Universidad de Valparaíso
CHILE

Facultad de Ciencias
Instituto de Estadística
Ingeniería en Estadística

Motor selector de técnicas de *machine learning* para la detección de anomalías de consumo eléctrico en clientes residenciales

ÁLVARO FRANCISCO FERNÁNDEZ ESCOBAR

Profesor Guía

Rodrigo Salas Fuentes, Ph.D.

Profesor Co-Guía

Cristóbal Roco Saavedra.

Proyecto de titulación para optar al:

título profesional de: *Ingeniero en Estadística*
minor en: *Minería de Datos*
grado académico en: *Licenciado en Estadística*

23 de diciembre de 2021

Agradecimientos

En primer lugar, quiero agradecer a aquellas personas que me permitieron tener una mejor vida y brindarme una educación tanto escolar como universitaria,a mis padres: Paola Escobar González y Marcos Fernández Alcayaga, sin ellos, sin sus palabras de aliento y apoyo incondicional, esto no hubiera sido posible, convirtiéndolos en los mayores responsables de que hoy culmine una etapa de mi vida de manera exitosa.

También a mis abuelos, Maria Teresa González Calderón y Orlando Escobar Rivera, quienes me criaron en mi niñez, formando mis valores como persona y, desde ese entonces, siempre han creído en mi y apoyándome cuando lo necesitaba.

Al cuerpo académico formado para mi proyecto de titulación; Rodrigo Salas y Cristóbal Roco, quienes me han guiado en este desafío tan complejo, por su disponibilidad y dedicación, muchas gracias.

A mi gran amor incondicional que empezó en mi época universitaria, Camila Ruz Olivares quien fue un pilar fundamental en esta etapa, mi flecha guía, mi luz, nunca faltaron sus palabras sabias en momentos en cuando me sentí en un hoyo, agradecer su perseverancia de esperarme en momentos difíciles cuando por temas académicos, el tiempo no estaba de nuestro lado, espero seguir compartiendo alegrías y metas contigo.

A aquellos personajes que hicieron mi día a día más divertido dentro de este ciclo, Luis Altamirano Pizarro y Angello Canales Olivares, espero poder seguir compartiendo alegrías y logros importantes, son personas que marcaron mi vida y se convirtieron en mi familia, llevando momentos buenos y malos, noches de estudio, frustración y felicidades, nos queda la etapa de vida más larga en donde de seguro compartiremos.

Finalmente, y no menos importante, a alguien que fue mi guía en estos años universitarios, un segundo padre, que siempre tuvo disponibilidad, apoyo y la palabra precisa hacia mi persona, el doctor Carlos Henríquez Roldán, gracias por confiar en mí y ayudarme cuando lo necesité.

Resumen

Las empresas distribuidoras de electricidad, se recogen una enorme cantidads de datos sobre sus clientes en particular, producidas mensualmente por medidores ubicados en cada propiedad. Cada cliente tiene un comportamiento diferente, lo que se verá reflejado en cada una de las categorías que se le atribuye a los clientes respectivos. Esta información es analizada para aportar a las empresas distribuidoras de electricidad, mediante la detección de consumos de energía anómalos. Estos consumos se conocen como órdenes de crítica. De esa forma, poder entender su origen y tomar decisiones al respecto. Entonces, una orden de crítica se refiere a aquella solicitud para analizar un cliente, el cual posee un consumo superior a los anteriores, bajo ciertas condiciones determinadas por la empresa. En ese sentido, este proyecto está basado en la selección de un modelo de *machine learning*, a través de un motor selector que distingue la mejor metodología que se ajuste al conjunto de datos.

Dado ese contexto, son tres metodologías de predicción seleccionadas: suavización exponencial, boosting y redes neuronales. El primero trata del modelo predictivo Holt-Winters. El segundo, conocido como *Extreme Gradient Boosting (XGBOOST)*, reacondiciona y es más efectivo que otros modelos gracias a la optimización de sistemas y mejoras algorítmicas. Mientras que el modelo de redes neuronales está basado en redes *Long-Short Term Memory (LSTM)*, tiene la cualidad de detectar variaciones de comportamiento en el núcleo familiar.

Abstract

Within the electricity distribution companies, huge amounts of data are collected monthly produced by meters belonging to each property, which each have different behaviors that will be reflected in each of the categories attributed to the respective customers. This information is analyzed to contribute to electricity distribution companies by detecting anomalous energy consumption, in this case, known as critical orders, in order to understand the origin of this anomaly and make decisions about it. So, a criticism order refers to that request to analyze a customer, which has a higher consumption than previous consumption under certain conditions determined by the company. In this sense, this project is based on the selection of the machine learning model through a selector engine that distinguishes the best methodology that fits the data set.

Given this context, there are three forecasting methodologies to select; exponential smoothing, boosting and neural networks. The first deals with the Holt-Winters predictive model. The second, Extreme Gradient Boosting (XGBOOST), which reconditions these models, improving them through system optimization and algorithmic improvements. While the neural network model is based on textit Long-Short Term Memory (LSTM) networks where it has the ability to detect behavioral variations in the family nucleus.

Índice general

Índice de figuras

Índice de cuadros

Glosario

- **OC:** Órden de crítica
- ***ML:*** *Machine Learning*
- ***XGBoost:*** *Extreme Gradient Boosting*
- ***LSTM:*** *Long-Short Term Memory*
- ***LE:*** *Learning Element*
- ***PE:*** *Performance Element*
- ***RNN:*** Redes Neuronales Recurrentes
- ***RMSE:*** *Root mean squared error*

Capítulo 1

INTRODUCCIÓN

1.1. Descripción del problema

La problemática para la empresa surge cuando requiere de optimizar el tiempo de los operarios pertenecientes a la Subgerencia de Operaciones y Controles de Servicios, considerando el importante volumen de órdenes de crítica (OC) generadas mensualmente. Esta cantidad se atribuye a un mal rendimiento del modelo utilizado para tratar los consumos de los clientes masivos y potencia contratada, el cual estaría sobregenerando o sobredetectando variaciones de consumos. Este problema se presenta en la Fig.1.1, en donde se posee un mínimo de OC de 17.500 clientes, aproximadamente, quienes deben ser atendidos por un grupo reducido de operarios.

Es necesario precisar que abril de 2020 y enero de 2021, no fueron tomados en cuenta para el análisis descriptivo, debido a factores externos que influenciaron la detección de consumos anómalos. En el caso de abril, el efecto COVID-19 trajo consigo problemáticas de lecturas de medidores, dado las cuarentenas obligatorias propiciadas por el Estado. Por lo tanto, la mayoría de los clientes fueron estimados implicando una notoria disminución de OC. Por otro lado, en enero, una actualización del software de la empresa trajo consigo errores para la detección de variaciones de consumo, el cual significó un volumen irreal de OC, cuyo valor no figura en la gráfica para efectos de visualización.

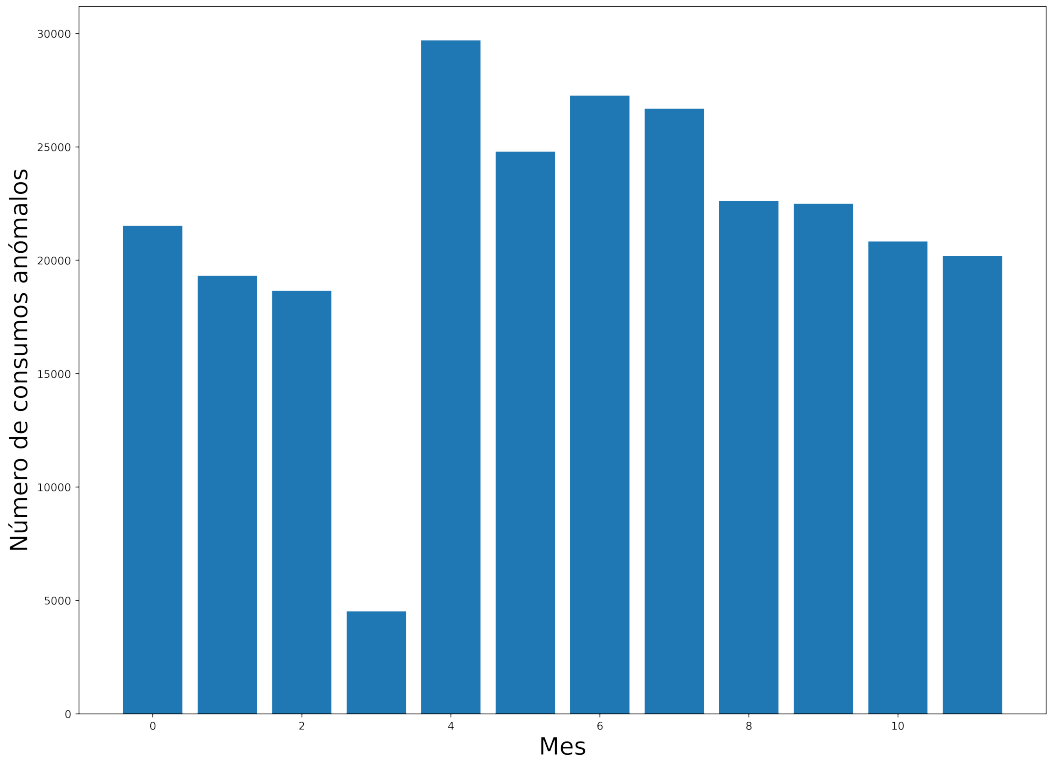

Figura 1.1: Comparación del volumen de OC 2020 vs 2021.
Fuente: Elaboración propia.

Sobre la base de lo mencionado, se utilizarán metodologías de *machine Learning* (ML) para solventar los problemas de tiempo en los operarios, mediante la mejora de los sistemas de detección de consumos anómalos a través de modelos de *forecasting*: *Holt Winters*, *XGBoost* y *LSTM*. Para el presente trabajo de titulación, el conjunto de datos fue facilitado por una empresa distribuidora de electricidad de Chile, para ser estudiada para efectos de aplicación de los modelos propuestos. Los datos utilizados en la investigación son confidenciales.

1.2. ESTADO DEL ARTE

Li y cols. (2019) proponen un modelo de pronóstico combinado basado en *LSTM* y *XGBoost*. En primer lugar se entrenan modelos XGBoost y LSTM respectivamente para predecir la carga de la potencia mediante una metodología recíproca del error para combinar los resultados de ambos modelos.

Fenza, Gallo, y Loia (2019) desarrollan un modelo *RNN* basado en *LSTM* para perfilar y pronosticar los comportamientos de consumo de los usuarios finales, usando sus datos de consumo recientes y pasados. Los investigadores se centran en la necesidad de desarrollar una metodología de detección de anomalías que sea capaz de detectar cambios estructurales familiares u hogares.

El seguimiento continuo de los errores de predicción del consumo permite distinguir entre posibles anomalías y cambios en el comportamiento normal que corresponden a diferentes motivos de error. Los resultados experimentales comprueban su aptitud, al detectar anomalías después de un periodo de entrenamiento determinado.

Jiang, Wu, Gong, Yu, y Zhong (2020) crean un modelo predictivo basado en el suavizado exponencial de Holt-Winters, dejando pronosticar con precisión series periódicas con relativamente pocas muestras de entrenamiento, el cual, con algoritmos de optimización *fruit fly*, seleccionan mejores parámetros de suavizado para el modelo Holt-Winters. Esta propuesta fue utilizada con datos de consumo eléctrico de una ciudad de China, obteniendo buenos resultados, inclusive con pocas muestras de entrenamiento.

Almazrouee, Almeshal, Almutairi, Alenezi, y Alhajeri (2020) presentan una investigación sobre el desempeño del modelo Prophet en el pronóstico de carga máxima a largo plazo de Kuwait comparándolo con el modelo clásico Holt Winters evaluando la viabilidad y precisión de ambos para la redicicón de cargas máximas a largo plazo.

1.3. Propósito de la investigación

1.3.1. Hipótesis

Mediante el estudio de variables relacionadas al comportamiento de consumo de energía de clientes de una empresa distribuidora de electricidad, se logrará detectar clientes que posean alzas de consumos anómalas de manera más eficaz, a través de un modelo de aprendizaje automático que será escogido por un motor selector.

1.3.2. Objetivo general

Desarrollar un mecanismo automatizado que seleccione el mejor modelo de *machine learning* que detecte las alzas de consumo anómalas de manera eficiente.

1.3.3. Objetivos específicos

1. Realizar un estudio relacionado a la industria eléctrica que permita identificar y comprender variables que influyen en el consumo del cliente.
2. Desarrollar modelos de machine learning para la detección de anomalías.
3. Comparar el desempeño de las técnicas de *machine learning* propuestas.

Capítulo 2

MARCO TEÓRICO

2.1. Aprendizaje automático

El aprendizaje automático se ha convertido en una de las ramas de inteligencia artificial más importantes y prolíficas. No es de extrañar que día a día sus aplicaciones estén cada vez más extendidas en todos los sectores de negocio, siempre con nuevas y más potentes herramientas y resultados para ocupar estas metodologías (Bonaccorso, 2017). Esto posibilita obtener resultados eficaces que permitan a las empresas una mejor toma de decisiones, encontrando patrones naturales provenientes de los conjuntos de datos. De esta manera, se genera información adicional que contribuye a solucionar un problema complejo que involucra grandes cantidades de datos y muchas variables, pero sin una ecuación existente.

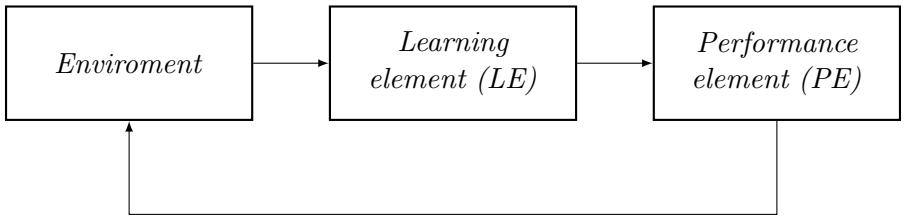

Figura 2.1: Modelo genérico de aprendizaje automático
Fuente: Yao y Liu (2014)

Para facilitar la conexión con el proyecto de investigación, los modelos a tratar están sujetos a una estructura genérica (Fig. 2.1). Estos poseen dos componentes claves: el elemento de aprendizaje (LE) y el elemento de desempeño (PE). El entorno de aprendizaje automático brindará la información al LE para luego, mediante la información aprendida, modificar el PE que permitirá que este último tome mejores decisiones (Yao y Liu, 2014). En otras palabras, los modelos a utilizar recogerán información valiosa de los clientes que se les brinde y aprenderán de esta, logrando identificar patrones para llegar a resoluciones eficientes. En este caso, clasificar o detectar si el consumo del mes a tratar es anómalo o no.

2.2. Orden de crítica (OC)

Para comprender en profundidad el proyecto y la problemática en donde ocurre, es necesario precisar definiciones y procesos técnicos procedentes al rubro. Se atribuye una orden de crítica a aquel cliente que posea un consumo anómalo. Es así que es definido como una variación positiva en

KWh respecto a una serie de condiciones dadas por la empresa. Por lo tanto, este último pasará a ser analizado a través de actividades de control, que permitirán comprender el origen de la variación.

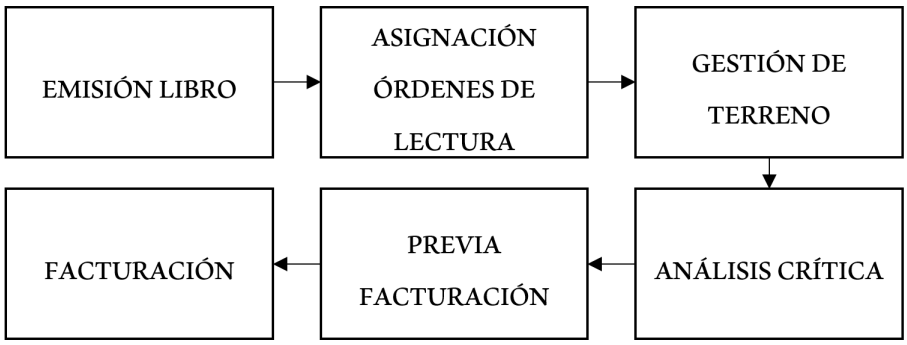

Figura 2.2: Proceso de lectura y facturación de consumos eléctricos
Fuente: Elaboración propia

En este proceso (Fig. 2.2) se gestionan las actividades mediante las cuales los clientes reciben sus facturas o boletas, que representan los consumos del periodo correspondiente.

