse de la última/primera semana del mes y 0 en caso contrario. Esto es para dar cuenta del comportamiento de las adquisiciones durante el mes, las cuales suelen tener una disparada en la última semana, mientras que en la primera su cantidad suele ser más baja.

Además, se realiza *one-hot-encoding* de la variable day_of_week, de modo de poderle pasar al modelo *features* numéricos.

7.1.5.2. *Rendimiento XGBoost*

Probando, entonces, el modelo de XGBoost, e incluyendo todos los *features* mencionados anteriormente, obtenemos los siguientes resultados:

Country_Name	AccountSource	MSE
Argentina	SSU	273.30
Bolivia	SSU	4.70
Chile	SSU	34.29
Costa Rica	SSU	2.17
Ecuador	SSU	15.27
El Salvador	SSU	3.06
Guatemala	SSU	2.94
Honduras	SSU	1.51
Nicaragua	SSU	3.46
Panamá	SSU	1.74
Paraguay	SSU	9.66
Perú	SSU	22.03
República Dominicana	SSU	17.93
Uruguay	SSU	1.40
Venezuela	SSU	6.47
Argentina	Field	96.19
Bolivia	Field	14.69
Chile	Field	16.64
Costa Rica	Field	5.00
Ecuador	Field	32.69
El Salvador	Field	25.30
Guatemala	Field	69.46
Honduras	Field	6.46
Nicaragua	Field	16.90
Panamá	Field	8.76
Paraguay	Field	6.71
Perú	Field	115.04
República Dominicana	Field	13.13
Uruguay	Field	6.11
Venezuela	Field	27.29
Argentina	Must Have	2.84
Bolivia	Must Have	0.23
Chile	Must Have	0.96
Costa Rica	Must Have	1.47
Ecuador	Must Have	0.40
El Salvador	Must Have	22.94
Guatemala	Must Have	10.79
Honduras	Must Have	0.21
Nicaragua	Must Have	0.09
Panamá	Must Have	0.00
Paraguay	Must Have	11.11
Perú	Must Have	2.27
República Dominicana	Must Have	0.01
Uruguay	Must Have	0.10
Venezuela	Must Have	0.03

Figura 48 - XGBoost: MSE por país y canal (en conjunto de validación)

La mejoría es notable. Argentina en sus dos canales y Perú *Field*, siguen siendo los casos de mayor MSE, pero son menores que con las estrategias anteriormente probadas. Se destaca la gran mejoría en el caso de Argentina *Field*, que pasa de un 154.84 en Prophet a tan solo un 96.19 con XGBoost. Si bien el rendimiento en Argentina *SSU* es levemente inferior, el de Perú *Field* es superador: 115.04 vs 262.4. Además, hay una mejora en términos generales, dado que el MSE global pasa de 21.59 a 20.31.

Por otro lado, se percibe cómo, en algunos casos, se logra detectar la subida de adquisiciones hacia fin de mes y poder proyectar un nivel más ajustado a la realidad (Figura 49). Esto probablemente se deba a la inclusión de los *features* de end_of_month y last_week_of_month mencionados.

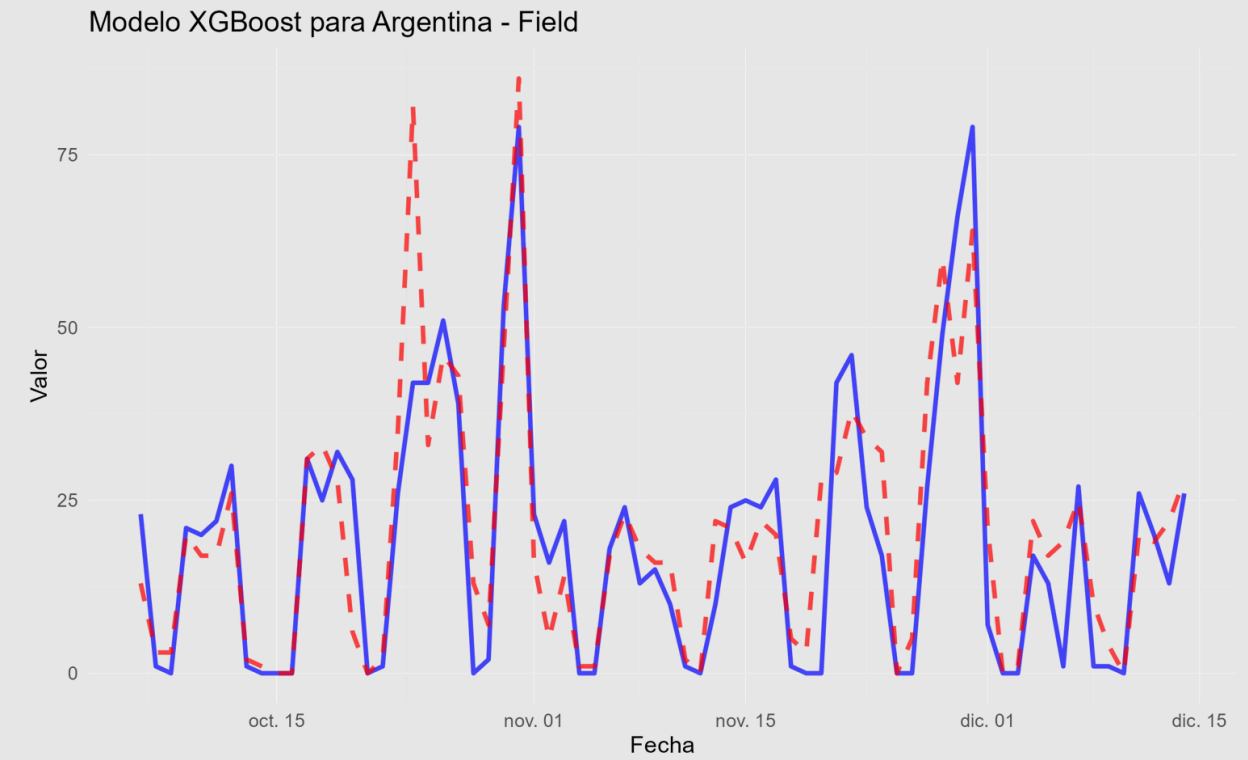

Figura 49 - XGBoost: Ajuste del modelo (rojo) a los datos reales (azul) en Argentina Field (en conjunto de validación)

7.1.5.3. *XGBoost: grilla de hiperparámetros*

Probamos una grilla de hiperparámetros, con la intención de mejorar el rendimiento de este modelo. En la misma, introducimos distintos valores para:

- nrounds
- max_depth
- eta
- gamma

Las ventanas temporales usadas son las mismas que en 7.1.4.3 Prophet con grilla de hiperparámetros. Nos quedamos con la combinación de estos cuatro hiperparámetros que mejora el MSE en la muestra de validación, para luego entrenar un modelo en el conjunto de entrenamiento + validación con dichos hiperparámetros. Igual que se hizo en el caso de Prophet con grilla de hiperparámetros, se eligen los hiperparámetros óptimos para cada mercado-canal y luego se aprenden los parámetros del modelo, mediante un entrenamiento también separado para cada mercado-canal (dando como resultado 45 modelos diferentes). Finalmente, probamos su rendimiento en el conjunto de prueba y observamos una mejora en términos generales:

Country_Name	AccountSource	MSE
Argentina	SSU	216.56
Bolivia	SSU	5.70
Chile	SSU	31.70
Costa Rica	SSU	1.96
Ecuador	SSU	16.23
El Salvador	SSU	3.30
Guatemala	SSU	2.93
Honduras	SSU	1.40
Nicaragua	SSU	4.19
Panamá	SSU	1.53
Paraguay	SSU	6.93
Perú	SSU	37.13
República Dominicana	SSU	16.60
Uruguay	SSU	1.19
Venezuela	SSU	2.80
Argentina	Field	79.44
Bolivia	Field	8.47
Chile	Field	15.46
Costa Rica	Field	3.47
Ecuador	Field	31.60
El Salvador	Field	29.30
Guatemala	Field	46.49
Honduras	Field	3.59
Nicaragua	Field	7.33
Panamá	Field	6.20
Paraguay	Field	5.14
Perú	Field	130.27
República Dominicana	Field	12.01
Uruguay	Field	5.44
Venezuela	Field	26.90
Argentina	Must Have	1.77
Bolivia	Must Have	0.20
Chile	Must Have	1.01
Costa Rica	Must Have	1.40
Ecuador	Must Have	0.24
El Salvador	Must Have	22.89
Guatemala	Must Have	0.81
Honduras	Must Have	0.20
Nicaragua	Must Have	0.07
Panamá	Must Have	0.00
Paraguay	Must Have	8.36
Perú	Must Have	1.00
República Dominicana	Must Have	0.01
Uruguay	Must Have	0.06
Venezuela	Must Have	0.03

Figura 50 - XGBoost con grilla de hiperparámetros: MSE por país y canal (en conjunto de validación)

Apreciamos cómo, en general, los mercados-canales presentan un MSE inferior y, si bien sigue superior al resto, Argentina-*SSU* baja de 273.3 a 216.56. Notable, también, es la mejora de Argentina-*Field*: de 96.19 a 79.44.

Además, a nivel global, el MSE pasa de 20.31 a 17.77, bajando drásticamente. Sin dudas, este modelo es mejor que el inicialmente planteado.

A continuación, un ejemplo del buen ajuste del modelo en general, representado por Argentina-*Field*:

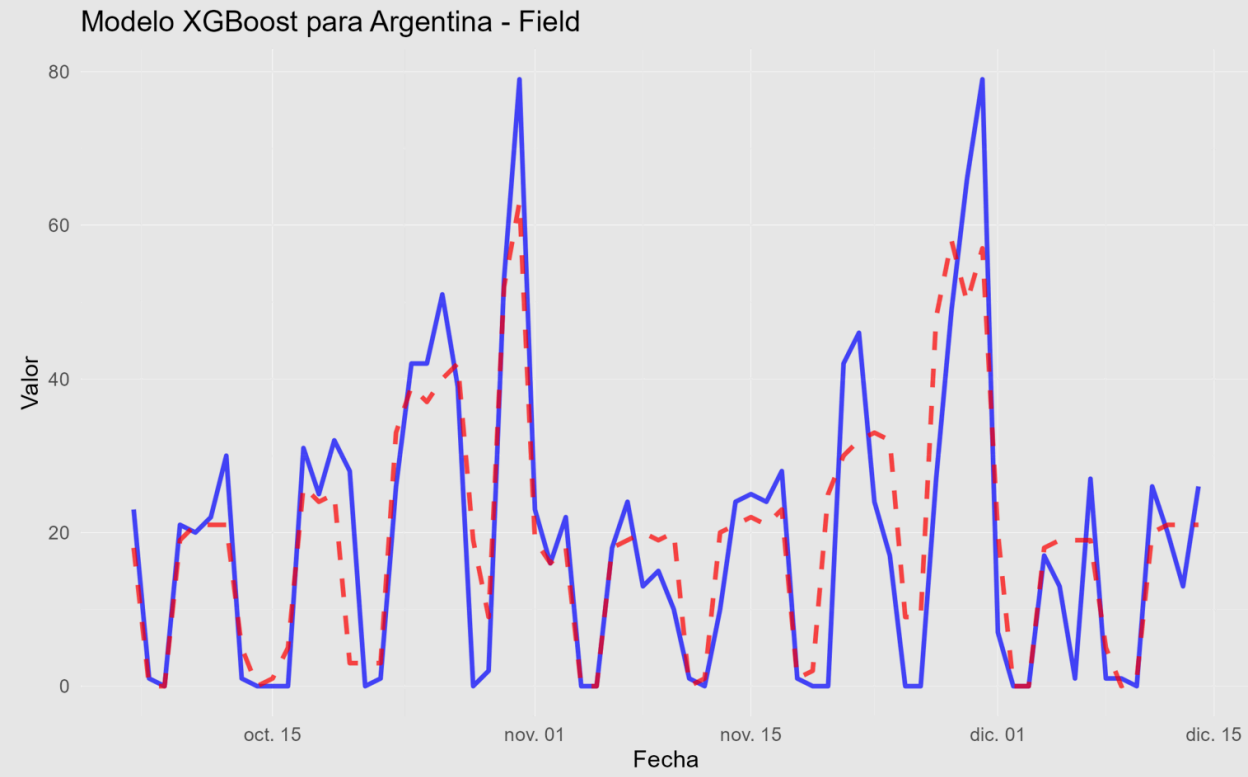

Figura 51 - XGBoost con grilla de hiperparámetros: Ajuste del modelo (rojo) a los datos reales (azul) en Argentina Field (en conjunto de validación)

7.1.5.4. *XGBoost: agregado de rezagos y medias móviles*

Con el objeto de captar aún mejor el fenómeno tendencia/estacionalidad, es que probamos el siguiente modelo. El mismo consiste en el mismo XGBoost original probado en la sección 7.1.5.2., con la diferencia de que se le agregan, como *features* adicionales:

- Valores de rezago de Acquisitions: se agregan siete *features* que consisten en la variable Acquisitions rezagada de uno a siete días. De este modo, se busca captar la tendencia que viene siguiendo la variable a predecir. Para el conjunto de entrenamiento, la variable rezagada se calcula conociendo los valores de la misma variable en los días inmediatamente anteriores. En cambio, en el caso

del conjunto de prueba, dado que el algoritmo “no conoce” el valor de la variable Acquisitions, se utilizan las mismas predicciones como valores rezagados de Acquisitions en una fecha posterior. De este modo, lo pronosticado, por ejemplo, para el día 1 de diciembre de 2023 es el rezago -1 que se usa como *feature* al pronosticar Acquisitions para el día 2 de diciembre de 2023. Esto se realiza con el objeto de no utilizar data conocida de la variable a predecir y evitar el *data leakage*.

