UNIVERSIDAD ESAN

FACULTAD DE INGENIERÍA

INGENIERÍA INDUSTRIAL Y COMERCIAL

Predicción de los valores de la demanda máxima de energía eléctrica empleando técnicas de machine learning para la empresa Nexa Resources – Cajamarquilla

Trabajo de Suficiencia Profesional presentado en satisfacción parcial de los requerimientos para obtener el título profesional de Ingeniero Industrial y Comercial

AUTORES

Alexis Alfredo Bustinza Barrial

Anghy Mabel Bautista Abanto

Diego Alexis Alva Alfaro

Giovanni Mauricio Villena Sotomayor

Jeanpiere Manuel Trujillo Sabrera

ASESOR

Junior John Fabián Arteaga

ORCID Nº 0000-0001-9804-7795

Octubre, 2022

Dedicatoria

Este trabajo está dedicado a nuestros padres y familia que sin su apoyo y constante motivación no tendríamos la oportunidad de salir afrontar los retos académicos y profesionales que se presentan en el camino. También agradecemos a todos los involucrados que apoyaron de forma directa con este trabajo de suficiencia profesional. Dios mediante y la Virgen María que guíe e ilumine nuestros caminos frente a cualquier dificultad.

RESUMEN

Nexa Resources Cajamarquilla es una empresa dedicada a la extracción, tratamiento y transformación de metales. Actualmente, el precio de metales eco amigables viene en aumento debido a las restricciones en el transporte marítimo de combustibles desde Rusia, por lo que se han incrementado los costos de petróleo, gasolina y otros. Las operaciones de las empresas que dependen de energía eléctrica generada por estos combustibles ha aumentado, es en este sentido que se ha propuesto disminuir su consumo de energía eléctrica aplicando herramientas de Machine Learning para pronosticar sus puntos máximos de demanda de energía y poder dosificar su producción. En el presente estudio se aplicó una metodología basada en una estructura cuantitativa relacionando de dos a más variables con un diseño experimental, la variable dependiente y a predecir es el consumo de energía la cual dependerá de periodos de tiempo y tipo de días de la semana (festivo, laborables). Finalmente, los resultados nos ayudaron a elaborar un modelo matemático que nos ayuda a conocer el comportamiento de la demanda de energía; por lo tanto, se pueden anticipar los consumos máximos y de esta manera dosificar su uso para reducir costos y efectos secundarios en los procesos de producción.

Palabras clave: consumo de energía eléctrica, machine learning, inteligencia artificial, series de tiempo, dosificación de producción.

ABSTRACT

Nexa Resources Cajamarquilla is a company dedicated to the extraction, treatment and transformation of metals. Currently, the price of eco-friendly metals is increasing due to restrictions on the maritime transport of fuels from Russia, which has increased the costs of oil, gasoline and others. The operations of companies that depend on electrical energy generated by these fuels have increased, it is in this sense that it has been proposed to reduce their consumption of electrical energy by applying Machine Learning tools to forecast their maximum energy demand points and be able to dose their production. In the present study, a methodology based on a quantitative structure was applied, relating two or more variables with an experimental design, the dependent variable and to be predicted is the energy consumption, which will depend on periods of time and type of days of the week (holiday, weekdays). Finally, the results helped us to elaborate a mathematical model that helps us to know the behavior of the energy demand; therefore, maximum consumption can be anticipated and in this way its use can be dosed to reduce costs and side effects in production processes.

Key Words: electrical energy consumption, machine learning, artificial intelligence, time series, production dosage.