Las actividades de lectura comienzan desde que se emite un libro que genera automáticamente las órdenes de lecturas por ciclos (un ciclo se refiere a un plan estratégico desarrollado por la empresa que permite al lector tomar una mejor ruta). Luego, son asignadas a cada lector mediante una gestión de terreno. Este procedimiento considera la captura del consumo en equipos de medicion; el análisis del valor numérico, para verificar magnitud, rangos y evitar consumos erróneos; y que las lecturas abarquen la totalidad de los clientes. Finalmente, en la facturación se toman en cuenta las actividades de valorización en la obtención de los consumos, acorde a tarifas autorizadas por la entidad reguladora y contratos de cada cliente, respectivamente.

El proceso de lectura se verifica dentro de la Subgerencia de Centros de Servicio y Operaciones, para evitar reclamos ocasionados por cobros excesivos como también, validar consumos atípicos. Cabe destacar que este proyecto considera a los clientes tipo masivo y de potencia contratada. Dichas clasificaciones surgen a partir de la tarifa contratada por estos mismos, las que están regidas por un ente regulador que las pre-definen. De esta manera, incluye las tarifas BT-1 para el primer caso y, por otra parte, AT-4 y BT-2 (Chilquinta, 2021). Estas se definen como:

Clientes masivos

- **BT-1:** opción de tarifa simple, en baja tensión, para clientes residenciales con medición de energía que cuente con potencia conectada inferior o igual a 10 KW o con un imitador de potencia que permita cumplir esta condición. Aplica a clientes abastecidos por concesionarias cuya demanda máxima anual de consumos se produce en meses en que se han definido horas de punta y a clientes abastecidos por empresas cuya demanda máxima anual de consumos se produce en meses en que no se hayan definido horas de punta y cuyo factor de clasificación sea igual o inferior a 2,5.

Clientes potencia contratada

- **AT-4:** opción de tarifa horaria en alta tensión, para clientes con al menos medición de energía y demanda máxima de potencia contratada o leída, y demanda máxima de potencia contratada

o leída en horas de punta del sistema eléctrico. En esta opción existen tres modalidades de medición: AT4.1, medición de la energía mensual total consumida, y contratación de la demanda máxima de potencia en horas de punta y de la demanda máxima de potencia. AT4.2, Medición de la energía mensual total consumida y de la demanda máxima de potencia leída en horas de punta, y contratación de la demanda máxima de potencia. AT4.3, medición de la energía mensual total consumida, de la demanda máxima de potencia leída en horas de punta y de la demanda máxima de potencia suministrada.

La demanda máxima de potencia que contrate el cliente deberá ceñirse a las capacidades de limitadores disponibles en el mercado. Se entenderá por demanda máxima de potencia leída del mes como el más alto valor de las demandas integradas en periodos sucesivos de 15 minutos. La demanda máxima de potencia de cada hora corresponderá al máximo valor de los registros leídos que se encuentren dentro de esta.

- **BT-2:** opción de tarifa en baja tensión con potencia contratada para clientes con al menos medición de energía y potencia contratada. Los clientes que decidan optar por la presente tarifa, podrán contratar libremente una potencia máxima con la respectiva concesionaria, la que regirá por un plazo de 12 meses. Durante dicho periodo, los consumidores no podrán disminuir ni aumentar su potencia contratada sin el acuerdo de la concesionaria. Al término de la vigencia anual de la potencia contratada, los clientes tienen la opción de adquirir una nueva potencia. Los consumidores podrán utilizar la potencia contratada sin restricción en cualquier momento durante el período de vigencia de dicha potencia y deberá ceñirse a las capacidades de limitadores disponibles en el mercado.

Actualmente, el algoritmo genera una orden de crítica a aquellos clientes de tarifas mencionadas anteriormente, que posean un consumo superior a un delta-consumo de 300 KWh, para que, finalmente, sean gestionados mediante operarios atribuidos a estos procesos. De esta forma, se asegura que las lecturas y consumos a cada cliente sean exactos, existiendo una correlación lógica en sus cuentas respectivas, acorde al consumo y su contrato. No obstante, existen factores que permiten a las OC ser facturadas directamente, sin pasar por un análisis exhaustivo. Estos son:

- **Factor 1:** todas las órdenes de crítica generadas que no tienen recuperación de consumo de meses anteriores, no poseen solicitudes de reclamos asociadas en los últimos 6 meses y que su factor respecto al mismo periodo del año anterior sea menor o igual a 2,5 y mayor a 0, se facturan.

- **Factor 2:** todas las órdenes de crítica que generan consumos menores o igual a 500 Kwh sin meses de recuperación, no poseen órdenes de reclamos asociadas en los últimos 6 meses y que su factor de consumo respecto al mismo periodo año anterior es menor o igual a 3,5, y mayor a 0, se facturan.

- **Factor 3:** todas las órdenes de crítica generadas que no tienen recuperación de consumo de meses anteriores, no poseen solicitudes de reclamos asociadas en los últimos 6 meses y que su factor respecto al consumo del mes anterior sea menor o igual a 2,5 y mayor a 0, se facturan.

- **Clientes retirados:** todos aquellos clientes que presenten en el estado de producto una condición de “retirado”.

El proceso culmina en la entrega de estos registros al proveedor para efectuar la impresión de boletas y facturas distribuidas a las oficinas. Posteriormente, son retiradas por los contratistas que realizan la distribución física o mediante vías electrónicas.

2.2.1. Tipos de consumos anómalos

Para poder entender y abarcar de mejor forma el concepto de anomalía de consumo, es necesario definir clasificaciones referentes a las órdenes de crítica. Dentro de la empresa, en cuestión, se clasifican y definen algunos consumos atípicos:

- **Anómalo estacional:** es aquel cliente que presenta alzas de consumo en períodos estivales, como lo son Navidad, Año Nuevo, invierno o verano; ya que, presentan altas cantidades de consumos adicionales al comportamiento habitual.
- **Sin promedio:** pertenece a la categoría de clientes que poseen menos de seis meses integrados a la empresa distribuidora de electricidad, por lo que se desconoce si posee consumos atípicos.
- **Carta sobreconsumo:** siendo uno de los más comunes, tal como dicta su nombre, en esta categoría pertenecen todos aquellos clientes que tuvieron un consumo excesivo en períodos habituales, donde no debiesen existir estas alzas.

2.3. Clustering

Consiste en una técnica de aprendizaje automático que en los últimos años se ha vuelto muy atractiva para su implementación dado la simplicidad que esta posee. Se le considera como un método de aprendizaje automático no supervisado, debido a que, con el fin de medir su rendimiento no se tiene una salida predefinida con anterioridad que permita compararla con el resultado brindado por el algoritmo en cuestión.

Esta metodología agrupa datos, en este caso, observaciones de consumos de energía en varios grupos, permitiendo clasificar estos en anormales o normales. Para ello se evalúa la similitud entre las muestras midiendo las distancias entre los puntos en el espacio de medición (Lavine, 2006), siendo la distancia euclidiana una de las mediciones más comunes para variables continuas.

La elección de un algoritmo de agrupación depende tanto del tipo de datos disponibles como del propósito particular (Kaufman y Rousseeuw, 2009). En este proyecto, dada la literatura y la popularidad que posee el algoritmo demostrado posteriormente, se trabajará con un *cluster* con metodología de particionamiento.

Metodología de particionamiento

Este algoritmo básico consiste en construir agrupaciones clasificando la información acorde a similitudes que posean los datos. Dado un conjunto D expresado como $D = \{x_1, ..., x_i\}$, siendo x_i una observación $\in R^n$, se construyen k particiones; las cuales, cada una representa un clúster y una región en particular (Fig.2.3). Usualmente, los analistas o investigadores son los que deben especificar el número de clústeres que se deben generar, no obstante, esta investigación propone un método que define la cantidad de clústeres por sí solo según los hiperparámetros que condicionan dicha agrupación.

2.3.1. K-Means

Se considera uno de los algoritmos de agrupación más populares de la época debido a su simplicidad (Fenza y cols., 2019). Su nombre deriva en base a que el algoritmo encuentra una partición tal que se minimiza el error al cuadrado entre la media empírica de un grupo y los puntos en el

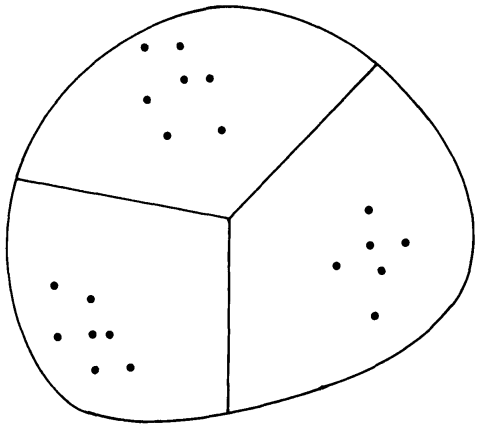

Figura 2.3: Metodología de partición
Fuente: Kaufman y Rousseeuw (2009)

grupo . A grandes rasgos, se determina un número inicial de K *clusters* que organiza las observaciones de entrada mediante iteraciones, en donde, cada observación se relaciona al centroide que se encuentre más cercano según la distancia que posean entre ambos. De manera más formal, teniendo $X = \{x_i\}, i = 1, ..., n$ las observaciones que serán agrupadas en K *clusters*, $C = \{c_k\}, k = 1, ..., K$. Siendo μ_i la media del cluster c_k, se minimiza la suma del error cuadrático medio para todos los K Clúster (Jain, 2010), lo cual, está definido como:

$$min \sum_{k=1}^{K} \sum x_i \in C_k ||x_i - \mu_k||^2 \quad . \tag{2.1}$$

En resumen, el algoritmo inicia en una partición inicial de K *clusters* aleatorios, asignando a las observaciones a su centroide más cercano, iterando deste último proceso hasta estabilizar los centroides de manera que las iteraciones ya no generen cambio alguno.

2.4. *Boosting*

Esta metodología proviene de la familia *Ensemble Methods*. Utiliza muchos *learners* para mejorar el desempeño de cada uno de ellos, individualmente. De este modo, se pueden describir como técnicas que utilizan un grupo de *weak learners* (aquellos que, en promedio, logran sólo resultados ligeramente mejores que un modelo aleatorio), con el fin de crear uno más fuerte y agregado. Su popularidad surge debido a la habilidad que se les reconoce de aumentar la estabilidad de un modelo, a través de la reducción de la varianza y el sesgo.

El modelo *Boosting* se refiere a un método general y demostrablemente efectivo para producir una regla de predicción muy precisa, mediante la combinación de reglas generales aproximadas y moderadamente inexactas (Schapire, 2003). Aquí, se presume la existencia de un algoritmo de aprendizaje débil, en donde, en consecuencia, produciría un clasificador o predictor débil, según ejemplos de entrenamientos que se utilicen.

El objetivo de *Boosting* es mejorar el rendimiento del algoritmo de aprendizaje débil, mientras se trata como una "caja negra" que se puede llamar repetidamente, como una subrutina, pero cuyas características no se pueden observar ni manipular (Schapire y Freund, 2012).

Algoritmo general de entrenamiento:

Es evidente que cada metodología perteneciente a la familia de *boosting*, tiene diferencias en cuanto al algoritmo de aprendizaje, sin embargo, Z-Ai (2020) resume de manera general, creando un patrón representativo:

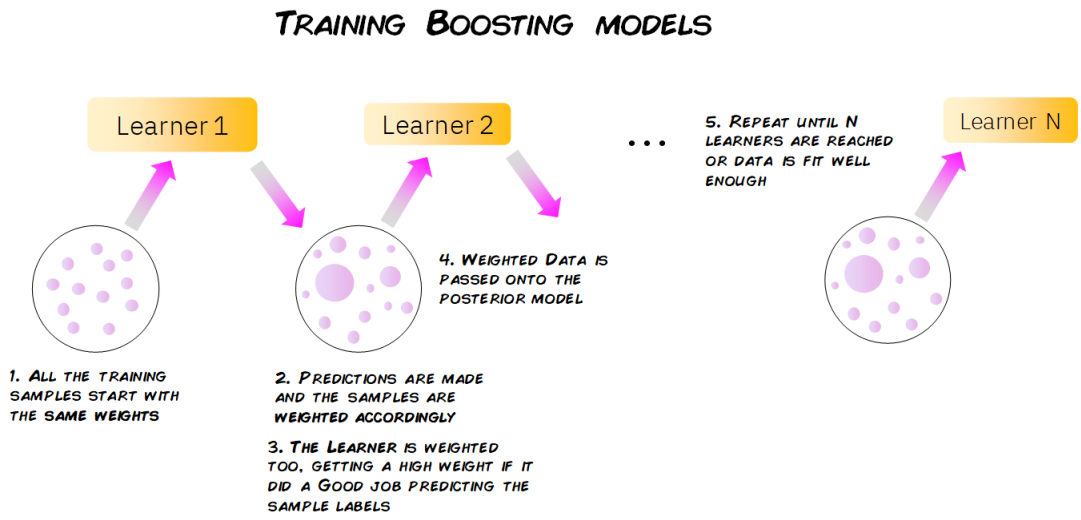

Figura 2.4: Algoritmo de entrenamiento general para modelos *Boosting*
Fuente: Z-Ai (2020)

En primera instancia, las muestras del modelo serán utilizadas para entrenar algún tipo de modelo individual, las cuales comienzan con las mismas ponderaciones (Fig 2.4). Luego, se calcula un error de predicción perteneciente a cada muestra, permitiendo aumentar los ponderadores provenientes de aquellas muestras que hayan tenido un mayor error. Estas, consecuentemente, se tornan más importantes para el entrenamiento del modelo individual siguiente. Por lo que, mientras más certeras sean las predicciones, mejor ponderación tendrán, lo que implica una mayor influencia en la predicción final. Todo este proceso ocurre para que, finalmente, los datos que fueron ponderados, repiten la lógica anterior de asignación de pesos. Esta acción se repetirá hasta que el error del modelo esté por debajo de un umbral predefinido. Como último paso, se realiza una predicción final por combinaciones de predicciones escaladas, según las ponderaciones de cada modelo individual.

2.4.1. *Extreme Gradient Boosting (XGBOOST)*

Diseñado por Chen y Guestrin (2016), es uno de los modelos de *Boosting* más popurales y utilizados dentro de la industria del *machine learning*. Se trata de una implementación escalable y precisa de árboles que son potenciados por gradientes, creada explícitamente para optimizar la velocidad computacional y el rendimiento del modelo (Zhao y Hryniewicki, 2018).

A diferencia de los modelos *Gradient Boosting Machine* tradicionales, *XGBoost* los reacondiciona, mediante una optimización de sistemas y mejoras algorítmicas (Morde, 2019), convirtiéndolo en una metodología eficaz, en relación al tiempo de ejecución (Fig. 2.5).

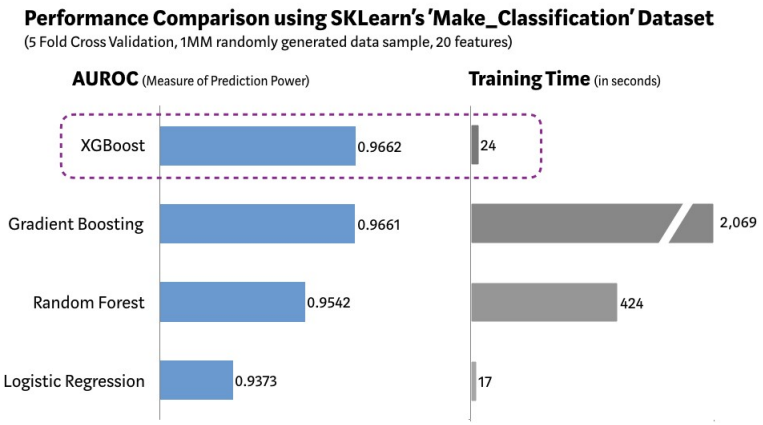

Figura 2.5: Comparación de eficiencia de *XGBoost* vs otros algoritmos
Fuente: Morde (2019)

Función objetivo

Se propone una función objetivo, la cual está compuesta por una función de pérdida (l) y un factor de regularización (Ω), en donde, la metodología tiene como objetivo minimizar el número de iteraciones (t) de la función en Ec. 2.2.