- Valores de rezago de Marketing_Cost_Last_30_days: se decide que esta variable, la cual también varía día a día, sea sometida a un proceso de generación de siete rezagos, similar a como se hace con Acquisitions. La diferencia aquí radica en que, como esta no es la variable a predecir, se puede simplemente usar sus siete rezagos partiendo de la base original, sin tener que utilizar un pronóstico como rezago de un registro con fecha posterior.
- Valores de medias móviles de Acquisitions: para la observación de cada día, se toman los valores de Acquisitions de los siete días inmediatamente anteriores y se calcula su promedio. Este valor es el que sirve como nuevo *feature* para el modelo. En el caso del conjunto de prueba, lo que se utilizan para calculas las medias móviles son los valores de las predicciones, para así evitar el *data leakage*.
- Valores de medias móviles de Marketing_Cost_Last_30_days: para la observación de cada día, se toman los valores de Marketing_Cost_Last_30_days de los siete días inmediatamente anteriores y se calcula su promedio. Este valor es el que sirve como nuevo *feature* para el modelo.

Resulta prudente realizar algunas aclaraciones. La primera tiene que ver con la pérdida de datos. Al implementar siete rezagos diarios y medias móviles, los primeros siete días del conjunto de datos (para cada mercado-canal) no tendrán su valor en las nuevas *features* agregadas. Por lo tanto, nos vemos forzados a simplemente eliminar del conjunto de datos de entrenamiento estos registros. No habiendo abundancia de datos, esto impacta en el tamaño del conjunto de entrenamiento.

Por otro lado, cabe mencionar que las variables elegidas son estas dos (Acquisitions y Marketing_Cost_Last_30_days) porque son aquellas variables numéricas que varían día a día y cuya tendencia consideramos que vale la pena captar. Variables como Seller_Head_Count_exp, entre otras, no varían de un día a otro del mes (en cada mes se tiene el mismo valor) por lo que realizar el trabajo de rezagos y/o medias móviles sobre ellas no tendría sentido.

Al entrenar el modelo y observar sus resultados sobre el conjunto de prueba obtenemos:

Country_Name	AccountSource	MSE
Argentina	SSU	364.41
Bolivia	SSU	5.30
Chile	SSU	33.17
Costa Rica	SSU	2.46
Ecuador	SSU	34.69
El Salvador	SSU	3.01
Guatemala	SSU	5.14
Honduras	SSU	1.24
Nicaragua	SSU	3.44
Panamá	SSU	1.37
Paraguay	SSU	7.33
Perú	SSU	29.97
República Dominicana	SSU	27.90
Uruguay	SSU	0.99
Venezuela	SSU	5.93
Argentina	Field	65.26
Bolivia	Field	12.94
Chile	Field	30.53
Costa Rica	Field	7.50
Ecuador	Field	30.67
El Salvador	Field	13.61
Guatemala	Field	49.87
Honduras	Field	6.86
Nicaragua	Field	18.26
Panamá	Field	6.21
Paraguay	Field	6.34
Perú	Field	147.57
República Dominicana	Field	23.57
Uruguay	Field	5.89
Venezuela	Field	30.77
Argentina	Must Have	2.51
Bolivia	Must Have	0.20
Chile	Must Have	1.16
Costa Rica	Must Have	165.31
Ecuador	Must Have	0.40
El Salvador	Must Have	22.89
Guatemala	Must Have	6.90
Honduras	Must Have	0.27
Nicaragua	Must Have	0.11
Panamá	Must Have	0.00
Paraguay	Must Have	8.31
Perú	Must Have	2.17
República Dominicana	Must Have	0.01
Uruguay	Must Have	0.09
Venezuela	Must Have	0.11

Figura 52 - XGBoost con rezagos y medias móviles: MSE por país y canal (en conjunto de validación)

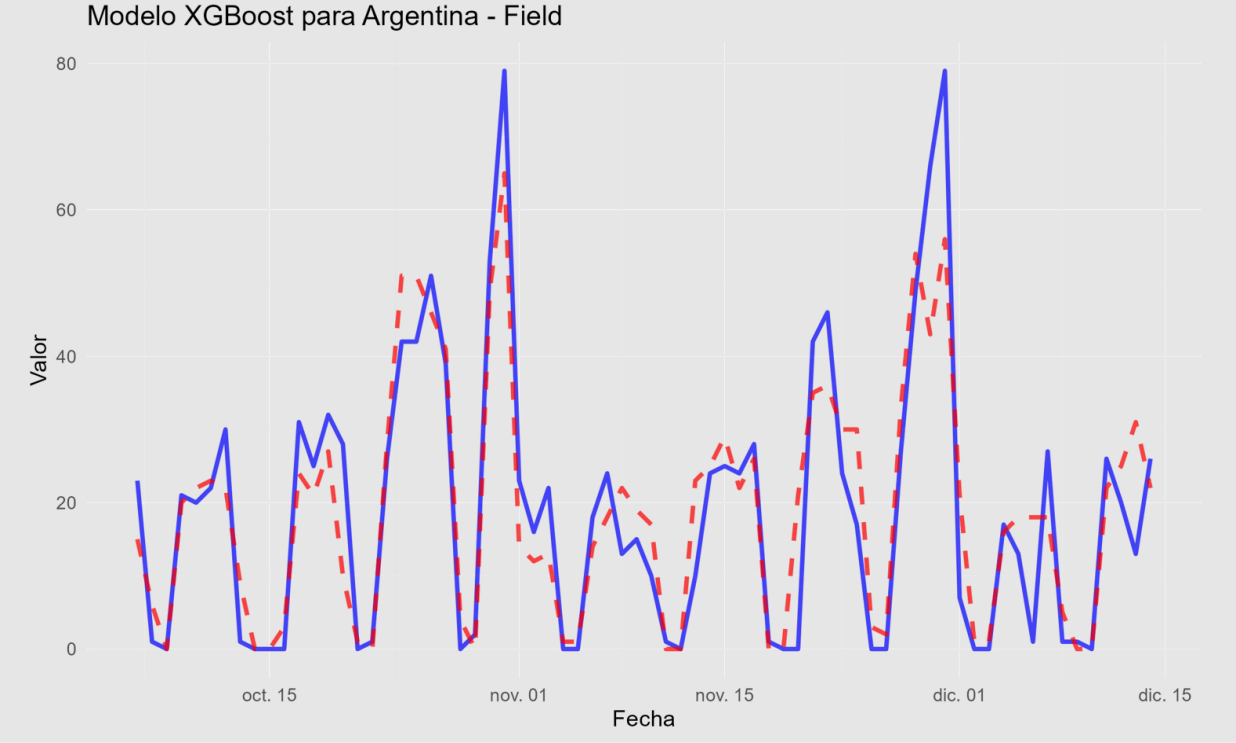

Figura 53 - XGBoost con rezagos y medias móviles: Ajuste del modelo (rojo) a los datos reales (azul) en Argentina Field (en conjunto de validación)

Obtenemos, en general, un peor desempeño que el de XGBoost original (MSE de 26.50 vs 20.31), lo cual se puede deber a la pérdida de data mencionada, así como a la posibilidad de que las variables agregadas no aporten más o mejor información al modelo. Sin embargo, a nivel particular, algunos mercados-canales tienen un desempeño levemente superior.

7.1.5.5. *XGBoost: agregado de rezagos y medias móviles con grilla de hiperparámetros*

Probamos, ahora, este mismo modelo (sección 7.1.5.4.), pero optimizando sus hiperparámetros mediante el uso de una grilla. La metodología usada es la misma que en los otros casos en que se usa grilla (secciones 7.1.4.3. y 7.1.5.3.).

Una salvedad a realizar tiene que ver con la ventana temporal. El conjunto de prueba mantiene su tamaño en un 20% de los datos (del 6 de octubre de 2023 al 14 de diciembre de 2023) con el objeto de tener una misma ventana de tiempo comparable vs. el resto de los modelos. Tras esta separación del conjunto de prueba, se toma el 80% restante de los datos y se le remueven los registros con valores faltantes a causa del agregado de rezagos y medias móviles. Solo después de realizar esta remoción de registros, es que aplicamos la división de los datos en conjunto de entrenamiento y de validación, destinando un 87,50% de estos datos al primer conjunto y el resto al segundo.

Además, cabe mencionar que la utilización de las predicciones como valores rezagados y componentes de la media móvil para observaciones de una fecha posterior se realiza tanto en el conjunto de validación como en el de prueba, con el objeto de evitar *data leakage*.

Finalmente, observamos los resultados obtenidos en el conjunto de prueba:

Country_Name	AccountSource	MSE
Argentina	SSU	201.13
Bolivia	SSU	6.79
Chile	SSU	60.76
Costa Rica	SSU	1.91
Ecuador	SSU	14.37
El Salvador	SSU	2.43
Guatemala	SSU	3.79
Honduras	SSU	1.49
Nicaragua	SSU	4.23
Panamá	SSU	1.63
Paraguay	SSU	6.03
Perú	SSU	25.97
República Dominicana	SSU	23.47
Uruguay	SSU	0.90
Venezuela	SSU	6.80
Argentina	Field	116.73
Bolivia	Field	13.66
Chile	Field	43.83
Costa Rica	Field	3.70
Ecuador	Field	30.74
El Salvador	Field	10.04
Guatemala	Field	55.47
Honduras	Field	4.40
Nicaragua	Field	19.40
Panamá	Field	18.16
Paraguay	Field	6.23
Perú	Field	223.01
República Dominicana	Field	38.46
Uruguay	Field	6.10
Venezuela	Field	30.30
Argentina	Must Have	2.47
Bolivia	Must Have	0.30
Chile	Must Have	1.47
Costa Rica	Must Have	1.67
Ecuador	Must Have	0.57
El Salvador	Must Have	33.11
Guatemala	Must Have	16.07
Honduras	Must Have	0.09
Nicaragua	Must Have	0.07
Panamá	Must Have	0.00
Paraguay	Must Have	9.76
Perú	Must Have	6.44
República Dominicana	Must Have	0.03
Uruguay	Must Have	0.06
Venezuela	Must Have	0.20

Figura 54 - XGBoost con rezagos y medias móviles - grilla de hiperparámetros: MSE por país y canal (en conjunto de validación)