ÍNDICE DE CONTENIDOS

Introducción 1

Capítulo I: Planteamiento del Problema 2

1.1 Descripción de la Realidad Problemática 2

1.2 Justificación de la Investigación 6

1.3 Delimitación de la Investigación 8

Capítulo II: Marco Teórico 11

2.1 Antecedentes de la Investigación 11

2.2 Bases Teóricas 53

2.2.1 Inteligencia Artificial 53

2.2.2 Machine Learning 54

2.2.3 Aprendizaje Supervisado 55

2.2.4 Series Tiempo 57

2.2.5 Demanda 67

2.2.6 Pronóstico 68

2.2.7 Energía Eléctrica 69

Capítulo III: Entorno Empresarial 70

3.1 Descripción de la empresa 70

3.1.1 Reseña histórica y actividad económica 70

3.1.2 Descripción de la organización 71

3.1.3 Datos generales estratégicos de la empresa 72

3.2 Modelo de negocio actual (CANVAS) 78

3.3 Mapa de procesos actual 80

Capítulo IV: Metodología de la Investigación 82

4.1 Diseño de la Investigación 82

4.1.1 Enfoque de la investigación 82

4.1.2 Alcance de la investigación 82

4.1.3 Tipo de investigación 82

4.1.4 Población y muestra 82

4.2 Metodología de implementación de la solución 83

4.2.1 Recopilación de datos 83

4.2.2 Pre-Procesamiento 83

4.2.3 Modelado 84

4.2.4 Evaluación del Modelo 84

4.3 Metodología para la medición de resultados de la implementación 84

4.4 Cronograma de actividades y presupuesto 85

Capítulo V: Desarrollo de la Solución 88

5.1 Propuesta solución 88

5.1.1 Planteamiento y descripción de Actividades 88

5.1.2 Desarrollo de actividades. Aplicación de herramientas de solución 89

5.2 Medición de la solución 109

5.2.1 Análisis de Indicadores cuantitativo y/o cualitativo 110

5.2.2 Simulación de solución. Aplicación de Software 114

Capítulo VI: Conclusiones y Recomendaciones 136

6.1 Conclusiones 136

6.2 Recomendaciones 137

Referencia Bibliográfica 139

Anexos 143

Índice de Figuras

Figura 1: Estimación del precio de Litio, Cobalto, Níquel y Cobre (20221-2040) 3
Figura 2: Kwh per Cápita - Perú. 5
Figura 3: Consumo de Energía dentro de la organización (2015-2017) 6
Figura 4: Delimitación Espacial. 9
Figura 5: Mapa de operaciones y proyectos. 10
Figura 6: Flujo de información hacia la nube de Google Cloud. 12
Figura 7: Flujo de información de SAP a Google Cloud. 12
Figura 8: Pareto de Jerarquía 1. 13
Figura 9: Gráfico de ventas mensuales según jerarquías de calzado del 2014 al 2020. 14
Figura 10: Modelo NNAR o redes neuronales. 15
Figura 11: Gráfica de datos y tratamiento con modelo NNAR. 15
Figura 12: Gráfico de predicción con modelos NNAR y ETS. 16
Figura 13: Gráfica de predicción según modelo ETS y NNAR. 17
Figura 14: Metodología del proyecto. 19
Figura 15: Cuadro de correlación entre variables. 20
Figura 16: Gráfico de barras para obtener picos y valles de carga eléctrica. 20
Figura 17: Gráficas de grado de correlación de datos con una hora de anticipación. 21
Figura 18: Gráficas de grado de correlación de datos con un día de anticipación de picos altos y valles. 22
Figura 19: El RMSE mensual del año 2018 para el pronóstico de carga eléctrica por hora en el Caso 1. 24
Figura 20: Proyección de datos con modelos SVM y LSTM. 25
Figura 21: Gráfico de barras de datos con modelos LSTM y SVM Caso 2. 26
Figura 22: Proyección de datos con modelos SVM y LSTM Caso 2. 27
Figura 23: El RMSE mensual de 2018 para el pronóstico diario de carga pico y valle en el Caso 3. 28
Figura 24: Tabla de comparación de casos y sus respectivas métricas de error. 28
Figura 25: Desglose de las tecnologías de generación eléctrica en España en marzo del 2019. 29
Figura 26: Distribución del sentido de los desvíos. 31
Figura 27: Distribución de los desvíos (1 para desvíos a subir y -1 para desvíos a bajar) agrupados en diferentes resoluciones temporales. 31

Figura 28: Precio de los desvíos a subir y bajar agrupados por horas y días de la semana. 32
Figura 29: Precio de los desvíos a subir y a bajar agrupados por hora y mes. 33
Figura 30: Flujo de trabajo del modelo propuesto. 38
Figura 31: Demanda máxima de energía eléctrica de los electrodomésticos. 41
Figura 32: % de error de predicción de diferentes modelos predictivos. 41
Figura 33: Arquitectura de la Herramienta de Apoyo a la Decisión para la Gestión Dinámica de Inventario. 43
Figura 34: Ecuación 1: Modelamiento 44
Figura 35: Ecuación 2 Programa Lineal 45
Figura 36: Ecuación 3 Proceso autoregresivo 45
Figura 37: Ecuación 4 Proceso de promedio móvil 45
Figura 38: Ecuación 5 Fusión 46
Figura 39: Configuración del modelo NARNN para la previsión de la demanda. 46
Figura 40: Ecuación 6 Precio 47
Figura 41: Ecuación 7 Resultado de predicción 47
Figura 42: Comparación entre la gestión de inventarios con y sin la herramienta de decisión propuesta. 48
Figura 43: Promedio de ventas agregadas por sección. 49
Figura 44: Ejemplos de comportamientos de productos. 50
Figura 45: MAE por modelo a distintos horizontes. 51
Figura 46: Variables cuantitativas o cualitativas. 55
Figura 47: Clasificación. 56
Figura 48: Regresión Lineal. 57
Figura 49: Componentes de una serie de tiempo. 58
Figura 50: Coeficiente del modelo en función de la regularización. 59
Figura 51: Ecuación de MSE. 60
Figura 52: Ecuación RMSE. 61
Figura 53: Ecuación MAE. 61
Figura 54: Diagrama de cajas y bigotes. 64
Figura 55: Mapa conceptual de diagrama de caja y bigote 65
Figura 56: Diagrama de dispersión 66
Figura 57: Diagrama de Violín. 67
Figura 58: Curva de demanda 68
Figura 59: Modelo de pronóstico cuantitativo. 69