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y_i}^{(t-1)} + f_t(x_i)) + \Omega(f_t) \quad , \tag{2.2}$$

donde la función de pérdida (l) es convexa diferenciable, la cual mide la diferencia entre la predicción ($\hat{y_i}$) y el valor real (y_i). Siendo $\Omega(f) = \gamma T + \frac{1}{2}\lambda||w||^2$, la que penaliza la complejidad del modelo mediante el suavizamiento de los pesos finales aprendidos. Así, se evita el sobreajuste y, consecuentemente, se seleccionarán funcionan simples que proporcionan predicciones más acertadas.

En la ecuación anterior (Ec. 2.2), se puede notar que esta metodología incluye funciones en sus parámetros, las cuales no pueden ser optimizadas con métodos tradicionales en un espacio euclidiano. Por lo tanto, se utiliza la aproximación de Taylor para transformar la función objetivo original a una que pertenezca al dominio Euclideo. En este caso, se utiliza una aproximación de Taylor de segundo orden (Eq. **??**), permitiendo optimizar rápidamente la función objetivo. Entonces, los estadísticos de gradiente de primer y segundo orden de la función de pérdida son $g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y_i}^{(t-1)})$ y $h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y_i}^{(t-1)})$, respectivamente.

$$\mathcal{L}^{(t)} \simeq \sum_{i=1}^{n} [l(y_i, \hat{y_i}^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) \quad . \tag{2.3}$$

Luego, se eliminan las constantes, dejando simplificar el objetivo en t

$$\tilde{\mathcal{L}}^{(t)} = \sum_{i=1}^{n} [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) \quad . \tag{2.4}$$

Como se puede observar en Ec. 2.4, su composición es una suma de funciones cuadráticas simples de una variable, la cual, es factible minimizarla con técnicas tradicionales. Por lo tanto, la problemática siguiente, surge en buscar un próximo *learner* que minimice la función de pérdida en la iteración t.

Chen y Guestrin (2016) definen a I_j como el conjunto de instancias de la hoja j, y w_j como el peso óptimo de la hoja j, donde:

$$I_j = \{i|q(x_i = j\} \quad ,$$
$$w_j^* = -\frac{\sum_{i\in I_j} g_i}{\sum_{i\in I_j} h_i + \lambda} \quad .$$

Lo que permite reescribir la Ec 2.4 mediante la descomposición de Ω:

$$\begin{aligned}\tilde{\mathcal{L}}^{(t)} &= \sum_{i=1}^{n}[g_i f_t(x_i) + \frac{1}{2}h_i f_t^2(x_i)] + \gamma T + \frac{1}{2}\lambda\sum_{j=1}^{T} w_j^2 \\ &= \sum_{j=1}^{T}[(\sum_{i\in I_j} g_i)w_j + \frac{1}{2}(\sum_{i\in I_j} h_i + \lambda)w_j^2] + \gamma T \quad .\end{aligned} \tag{2.5}$$

Para, finalmente, calcular el valor óptimo correspondiente con la Eq **??**, conocida como la "Función de Puntuación de Calidad". Esto hace referencia a evaluar la calidad de cada estructura de un árbol q, a través de valores similares a los *Impurity Scores* provienentes de los árboles de decisiones, con la diferencia de que en *XGBoost* deriva a una gama más amplia de funciones objetivas. En definitiva esta última va a devolver un mínimo valor de pérdida enfocado a una estructura de árbol determinada, teniendo, como consecuencia, una evaluación de la función de pérdida original mediante valores de pesos óptimos.

$$\tilde{\mathcal{L}}^{(t)}(q) = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j + \lambda} + \gamma T \quad , \tag{2.6}$$

siendo $G_j = \sum_{i\in I_j} g_i$ y $H_j = \sum_{i\in I_j} h_i \quad .$

Para corregir la dificultad de enumerar todas las posibles estructuras de árboles q, se utiliza un algoritmo *Greedy*, el cual inicia desde una sola hoja que contiene todos los ejemplos de entrenamiento para luego, iterar todas las características y valores de las características, como también agregar nuevas ramas al árbol de decisión. Estas evalúan las posibles reducciones de pérdida divididas (ver Ec.2.7), la cual esta compuesta por la diferencia entre la pérdida de las instancias principales y la suma de las pérdidas de la rama derecha e izquierda.

$$\mathcal{L}_{split} = \frac{1}{2}\left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda}\right] - \gamma \quad . \tag{2.7}$$

En Ec.2.7 las ramas del árbol dejarán de crecer cuando la ganancia para la mejor división no sea positiva.

2.5. Redes neuronales

Originalmente, las redes neuronales fueron fue propuestas teóricamente por McCulloch y Pitts (1943). Ellos diseñaron el primer modelo de redes neuronales (*NN*). Es así que se encuentra basado en modelos y algoritmos matemáticos, que, a causa de la fecha, no pudo ser probado por las carencias en recursos computacionales de la época.

Las primeras definiciones de esta metodología surgen a partir de la similitud que estas poseen con las redes neuronales biológicas (Shanmuganathan, 2016), teniendo en cuenta que el cerebro humano es aún mas complejo, dado que hasta la fecha no se conoce en su totalidad sus funciones. No obstante, se asemejan, según el autor, en 6 características principales: aprendizaje, adaptación , generalización, paralelismo masivo, robustez, almacenamiento asociativo de información y procesamiento de información espacio-temporal.

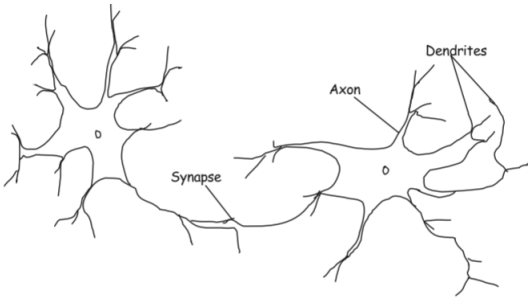

Figura 2.6: Red neuronal biológica
Fuente: Shanmuganathan (2016)

Una neurona genérica consiste en una capa de entrada (*input layer*), capas ocultas y una final, que tiene como característica, neuronas de salida (*output layers*) (Wang, 2003), tal como es demostrado en la Figura 2.7.

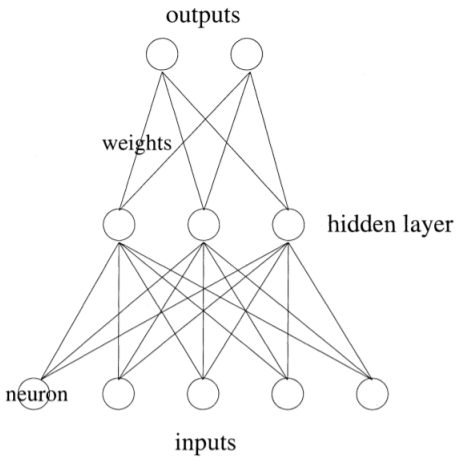

Figura 2.7: Arquitectura de una red neuronal
Fuente: Wang (2003)

Según Suryansh (2020), existen tres partes esenciales para formar la arquitectura de una neurona: neuronas, conexiones con sus pesos, parámetros y sesgos. La primera está construida sobre la base

de funciones que contienen pesos y sesgos a la espera de la entrada de datos, para así, realizar los cálculos determinados. Luego, pasan por una función de activación para filtrar los datos a un rango predeterminado. Los pesos, en términos generales, son valores que la red aprende para generalizar un problema. El sesgo corresponde a valores que representan lo que la red supone que se deberían sumar a la multiplicación de los pesos con los datos, aprendiendo a detectar los sesgos óptimos.

2.5.1. Redes neuronales *Long-Short Term Memory (LSTM)*

Este concepto fue propuesto por Hochreiter y Schmidhuber (1997) como una respuesta ante la problemática de memoria que poseen las redes neuronales. Los autores señalan que existen dificultades en cuanto a *Backpropagation Trought Time* o *Real-Time Recurrent Learning*, es decir, falencias que se hacen notorias en función del transcurso de secuencias o tiempo. Mencionan que los errores de los modelos tienden a desaparecer debido a que su evolución temporal proveniente del aprendizaje *Backpropagation*, depende exponencialmente de las ponderaciones. Dicha situación se puede ejemplificar mediante Redes Neuronales Recurrentes (RNN)

Redes Neuronales Recurrentes (RNN)

Esta metodología es una generalización de la red neuronal *feedforward*, con la diferencia de que esta, tiene una memoria interna. Este modelo realiza la misma función para cada entrada de datos, mientras que la salida de cada entrada respectiva depende del último cálculo, lo que las diferencia con otras redes neuronales, ya que estas son independientes entre sí a diferencia de RNN. Después de producir la salida, se copia y se envía de nuevo a la red recurrente. Finalmente, para generar una predicción o una clasificación estas considerarán las entradas de cada neurona y la información de la salida que ha aprendido de la entrada anterior.

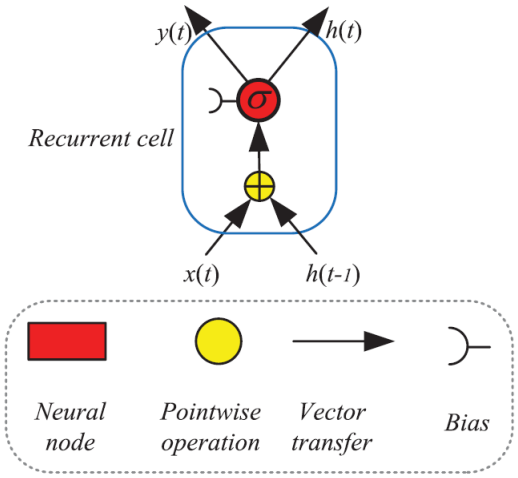

Figura 2.8: Arquitectura de una RNN
Fuente: Yu, Si, Hu, y Zhang (2019)

La arquitectura anterior (Fig. 2.8), representa una neurona estándar con una celda recurrente sigma, la cual está definida como:

$$\begin{aligned} h_t &= \sigma(W_h h_{t-1} + W_x x_t + b) \quad , \\ y_t &= h_t \quad , \end{aligned} \tag{2.8}$$

donde x_t es la entrada, h_t la información recurrente, y_t la salida de la celda en la iteración t, W_h y W_x los pesos y b el sesgo. No obstante, existen otras arquitecturas matemáticas más complejas para clasificaciones, donde la generación de la información recurrente (h_t) pasa por una transformación lineal y usando una transformación tangente hiperbólica.

$$h_t = tanh(W_h h_{t-1} + W_x x_t + b) \tag{2.9}$$

Luego, se realiza la predicción, utilizando la información recurrente anterior y aplicando nuevamente una transformación lineal, que posteriormente pasa por una función *softmax*.

$$y_t = softmax(W_y h_t + b) \tag{2.10}$$

Dependencia de Largo Plazo

Las redes recurrentes que constan de celdas recurrentes estándar no son capaces de manejar dependencias a largo plazo: a medida que crece la brecha entre las entradas relacionadas, es difícil aprender la información de conexión (Yu y cols., 2019). Esto quiere decir que mientras se vayan agregando más secuencias dentro de las neuronas, el primer valor oculto de la primera neurona resultará poco influenciable para la clasificación final. Tal como se demuestra en la Figura 2.9; donde x_0 y x_1, al pasar por una secuencia de neuronas, dejan de tener influencia sobre h_{t+1}.

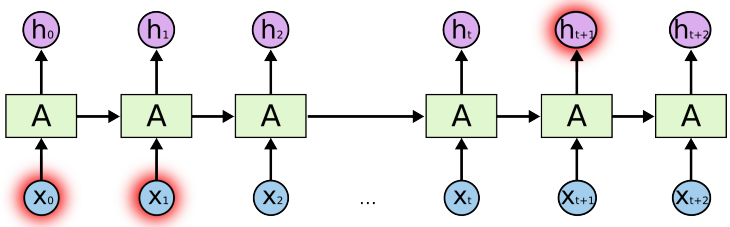

Figura 2.9: Red neuronal con problemas de dependencia a largo plazo
Fuente: (Olah, 2015)

Esto se puede demostrar matemáticamente. Si se consideran tres neuronas de la red de la Fig. 2.9, cada una deberá calcular la información recurrente anterior, por lo que, si se toma en cuenta la Ec.2.10 y se aumenta paulatinamente la secuencia de neuronas, se tendría:

$$y_t = softmax(\ldots tanh(\ldots tanh(\ldots tanh(h_0 \ldots) \ldots) \ldots) \ldots) \tag{2.11}$$

A medida que se se incorporan nuevas secuencias de neuronas, la información recurrente deberá pasar por funciones tangente hiperbólicas por cada neurona y, luego, por una función *softmax*. La memoria inicial de la neurona (h_0) será escalada por un número mucho menor a cero, esto es debido a que las funciones *tanh* están anidadas. Están las cuales poseen un valor que, en el mejor de los casos, será cercano a uno, por lo que la influencia de h_0 irá disminuyendo en la predicción final mientras que se incorporen nuevas secuencias dentro de la red.

Solución al problema de dependencia a largo plazo

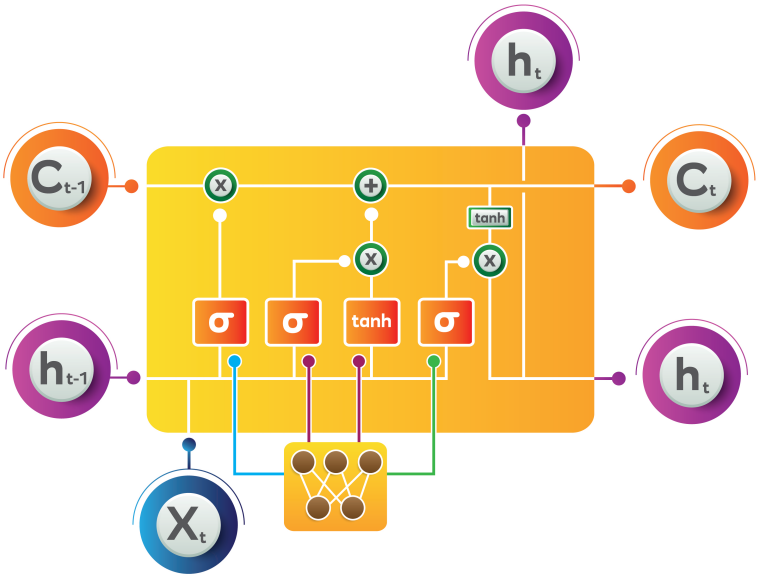

Figura 2.10: Estructura de una red neuronal *LSTM* (Elaboración propia)
Fuente: Elaboración propia

Las *LSTM* están diseñadas explícitamente para evitar el problema de la dependencia a largo plazo. Recordar información durante periodos de tiempo extendidos es prácticamente su comportamiento predeterminado (Olah, 2015). Lo que diferencia de la red neuronal recurrente, es la implementación de una vía que conecta todas las secuencias de neuronas llamada *Cell State*; la cual permite la incorporación y eliminación de información, mediante compuertas que se definen a continuación.

Compuerta de olvido

En sus inicios, estas redes no poseían esta compuerta. Fue introducida por Gers, Schmidhuber, y Cummins (2000). La puerta de olvido puede decidir qué información se eliminará del estado de la celda. Cuando el valor de la puerta de olvido, f_t es 1, conserva esta información; mientras tanto, el valor de 0 significa que se deshace de toda la información (Yu y cols., 2019).

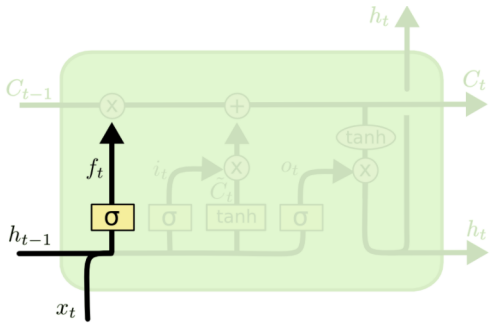

Figura 2.11: Compuerta de olvido de una red *LSTM*
Fuente: Olah (2015)

Esta compuerta decide la información que será eliminada del *Cell State* gracias a una función de activación sigmoidea, la cual toma en cuenta a h_{t-1} y x_t para que, mediante una transformación lineal, se genere un número entre 0 y 1. Estos valores significan la imposibilidad del paso de esta información y la aprobación para el paso de esta última respectivamente, a través de C_{t-1}.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad . \tag{2.12}$$

Compuerta de actualización

En esta compuerta se decide qué información se desea incorporar en la *Cell State*; es decir, actualizar. En primera instancia, está compuesta por una función sigmoidea qué indica que valores de la información recurrente requiere la actualización.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2.13}$$

Luego, una capa tangente hiperbólica generará un vector con los posibles valores candidatos para ser incluidos en C_t.