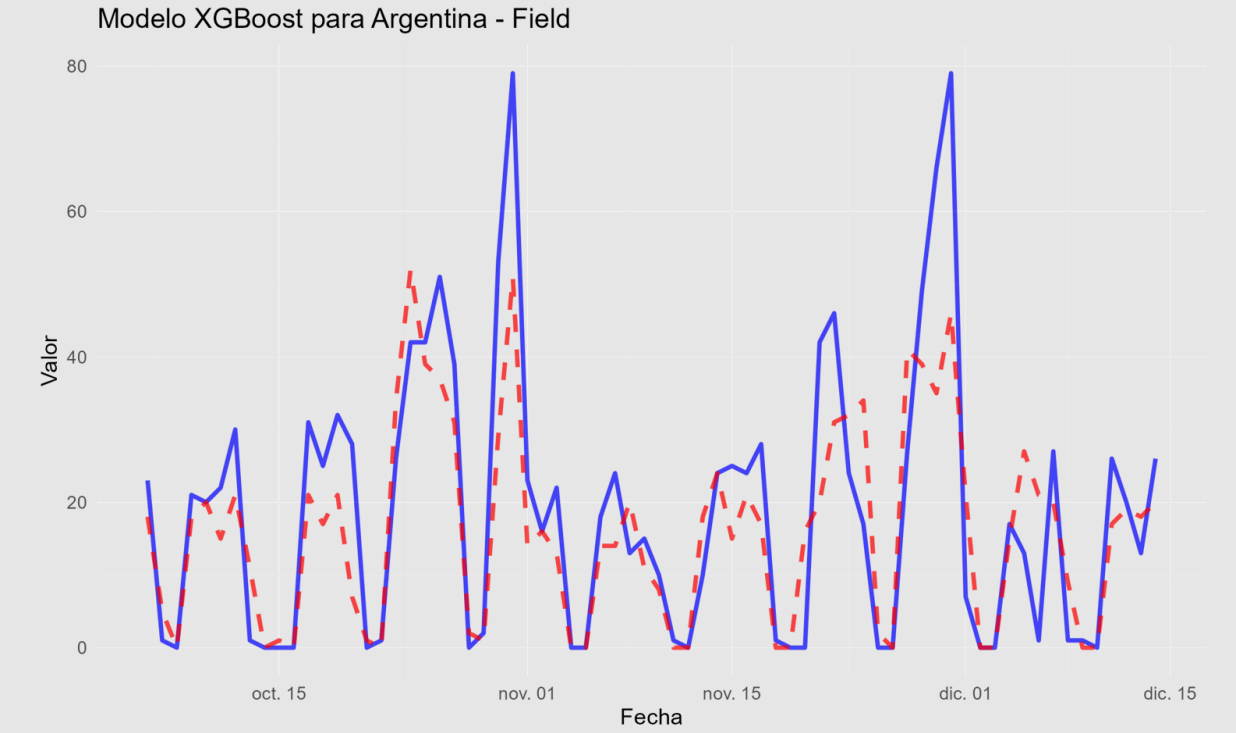

Figura 55 - XGBoost con rezagos y medias móviles - grilla de hiperparámetros: Ajuste del modelo (rojo) a los datos reales (azul) en Argentina Field (en conjunto de validación)

El MSE a nivel general es de 23.43, mejor que antes de optimizar hiperparámetros, pero sigue siendo peor que el modelo XGBoost original, por más que en algunos mercados-canales particulares el rendimiento sea superador. Esto puede deberse, como se menciona en la sección anterior, a la pérdida de datos y a que las variables agregadas no parecen aportar mejor información predictiva al modelo. Además, en un escenario de división de los datos en entrenamiento-validación-prueba, la cantidad de observaciones para entrenar termina siendo bastante menor.

7.1.6. Light GBM

Continuando la línea del *boosting*, o ensamble, probamos ahora un modelo Light GBM (*Gradient Boosting Machine*) para comparar con XGBoost.

El modelo utilizado de Light GBM es uno con 100 iteraciones (probado como valor inicial e iterado, sin obtener mejoras en rendimiento) y trata, como siempre, de minimizar el MSE (error cuadrático medio).

Previo al entrenamiento del modelo se realiza un *feature engineering* para obtener, a partir de Acquisition_Date, otros *features* relacionados a la fecha como los usados en XGBoost y así aportar a la estacionalidad del modelo. Por otro lado, cabe mencionar que en este caso no hace falta hacer *one-hot-encoding* de variables categóricas, ya que este tipo de *features* son bien manejadas por el modelo. Los resultados son los siguientes:

Country_Name	AccountSource	MSE
Argentina	SSU	307.99
Bolivia	SSU	9.91
Chile	SSU	85.71
Costa Rica	SSU	2.33
Ecuador	SSU	19.69
El Salvador	SSU	3.37
Guatemala	SSU	4.17
Honduras	SSU	1.31
Nicaragua	SSU	3.21
Panamá	SSU	1.84
Paraguay	SSU	10.67
Perú	SSU	40.47
República Dominicana	SSU	27.89
Uruguay	SSU	1.07
Venezuela	SSU	4.80
Argentina	Field	250.14
Bolivia	Field	17.43
Chile	Field	25.60
Costa Rica	Field	4.40
Ecuador	Field	30.71
El Salvador	Field	12.33
Guatemala	Field	70.94
Honduras	Field	4.96
Nicaragua	Field	8.96
Panamá	Field	6.61
Paraguay	Field	10.53
Perú	Field	227.39
República Dominicana	Field	22.30
Uruguay	Field	6.81
Venezuela	Field	41.16
Argentina	Must Have	2.59
Bolivia	Must Have	0.20
Chile	Must Have	1.14
Costa Rica	Must Have	5.71
Ecuador	Must Have	0.23
El Salvador	Must Have	23.10
Guatemala	Must Have	0.76
Honduras	Must Have	0.06
Nicaragua	Must Have	0.06
Panamá	Must Have	0.03
Paraguay	Must Have	8.33
Perú	Must Have	1.31
República Dominicana	Must Have	0.01
Uruguay	Must Have	0.04
Venezuela	Must Have	0.07

Figura 56 - Light GBM: MSE por país y canal (en conjunto de validación)

En general, no superan a los resultados obtenidos con XGBoost. Por otro lado, el MSE a nivel global es de 29.07.

Cuando nos enfocamos en los gráficos (Figura 57), para observar el ajuste que hace el modelo, notamos que si bien detecta cierta estacionalidad, en general se mantiene en niveles no tan variables, no muy alejados de la media. Tiene picos y caídas pero muy suavizados. Al modificar los valores de los hiperparámetros, el rendimiento no es superador.

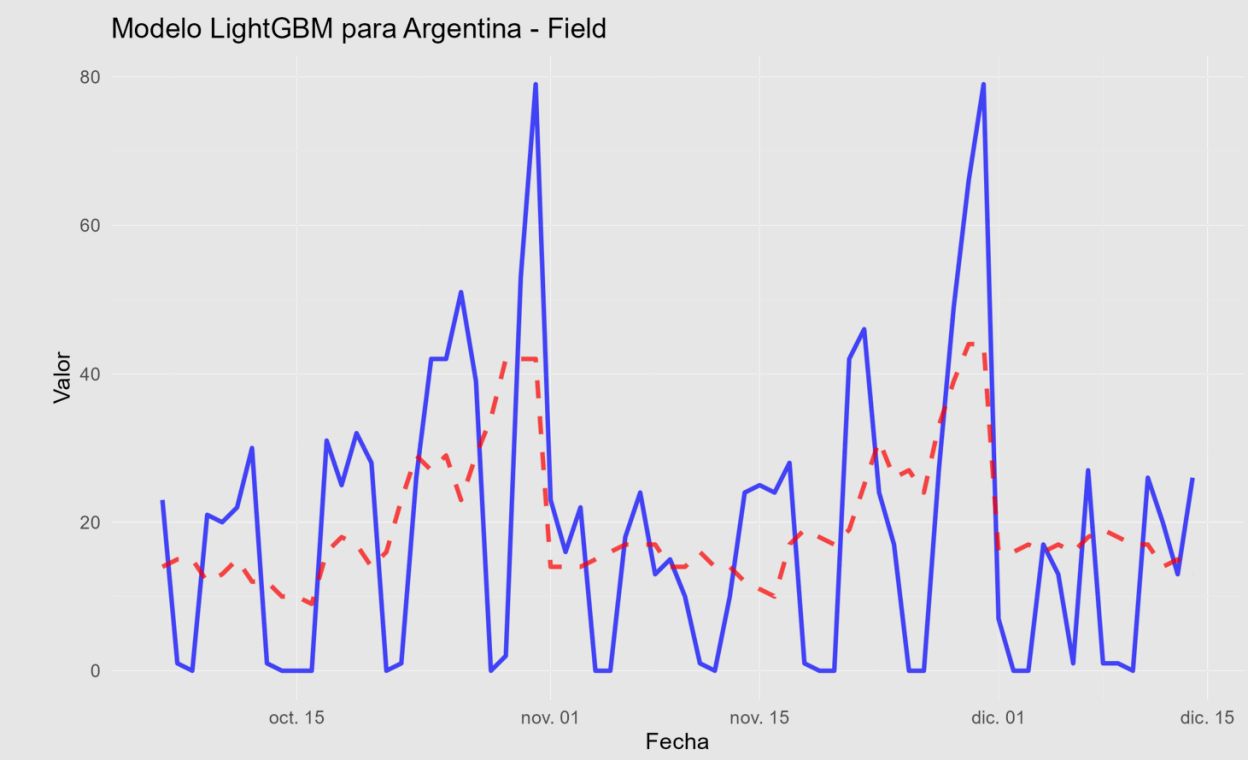

Figura 57 - Light GBM: Ajuste del modelo (rojo) a los datos reales (azul) en Argentina Field (en conjunto de validación)

7.2. Modelos pool generales

En esta sección, se busca volver sobre dos estrategias abordadas anteriormente: XGBoost y Light GBM, pero con un abordaje distinto. En este caso, en lugar de entrenar 45 modelos por separado (uno por cada combinación de mercado-canal), lo que se busca es predecir Acquisitions con un modelo pool, que sea único para todos los mercados-canales. El mismo, los incorpora como otras *features* explicativas (Country_Name y AccountSource) que se suman a las que ya tenemos.

Los modelos elegidos para este ejercicio son aquellos que admiten la inclusión features adicionales como XGBoost (en su variante de uso de grilla de hiperparámetros, debido a su probado rendimiento superior) y Light GBM.

Para asegurar una óptima división de la base de datos en conjuntos de entrenamiento, validación (cuando corresponde) y prueba, lo que se hace es, primero, subdividir la base de datos original en 45 bases (una por cada combinación mercado-canal). Dentro de esta iteración por mercado-canal, se realiza la división:

- En XGBoost con grilla de hiperparámetros, se usa 70% de los datos para entrenamiento, 10% para validación y 20% para prueba.
- En Light GBM, se usa 80% para entrenamiento y 20% para prueba.

Estas divisiones (y su temporalidad) son las mismas que las usadas a lo largo de todo el trabajo, en todas las estrategias probadas.

Finalmente, se compilan los conjuntos de datos de los distintos mercados-canales, obteniendo:

- Un único conjunto de entrenamiento.
- Un único conjunto de validación.
- Un único conjunto de evaluación.

Al hacer la división de los conjuntos al interior de la iteración por mercado-canal se garantiza que la división de la base original en estos tres conjuntos tenga suficientes datos de cada mercado-canal y que no ocurra que, debido a una cuestión de orden de los registros, algunos mercados-canales se vean subrepresentados.

Como en la anterior sección, todas las referencias a MSE y comparativas entre datos reales y pronosticados, son relativos a la ventana de tiempo correspondiente al conjunto de validación.

7.2.1. Pool: XGBoost con grilla de hiperparámetros

Se implementa un modelo similar al aplicado en la sección 7.1.5.3., con la diferencia de que, en este caso, se hace un trabajo adicional de *one-hot-encoding* de las variables Country_Name y AccountSource, ya que las mismas se suman con *features* explicativas y XGBoost solo acepta *input* de variables numéricas.

Tras entrenar el modelo, se encuentra cuál es la combinación única de hiperparámetros que mejor predice Acquisitions en conjunto de validación. Luego, se

calcula el MSE en el conjunto de prueba. Finalmente, se deshace el *one-hot-encoding* para volver a la estructura de datos original.