Figura 60: Organigrama 71
Figura 61: Cadena de Suministro Nexa Resources. 72
Figura 62: Leyenda de Mapa de procesos 80
Figura 63: Mapa de procesos de Nexa Resources Cajamarquilla 81
Figura 64: Metodología de implementación de la solución. 83
Figura 65: Formato de data. 90
Figura 66: Comportamiento de la demanda eléctrica periodo 2017-2022 91
Figura 67: Zoom comportamiento de la demanda mes de febrero 2019. 91
Figura 68: Distribución de la demanda por mes 92
Figura 69: Distribución de la demanda por semana 92
Figura 70: Distribución de la demanda por hora del día 93
Figura 71: Distribución de la demanda entre días festivos y no festivos 94
Figura 72: Gráfica de autocorrelación 94
Figura 73: Gráfica de autocorrelación parcial 95
Figura 74: Base de datos de demanda eléctrica de Nexa Resources 96
Figura 75: Base de datos procesada 97
Figura 76: Verificación de data completa 98
Figura 77: Completar datos incompletos o vacíos 98
Figura 78: Conversión al formato fecha 98
Figura 79: Delimitación de la data 99
Figura 80: Asignación de variables 100
Figura 81: Separación de datos en train y test 100
Figura 82: Modelamiento de la regresión lineal 101
Figura 83: Sklearn y Skforecast 101
Figura 84: Entrenamiento del forecaster 102
Figura 85: Grid Search 102
Figura 88: Variable Exógena Feriados 105
Figura 89: Gráfica de dispersión regresión lineal 106
Figura 91: Gráfica de predicción vs demanda real primer forecast intervalo predicción 107
Figura 92: Gráfica de predicción vs demanda real Grid Search 107
Figura 93: Gráfica de predicción vs demanda real predicción diaria anticipada 108
Figura 94: Gráfica de predicción vs demanda real variable exógena 108
Figura 95: Gráfica de predicción vs demanda real variable exógena con predicción anticipada 109

Figura 96: MAE 110
Figura 97: MSE 111
Figura 98: RMSE 111
Figura 99: Varianza 111
Figura 100: Error backtest forecast inicial 112
Figura 101: Cobertura del intervalo predicho forecast inicial 112
Figura 102: Error backtest Grid Search 113
Figura 103: Error backtest predicción diaria anticipada 113
Figura 104: Error backtest variable exógena 113
Figura 105: Error backtest variable exógena con predicción diaria anticipada 114
Figura 106: Importación de librerías y algoritmos a utilizar 114
Figura 107: Importación de datos 115
Figura 108: Conversión del formato fecha y delimitación de data 115
Figura 109: Variable independiente 116
Figura 110: Variable dependiente 116
Figura 111: Definición de set de entrenamiento y set de prueba 117
Figura 112: Elegir el modelo - alumno 117
Figura 113: Entrenar al alumno 117
Figura 114: Examen - predicción 117
Figura 115: MSE, MAE, RMSE, coeficientes de la recta y varianza 118
Figura 116: Gráfico de dispersión 119
Figura 117: Importación de librerías y algoritmos a utilizar 120
Figura 118: Importación de datos 121
Figura 119: Conversión del formato fecha y la verificación de la data completa 121
Figura 120: Convirtiendo data a intervalos de 1 hora 122
Figura 121: Convirtiendo variable exógena a entero 122
Figura 122: Adición de nuevos parámetros 123
Figura 123: Paso 1: Separando los datos en train y test 123
Figura 124: Paso 2 y 3: Eligiendo el modelo y entrenando al alumno 124
Figura 125: Paso 4: Examen - predicción 124
Figura 126: Paso 2 y 3: Eligiendo el modelo y entrenando al alumno Grid Search 125
Figura 127: Paso 4: Examen – predicción Grid Search 126
Figura 128: Paso 2 y 3: Eligiendo el modelo y entrenando al alumno predicción anticipada 126
Figura 129: Paso 4: Examen – predicción diaria anticipada 127