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_i) \quad . \tag{2.14}$$

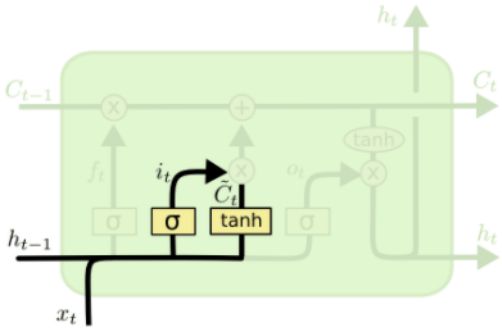

Figura 2.12: Compuerta de actualización de una *LSTM*
Fuente: Olah (2015)

Toda la información que fue recopilada, ya sea en la primera compuerta utilizada para eliminar información poco relevante o la segunda para ingresar nueva información que pueda aportar al modelo debe ingresarse a la *Cell State*.

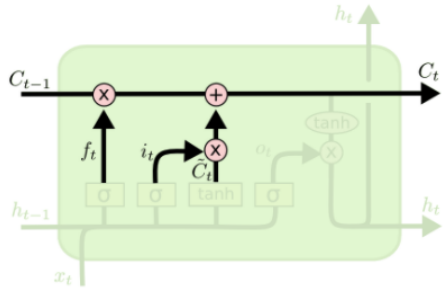

Figura 2.13: Actualización del *Cell State* de una *LSTM*
Fuente: Olah (2015)

Para eliminar la información del *Cell State*, se multiplica el *State* anterior por el resultado de la compuerta f_t.

$$f_t * C_{t-1} \quad . \tag{2.15}$$

Luego, se escalan los nuevos valores candidatos de manera predefinida para luego incorporarlos al *Cell State*.

$$i_t * \tilde{C}_t \quad . \tag{2.16}$$

2.5.2. Compuerta de salida

Tal como lo menciona su nombre, es una compuerta que genera información hacia la próxima neurona mediante el *Cell State* pero de manera filtrada, es decir, la nueva información recurrente (h_t). En primera instancia, en la Figura 2.14 se observa una capa sigmoidea, que, a través de una transformación lineal, decide qué información del *Cell State* se generará.

$$o_t = \sigma(W_o[h_{t-1,x_t}] + b_o) \quad . \tag{2.17}$$

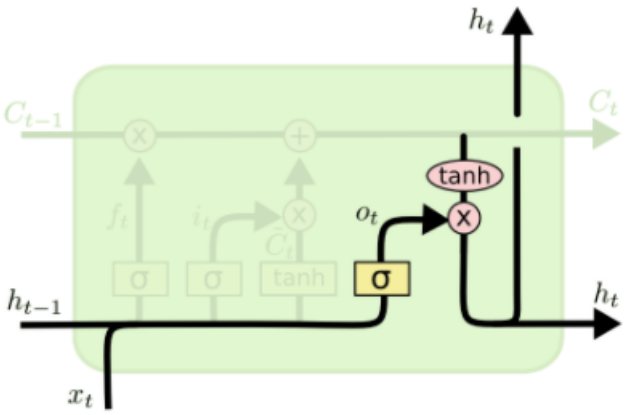

Figura 2.14: Compuerta de salida de una LSTM
Fuente: Olah (2015)

Luego, se escala el *Cell State* para garantizar que este fructúe entre -1 y 1 (rango que contiene h_t). Esto se realiza mediante una función tangente hiperbólica.

$$h_t = o_t * tanh(C_t) \quad . \tag{2.18}$$

Finalmente, se filtran los valores del *Cell State* (o_t) con el vector generado por la compuerta de salida (h_t).

$$o_t \times h_t \quad . \tag{2.19}$$

2.6. Suavizamiento exponencial Holt Winters

Holt-Winters es un modelo de comportamiento de series de tiempo (Winters, 1960), siendo una extensión de la metodología realizada por Holt (Holt, 2004), en donde su literatura fue actualizada a la fecha de 2004, el cual consiste en usar suavizamiento exponencial para codificar valores del pasado entrenándolos. De esa manera, poder predecir valores. Esta suavización es utilizado mediante la media móvil ponderada exponencialmente, en donde se modela tres aspectos de estas series: el promedio, la tendencia y un patrón repetitivo cíclico conocido como estacionalidad (Solar Winds, 2021). Con una ecuación dada como:

$$\hat{y}_{t+m} = T_t + b_t m + S_{t+m-L} \quad , \tag{2.20}$$

donde los $\hat{y}_{t+m}$ corresponden a los datos predichos con m-tiempos hacia atrás, t corresponde al período de datos de entrenamiento en donde se incluyen tres variables de carácter temporal: T_t como estacionareidad, b_t como la tendencia, S_t estima índices estacionales para excluir la interferencia aleatoria (Jiang y cols., 2020) y L son las *seasons*. La ecuación 2.19 tiene como característica poseer múltiples variaciones acorde a los tipos de series de tiempo (Ariton, 2021).

Para el caso de la estacionalidad positiva se emplea la ecuación:

$$\hat{y}_{t+m} = T_t + (b_t \cdot m) + S_{t+m-L} \quad , \tag{2.21}$$

en caso de que la estacionalidad sea multiplicativa su fórmula está definida como:

$$\hat{y}_{t+m} = [T_t + (b_t \cdot m)] \cdot S_{t+m-L} \quad . \tag{2.22}$$

Según Ariton (2021), lo llamativo del modelo es la independencia que este posee entorno a sus componentes. Es decir, por ejemplo en el caso de que la estacionalidad sea multiplicativa, no implica que la tendencia sea con las mismas características, al contrario esta última puede ser aditiva. Dicho esto, entrega la libertad de realizar múltiples combinaciones para encontrar el modelo más adecuado para las series temporales a estudiar. En el caso de este proyecto, se utiliza un *Grid Search* evalúa las posibilidades y escoge el mejor modelo para cada cliente.

Esta metodología, incorpora una tercera componente a actualizar la estacionalidad:

$$\begin{aligned} \text{Multiplicativo:} \quad & S_t = \gamma \frac{y_t}{T_t} + (1-\gamma) S_{t-L} \quad , \\ \text{Aditivo:} \quad & S_t = \gamma (y_t - T_t) + (1-\gamma) S_{t-L} \quad , \end{aligned} \tag{2.23}$$

donde, γ es la nueva constante de suavizado. Por otra parte, las otras dos compontentes siguien siendo iguales a la metodología de doble suavizamiento exponencial, donde y_t son las observaciones:

$$\begin{aligned} & b_t = \beta (T_t - T_{t-1}) + (1_\beta) b_{t-1} \quad , \\ \text{Multiplicativo:} \quad & T_t = \alpha \frac{y_t}{S_{t-L}} + (1-\alpha)(T_{t-1} + b_{t-1}) \quad , \\ \text{Aditivo:} \quad & T_t = \alpha (y_t - S_{t-L}) + (1-\alpha)(T_{t-1} + b_{t-1}) \quad . \end{aligned} \tag{2.24}$$

Resumiento α, β y γ son parámetros de suavizado del modelo Holt Winters comprendidos en un rango de [0,1].

Es importante destacar que en la ecuación 2.23, para la metodología Holt Winters, a diferencia de sus versiones anteriores, incorpora una modificación en la componente de estacionareidad (T_t), corrigiendo el error que poseía con series estacionales de manera que se produce el efecto inverso a estacionalizar la serie.

Capítulo 3

MATERIALES Y MÉTODOS

3.1. Materiales

3.1.1. Conjunto de datos

Para el proyecto se tiene un conjunto de datos de consumos mensuales desde enero del 2016 a diciembre del 2020, contando con un total de 535610 cliente. El conjunto de datos sólo posee variables categóricas y una variable tipo *string* que corresponde a la serie del medidor en cuestión.

3.1.2. *Kaggle*

Debido a la limitación tecnológica que pueda poseer un investigador en su computador personal, esta plataforma ofrece un servicio computacional en línea, mediante un computador de códigos, según el lenguaje de programación de que se desea optar. En nuestro caso, es Python, el cual posee especificaciones técnicas suficientes para desarrollar la investigación.

3.1.3. *Python*

Es uno de los lenguajes de programación más conocidos dentro de la industria debido a su amplia capacidad para resolver diferentes temáticas que se le atribuyan. Dentro de la investigación, se utiliza mediante Jupyter Notebook para el desarrollo de los modelos de *Machine Learning*, para que, finalmente, se elabore un motor selector del mejor modelo.

3.1.4. Librerías utilizadas

Pandas

Pandas pretende ser el bloque de construcción fundamental de alto nivel para realizar análisis de datos prácticos del mundo real en Python. Además, tiene el objetivo más amplio de convertirse en la herramienta de análisis / manipulación de datos de código abierto más potente y flexible disponible en cualquier idioma (Pandas, 2008). Es decir, provee estructuras de datos generando gráficos de muy buena calidad junto con la librería *Matplotlib*, ademas se integra facilmente con otras bibliotecas que se especializan en trabajar los datos dentro de matrices como *Numpy*.

Numpy

NumPy es el paquete fundamental para la computación científica en Python. Es una biblioteca de *Python* que proporciona un objeto de matriz multidimensional, varios objetos derivados (como

matrices y matrices enmascaradas) y una variedad de rutinas para operaciones rápidas en matrices, que incluyen manipulación matemática, lógica, de formas, clasificación, selección, transformadas discretas de Fourier, álgebra lineal básica, operaciones estadísticas básicas, simulación aleatoria y mucho más (NumPy, 2008). En otras palabras, esta libreria es útil para la realización de calculos y análisis numéricos a gran escala de datos. La cual, incorpora una nueva funcionalidad basada en *arrays*, permitiéndo de manera eficiente la manipulación de estos.

Matploblib

Matplotlib es una biblioteca completa para crear visualizaciones estáticas, animadas e interactivas en Python. Matplotlib hace que las cosas fáciles sean fáciles y las difíciles sean posibles. Crea gráficos de calidad de publicación. Hace figuras interactivas que puedan hacer zoom, desplazarse, actualizar. Personaliza el estilo y el diseño visual. Exporta a muchos formatos de archivo. Incrusta en JupyterLab e interfaces gráficas de usuario. Utiliza una amplia variedad de paquetes de terceros creados en *Matplotlib* (Matplotlib, 2003).

Keras

Keras es una *API* de aprendizaje profundo escrita en Python, que se ejecuta sobre la plataforma de aprendizaje automático *TensorFlow*. Fue desarrollado con un enfoque en permitir una experimentación rápida. Ser capaz de pasar de la idea al resultado lo más rápido posible es clave para hacer una buena investigación (Team, 2015). Esta librería fue utilizada para realizar pronósticos mediante redes neuronales *Long-Short Term Memory*.

Sklearn

*Scikit-lear*n (*Sklearn*) es la biblioteca más útil y robusta para el aprendizaje automático en *Python*. Proporciona una selección de herramientas eficientes para el aprendizaje automático y el modelado estadístico que incluyen clasificación, regresión, agrupación y reducción de dimensionalidad a través de una interfaz de consistencia en *Python*. Esta biblioteca, que está escrita en gran parte en *Python*, se basa en *NumPy*, *SciPy* y *Matplotlib* (Tutorialspoint, s.f.). Tal como se menciona anteriormente, mediante la utilización de herramientas aprendizaje automático, se utilizó en este proyecto para la implementación de métricas de desempeño para la comparación de modelos.

Statsmodels

Statsmodels es un módulo de Python que proporciona clases y funciones para la estimación de muchos modelos estadísticos diferentes, así como para realizar pruebas estadísticas y exploración de datos estadísticos (Taylor, 2009).

XGBoost

XGBoost es una biblioteca optimizada de aumento de gradiente distribuida diseñada para ser altamente eficiente , flexible y portátil . Implementa algoritmos de aprendizaje automático bajo el marco *Gradient Boosting* (The XGBoost Contributor, 2014).

Seaborn

Seaborn es una biblioteca para hacer gráficos estadísticos en Python. Se basa en matplotlib y se integra estrechamente con las estructuras de datos de pandas. Esta libreria ayuda a explorar

y comprender sus datos. Sus funciones de trazado operan en marcos de datos y matrices que contienen conjuntos de datos completos y realizan internamente el mapeo semántico y la agregación estadística necesarios para producir gráficos informativos. Su API declarativa, orientada a conjuntos de datos, le permite concentrarse en lo que significan los diferentes elementos de sus gráficos, en lugar de en los detalles de cómo dibujarlos (Waskom, 2012). Entonces, la librería ofrece una interfaz optimizada para la visualización de datos de buena calidad, con la finalidad de hacer un eje central la visualización de datos como una parte fundamental de la exploración de datos.

3.1.5. *Microsoft Excel*

Si bien es un software de hojas de cálculo, no fue utilizado de manera directa para la resolución de los desafíos planteados. No obstante, fue parte de la investigación en la etapa preliminar, en donde la empresa proovedora de los conjuntos de datos los brindaba mediante la generación automática de QUERYS en formato *.csv*.

3.2. Metodología

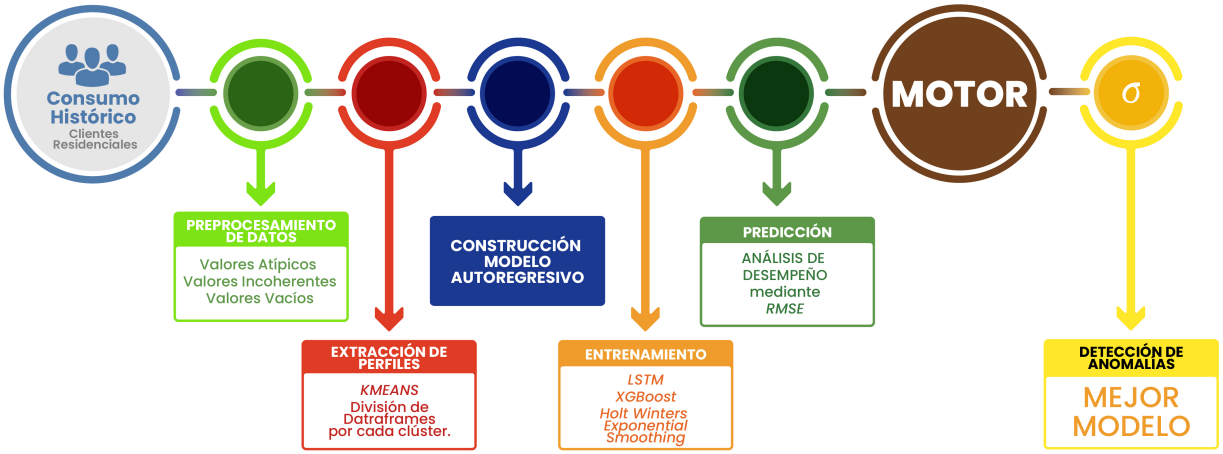

Figura 3.1: Diagrama explicativo sobre la metodología para la detección de consumos anómalos
Fuente: Elaboración propia

3.2.1. Proceso de detección de anomalías de consumo

La idea general de la metodología está representada en la Figura 3.1, la cual tiene como núcleo predecir series de tiempo. En este caso, se centra en los consumos de clientes residenciales; en donde, mediante las técnicas abordadas en la investigación, permiten desarrollar un modelo que detecta alzas de consumo no esperadas. Cabe destacar que los modelos de alerta son personalizados para cada consumo histórico, es decir, aprenden en base a los comportamientos que posee cada persona. A continuación, se definirá de manera general las etapas de modelo, comprendidas a partir de la imagen mencionada anteriormente.

Etapa Previa:

Corresponde a la recolección de información realizada por empresas contratistas, reguladas por la distribuidora de energía. En esta fase se desarrollan planes de lecturas de consumos mensuales

que son llevados a una base de datos. Aquí, mediante softwares de consulta, se otorgan los consumos históricos de cada cliente para que, posteriormente, sean aplicados a la metodología propuesta.

Preprocesamiento de los datos

Cuando ya es recibida la información necesaria, llega el momento de estudiar la calidad de los datos. Esto se efectúa a través de técnicas de administración de datos, en donde se analizan los valores perdidos, incoherentes, duplicados, etc y son solucionados mediante medidas tomadas por la empresa en cuestión.

Extracción de perfiles de consumo

Mediante técnicas de clustering, en este caso *K-Means* se extraen los perfiles de consumo del conjunto de datos general, con la finalidad de alivianar la carga al compilador utilizado, a través de la división en subconjuntos acorde a cada *cluster* generado y, a su vez, aportar nueva información al modelo, en donde en la siguiente etapa, aportará información adicional a este último.