Los resultados observados son los siguientes:

Country_Name	AccountSource	MSE
Argentina	SSU	237.43
Bolivia	SSU	49.93
Chile	SSU	202.40
Costa Rica	SSU	1.63
Ecuador	SSU	15.03
El Salvador	SSU	2.49
Guatemala	SSU	3.39
Honduras	SSU	1.26
Nicaragua	SSU	2.06
Panamá	SSU	1.51
Paraguay	SSU	10.24
Perú	SSU	30.46
República Dominicana	SSU	26.37
Uruguay	SSU	1.37
Venezuela	SSU	2.60
Argentina	Field	81.54
Bolivia	Field	9.27
Chile	Field	15.27
Costa Rica	Field	2.64
Ecuador	Field	18.41
El Salvador	Field	5.20
Guatemala	Field	24.90
Honduras	Field	3.77
Nicaragua	Field	4.44
Panamá	Field	3.74
Paraguay	Field	9.13
Perú	Field	85.97
República Dominicana	Field	8.73
Uruguay	Field	5.84
Venezuela	Field	27.76
Argentina	Must Have	3.71
Bolivia	Must Have	0.33
Chile	Must Have	2.67
Costa Rica	Must Have	1.14
Ecuador	Must Have	1.16
El Salvador	Must Have	23.09
Guatemala	Must Have	0.51
Honduras	Must Have	0.17
Nicaragua	Must Have	0.40
Panamá	Must Have	0.10
Paraguay	Must Have	8.50
Perú	Must Have	4.81
República Dominicana	Must Have	0.33
Uruguay	Must Have	0.11
Venezuela	Must Have	0.11

Figura 58 - Pool: XGBoost con grilla de hiperparámetros: MSE por país y canal (en conjunto de validación)

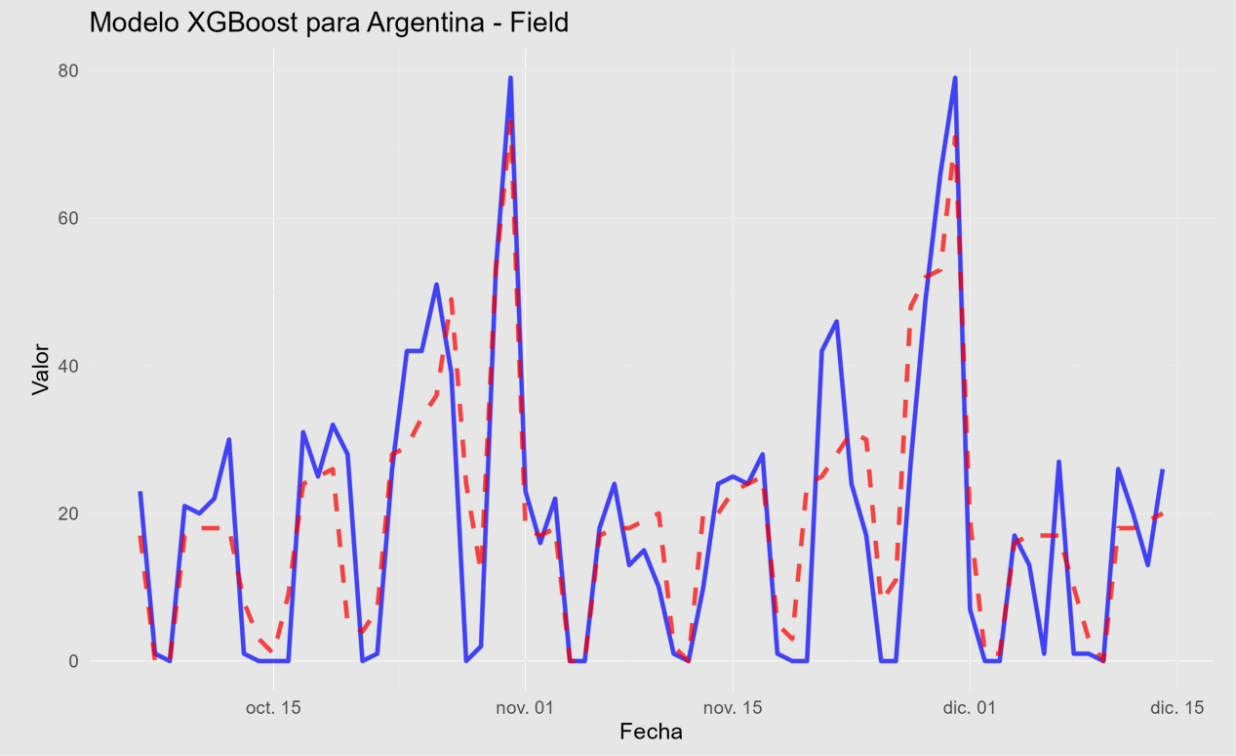

Figura 59 - Pool: XGBoost con grilla de hiperparámetros - Ajuste del modelo (rojo) a los datos reales (azul) en Argentina Field (en conjunto de validación)

El MSE a nivel general es de 20.93, el cual es mayor que el modelo XGBoost con grilla de la sección 7.1.5.3. Sin embargo, para el caso de algunos mercados-canales en particular, este modelo tiene un mejor desempeño.

7.2.2. Pool: Light GBM

Ahora probamos un modelo Light GBM replicando lo hecho en la sección 7.1.6., agregando Country_Name y AccountSource como *features* explicativos.

En este caso, no hace falta realizar *one-hot-encoding* tampoco para estas nuevas variables explicativas, dado que Light GBM es capaz de recibir como entrada *features* no numéricas.

El proceso de división en conjuntos de entrenamiento y prueba sigue la misma lógica que el explicado en 7.2.1., obteniendo finalmente un único conjunto de entrenamiento y un único conjunto de prueba.

Al evaluar el rendimiento del modelo en el conjunto de prueba, observamos:

Country_Name	AccountSource	MSE
Argentina	SSU	314.39
Bolivia	SSU	39.80
Chile	SSU	95.70
Costa Rica	SSU	2.89
Ecuador	SSU	24.44
El Salvador	SSU	3.43
Guatemala	SSU	5.41
Honduras	SSU	5.04
Nicaragua	SSU	2.97
Panamá	SSU	2.03
Paraguay	SSU	11.67
Perú	SSU	83.37
República Dominicana	SSU	50.21
Uruguay	SSU	3.14
Venezuela	SSU	1.56
Argentina	Field	273.89
Bolivia	Field	40.16
Chile	Field	29.84
Costa Rica	Field	6.50
Ecuador	Field	55.80
El Salvador	Field	14.79
Guatemala	Field	62.96
Honduras	Field	9.11
Nicaragua	Field	7.84
Panamá	Field	6.93
Paraguay	Field	11.40
Perú	Field	212.39
República Dominicana	Field	24.70
Uruguay	Field	10.36
Venezuela	Field	38.10
Argentina	Must Have	3.06
Bolivia	Must Have	0.26
Chile	Must Have	11.50
Costa Rica	Must Have	1.49
Ecuador	Must Have	0.39
El Salvador	Must Have	23.09
Guatemala	Must Have	0.53
Honduras	Must Have	0.26
Nicaragua	Must Have	0.26
Panamá	Must Have	0.21
Paraguay	Must Have	8.53
Perú	Must Have	7.63
República Dominicana	Must Have	0.21
Uruguay	Must Have	1.74
Venezuela	Must Have	0.23

Figura 60 - Pool: Light GBM: MSE por país y canal (en conjunto de validación)

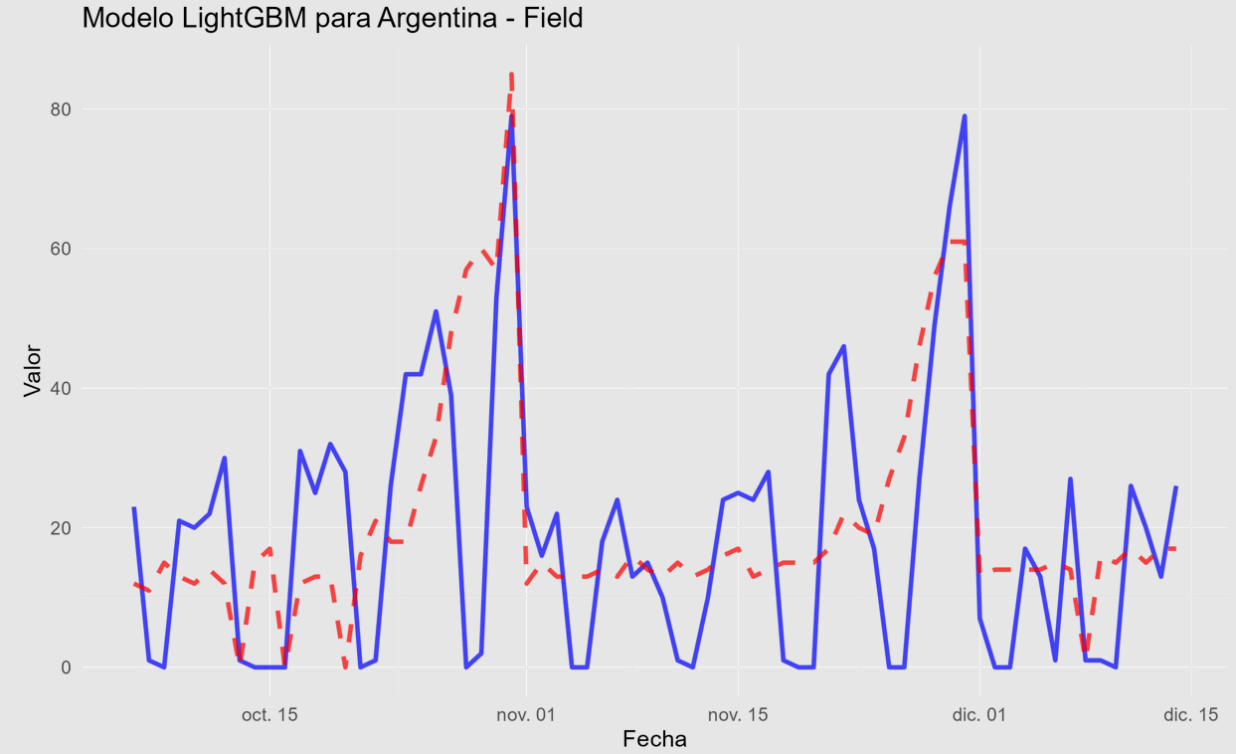

Figura 61 - Pool: Light GBM - Ajuste del modelo (rojo) a los datos reales (azul) en Argentina Field (en conjunto de validación)

El MSE a nivel general es de 33.56 (mayor al mostrado por el modelo Light GBM por mercado-canal). Asimismo, al ir al detalle particular, en la mayoría de los mercados-canales el rendimiento también es peor con un único modelo general que con uno entrenado para cada mercado-canal. Es posible que, al tratar de aprender parámetros comunes a todos los mercados-canales, el modelo no pueda captar la realidad de todos estos escenarios al mismo tiempo.

8. Modelo Ganador

Se recopilan los datos de los MSE de cada estrategia usada, para cada combinación mercado-canal y se presentan en la siguiente figura.