Construcción de un modelo autorregresivo

En primera instancia, se realiza un análisis exploratorio en los datos para series temporales, con la finalidad de estudiar los comportamientos históricos. Luego, para la correcta utilización y predicción con técnicas de *machine learning*, se transforman los conjuntos generados en la etapa anterior a un modelo autorregresivo que permita el correcto entrenamiento.

Entrenamiento

Durante esta etapa, es crucial un buen desarrollo para el correcto entendimiento de la problemática actual. Aquí se divide cada subconjunto en uno de prueba y otro de validación correspondiente a cada usuario. Estos dos grupos son series de tiempo, indicando cuantos meses se espera predecir a partir del entrenamiento en meses previos. Para la investigación serán tres modelos los que aprenderán de la información de cada usuario, redes neuronales, modelos de *boosting* y de suavización exponencial.

Predicción

Tal como lo indica su nombre, tras haber entrenado correctamente el modelo, se predicen los próximos consumos, con el fin de que estos sean los esperados respectivamente. De esta forma, se entrega el primer indicio de detección de anomalía, en caso de que el consumo real sea completamente diferente de lo que se esperaba. Luego, se define de manera más completa esta afirmación.

Motor selector

Mediante métricas de desempeño de los modelos, se selecciona el más óptimo para cada usuario que pasará por el proceso de detección de anomalías.

Detección de anomalías

Finalmente, se proponen reglas óptimas relacionadas a la desviación estándar del error de predicción para alertar sobre posibles alzas de consumo. Así, posteriormente, será notificado a un analista experto del área, el cual concluirá la causa de este cambio de comportamiento.

3.3. Administración de datos

Tal como lo representa la figura anterior, esta sección corresponde al correcto desarrollo de un conjunto de datos. Este debe ser idóneo para la problemática y la correcta realización de los modelos.

La limpieza de los datos y la atención a las suposiciones también pueden tener importantes efectos beneficiosos sobre el poder, el tamaño del efecto y la precisión de las estimaciones de población y, por lo tanto, la replicabilidad de los resultados (Osborne, 2013). Dada la cita anterior, antes de realizar una limpieza y administración de datos, es necesario tener un conjunto de datos. Para ello, se inició con 60 dataframes, que corresponden a los consumos mensuales de los usuarios desde el inicio del 2016 hasta el fin del 2020. Por ende, se construyó un conjunto a partir de los consumos de los últimos mediante su NIS (ID) y la serie del medidor.

La iniciativa de no sólo tomar en cuenta el ID se debe a que algunos clientes pueden tener más de un medidor de electricidad en su casa. Por lo tanto, la metodología empleada en la investigación se aplicó a cada medidor de la empresa. Posteriormente, concluida la construcción del dataframe a estudiar, la ID del cliente es eliminada, dejando a la serie de medidor como identificador.

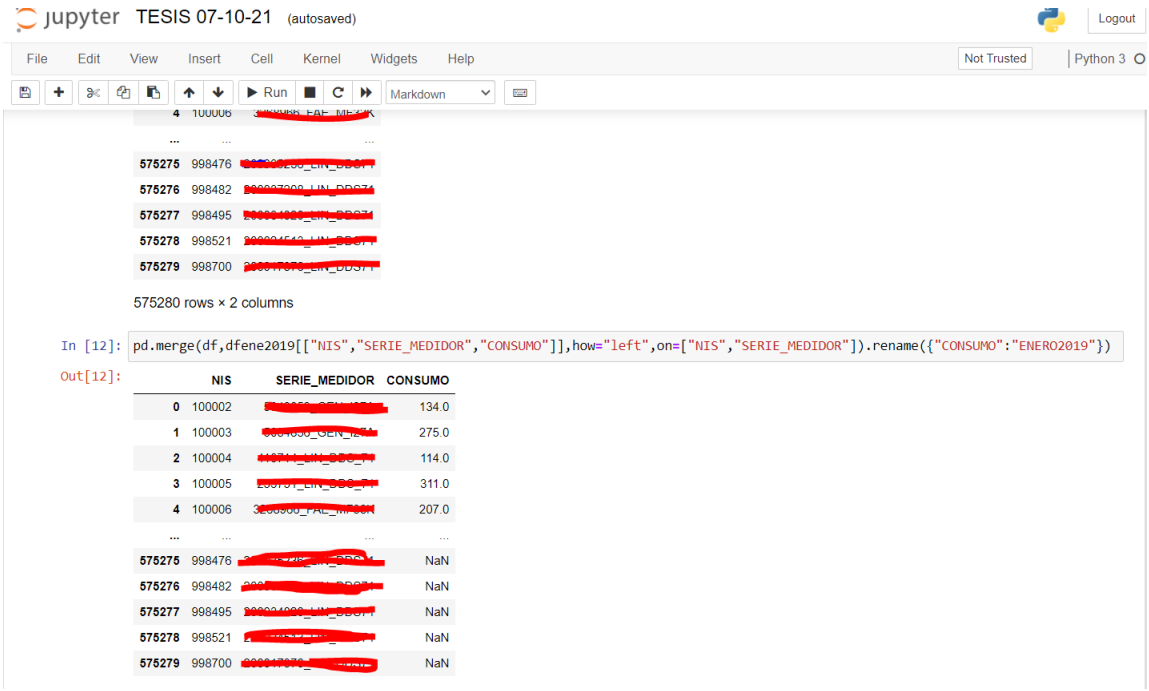

Figura 3.2: Ejemplo de salida del kérnel de Python para la generación del conjunto de datos
Fuente: Elaboración propia

Luego, se optó trabajar sólo con la categoría residencial de usuario, ya que representa más de un 90 % del universo de clientes (Tabla 3.1). No obstante, es posible que para futuras investigaciones se incorporen más categorías, aportando nueva información a la metodología planteada.

Se emplea la misma lógica dentro de la categoría residencial, en donde la subcategoría Casas y Departamento representan la mayoría del total de observaciones (Tabla 3.2), por lo que se opta filtrar por estas últimas. Es necesario precisar que la subcategoría Otros, representa a distintos grupos por separados, en donde cada uno posee una frecuencia menor o igual a 5.

Categorías	Frecuencia
Residencial	544.045
Otras	31.247

Cuadro 3.1: Comparación de frecuencias de clientes entre la categoría residencial y otras.
Fuente: Elaboración propia

Subcategoría	Frecuencia
Casa-Habitación	421.650
Departamento	115.423
Campamento	2.612
Servicio común	2.551
Catastro	1.290
Provisorio	310
Cliente Regularización	172
Casa Negocio	37
Otros	menor a 5 por categoría

Cuadro 3.2: Comparación sobre la cantidad de observaciones por subcategoría de cada cliente.
Fuente: Elaboración propia

Principalmente, los valores perdidos del conjunto final están relacionados con el tiempo de vida que tenga un usuario dentro de la empresa. Por ejemplo, si este inicio su contrato en el 2018, es seguro que para las fechas anteriores posea missing values, los cuales, para efectos del desarrollo del modelo, fueron rellenados con 0. Para el caso de outliers, existían situaciones en donde habían valores con consumos irreales, resultando incoherentes, como valores negativos o valores que corresponden a un consumo tope. Por lo tanto, fueron eliminado, ya que están relacionado a políticas de devolución o de estimación de consumo. Entonces, no representan un consumo real, por lo que no funcionan en metodologías de predicción de tiempo.

3.4. Aplicación de modelos predictivos

Ya cumplida la etapa de administración de datos y selección de las categorías de usuarios, se procede al modelamiento para técnicas de predicción propuestas. Estas corresponden a las redes neuronales *Long-Short Term Memory*, *XGBoost* y Suavizamiento exponencial Holt Winters. Antes de aplicar directamente las metodologías, es necesario transformar los datos para ajustar un modelo autorregresivo, ya que se está utilizando técnicas de *machine learning* para predecir series temporales. Para ello, se prepara el conjunto, transformando la serie en un problema de aprendizaje supervisado. Entonces, se adaptan los datos para que sean estacionarios y, finalmente, se escalan. Los modelos de aprendizaje supervisado predictivo empleados, poseen una estructura definida con un $input(X)$ y $output(Y)$. Dada la naturaleza de una serie temporal, no es trivial predecir de manera inmediata. En ese sentido, se transforma el problema en un modelo autorregresivo para poder realizar un one-step forecasting. En este último, los n valores anteriores de la serie están disponibles y el problema de pronóstico se puede proyectar en la forma de un problema de regresión genérico Billah, King, Snyder, y Koehler (2006), como se muestra en la Figura 3.3.

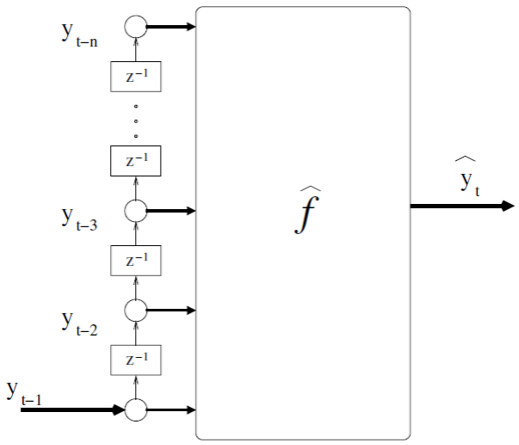

Figura 3.3: Mecanismo para transformar series de tiempo en un problema supervisado (Billah y cols., 2006)

Según el autor, el aproximador f devuelve la predicción del valor de la serie de tiempo en el tiempo $t+1$, en función de los n valores anteriores(Fig.3.3).

$$X = \begin{bmatrix} y_{N-1} & y_{N-2} & \cdots & y_{N-n-1} \\ y_{N-2} & y_{N-3} & \cdots & y_{N-n-2} \\ \vdots & \vdots & \vdots & \vdots \\ y_n & y_{n-1} & \cdots & y_1 \end{bmatrix} \quad ; \quad Y = \begin{bmatrix} y_N \\ y_{N-1} \\ \vdots \\ y_{n+1} \end{bmatrix} \quad . \tag{3.1}$$

Cabe destacar que esta configuración presenta una variable denominada lag, la cual indica cuántos saltos hacia atrás en el tiempo se desea dar para la respuesta del vector Y. En términos simples, relacionándolo con la problemática actual, si se considera un lag de valor 2, se está designando como matriz de entrada dos consumos mensuales previos que explican el valor del tercer mes. Finalmente, luego de haber realizado la transformación el conjunto, se modifica a estacionario y se escala. Dentro de la etapa de entrenamiento, se separa el conjunto de datos en subconjuntos de aprendizaje y validación, siendo este último definido como el pronóstico de los últimos tres meses, indicando el perfil de consumo del cliente. Por consiguiente, se entrena con cada uno de los modelos predictivos respectivos, en la primera parte, con las redes neuronales *LSTM* y *XGBoost* aplicando la modificación del conjunto de datos a diferencia del tercer modelo, Holt Winters.

3.5. Motor selector y detección de anomalías

Posteriormente, se automatiza la selección del modelo óptimo acorde a cada perfil de consumo de cada cliente. Esto quiere decir que cada cliente tendrá un modelo de predicción que fue entrenado de manera personalizada, acorde a su consumo. Para seleccionar el mejor modelo, se tomó en consideración la raíz error cuadrático medio ($RMSE$), donde y'_t es el valor de la predicción, y_t es el valor del conjunto de test y n el número de instancias Fenza y cols. (2019). Por lo tanto, el modelo adecuado para el usuario será el que tenga el $RMSE$ de menor valor.

$$RMSE = \sqrt{\frac{\sum_{t-1}^{n}(y'_t - y_t)^2}{n}} \quad . \tag{3.2}$$

La detección de anomalías está fundamentada las condiciones inspiradas por artículos y políticas de la empresa. En donde se aplica una condición que señala que si un cliente posee un índice de *RMSE*, para el próximo mes, mayor que la suma de 300 KWh y el RMSE del test del model, este último consumo alertaría una alza anómala de consumo. La lógica de emplear una base de 300 KWh, se justifica en que actualmente, para detectar un cliente con órden de crítica, es si éste consume el valor mencionado anteriormente, por sobre el mes anterior, entonces, partiendo de la base de los objetivos en donde se pretende disminuir el volúmen de OC, el no contar con este valor, incorporaría consumos anómalos en clientes que tengan una diferencia menor a 300 KWh.

$$\text{Anomalía} = \sqrt{\frac{\sum_{t-1}^{n}(y'_t - y_t)^2}{n}} + 300 < RMSE_{t+1} \quad . \tag{3.3}$$

Entonces, de acuerdo a lo mencionado anteriormente; por ejemplo, si el modelo posee un buen desempeño con un *RMSE* de 5 KWh, donde se espera que el próximo consumo real sea acorde al valor predicho realizado en un *out-of-sample*. De lo contrario, si este valor real difiere por sobre el valor predicho de manera significativa poseerá un *RMSE* alto, en donde si supera la suma entre 300KWh y el *RMSE* del test, este consumo es considerado anómalo. Cabe destacar que para la utilización continua de esta metodología se sugiere que los pronósticos a futuro y los re-entrenamientos se realicen cada tres meses, no obstante, esta es una decisión que deba tomar la empresa a cargo de la problemática, monitoreando la metodología en una ventana temporal que se estime conveniente.

Capítulo 4

RESULTADOS

4.1. Preprocesamiento de datos

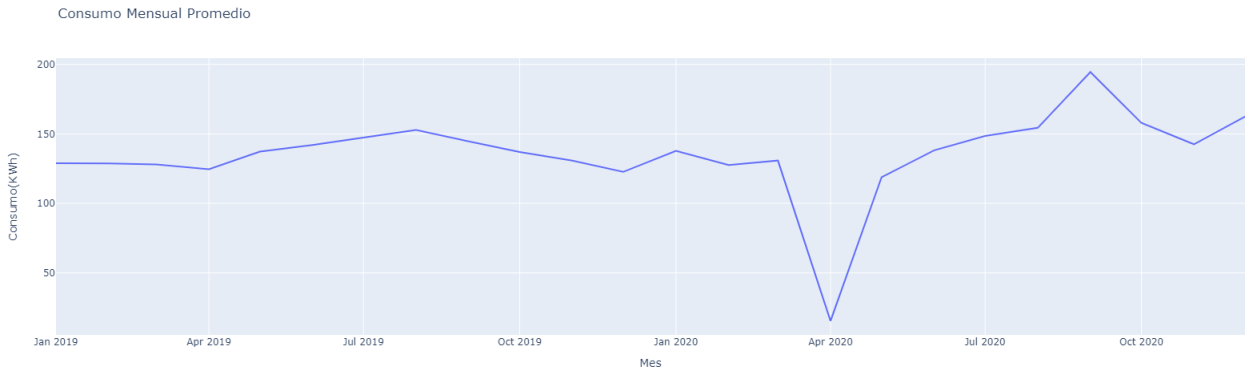

Figura 4.1: Serie de tiempo del consumo mensual promedio de los clientes del conjunto de datos, con error en el mes de abril.

Fuente: Elaboración propia

Cuando comenzó la pandemia, se aplicó una cuarentena en Chile que comenzó en abril del 2020. Bajo este contexto, no se registraron lecturas de consumo eléctrico, lo cual se ve reflejado en el gráfico anterior. Esto puede representar mucho ruido para el entrenamiento correcto de los modelos.

Dentro de la empresa existe la política de que, en caso de que exista falta de información en un consumo mensual, se debe estimar mediante el promedio de los últimos 6 meses, medida que fue acogida. Con esto presente, resultó la siguiente gráfica, en donde se logra ver un promedio de consumos más estable.

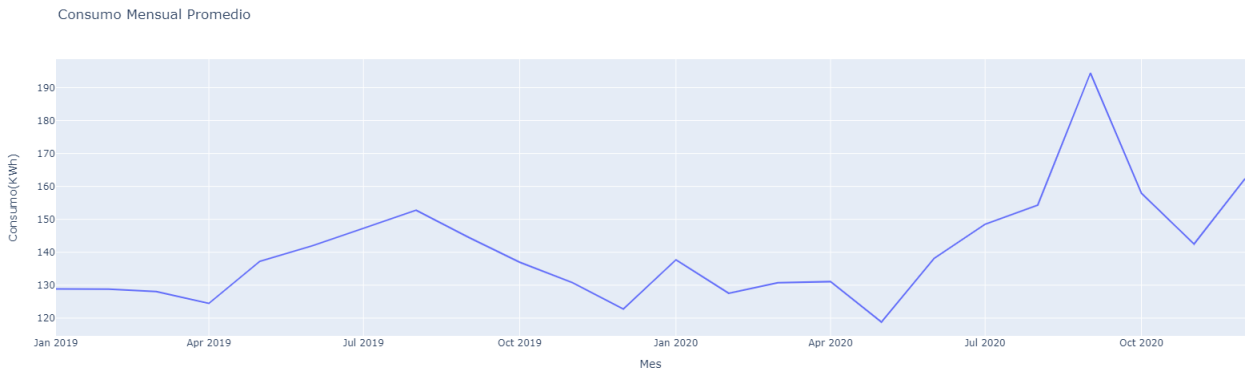

Figura 4.2: Serie de tiempo sobre el consumo mensual promedio de los clientes con el mes de abril estimado.

Fuente: Elaboración propia

4.2. Extracción de perfiles

A través de la metodología de Clusterización *K-Means*, utilizada por su popularidad, se extraen y se agrupan los perfiles similares de cada cliente, reduciendo su dimensionalidad segmentando perfiles en distintos entrenamientos. Para ello, mediante la técnica del Codo (Elbow's Method) se predefine el número de conglomerados para segmentar, lo que indica la existencia de 4 *clusters* representativos el siguiente gráfico:

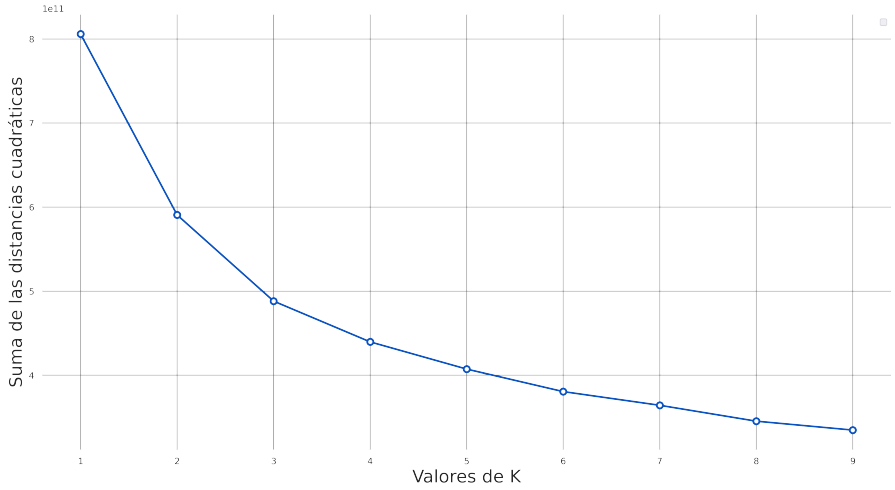

Figura 4.3: Método del codo para determinar la cantidad de *clusters* para la metodología *KMeans*
Fuente: Elaboración propia

Cada *cluster* agrupó clientes acorde a comportamientos similares en base a sus consumos (ver Tabla 4.5).

Cluster	**Consumo Promedio (KWh)**
1	61.4312
2	350.3073
3	164.3526
4	938.2067

Cuadro 4.1: Consumos promedios de cada *cluster* por algoritmo de agrupación *K-Means*
Fuente: Elaboración propia

Los resultados gráficos de la agrupación son los siguientes:

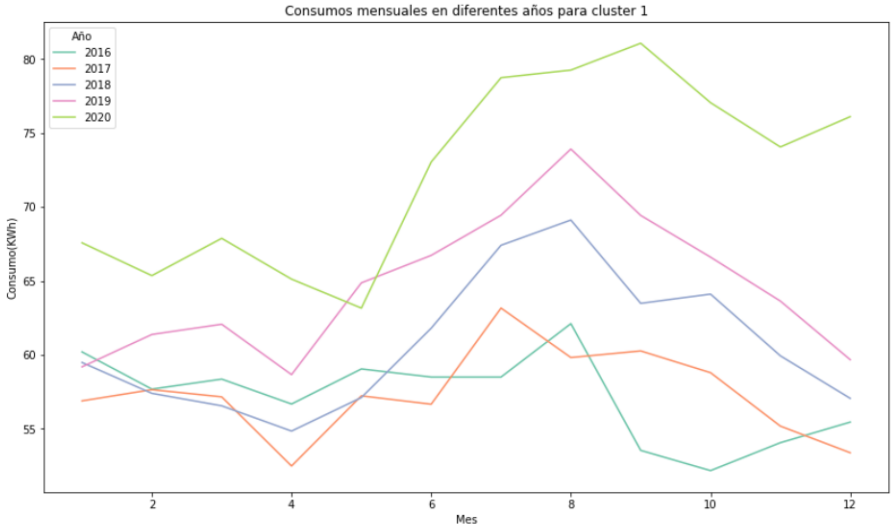

Figura 4.4: Comparación de consumos mensuales promedios por año para el *cluster* 1
Fuente: Elaboración propia

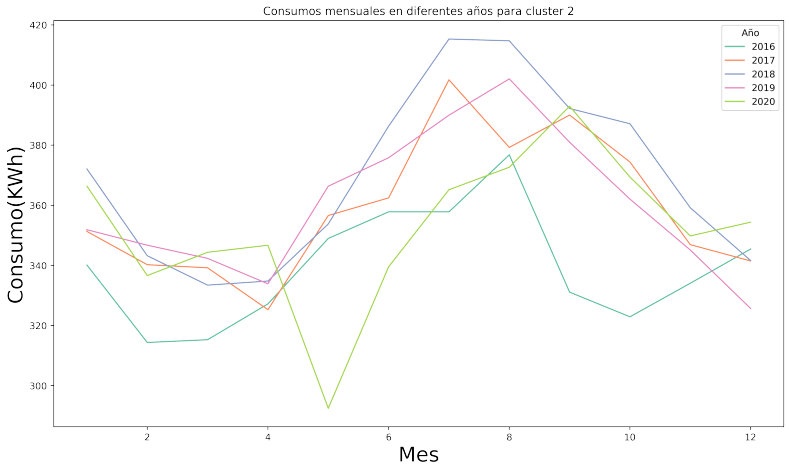

Figura 4.5: Comparación de consumos mensuales promedios por año para el *cluster* 2
Fuente: Elaboración propia

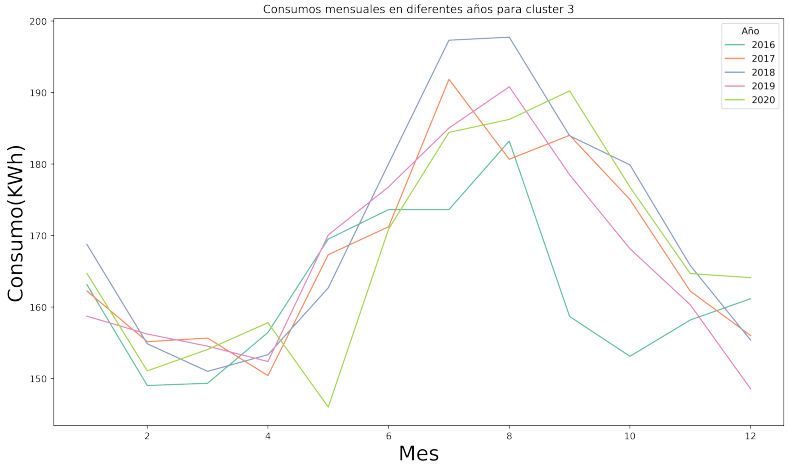

Figura 4.6: Comparación de consumos mensuales promedios por año para el *cluster* 3
Fuente: Elaboración propia

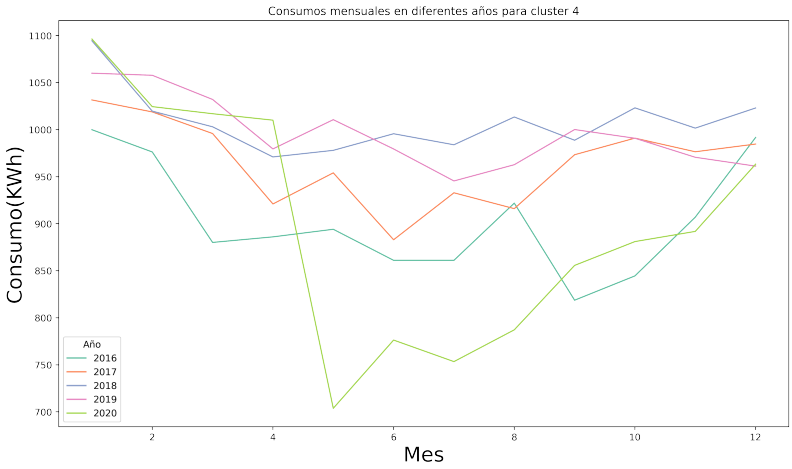

Figura 4.7: Comparación de consumos mensuales promedios por año para el *cluster* 4
Fuente: Elaboración propia

En primer lugar, se observa que los consumos para el *cluster* 1 (Fig.4.4) fructúan entre 50 y 80 KWh, aproximadamente. El *cluster* 2 (Fig.4.5) entre 280 a 420 KWh. El tercer *cluster*(Fig. 4.6) en un rango de 120 a 200 KWh y por último, el *cluster* 4 (Fig.4.7), oscila entre 700 KWh y 1100 KWh de manera aproximada, siendo la agrupación con mayor nivel de consumo. Cabe destacar que entre julio y agosto suelen ser los meses con mayor consumo para todos los años, esto es debido a la ubicación geográfica en la cual son los meses con menor temperatura según Weather Spark (2021), lo que ocasiona el mayor uso de artefactos tecnológicos eléctricos para mantener una temperatura ideal dentro de los hogares. Los primeros tres clusters poseen un patrón de consumo similar por año a diferencia del cuarto quien posee mayores diferencias de consumos promediso anuales.

Por último, es importante hacer notar que el año 2020 presenta mayor variabilidad que años anteriores, esto es ocasionado por el impacto que el COVID-19 produce a las empresas ya sea por irregularidades por falta de información o cambios en comportamientos de clientes (World Bank, 2021), como lo es por ejemplo el mes de abril, donde no se registró información de consumo producto de cuarentenas, lo que conlleva a estimar el mes en cuestión.

4.3. Análisis exploratorio

Para el resto del desarrollo de la investigación se trabajó con muestras aleatorias de tamaño 2 por cada cluster, debido a limitaciones tecnológicas que conlleva realizar este tipo de procedimiento; es decir, se estudiará el motor selector del mejor modelo para detectar anomalías en 2 clientes por cada grupo.

Para efectos de entendimiento, a cada usuario correspondiente a cada *cluster* respectivo, se le asigna una numerificación para asignarlo de la siguiente forma:

- *Cluster* 1: cliente 1.1 y cliente 1.2
- *Cluster* 2: cliente 2.1 y cliente 2.2

- *Cluster* 3: cliente 3.1 y cliente 3.2
- *Cluster* 4: cliente 4.1 y cliente 4.2

4.3.1. Asimetría y curtosis

Cluster	**Cliente**	**Asimetría**	**Curtosis**
1	1.1	-0.12	0.78
	1.2	0.46	2.20
2	2.1	1.23	4.51
	2.2	0.48	0.34
3	3.1	0.93	1.77
	3.2	0.02	2.94
4	4.1	0.40	0.22
	4.2	0.13	2.65

Cuadro 4.2: Resumen de análisis de curtosis y simetría para clientes de cada *cluster*
Fuente: Elaboración propia

En primera instancia, se estudió el comportamiento de los datos mediante el coeficiente de asimetría de Fisher y la curtosis, respectivamente (Cain, Zhang, y Yuan, 2016) (Tabla 4.2). En primer caso, se observa que los datos se comportan de manera asimétrica teniendo una cola más pesada hacia la derecha a diferencia del cliente 1.1, quien posee una leve asimetría hacia la izquierda. Por otra parte, en el coeficiente de curtosis, se envidencia que en el total de clientes extraídos, poseen una distribución leptocúrtica, en la cual, existe una mayor concentración de observaciones en torno a su media, destacándose el cliente 2.1 en este ámbito, no obstante el usuario que más se asemeja a una distribución gaussiana es el cliente 4.1.

4.3.2. Normalidad y estacionareidad

Para series temporales es crucial tener en cuenta el comportamiento de los datos en relacion a la normalidad y estacionareidad de los datos. El hecho de cumplir este supuesto permite tener una correcta modelación en machine learning en torno a los datos utilizados (Palachy, 2021) y (Tavgen, 2018), de tal forma de tener predicciones a futuros más certeras.

Para estudiar la normalidad, se utilizó la prueba K-cuadrado de D'Agostino, la cual se deriva utilizando la curtosis y la asimetría de la muestra y se considera la transformación para la asimetría de la muestra Ahmad y Al-Mutairi (2017). En donde el test de hipótesis, con un nivel de significancia del 0,05, es:

- H_0 : Los datos se distribuyen normal.
- H_1 : Los datos no se distribuyen normal.

Para el análisis de estacionareidad, se utilizó una de las técnicas más populares, la prueba de Dickey Fuller Aumentada, la cual, es la versión extendida de la prueba Dickey Fuller simple. El motivo de extender la versión, recae en incluir más rezagos en términos de las variables dependientes para eliminar el problema de la autocorrelación Mushtaq (2011). Siendo las de hipótesis, con un nivel de significancia de 0,05:

- H_0: Los datos tienen una raíz unitaria y no son estacionarios.
- H_1: Los datos no tienen raíz unitaria y son estacionarios.

		Normalidad		Estacionareidad	
Cluster	**Cliente**	**P-valor**	**Conclusión**	**P-valor**	**Conclusión**
1	1.1	0.307	No rechaza H_0	0.0000	Rechaza H_0
	1.2	0.010	Se rechaza H_0	0.0000	Rechaza H_0
2	2.1	0.000	Rechaza H_0	0.1496	No rechaza H_0
	2.2	0.176	No rechaza H_0	0.0019	Rechaza H_0
3	3.1	0.001	Rechaza H_0	0.0237	Rechaza H_0
	3.2	0.011	Rechaza H_0	0.2568	No rechaza H_0
4	4.1	0.297	No rechaza H_0	0.2847	No rechaza H_0
	4.2	0.015	No rechaza H_0	0.0000	Rechaza H_0

Cuadro 4.3: Resumen de análisis de normalidad mediante la prueba de K-Cuadrado de D'Angostino y estacionareidad a través de la prueba de Dickey Fuller Aumentada para los clientes de cada clúster con un α de 0.05

Fuente: Elaboración propia

Si bien en el cuadro anterior (Tabla 4.3) ofrece una visual en cuanto al comportamiento de las series temporales de cada cliente, en donde en casos se cumplen los supuestos, como en otros no. No obstante, estos incumplimientos son subsanados mediante técnicas de escalado y diferenciaciones propias del área del *machine learning*.

4.4. Motor selector de las técnicas empleadas

Es necesario recordar que se estudió la *performance* mediante la raíz del error cuadrático medio (*RMSE*) de cada modelo por cada uno de los cuatro clusters generados, debido al aporte que este índice de desempeño que ofrece al momento de observar el consumo real comparado con el predicho, permitiéndo la utilización de la metodología planteada para la detección de anomalías de consumo, en los cuales se extrajo una muestra aleatoria de dos clientes, respectivamente. En donde, se entrenó cada modelo y se evaluó. El modelo con mejor desempeño se guarda para futuras predicciones, lo cual, es competencia de la empresa en cuestión decidir cada cuanto tiempo estiman prudente reentrenar el modelo acorde a nuevos comportamientos de los usuarios.

4.4.1. Resultados motor selector

A continuación se presentan la tabla resultante para cada uno de los *clusters* en donde se entrenó en un segmento temporal compuesto de 3 meses con la finalidad de extraer un perfil de consumo, no obstante, es importante señalar que la empresa es la que toma la decisión final para la ventana temporal utilizada como perfil de consumo de los clientes. Esto debe ser estudiado mediante un montireo que permita obtener el segmento de tiempo más adecuado a utilizar.