Country_Name	AccountSource	ARIMA	SARIMA	ETS	Prophet Inicial	Prophet con estacionalidad mensual	Prophet con grilla de HP	XGBoost	XGBoost con grilla de HP	XGBoost con rezagos y mm	XGBoost con rezagos y mm - grilla de HP	Light GBM	Pool: XGBoost con grilla	Pool: Light GBM
Argentina	SSU	338.01	223.19	208.14	218.06	165.17	162.09	273.30	216.56	364.41	201.13	307.99	237.43	314.39
Bolivia	SSU	10.51	6.69	5.71	6.51	5.86	6.19	4.70	5.70	5.30	6.79	9.91	49.93	39.80
Chile	SSU	82.84	36.74	42.74	37.31	39.59	38.24	34.29	31.70	33.17	60.76	85.71	202.40	95.70
Costa Rica	SSU	2.33	2.23	2.27	2.44	1.83	1.86	2.17	1.96	2.46	1.91	2.33	1.63	2.89
Ecuador	SSU	27.44	23.36	30.86	24.30	19.63	19.51	15.27	16.23	34.69	14.37	19.69	15.03	24.44
El Salvador	SSU	3.57	2.46	2.83	2.46	2.34	2.76	3.06	3.30	3.01	2.43	3.37	2.49	3.43
Guatemala	SSU	4.43	3.90	3.79	3.91	3.29	3.23	2.94	2.93	5.14	3.79	4.17	3.39	5.41
Honduras	SSU	1.59	1.30	1.39	1.30	1.34	1.24	1.51	1.40	1.24	1.49	1.31	1.26	5.04
Nicaragua	SSU	3.49	2.83	2.97	2.83	2.53	2.47	3.46	4.19	3.44	4.23	3.21	2.06	2.97
Panamá	SSU	1.31	1.14	1.17	1.17	1.20	1.20	1.74	1.53	1.37	1.63	1.84	1.51	2.03
Paraguay	SSU	9.31	7.97	14.36	7.80	6.53	6.73	9.66	6.93	7.33	6.03	10.67	10.24	11.67
Perú	SSU	56.16	37.89	42.93	38.23	29.01	28.89	22.03	37.13	29.97	25.97	40.47	30.46	83.37
República Dominicana	SSU	40.50	33.24	49.53	33.17	26.93	25.16	17.93	16.60	27.90	23.47	27.89	26.37	50.21
Uruguay	SSU	0.90	0.70	1.04	0.70	0.83	1.01	1.40	1.19	0.99	0.90	1.07	1.37	3.14
Venezuela	SSU	3.60	2.71	1.47	2.81	3.84	3.64	6.47	2.80	5.93	6.80	4.80	2.60	1.56
Argentina	Field	365.84	224.87	257.56	225.03	154.84	161.09	96.19	79.44	65.26	116.73	250.14	81.54	273.89
Bolivia	Field	22.41	16.10	19.49	16.74	12.76	13.17	14.69	8.47	12.94	13.66	17.43	9.27	40.16
Chile	Field	37.49	24.49	25.30	26.13	24.86	25.57	16.64	15.46	30.53	43.83	25.60	15.27	29.84
Costa Rica	Field	3.69	3.50	3.74	3.44	3.71	4.51	5.00	3.47	7.50	3.70	4.40	2.64	6.50
Ecuador	Field	32.21	23.90	26.29	25.59	17.39	19.74	32.69	31.60	30.67	30.74	30.71	18.41	55.80
El Salvador	Field	9.44	7.77	9.76	7.24	6.59	7.17	25.30	29.30	13.61	10.04	12.33	5.20	14.79
Guatemala	Field	45.50	48.37	59.99	48.07	40.06	40.30	69.46	46.49	49.87	55.47	70.94	24.90	62.96
Honduras	Field	4.97	5.09	3.74	6.00	5.84	6.90	6.46	3.59	6.86	4.40	4.96	3.77	9.11
Nicaragua	Field	6.31	5.03	4.91	5.03	7.74	7.51	16.90	7.33	18.26	19.40	8.96	4.44	7.84
Panamá	Field	5.07	4.66	4.83	4.46	3.90	5.03	8.76	6.20	6.21	18.16	6.61	3.74	6.93
Paraguay	Field	15.39	13.81	17.33	12.40	7.57	8.29	6.71	5.14	6.34	6.23	10.53	9.13	11.40
Perú	Field	365.66	348.04	369.31	366.71	262.40	356.74	115.04	130.27	147.57	223.01	227.39	85.97	212.39
República Dominicana	Field	22.74	26.20	29.23	24.59	17.80	17.80	13.13	12.01	23.57	38.46	22.30	8.73	24.70
Uruguay	Field	7.57	5.01	5.67	5.33	4.99	5.00	6.11	5.44	5.89	6.10	6.81	5.84	10.36
Venezuela	Field	43.61	54.37	46.09	53.17	52.50	79.99	27.29	26.90	30.77	30.30	41.16	27.76	38.10
Argentina	Must Have	2.79	1.60	2.09	1.60	1.76	1.84	2.84	1.77	2.51	2.47	2.59	3.71	3.06
Bolivia	Must Have	0.20	0.20	0.20	0.20	0.20	0.20	0.23	0.20	0.20	0.30	0.20	0.33	0.26
Chile	Must Have	1.14	1.11	1.57	1.69	1.39	0.90	0.96	1.01	1.16	1.47	1.14	2.67	11.50
Costa Rica	Must Have	1.40	1.46	2.97	1.34	1.51	1.61	1.47	1.40	165.31	1.67	5.71	1.14	1.49
Ecuador	Must Have	0.86	0.36	0.19	0.17	0.23	0.19	0.40	0.24	0.40	0.57	0.23	1.16	0.39
El Salvador	Must Have	22.89	22.90	23.17	22.97	23.13	22.89	22.94	22.89	22.89	33.11	23.10	23.09	23.09
Guatemala	Must Have	0.90	0.99	1.56	0.90	0.90	0.46	10.79	0.81	6.90	16.07	0.76	0.51	0.53
Honduras	Must Have	0.06	0.06	0.14	0.06	0.06	0.06	0.21	0.20	0.27	0.09	0.06	0.17	0.26
Nicaragua	Must Have	0.00	0.06	0.06	0.06	0.06	0.06	0.09	0.07	0.11	0.07	0.06	0.40	0.26
Panamá	Must Have	8.30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.10	0.21
Paraguay	Must Have	8.30	8.30	8.30	8.30	8.31	8.30	11.11	8.36	8.31	9.76	8.33	8.50	8.53
Perú	Must Have	1.14	1.06	1.29	1.10	1.00	1.07	2.27	1.00	2.17	6.44	1.31	4.81	7.63
República Dominicana	Must Have	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.11	0.03	0.01	0.33	0.21
Uruguay	Must Have	0.04	0.04	0.04	0.04	0.04	0.04	0.10	0.06	0.09	0.06	0.04	0.11	1.74
Venezuela	Must Have	0.03	0.03	0.03	0.03	0.10	0.03	0.03	0.03	0.11	0.20	0.07	0.11	0.23
General		**35.86**	**27.46**	**29.68**	**27.78**	**21.59**	**24.46**	**20.31**	**17.77**	**26.50**	**23.43**	**29.07**	**20.93**	**33.56**

Figura 62 - MSE por mercado-canal, para cada estrategia probada

Observando la Figura 62, si buscamos el MSE mínimo en cada caso, llegamos a la conclusión de que los modelos ganadores para cada mercado-canal son los siguientes:

Country_Name	AccountSource	Modelo/s Ganador/es
Argentina	SSU	Prophet con grilla de HP
Bolivia	SSU	XGBoost
Chile	SSU	XGBoost con grilla de HP
Costa Rica	SSU	Pool: XGBoost con grilla
Ecuador	SSU	XGBoost con rezagos y mm - grilla de HP
El Salvador	SSU	Prophet con estacionalidad mensual
Guatemala	SSU	XGBoost con grilla de HP
Honduras	SSU	Prophet con grilla de HP, XGBoost con rezagos y mm
Nicaragua	SSU	Pool: XGBoost con grilla
Panamá	SSU	SARIMA
Paraguay	SSU	XGBoost con rezagos y mm - grilla de HP
Perú	SSU	XGBoost
República Dominicana	SSU	XGBoost con grilla de HP
Uruguay	SSU	SARIMA, Prophet Inicial
Venezuela	SSU	ETS
Argentina	Field	XGBoost con rezagos y mm
Bolivia	Field	XGBoost con grilla de HP
Chile	Field	Pool: XGBoost con grilla
Costa Rica	Field	Pool: XGBoost con grilla
Ecuador	Field	Prophet con estacionalidad mensual
El Salvador	Field	Pool: XGBoost con grilla
Guatemala	Field	Pool: XGBoost con grilla
Honduras	Field	XGBoost con grilla de HP
Nicaragua	Field	Pool: XGBoost con grilla
Panamá	Field	Pool: XGBoost con grilla
Paraguay	Field	XGBoost con grilla de HP
Perú	Field	Pool: XGBoost con grilla
República Dominicana	Field	Pool: XGBoost con grilla
Uruguay	Field	Prophet con estacionalidad mensual
Venezuela	Field	XGBoost con grilla de HP
Argentina	Must Have	SARIMA, Prophet Inicial
Bolivia	Must Have	ARIMA, SARIMA, Prophet Inicial, Prophet con estacionalidad mensual, Prophet con grilla de HP, XGBoost con grilla de HP, XGBoost con rezagos y mm, Light GBM
Chile	Must Have	Prophet con grilla de HP
Costa Rica	Must Have	Pool: XGBoost con grilla
Ecuador	Must Have	Prophet Inicial
El Salvador	Must Have	ARIMA, Prophet con grilla de HP, XGBoost con grilla de HP, XGBoost con rezagos y mm
Guatemala	Must Have	Prophet con grilla de HP
Honduras	Must Have	ARIMA, SARIMA, Prophet Inicial, Prophet con estacionalidad mensual, Prophet con grilla de HP, Light GBM
Nicaragua	Must Have	ARIMA
Panamá	Must Have	SARIMA, ETS, Prophet Inicial, Prophet con estacionalidad mensual, Prophet con grilla de HP, XGBoost, XGBoost con grilla de HP, XGBoost con rezagos y mm, XGBoost con rezagos y mm - grilla de HP
Paraguay	Must Have	ARIMA, SARIMA, ETS, Prophet Inicial, Prophet con grilla de HP
Perú	Must Have	Prophet con estacionalidad mensual, XGBoost con grilla de HP
República Dominicana	Must Have	ARIMA, SARIMA, ETS, Prophet Inicial, Prophet con estacionalidad mensual, Prophet con grilla de HP, XGBoost, XGBoost con grilla de HP, Light GBM
Uruguay	Must Have	ARIMA, SARIMA, ETS, Prophet Inicial, Prophet con estacionalidad mensual, Prophet con grilla de HP, Light GBM
Venezuela	Must Have	ARIMA, SARIMA, ETS, Prophet Inicial, Prophet con grilla de HP, XGBoost, XGBoost con grilla de HP
General		**XGBoost con grilla de HP**

Figura 63 - Modelo ganador (estrategia ganadora) por mercado-canal

Con el objeto de presentar esta información de modo resumido, contamos, para cada estrategia probada, la cantidad de mercados-canales en los cuales dicha estrategia fue exitosa, es decir, la cantidad de veces en las que su MSE fue el mínimo entre todas. Los criterios de evaluación son los que siguen:

- Cantidad de mercados-canales ganados en general.
- Cantidad de mercados-canales ganados por canal:
  - *SSU*
  - *Field*
  - *Must Have*

Así, llegamos al siguiente cuadro sintético:

Criterio	ARIMA	SARIMA	ETS	Prophet Inicial	Prophet con estacionalidad mensual	Prophet con grilla de HP	XGBoost	XGBoost con grilla de HP	XGBoost con rezagos y mm	XGBoost con rezagos y mm - grilla de HP	Light GBM	Pool: XGBoost con grilla	Pool: Light GBM
Cantidad de mercados-canales ganados	8	10	7	10	9	12	5	13	5	3	4	11	0
Cantidad ganados para SSU	0	2	1	1	1	2	2	3	1	2	0	2	0
Cantidad ganados para Field	0	0	0	0	2	0	0	4	1	0	0	8	0
Cantidad ganados para Must Have	8	8	6	9	6	8	3	6	3	1	4	1	0

Figura 64 - Modelo ganador (estrategia ganadora) por mercado-canal

En la Figura 64, podemos observar cómo los modelos univariados/autorregresivos en general son más exitosos en el canal *Must Have*, mientras que los multivariados tienen más casos de éxito entre los canales *SSU* y *Field*. Particularmente, el modelo Prophet inicialmente probado, es bastante exitoso dentro del canal *Must Have*, mientras que

XGBoost con grilla de hiperparámetros es exitoso en los otros canales con sus dos variantes: en *SSU* con la elaboración de modelos a nivel mercado-canal y en *Field* en su variante de único modelo pool, con los mercados-canales como *features* explicativos.

En términos generales, observamos cómo XGBoost con grilla de hiperparámetros es la estrategia que termina teniendo un rendimiento más satisfactorio, ya que tiene el MSE general más bajo (17.77), es el que más mercados-canales ganados tiene (13) y también sale favorecido en el canal digital *SSU* (3).

Si bien una conclusión posible de estos datos sería la aplicación, para cada mercado-canal, del modelo de mejor rendimiento, dado el buen rendimiento en general de la estrategia de XGBoost con grilla de hiperparámetros, consideramos coherente implementarla para todos los casos. El implementar una única estrategia (la cual tiene modelos adaptables a cada mercado-canal) combina la posibilidad de adaptación a realidades locales, con la coherencia general del uso de un modelo único a nivel regional y la no dependencia de los datos usados en el tiempo. Si se adoptara la decisión de utilizar el modelo de mejor rendimiento para cada mercado-canal, se estaría sujeto a la existencia de hasta trece posibles estrategias distintas en 45 mercados-canales distintos, las cuales incluso podrían variar en el pronóstico hecho mes a mes. Resulta, entonces, más coherente adoptar la estrategia de XGBoost con grilla para todos ellos y mantenerla en el tiempo, al menos mientras el rendimiento del modelo acompañe. Por lo tanto, si hay que elegir un único modelo, entonces decimos que XGBoost con grilla de hiperparámetros es el ganador.

Si miramos más detalladamente los resultados de este modelo ganador, vemos cómo el patrón de mayor cantidad de adquisiciones a fin de mes, el cual algunos modelos no captaba y otros sólo parcialmente, es detectado mejor en el caso de XGBoost y eso ayuda a un mejor rendimiento. Una mirada a los gráficos comparativos Real vs Pronóstico también nos permite entender visualmente cómo este modelo ajusta mucho mejor que el resto.

Con respecto a los *features* utilizados, se hace un análisis de importancia de *features* basado en el *gain*. El *gain* representa la mejora en la precisión de las predicciones que se obtiene al añadir cada *feature* en particular a un nodo del árbol ensamblado en el modelo. A modo de ejemplo, se incluyen algunos de los gráficos que representan el *gain* en cada mercado-canal (Figuras 65, 66, 67 y 68):

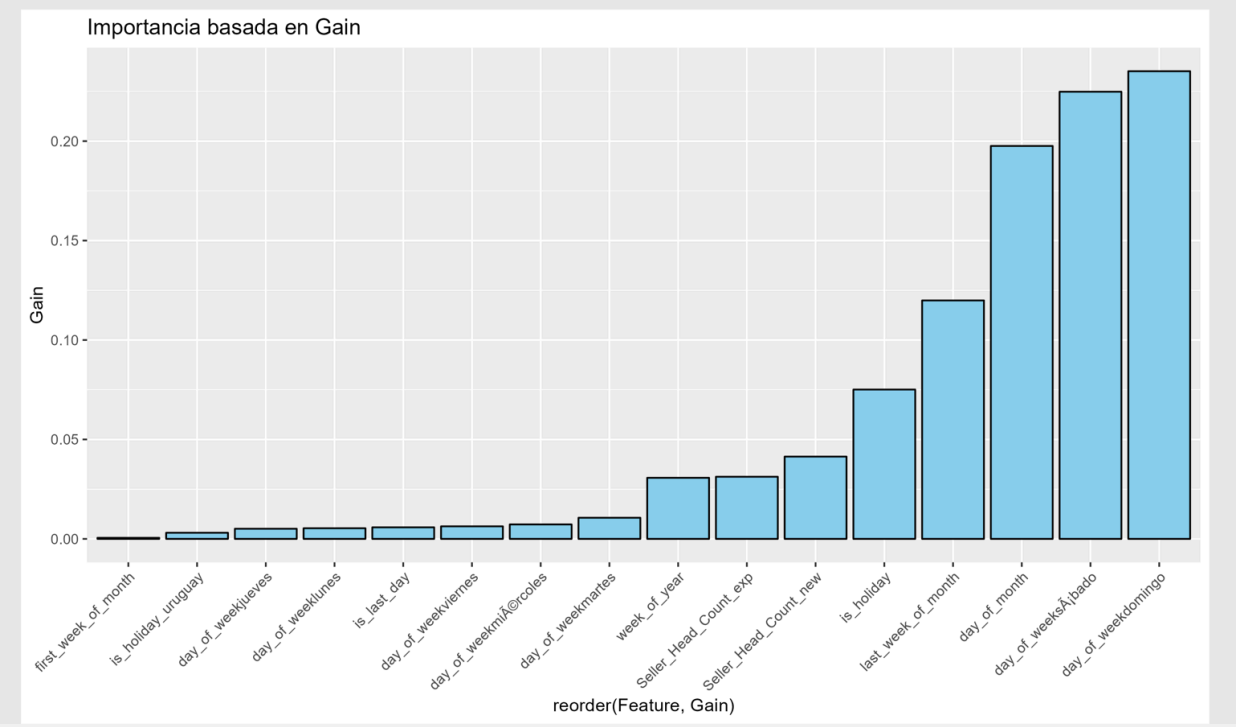

Figura 65 - Importancia de features XGBoost en Argentina Field

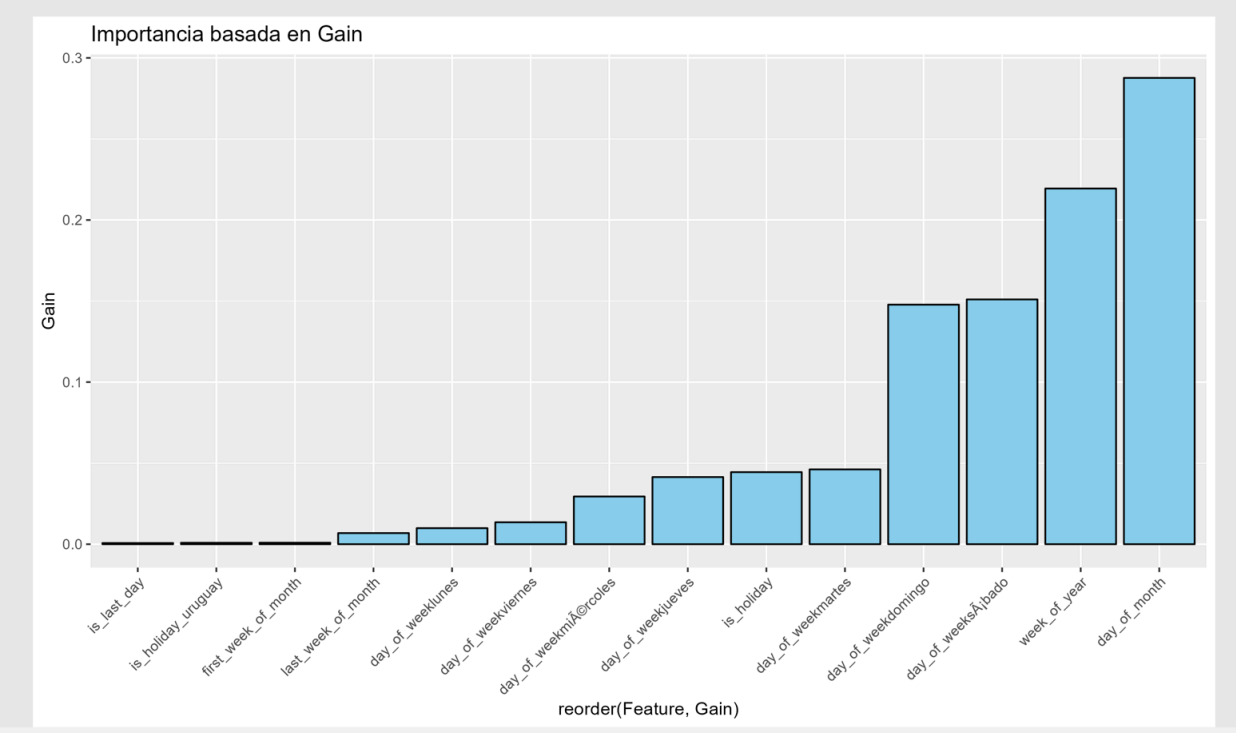

Figura 66 - Importancia de features XGBoost en Argentina Must Have

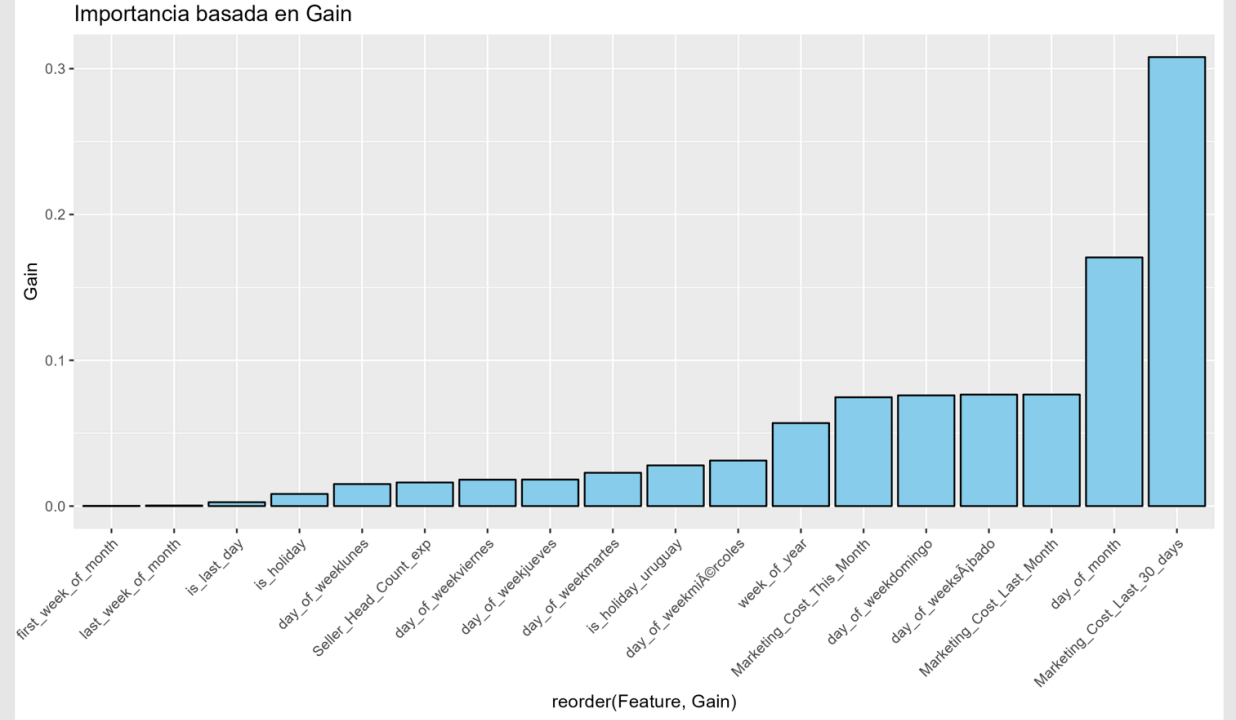

Figura 67 - Importancia de features XGBoost en El Salvador SSU

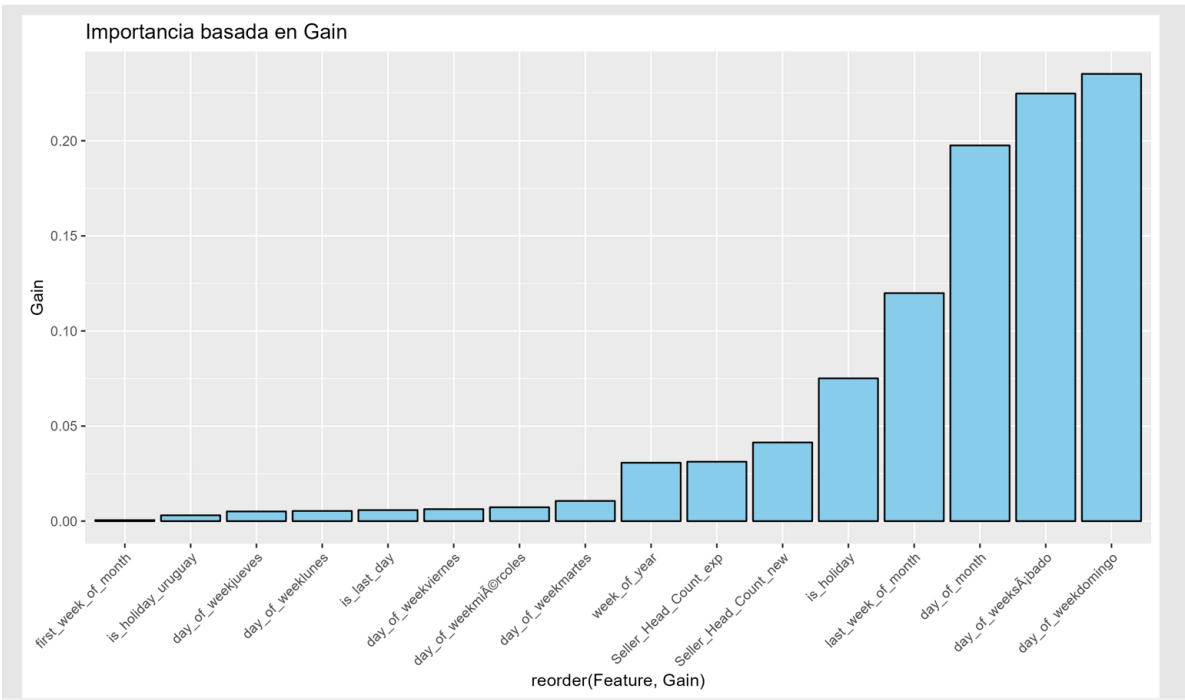

Figura 68 - Importancia de features XGBoost en Nicaragua Field

Del análisis de la importancia de *features*, podemos extraer:

- Las principales variables explicativas de las adquisiciones son:
    - *Features* de días de la semana
    - day_of_month
    - week_of_year
    - last_week_of_month
    - seller_head_count_new/exp
    - *Features* relacionadas a costos de marketing (Marketing_Cost_This_Month, Marketing_Cost_Last_Month, Marketing_Cost_Last_30_days)

- Entre estos *features*, se nota una preponderancia de aquellos relacionados al tiempo y la estacionalidad. La proporción de estos *features* entre los primeros puestos, con respecto a variables exógenas es marcadamente mayor. Sin embargo, el aporte de las exógenas a la predicción no es despreciable ni mucho menos, por lo que es aconsejable conservarlas.