Cluster	Cliente	Fecha	Consumo	*RMSE*	Consumo predicho	Modelo Ganador
1	1.1	2020-10-01	100.0	5.087167	98.916445	HoltWinters
	1.1	2020-11-01	116.0	5.087167	108.746579	HoltWinters
	1.1	2020-12-01	103.0	5.087167	107.883809	HoltWinters
	1.2	2020-10-01	68.0	42.545987	103.723267	HoltWinters
	1.2	2020-11-01	46.0	42.545987	95.471534	HoltWinters
	1.2	2020-12-01	137.0	42.545987	95.685373	HoltWinters
2	2.1	2020-10-01	223.0	7.633415	224.761966	HoltWinters
	2.1	2020-11-01	215.0	7.633415	223.336767	HoltWinters
	2.1	2020-12-01	211.0	7.633415	221.109444	HoltWinters
	2.2	2020-10-01	622.0	55.647652	682.152910	HoltWinters
	2.2	2020-11-01	609.0	55.647652	638.650892	HoltWinters
	2.2	2020-12-01	547.0	55.647652	616.227419	HoltWinters
3	3.1	2020-10-01	326.0	18.667953	337.182695	HoltWinters
	3.1	2020-11-01	284.0	18.667953	301.530327	HoltWinters
	3.1	2020-12-01	263.0	18.667953	287.761106	HoltWinters
	3.2	2020-10-01	175.0	8.788101	161.058571	HoltWinters
	3.2	2020-11-01	144.0	8.788101	138.633759	HoltWinters
	3.2	2020-12-01	162.0	8.788101	159.079013	HoltWinters
4	4.1	2020-10-01	441.0	35.177464	437.848214	HoltWinters
	4.1	2020-11-01	448.0	35.177464	387.411120	HoltWinters
	4.1	2020-12-01	586.0	35.177464	591.604980	HoltWinters
	4.2	2020-10-01	956.0	72.750465	930.473244	HoltWinters
	4.2	2020-11-01	988.0	72.750465	1014.945518	HoltWinters
	4.2	2020-12-01	1229.0	72.750465	1108.583164	HoltWinters

Cuadro 4.4: Tabla resultado del motor selector del mejor modelo de forecasting para la detección de consumos anómalos.

Fuente: Elaboración propia

A través del recuadro anterior, lo primero a resaltar es que en todos los casos, salió como mejor modelo la metodología de suavizamiento exponencial de Holt Winters. Según la empresa norteamericana de desarrollo de softwares, SolarWinds, define este modelo como poderoso y simple, en donde se diferencia del resto de modelos empleados. Este posee la facultad de manejar muchos patrones estacionales complicados, como lo representa el conjunto de datos de la investigación, esto se realiza encontrando el valor central y luego agregando los efectos de la pendiente y la estacionalidad (Solar Winds, 2021), quien a diferencia de las otras metodologías, el poseer comportamientos estacionales irregulares, puede significar mucho ruido para el entrenamiento de estos. El efectivo desempeño de este modelo recae en la buena selección de parámetros, en donde, dentro de este modelo, se utiliza un *Grid Search*, quien es el encargado de designar los mejores parámetros para modelar.
Ahora bien, a pesar de resultar ser el modelo protagonista, sólo tres de ocho clientes poseen un desempeño que puede considerarse adecuado (Cuadro 4.4), poseyendo raices de errores cuadráticos medios inferiores a 9 KWh para los últimos 3 meses, por lo que se graficarán para una mayor comprensión del desempeño en las siguientes figuras, las cuales, mediante el análisis gráfico, se puede argumentar el desempeño de las otras metodologías plantedas, el donde al ser series temporales irregulares, teniendo picos de alzas y bajas de consumos producen demasiado ruido para estos, a diferencia del modelo protagónico, a su vez destacar el buen desempeño para el cliente proveniente del *cluster* 3 (Fig.4.10).

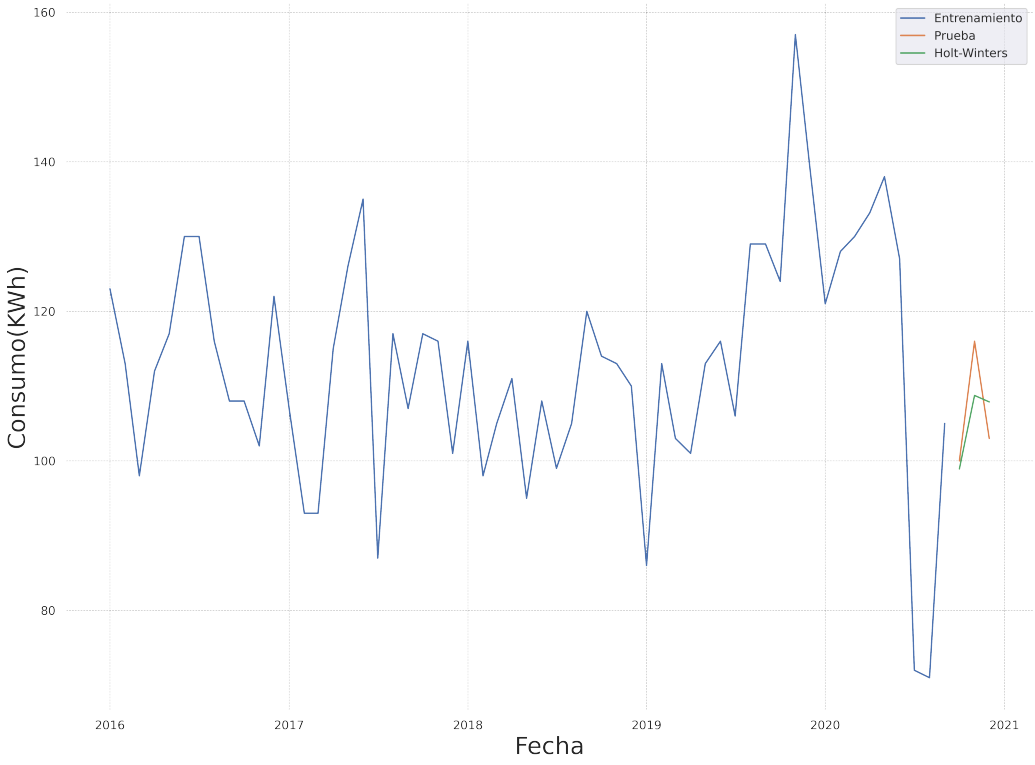

Figura 4.8: Estimación de los últimos tres meses para el cliente 1.1 mediante suavizamiento exponencial Holt Winters

Fuente: Elaboración propia

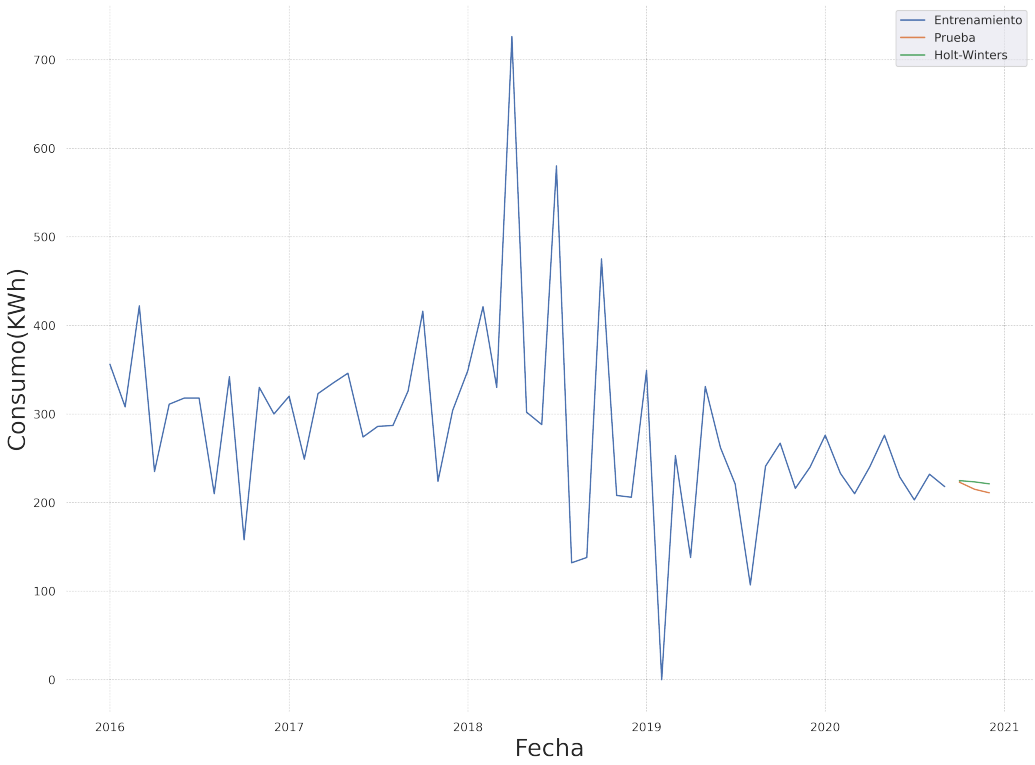

Figura 4.9: Estimación de los últimos tres meses para el cliente 2.1 mediante suavizamiento exponencial Holt Winters

Fuente: Elaboración propia

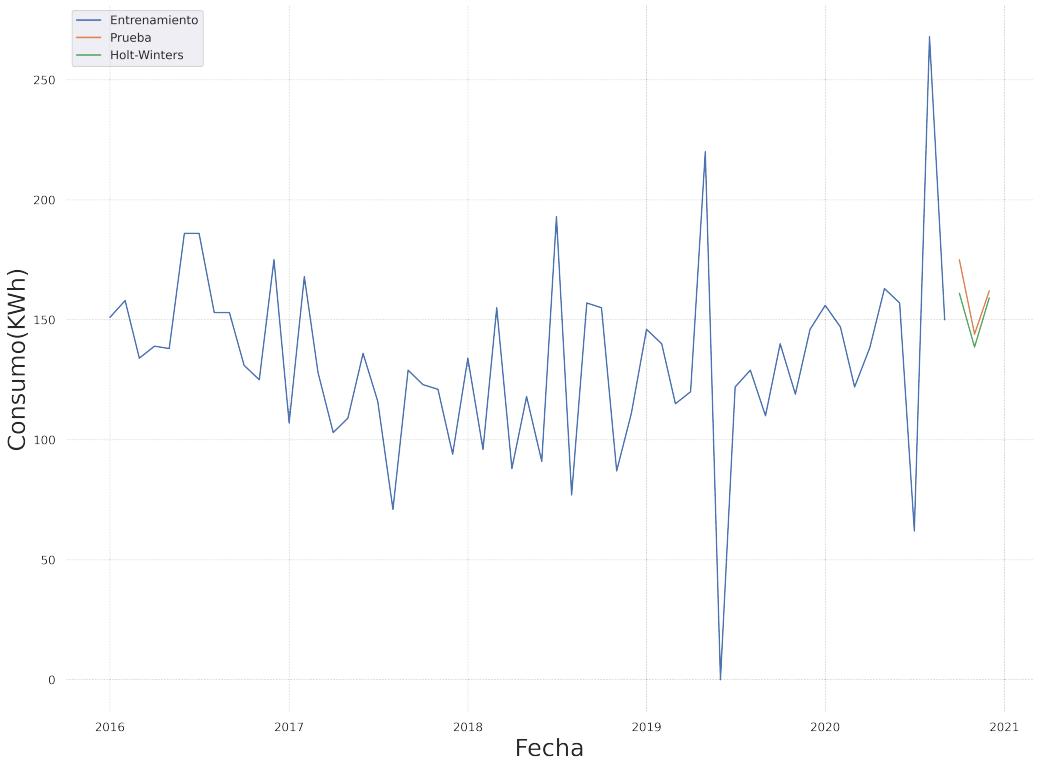

Figura 4.10: Estimación de los últimos tres meses para el cliente 3.2 mediante suavizamiento exponencial Holt Winters

Fuente: Elaboración propia

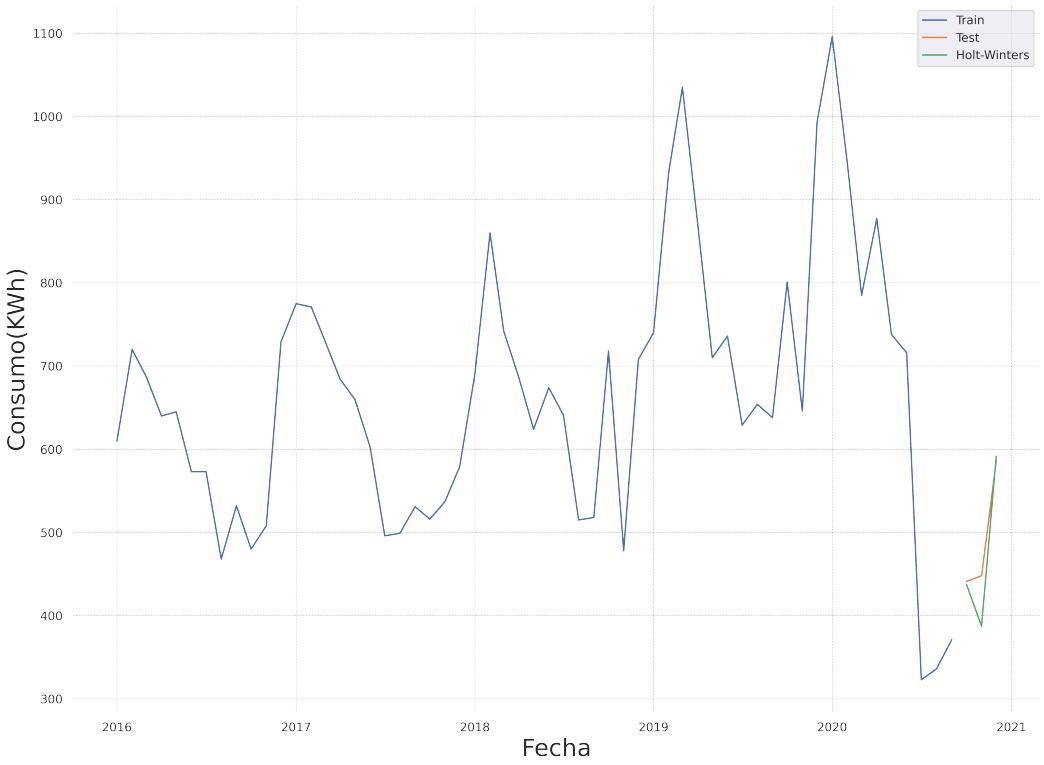

Figura 4.11: Estimación de los últimos tres meses para el cliente 4.1 mediante suavizamiento exponencial Holt Winters

Fuente: Elaboración propia

4.5. Detección de anomalías

Para corroborar que la metodología propuesta este en lo correcto, se desarrolló dos situaciones ficticias a partir del cliente con modelo con mejor desempeño (Fig.4.12), al cual se le realiza un *out-of-sample* para luego ser comparado con dos valores, uno muy elevado respecto a su perfil de consumo y otro semejante.