- Asimismo, se nota cómo los *features* más explicativos varían de modelo a modelo. Esto quiere decir que en cada mercado y en cada canal, las realidades de ese Country_Name/canal pueden ser diferentes y, lo que es bueno para predecir en un lugar, no lo es tanto en otro (o bien lo sigue siendo, pero hay un *feature* que le roba más importancia). Por lo tanto, usar un modelo separado para cada uno de los 45 casos es una buena práctica.

9. Conclusiones

En función de lo expuesto, lo investigado y lo experimentado, se llega a la conclusión de que el mercado de adquisición de partners en el mundo del delivery de comida se comporta con cierta tendencia y (más marcadamente) estacionalidad.

Si bien la tendencia a lo largo de los años ha sido una de crecimiento, disponiendo de datos solamente de 2023, no podemos llegar a reafirmar esta afirmación que se ha encontrado en la literatura sobre el tema, o, al menos, no tenemos elementos para confirmar o desmentir una tendencia a largo plazo. En términos anuales, podemos decir que la tendencia es bastante constante, y no se finaliza el año en un nivel mucho más alto o bajo que en sus comienzos.

Con respecto a la estacionalidad, sí que llegamos a ver marcadamente, en la data, cómo operan, al menos dos. En relación a esto, podemos destacar lo siguiente:

- Presencia de estacionalidad mensual: con un comienzo bien bajo de adquisiciones a inicios de mes, que va progresivamente aumentando, hasta dar un salto más bien exponencial sobre fin de mes.
- Presencia de estacionalidad semanal: con un comportamiento que inicia en bajos niveles a inicios de semana, sube rápidamente hacia mitad/final de los días hábiles y baja, también rápidamente, en el fin de semana.
- A nivel anual, podría presumirse cierta estacionalidad también, de similar comportamiento al de la semanal: inicio en niveles bajos durante el primer mes del año, subida (un poco más progresiva y lenta) a mitad de año y caída (más suavizada, pero caída al fin) en los últimos meses. Sin embargo, disponiendo solamente de data de 2023, no podemos llegar a confirmar esta presunción que, de momento, se mantiene más bien como una intuición y solo podrá ser confirmada con el paso del tiempo y la generación de más data histórica.

Por otro lado, la influencia de ciertas variables exógenas no es menor. El tamaño de la fuerza comercial tiene un impacto en la cantidad de adquisiciones esperadas. Con respecto a ello, se descubrió cómo no solo importa el tamaño en sí, sino también su experiencia: a más experiencia, más adquisiciones. Por lo tanto, resulta útil separar el *feature* de Seller_Head_Count de vendedores en nuevos / experimentados, para ser más precisos en las estimaciones.

También es importante mencionar la inversión en marketing, la cual estimula un mayor tráfico en el canal digital, que termina redundando, después de pasar por el proceso, en mayores adquisiciones.

Por lo tanto, para una predicción más certera de las adquisiciones, necesitamos, principalmente:

- Captar la tendencia y la estacionalidad de los datos
- Captar la influencia de variables exógenas en la variable a predecir

Por supuesto, los efectos de todas estas variables no son lineales y ahí radica la necesidad de un modelo que pueda amalgamar tendencias, estacionalidades y variables exógenas. Es por eso, que el boosting, particularmente con XGBoost, termina siendo el modelo ganador, porque permite utilizar todas las variables exógenas mencionadas, al mismo tiempo que se incorpora la tendencia y estacionalidad a través de un *feature engineering* realizado sobre la variable Acquisition_Date, extrayendo otras variables tales como día del mes, día de la semana, último día del mes, primer/última semana del mes, semana del año, etc. XGBoost permite obtener lo mejor de ambos mundos y predecir con un nivel de precisión superior al resto.

Asimismo, comprobamos lo que se había anticipado: un mejor rendimiento en el caso de XGBoost en comparación al resto, en parte sostenido por el hecho de que los datos

de adquisiciones presentan una distribución exponencial (Figura 5), y XGBoost no asume ninguna distribución en particular en los datos, lo cual le permite ser suficientemente flexible para adaptarse a ellos (en contraposición con los modelos ARIMA, SARIMA, ETS y Prophet, los cuales asumen normalidad en los residuos).

Por otro lado, merece la pena mencionar también la idoneidad de la construcción de un modelo diferente para cada mercado-canal. Esto se debe a que, como se vió, la influencia relativa de los mismos features en los distintos mercados-canales es bien diversa y lo que es un predictor top 1 en un país, puede que no lo sea en otro. Además, el uso de una grilla de hiperparámetros para ajustar el modelo XGBoost del mejor modo, es más personalizado cuando se crea un modelo particular para cada caso y no uno solo que generalice para todos. El tener un modelo para cada mercado-canal permite captar mejor las tendencias de cada realidad distinta. En nuestro caso de estudio, los distintos países pueden tener diferentes tendencias, pero donde más se nota es en la apertura por canal, donde si bien el canal físico es mayor al digital, la tendencia es la de crecimiento del share del digital sobre el físico.

Por último, cabe señalar cómo el comportamiento de un canal como lo es el de *Must Have*, es bien distinto al de los otros dos, con adquisiciones más esporádicas y que no sigue tendencias o estacionalidades similares a los canales físico común y digital. Asimismo, la productividad, en términos de números, es muy baja aquí, ya que se tienen vendedores dedicados a conseguir un puñado de nuevos partners al mes. Por lo tanto, tratar este headcount de vendedores como uno físico común, llevaría a una sobreestimación de las adquisiciones, al mismo tiempo que unificar este canal con el físico añadiría ruido a las estacionalidades. Es por eso que en este canal, debido a la menor influencia de las variables exógenas (inversión en marketing nula y cantidad de vendedores constante y baja a lo largo del tiempo), tiene sentido que modelos autorregresivos sean más exitosos que aquellos multivariados, que incorporan data de variables exógenas.

En conclusión, por todo lo expuesto y desarrollado, el presente trabajo encuentra en el *boosting*, particularmente XGBoost, una herramienta idónea para el pronóstico de nuevos partners de apps de delivery de comida, debido a que logra captar complejidades, no linealidades, incluir predictores relacionados con tiempo, estacionalidad y tendencia, a la vez que variables exógenas, y adaptarse a las realidades de cada mercado-canal de la manera más precisa. Además, debido a su potencia y velocidad, permite realizar predicciones más rápidamente que otros modelos y dar una respuesta rápida a un negocio que se mueve rápido y necesita una toma de decisiones ágil.

10. *Output*: entregables para el negocio

Volvemos, ahora entonces, al problema de negocio, para poder aplicar todo lo desarrollado en este presente trabajo. Nuestro objetivo, como se mencionó al inicio de la tesis, es obtener dos *outputs*, los cuales se presentan a continuación.

10.1. Seguimiento diario vs Pronóstico

Una vez encontrado el modelo ganador, utilizamos el mismo para hacer un pronóstico en nuestra ventana de predicción. Como *output*, obtenemos un *data frame* que exportamos a un archivo de Excel. El mismo consta de 4 columnas:

- Country_Name
- AccountSource
- Acquisition_Date
- Prediction
- Real (si es que ya se tiene alguna data real del mes para el cual se está haciendo pronóstico). Esta columna se va rellenando con la data real que se va registrando a lo largo del mes, manteniendo el pronóstico constante.

Como parte de la solución ofrecida al negocio, se provee de un archivo de Excel (Figura 69), el cual ya está preparado para recibir el *output* del modelo y calcular los números pertinentes de interés.

PEGAR EL OUTPUT AQUÍ					OBSERVAR RESULTADOS AQUÍ					
Country_Name	AccountSource	Acquisition_Date	Prediction	Real	Real MTD	Prediction MTD	Prediction EOM	Real vs FC Delta Day	Real vs FC Delta MTD	Real vs FC Delta EOM
Argentina	Field	2024-02-01	19	29	29	19	589	53%	53%	-95%
Argentina	Field	2024-02-02	21	25	54	40	589	19%	35%	-91%

Figura 69 - Output #1: Adquisiciones esperadas por día

Como se puede observar en la Figura 69, con solo pegar el *output* del modelo de R en las columnas de título amarillo, las columnas de título verde se actualizan automáticamente. En ellas, definimos los valores indicados con los siguientes criterios:

- **Real MTD** (*month to date*): las adquisiciones reales del mes al cual refiere Acquisition_Date, registradas hasta la fecha Acquisition_Date de dicho registro.
- **Prediction MTD** (*month to date*): lo que el modelo ha predecido para el mes al cual refiere Acquisition_Date, hasta la fecha Acquisition_Date de dicho registro.
- **Prediction EOM** (*end of month*): lo que el modelo ha predecido para el mes completo al cual refiere Acquisition_Date, es decir, hasta el último día del mes.
- **Real vs FC Delta Day**: la diferencia porcentual entre las adquisiciones registradas en la realidad y las proyectadas por el modelo. Calculado como: Real / Prediction -1. Sirve para entender qué tan por encima/debajo de lo esperado quedó el negocio ese día.

- **Real vs FC Delta MTD**: la diferencia porcentual entre lo realmente adquirido hasta el día del mes en cuestión y lo proyectado hasta esa misma fecha. Calculado como: Real MTD / Prediction MTD - 1. Sirve para entender si se está por encima/debajo de lo que se esperaba adquirir hasta ese día del mes. Suele responder a la pregunta del negocio: "¿Nos estamos rezagando/quedando cortos?".
- **Real vs FC Delta EOM**: la diferencia porcentual entre lo realmente adquirido hasta el día del mes en cuestión y lo proyectado para el mes completo. Calculado como: Real MTD / Prediction EOM -1. Sirve para responder a la pregunta del negocio: "¿Cuánto nos falta?"

En función de estos nuevos conocimientos aportados es que el negocio puede entonces tomar decisiones y actuar a tiempo (por ejemplo, si se nota que se están quedando rezagados respecto del pronóstico, ajustar la estrategia para impulsar las ventas). La detección temprana, entonces, de este tipo de alarmas, sirve para tomar una decisión y torcer el rumbo antes de que termine el mes y sea demasiado tarde.

10.2. Planilla mensual de FTEs necesarios por tarea

Por otro lado, el presente trabajo se propuso establecer un procedimiento automático para la determinación de la cantidad de FTEs necesarias en cada tarea relacionada a las adquisiciones.

En base a conversaciones y reuniones con expertos en cada uno de estos procesos del negocio se obtuvo que:

- *Quality Check*:
    - Es un proceso realizado por un equipo regional (el mismo personal atiende a todas las cuentas de LATAM).
    - El equipo trabaja sobre cuentas tanto de *Field* como de *SSU*, sin embargo, se estima que el tiempo dedicado a una cuenta *SSU* es un 40% de lo dedicado a una *Field*, debido a que, gracias a la registración por parte del partner en el canal digital, gran parte de este proceso es hecho por el mismo partner. En cambio, en el canal físico, es el agente de *Quality Check* quien realiza el 100% del proceso.
    - El volumen de cuentas promedio que puede manejar un agente de *Quality Check* al mes es de 550.

- *Menu Processing*:
    - Es un proceso también realizado por un equipo regional.

  - Solo atienden cuentas de *Field* (*SSU* es completamente gestionado por el mismo partner, por el canal digital).
  - El volumen de cuentas promedio que puede manejar un agente de *Menu Processing* al mes es de 220.

- *Onboarding*:
  - No hay un equipo regional, sino personal dedicado por cada mercado en particular, para atender las cuentas relativas a ese país en específico.
  - Atienden, principalmente, cuentas de *Field*. En pocas ocasiones, se ven obligados a atender también algunas cuentas de *SSU*. Esto último sucede cuando el partner se queda trabado en medio del proceso digital y no sabe o no puede avanzar. En ese escenario, el agente de *Onboarding* asiste al partner que está entrando por el canal digital del mismo modo que lo haría con uno de *Field*. Se estima que tan solo el 5% de las cuentas de *SSU* experimentan este tipo de inconvenientes que requieren ayuda del agente de *Onboarding*.
  - El volumen de cuentas promedio que puede manejar un agente de *Onboarding* al mes es de 180.

Por lo tanto, según los insights recopilados, necesitamos un *output* a nivel regional para *Quality Check* y *Menu Processing* y uno a nivel mercado para *Onboarding*.

A fines prácticos, consideramos *Must Have* como *Field*, dado que esta distinción sólo es útil a la hora de la estimación de vendedores y del pronóstico en sí, pero luego los procesos que siguen este tipo de adquisiciones se comportan del mismo modo que cualquier otra adquisición de *Field*.

Se provee, entonces, de una tabla dinámica inicial (Figura 70), la cual explica la cantidad de adquisiciones proyectada para el mes. Esto sirve solo a título informativo y para proveer contexto al negocio.