Mes	Consumo Predicho	Consumo real	RMSE
2021-01-01	159	500	340.13

Cuadro 4.5: Consumos promedios por algoritmos de agrupación K-Means
Fuente: Elaboración propia

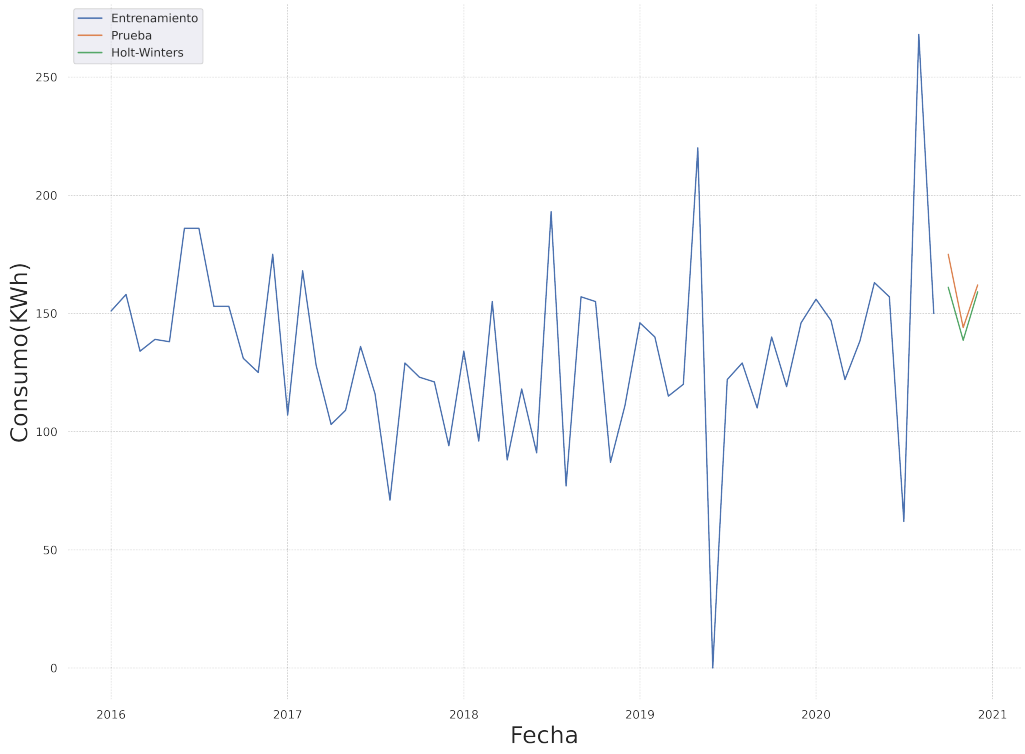

Figura 4.12: Estimación de los últimos tres meses para el cliente 3.2 mediante suavizamiento exponencial Holt Winters

Fuente: Elaboración propia

Luego de evidenciar el buen desempeño del modelo se procede a realizar una predicción *out of sample* para tres meses posteriores, resultando la siguiente gráfica:

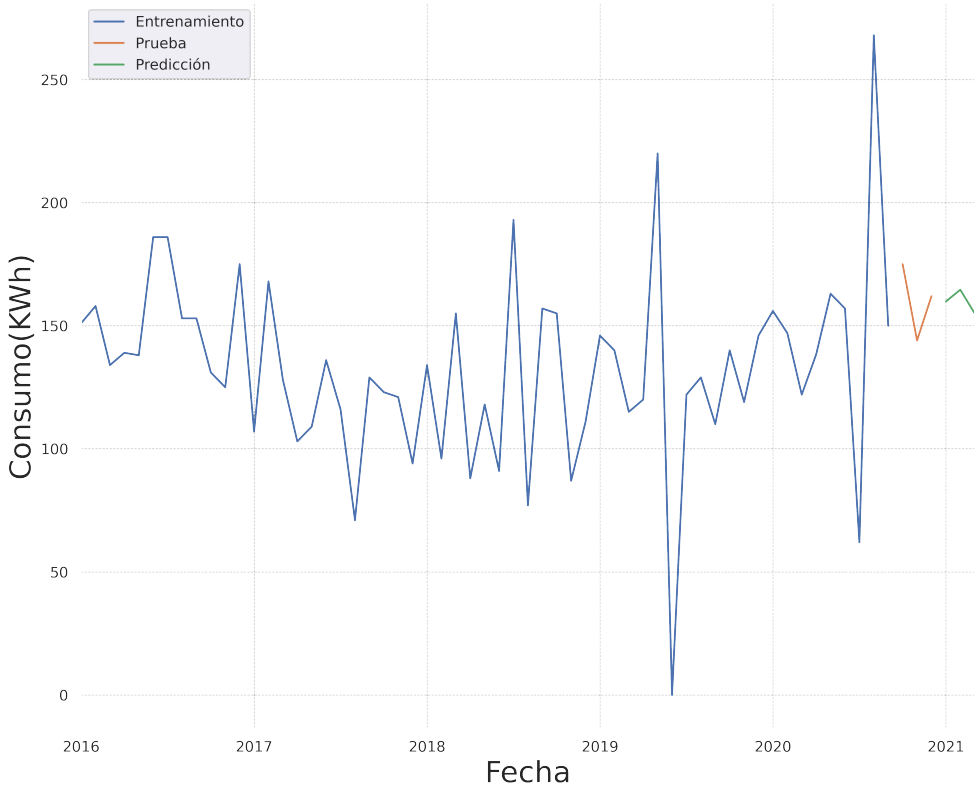

Figura 4.13: Entrenamiento, Prueba y observaciones predichas para tres meses usando Suavización Exponencial Holt Winters en cliente 3.2
Fuente: Elaboración propia

Ya teniendo los valores de los consumos (Cuadro 4.6) para los tres meses entrantes, se puede someter al clientes a casos ficticios de anomalías y consumos normales, mediante la fórmula propuesta.

Caso de anomalía

En primer lugar hay que tener en cuenta que para los meses correspondientes a octubre, noviembre y diciembre, asignados como test, poseen un *RMSE* de 8KWh, aproximadamente. Luego de haber realizado el proceso de lecturas en clientes residenciales, se indica que este usuario para el mes de enero, consumió 500 KWh, lo que puede atribuirse a visitas o al uso de artefactos electrónicos que puedan lidiar con la temperatura de la ubicación geográfica respectiva. No obstante, se esperaba un consumo de 159.84 KWh, aproximadamente, por lo que resulta:

$$RMSE = 340{,}16 \quad . \tag{4.1}$$

Ya teniendo el valor anterior se puede llevar a un proceso de detección de anomalías en donde se suma la raíz del error cuadrático medio de los tres meses correspondientes al test con los 300 KWh

Mes	Consumo
2021-01-01	159.838514
2021-02-01	164.625489
2021-03-01	154.990923

Cuadro 4.6: Valor en KWh de los consumos predichos para enero, febrero y marzo del año 2021
Fuente: Elaboración propia

bases y se evalúa frente al RMSE del nuevo mes ingresado. En conclusión, dado que la inecuación siguiente se cumple, este consumo se reflejaría como anómalo, lo que generaría una órden de crítica a analizar por parte de los analistas.

$$\begin{aligned} RMSE_{ant} + 300 &< RMSE_{actual} \quad , \\ 8 + 300 &< 340{,}16 \quad , \\ 308 &< 340{,}13 \quad . \end{aligned} \tag{4.2}$$

Teniendo en cuenta la ecuación anterior, para el caso en que el nuevo consumo ingresado no posea mayores diferencias que el consumo predicho por el modelo, la inecuación no se cumpliría por lo que el modelo no generaría una órden de crítica, figurando como un mes de consumo normal.

Capítulo 5

CONCLUSIONES

5.1. Sobre los resultados obtenidos

Resumiendo, dentro de la investigación se emplearon tres modelos con la finalidad predecir consumos eléctricos de clientes residenciales en forma de series temporales para luego detectar anomalías mediante la metodología seleccionada por un motor selector. En este caso, consumos históricos de clientes. La razón por la que se escogió estos instrumentos esta fundamentada en la popularidad que recae en estas debido a su gran desempeño y múltiples investigaciónes en donde son utilizadas, además de recomendaciones por parte de expertos en el área.

Si bien, se esperaba que las metodologías de *machine learning* tuviesen un mejor desempeño y pronóstico que herramientas clásicas como lo es el suavizamiento exponencial de Holt Winters, siendo esta última la protagonista con mejor evaluación mediante la raíz del error cuadrádico medio. La razón por la cual se utiliza esta métrica recae en el hecho que permite comparar distancias entre el valor real y el predicho, siendo esto muy útil a la hora de detectar consumos anómalos. Dicho lo anterior, queda demostrado que cuando se trata de datos demasiado irregulares los modelos de aprendizaje automático poseen mayor dificultad para poder entender el comportamiento de éstos. Ahora, pese a tener un motor selector funcional para la selección de modelo, no se obtuvo resultados alentadores que permitiesen demostrar en su cabalidad la eficiencia de las herramientas propuestas. Sin embargo, la investigación se desarrolló a buen nivel. Queda abierta a un futuro en donde se espera generar modelos más complejos que puedan tratar series temporales con comportamientos irregulares, para ser introducidos dentro del motor selector y tener una mayor eficiencia para la detección de anomalías.

A grandes rasgos la particularidad de tratar esta problemática a través de estos modelos, es la ventaja que ofrecen al personalizar las predicciones por cada cliente, es decir, cada cliente poseera un modelo que lo represente y permita estimar consumos futuros, para que en su posterioridad permitan detectar anomalías de consumo, a diferencia de la estrategia previamente utilizada donde solo consideraba un delta consumo para la alerta de consumos críticos. No obstante, se requiere una ventana temporal más amplia que permita monitorear la efectividad a lo largo del tiempo. En base a lo anteriormente señalado es importante que compañías de diferentes áreas del mercado innoven en técnicas actuales que pueden resultar benéficas para la optimización del tiempo y los recursos empleados para su realización.

5.2. Experiencia de la tesis

La realización de esta investigación resultó ser un gran desafío, dado a las limitaciones tecnológicas que surgían a la medida que se avanzaba en el proyecto. No obstante, gracias a eso pude conocer nuevas herramientas en línea como lo es Kaggle quien ofrece un servicio gratuito para aquellas personas que se sientan limitadas por recursos faltantes. En lo personal, ya finalizando mi proceso académico y el hecho de haber realizado esta temática, me permitió corroborar que no me equivoqué al elegir el área en la cual me quiero desempeñar, la inteligencia artificial.

Finalmente, esta experencia me permitió adquirir nuevos conocimientos como lo es enfrentarse y desenvolverse como Ingeniero Estadístico proveniente de la Universidad de Valparaíso, frente a una empresa y a datos reales que esta provee, haciendo notar la gran importancia y la utilidad del rubro a la hora de tomar decisiones correctamente para diversos tipos de compañía.

Referencias

Ahmad, M., y Al-Mutairi, A. (2017). Normality tests: A brief review. *Journal of ISOSS*, *3*(1), 35–44.

Almazrouee, A. I., Almeshal, A. M., Almutairi, A. S., Alenezi, M. R., y Alhajeri, S. N. (2020). Long-term forecasting of electrical loads in kuwait using prophet and holt–winters models. *Applied Sciences*, *10*(16). Descargado de https://www.mdpi.com/2076-3417/10/16/5627

Ariton, L. (2021). *A Thorough Introduction to Holt-Winters Forecasting.* Descargado de https://medium.com/analytics-vidhya/a-thorough-introduction-to-holt-winters-forecasting-c21810b8c0e6

Billah, B., King, M. L., Snyder, R. D., y Koehler, A. B. (2006). Exponential smoothing model selection for forecasting. *International Journal of Forecasting*, *22*(2), 239-247. Descargado de https://www.sciencedirect.com/science/article/pii/S016920700500107X

Bonaccorso, G. (2017). *Machine learning algorithms: A reference guide to popular algorithms for data science and machine learning.* Packt Publishing.

Cain, M. K., Zhang, Z., y Yuan, K.-H. (2016). Univariate and multivariate skewness and kurtosis for measuring nonnormality: Prevalence, influence and estimation. *Behavior Research Methods*, *49*(5), 1716–1735.

Chen, T., y Guestrin, C. (2016). Xgboost: A scalable tree bosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Chilquinta. (2021). *Tarifas.* Descargado de https://www.chilquinta.cl/tarifas

Fenza, G., Gallo, M., y Loia, V. (2019). Drift-aware methodology for anomaly detection in smart grid. *IEEE Access*, *7*, 9645-9657.

Gers, F. A., Schmidhuber, J. A., y Cummins, F. A. (2000). Learning to forget: Continual prediction with lstm. *Neural Comput.*, *12*(10), 2451–2471.

Hochreiter, S., y Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*, 1735-80.

Holt, C. C. (2004). Forecasting seasonals and trends by exponentially weighted moving averages. *International Journal of Forecasting*, *20*(1), 5-10. Descargado de https://www.sciencedirect.com/science/article/pii/S0169207003001134

Jain, A. (2010). Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, *31*, 651-666.

Jiang, W., Wu, X., Gong, Y., Yu, W., y Zhong, X. (2020). Holt–winters smoothing enhanced by fruit fly optimization algorithm to forecast monthly electricity consumption. *Energy*, *193*, 116779. Descargado de https://www.sciencedirect.com/science/article/pii/S0360544219324740

Kaufman, L., y Rousseeuw, P. (2009). *Finding groups in data: An introduction to cluster analysis.*

Lavine, B. (2006). Clustering and classification of analytical data..

Li, C., Chen, Z., Liu, J., Li, D., Gao, X., y Di. (2019). Power load forecasting based on the combined model of lstm and xgboost. En *Proceedings of the 2019 the international conference*

on pattern recognition and artificial intelligence (p. 46–51). New York, NY, USA: Association for Computing Machinery.

Matplotlib. (2003). *Matplotlib — Visualization with Python.* Descargado de https://matplotlib.org/

McCulloch, W. S., y Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*(4), 115–133.

Morde, V. (2019). *XGBoost Algorithm: Long May She Reign! - Towards Data Science.* Descargado de https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d

Mushtaq, R. (2011). Augmented Dickey Fuller Test. *SSRN Electronic Journal*.

NumPy. (2008). *What is NumPy? — NumPy v1.21 Manual.* Descargado de https://numpy.org/doc/stable/user/whatisnumpy.html

Olah, C. (2015). *Understanding LSTM Networks – colah's blog.* Descargado de https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Osborne, J. (2013). *Best practices in data cleaning: A complete guide to everything you need to do before and after collecting your data.*

Palachy, S. (2021). *Stationarity in time series analysis - Towards Data Science.* Descargado de https://towardsdatascience.com/stationarity-in-time-series-analysis-90c94f27322

Pandas. (2008). *pandas - Python Data Analysis Library.* Descargado de https://pandas.pydata.org/about/index.html

Schapire, R. E. (2003). The boosting approach to machine learning: An overview. En D. D. Denison, M. H. Hansen, C. C. Holmes, B. Mallick, y B. Yu (Eds.), *Nonlinear estimation and classification* (pp. 149–171). New York, NY: Springer New York.

Schapire, R. E., y Freund, Y. (2012). *Boosting: Foundations and algorithms.* The MIT Press.

Shanmuganathan, S. (2016). Artificial neural network modelling: An introduction. En S. Shanmuganathan y S. Samarasinghe (Eds.), *Artificial neural network modelling* (pp. 1–14). Cham: Springer International Publishing.

Solar Winds. (2021). *Holt-Winters Forecasting and Exponential Smoothing Simplified.* Descargado de https://orangematter.solarwinds.com/2019/12/15/holt-winters-forecasting-simplified/

Suryansh, S. (2020). *Neural Networks: All You Need to Know - Towards Data Science.* Descargado de https://towardsdatascience.com/nns-aynk-c34efe37f15a

Tavgen, A. (2018). *Time series modelling - AlexTavgen.* Descargado de https://medium.com/@ATavgen/time-series-modelling-a9bf4f467687

Taylor, J. (2009). *About statsmodels — statsmodels.* Descargado de https://www.statsmodels.org/stable/about.html#about-statsmodels

Team, K. (2015). *Keras documentation: About Keras.* Descargado de https://keras.io/about/

The XGBoost Contributor. (2014). *XGBoost Documentation — xgboost 1.5.1 documentation.* Descargado de https://xgboost.readthedocs.io/en/stable/

Tutorialspoint. (s.f.). *Scikit Learn Tutorial.* Descargado de https://www.tutorialspoint.com/scikit_learn/index.htm

Wang, S.-C. (2003). Artificial neural network. En *Interdisciplinary computing in java programming* (pp. 81–100). Boston, MA: Springer US.

Waskom, M. (2012). *An introduction to seaborn — seaborn 0.11.2 documentation.* Descargado de https://seaborn.pydata.org/introduction.html

Weather Spark. (2021). *Compare el clima y el tiempo en Valparaíso y en otra ciudad - Weather Spark.* Descargado de https://es.weatherspark.com/compare/y/25811/Comparaci%C3%

B3n-del-tiempo-promedio-en-Valpara%C3%ADso
Winters, P. R. (1960). Forecasting Sales by Exponentially Weighted Moving Averages. *Management Science*, *6*(3), 324–342.
World Bank. (2021). *Cómo la COVID-19 afecta a las empresas en todo el mundo.* Descargado de https://www.bancomundial.org/es/news/infographic/2021/02/17/how-covid-19is-affecting-companies-around-the-world
Yao, X., y Liu, Y. (2014). Machine learning. En E. K. Burke y G. Kendall (Eds.), *Search methodologies: Introductory tutorials in optimization and decision support techniques* (pp. 477–517). Boston, MA: Springer US.
Yu, Y., Si, X., Hu, C., y Zhang, J. (2019). A review of recurrent neural networks: Lstm cells and network architectures. *Neural Computation*, *31*, 1-36.
Z-Ai. (2020). *What is Boosting in Machine Learning? - Towards Data Science.* Descargado de https://towardsdatascience.com/what-is-boosting-in-machine-learning-2244aa196682
Zhao, Y., y Hryniewicki, M. K. (2018). Xgbod: Improving supervised outlier detection with unsupervised representation learning. *2018 International Joint Conference on Neural Networks (IJCNN)*.