Month	2024-02-01

Acquisitions Forecast	AccountSource			
Country	Field	Must Have	SSU	Total general
Argentina	589	27	429	1045
Bolivia	167	0	73	240
Chile	139	21	345	505
Costa Rica	41	3	31	75
Ecuador	220	8	145	373
El Salvador	111	8	21	140
Guatemala	157	11	29	197
Honduras	62	1	17	80
Nicaragua	36	0	23	59
Paraguay	99	2	87	188
Uruguay	41	1	20	62
Venezuela	80	0	40	120
Panamá	24	0	23	47
Perú	476	28	200	704
República Dominicana	102	0	112	214
Total general	**2344**	**110**	**1595**	**4049**

Figura 70 - Adquisiciones mensuales esperadas por país y canal

A continuación, se proporciona una tabla resumen (Figuras 71 y 72) con los FTEs necesarios para cada proceso, en la cual:

- **Cuentas por FTE**: proviene de los ya mencionados datos proporcionados por el mismo negocio.
- **FTEs por cuenta**: consiste en el cálculo 1 / Cuentas por FTE. El sentido de negocio es qué cantidad de trabajadores necesito para procesar 1 cuenta (todo en términos mensuales).
- **FTEs necesarios en el mes**: el *output* final que se busca, cuántos FTEs necesito para cada proceso en el mes. En el caso de:
    - *Quality Check*: se calcula como la suma del total de cuentas *Field* + *Must Have* proyectadas para el mes, multiplicado por los FTEs por cuenta, más la suma total de cuentas *SSU* proyectadas para el mes por un 40% de los FTEs por cuenta. Por ejemplo, para Febrero se necesitan 5.62 FTEs = (2344+110)*0.001818 + 1595*0.001818*0.4
    - *Menu Processing*: se calcula como la suma del total de cuentas *Field* + *Must Have* proyectadas para el mes, multiplicado por los FTEs por cuenta. Por ejemplo, para Febrero se necesitan 11.15 FTEs = (2344+110)*0.004545
    - *Onboarding*: el cálculo es a nivel mercado.

Proceso	Equipo	Cuentas por FTE	FTEs por cuenta	FTEs necesarios en el mes
Quality Check	Regional LATAM	550	0.001818	5.62
Menu Processing	Regional LATAM	220	0.004545	11.15
Onboarding	Por país	180	0.005556	Tabla por país

Figura 71 - Output #2: FTEs necesarios

En el caso, de *Onboarding*, se presenta una tabla a nivel país (Figura 60) con los FTEs necesarios del mes. Los mismos se calculan como la suma de de cuentas *Field* + *Must Have* proyectadas para el mes para dicho país, multiplicado por los FTEs por cuenta, más un 5% de las cuentas *SSU* proyectadas para el mes para dicho país, por los FTEs por cuenta. Por ejemplo, para Febrero, en Argentina, se necesitan 3.54 FTEs = (589+27)*0.005556 + 429*0.05*0.005556.

Country	Onboarding FTEs Feb
Argentina	3.54
Bolivia	0.95
Chile	0.98
Costa Rica	0.25
Ecuador	1.31
El Salvador	0.67
Guatemala	0.94
Honduras	0.35
Nicaragua	0.21
Paraguay	0.59
Uruguay	0.24
Venezuela	0.46
Panamá	0.14
Perú	2.86
República Dominicana	0.60

Figura 72 - FTEs necesarios Onboarding

De este modo, y solo pegando el *output* mensual del modelo en un archivo de Excel, la solución no solamente prepara el análisis real vs target, sino el estimado de FTEs necesarios para cada proceso, de modo que tanto los equipos locales como los regionales tienen la capacidad de planificar el mes, asignar y reasignar tareas en función de lo obtenido y, en caso de ser necesario, obtener nuevos recursos (FTEs) o ceder otros a otras tareas cuando se detecten discrepancias significativas entre lo necesario y lo disponible.

11. Reflexiones finales

En la presente tesis se ha pasado por distintos procesos con el objetivo de llegar a nuestro *output* final. Se comenzó planteando el problema: la necesidad de, en un negocio regido por la cantidad, conocer la cuantía de adquisiciones esperada en los distintos mercados y canales. La inexistencia de un número contra el cuál comparar los resultados parciales del mes, de forma de ir teniendo señales de qué tan bien/mal están yendo las ventas. Por otro lado, la imposibilidad de planificar de manera eficiente los equipos de trabajo de *Quality Check*, *Menu Processing* y *Onboarding*.

Se consultó la literatura existente sobre el tema y se descubrió que no hay mucho escrito acerca del nicho específico de pronóstico de nuevos partners en un negocio como el del delivery de comida.

Con lo aprendido de la bibliografía, se continuó realizando un análisis exploratorio de la data e, iterando varias veces, se llegó a la base de datos final, ajustando en repetidas ocasiones los criterios de medición y de agregación de la data, para mejorar la calidad del *input* que se le da al modelo.

Habiendo obtenido información útil del análisis exploratorio, se probaron distintas estrategias de pronóstico, pasando por ARIMA, SARIMA, ETS, Prophet, XGBoost, Light GBM y experimentando con distintas variantes de cada uno de ellos, para ver cuál se adapta mejor a la realidad del negocio en análisis.

Posteriormente, se llegó a encontrar el mejor modelo y se realizaron predicciones con el mismo. Se obtuvieron conclusiones interesantes que unen las ventajas del modelo con las necesidades del negocio y las particularidades de la data disponible.

Finalmente, con el pronóstico ya realizado, se elaboró la solución que se le presenta al negocio: se pasa de la complejidad de lo mencionado y se vuelve a la simplicidad de la necesidad del negocio. De este modo, se diseña una hoja de cálculo de excel, fácil de usar y de entender a nivel gerencial, que se puede alimentar con los *outputs* del modelo y, de manera automática, responder a las preguntas que se habían planteado inicialmente.

Existen algunos aspectos en los cuales se podría profundizar el estudio iniciado en la presente tesis. Uno de ellos tiene que ver con los *riders* (o repartidores) de las aplicaciones de *delivery*. Se ha relacionado la cantidad de adquisiciones con los recursos humanos necesarios directamente relacionados con el proceso de adquisición de partners, obviando el potencial impacto en otras áreas de la organización, incluyendo el personal de repartidores. ¿Tendría impacto, la cantidad de partners disponibles en la aplicación, en la demanda? ¿Impactaría, ello, de manera consecuente, en la cantidad de repartidores necesarios para atender a dicha demanda? Este tipo de preguntas depende de si uno considera que la demanda en las aplicaciones de *delivery* de comida depende de la variedad de oferta disponible (y en

qué medida). En cualquier caso, a los efectos de este trabajo, no se dispone de suficientes datos como para proveer respuestas a estas preguntas, las cuales serían interesantes para un posterior trabajo de investigación.

Otro de los aspectos sobre los cuales, en un futuro, se podría profundizar es el modelado de casos no exitosos en la adquisición de partners. El presente trabajo se enfoca únicamente en la adquisición exitosa de partners, pero se ignora por completo el fenómeno de las adquisiciones que quedan estancadas durante el proceso y no se llegan a concretar. Para estudiar y modelar este tipo de casos, sería necesario contar con algún tipo de medida de performance del proceso de adquisición con la que hoy no se cuenta, por lo que el trabajo se enfoca sobre la adquisición exitosa de partners, quedando como futura investigación la no exitosa.

Como nota final, queda mencionar que si bien se llegó a un resultado, lo positivo del presente caso es que, a medida que transcurre el tiempo, nueva data se va generando día a día. La misma será de mucha ayuda para aumentar el volumen de la base de datos inicial, lo cual, en el tiempo, aportará a un rendimiento cada vez mejor, derivando en resultados más precisos para tomar decisiones con más claridad y seguridad.

12. Bibliografía

1. Adhikari y Agrawal (2013). *An Introductory Study on Time Series Modeling and Forecasting*. [online] arXiv.org. Available at: https://arxiv.org/abs/1302.6613.
2. Asana (2023). *Is your business healthy? Sales forecasting can help.* [online] Asana. Available at: https://asana.com/es/resources/sales-forecast-template.
3. Bivona, E. (2022). Determinants of performance drivers in online food delivery platforms: a dynamic performance management perspective. *International Journal of Productivity and Performance Management*. doi:https://doi.org/10.1108/ijppm-10-2021-0606.
4. Capote Pérez, A.E. (2022). Estimación de las ventas de frutos secos. Un caso de estudio: La Gaviota Alimentación. *riull.ull.es*. [online] Available at: https://riull.ull.es/xmlui/handle/915/28704 [Accessed 21 Jun. 2024].
5. Chen, T. y Guestrin, C. (2016). XGBoost: a Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, pp.785–794. doi:https://doi.org/10.1145/2939672.2939785.
6. De Arce, R., y Mahía, R. (2003). Modelos Arima. *Programa CITUS: Técnicas de Variables Financieras*, 5-6.
7. Gardner, E.S. (1985). Exponential smoothing: The state of the art. *Journal of Forecasting*, 4(1), pp.1–28. doi:https://doi.org/10.1002/for.3980040103.
8. Hyndman, R. J., & Khandakar, Y. (2008). *Automatic time series forecasting: the forecast package for R. Journal of statistical software*, 27, 1-22.
9. Isa, N.F., Yusof, N.M.Y., Akhir, I.M. and Osman, S. (2021). The Effect of Consumer Experience on Food Delivery Apps. *International Journal of Academic Research in Business and Social Sciences*, 11(13). doi:https://doi.org/10.6007/ijarbss/v11-i13/8549.
10. Jaimes Campos, D.L. y López Zúñiga, E. (2021). Modelo de Forecast para predecir la demanda semanal de alimentos y bebidas de consumo masivo. *repositorio.uniandes.edu.co*. [online] Available at: https://repositorio.uniandes.edu.co/entities/publication/83fbf165-4b6a-4ebd-8100-28745d0b8c0e [Accessed 21 Jun. 2024].
11. Jia, H., Shen, S., Ramírez García, J.A. and Shi, C. (2022). Partner with a Third-Party Delivery Service or Not? A Prediction-and-Decision Tool for Restaurants Facing Takeout Demand Surges During a Pandemic. *Service Science*. doi:https://doi.org/10.1287/serv.2021.0294.
12. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. y Liu, T.-Y. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *Advances in Neural Information Processing Systems*, [online] 30. Available at:

https://proceedings.neurips.cc/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html.

13. Kerem Gülen (2022). *How is machine learning utilized for time series forecasting?* Dataconomy. [online]. Available at: https://dataconomy.com/2022/11/25/time-series-forecasting-machine-learning/.
14. Mejía Tovar, M.P. (2023). Modelo para el Forecast de una plataforma de Fast Delivery en Colombia. *repositorio.uniandes.edu.co*. [online] Available at: https://repositorio.uniandes.edu.co/entities/publication/471f50f2-034c-41fc-bff5-e81ea5181533/full.
15. Optisol (2023). *Top 5 Machine Learning Techniques for Sales Forecasting*. [online] OptiSol. Available at: https://www.optisolbusiness.com/insight/top-5-machine-learning-techniques-for-sales-forecasting#:~:text=How%20machine%20learning%20helps%20in [Accessed 7 Apr. 2024].
16. Pinzon Villanueva, L.M. (2023). Análisis comparativo de optimización de hiperparámetros por búsqueda en grilla y algoritmos genéticos para forecasting de XGBoost en ventas online. *CITAS: Ciencia, innovación, tecnología, ambiente y sociedad*, [online] 9(2), p.3. Available at: https://dialnet.unirioja.es/servlet/articulo?codigo=9177033 [Accessed 20 Jun. 2024].
17. Rock Content - ES. (2019). *Forecast de ventas: ¿qué es y cómo hacer uno con precisión?* [online] Available at: https://rockcontent.com/es/blog/forecast-de-ventas/.
18. Surendhranatha Reddy, C. and Aradhya, Dr.G.B. (2020). Driving Forces for the Success of Food Ordering and Delivery Apps: A Descriptive Study. *International Journal of Engineering and Management Research*, 10(02), pp.131–134. doi:https://doi.org/10.31033/ijemr.10.2.15.
19. Taylor, S.J. y Letham, B. (2017). *Forecasting at scale*. [online] peerj.com. Available at: https://peerj.com/preprints/3190/#.
20. Weingertner, G. (2023). *Time Series Forecasting with Facebook's Prophet in 10 Minutes — Part 1*. [online] Medium. Available at: https://towardsdatascience.com/time-series-forecasting-with-facebooks-prophet-in-10-minutes-958bd1caff3f.