Figura 130: Paso 2 y 3: Eligiendo el modelo y entrenando al alumno variable exógena 128
Figura 131: Paso 4: Examen – variable exógena 128
Figura 132: Paso 2 y 3: Eligiendo el modelo y entrenando al alumno variable exógena con predicción diaria anticipada 129
Figura 133: Paso 4: Examen – variable exógena con predicción diaria anticipada 129
Figura 134: Paso 5: Backtest e intervalo de cobertura forecast inicial 130
Figura 135: Paso 5: Backtest Grid Search 130
Figura 136: Paso 5: Backtest predicción anticipada diaria 130
Figura 137: Paso 5: Backtest variable exógena 131
Figura 138: Paso 5: Backtest variable exógena con predicción diaria anticipada 131
Figura 139: Paso 6: Programación y gráfica del forecast inicial 132
Figura 140: Gráfico intervalo de predicción del forecast inicial 132
Figura 141: Programación y gráfica del Grid Search 133
Figura 142: Programación y gráfica del predicción diaria anticipada 134
Figura 143: Programación y gráfica de la variable exógena 134
Figura 144: Programación y gráfica de la variable exógena con predicción diaria anticipada 135

Índice de tablas

Tabla 1: Tabla de resultados de predicción del sentido de los desvíos aplicando modelos de clasificación para multi-output. 35
Tabla 2: Tabla de resultados de predicción del precio de los desvíos aplicando modelos de regresión para multi-output. 36
Tabla 3: Descripción del valor medio de las siete características por aparato eléctrico. 40
Tabla 4: Comparación entre modelo de clasificación a una semana con modelos, entrenado con los mejores dos. 52
Tabla 5: Matriz FODA 74
Tabla 6: FODA cuantitativo 77
Tabla 7: Modelo de Negocio Actual. 79
Tabla 8: Población y muestra de la investigación. 82
Tabla 9: Metodología para la medición de resultados (REGRESIÓN LINEAL). 84
Tabla 10: Metodología para la medición de resultados (SERIES DE TIEMPO) 85
Tabla 11: Cronograma de actividades 86
Tabla 12: Presupuesto de la investigación 87
Tabla 13: Resultado del modelo de regresión lineal 110

Introducción

Actualmente muchas empresas están destinando sus esfuerzos a una correcta ejecución operacional, pero lo que muchas no involucran dentro de estas acciones es la correcta gestión de insumos la cual no solo se basa en materia prima sino también en recursos. Cada rubro tiene insumos críticos a los cuales se debe hacer un correcto seguimiento, pero en muchas ocasiones no se tiene mapeadas los recursos críticos empleados.

Cada empresa posee recursos variados de acuerdo al rubro dentro del cual está desempeñando sus funciones. Para el caso de una fundidora, los equipos críticos son los hornos, pero estos pueden funcionar mediante combustión con la ayuda de algún combustible, pero también los hornos pueden ser eléctricos. Es por eso la importancia de tener mapeado los equipos críticos y sus principales inputs que garantizaran su correcto funcionamiento. Dentro del rubro minero, la energía es vista como un recurso crítico dentro de operaciones por todo el impacto que genera en la planta de electrólisis.

La energía que se distribuye por toda la sección de Lima tienen un límite que no debe superarse en consumo, para esto existe una demanda potencial la cual no debe superarse debido a que el incumplimiento generaría costos excesivos de manera mensual impactando en la rentabilidad de la empresa. Por otro lado, es importante que las empresas orienten sus esfuerzos a una correcta gestión de recursos garantizando el correcto desempeño de las operaciones.

Frente a todo este contexto las áreas de planeamiento y control de producción forman parte importante haciendo el correcto seguimiento de los inputs del proceso para aumentar la productividad de las organizaciones.

En la actualidad existen muchas herramientas que ayudan a una correcta gestión de la empresa. Al igual que cualquier paso de un planeamiento, debe haber una estrategia a seguir para una correcta ejecución. Para esto existen múltiples modelos predictivos que generar identificar un día de demanda máxima para lo cual será necesario dosificar energía sin perder un correcto flujo continuo de producción.

Las empresas siempre estarán a disposición de sus recursos y es que la producción siempre será la excelencia si existe a la par una correcta gestión de insumos y recursos.

Capítulo I: Planteamiento del Problema

1.1 Descripción de la Realidad Problemática

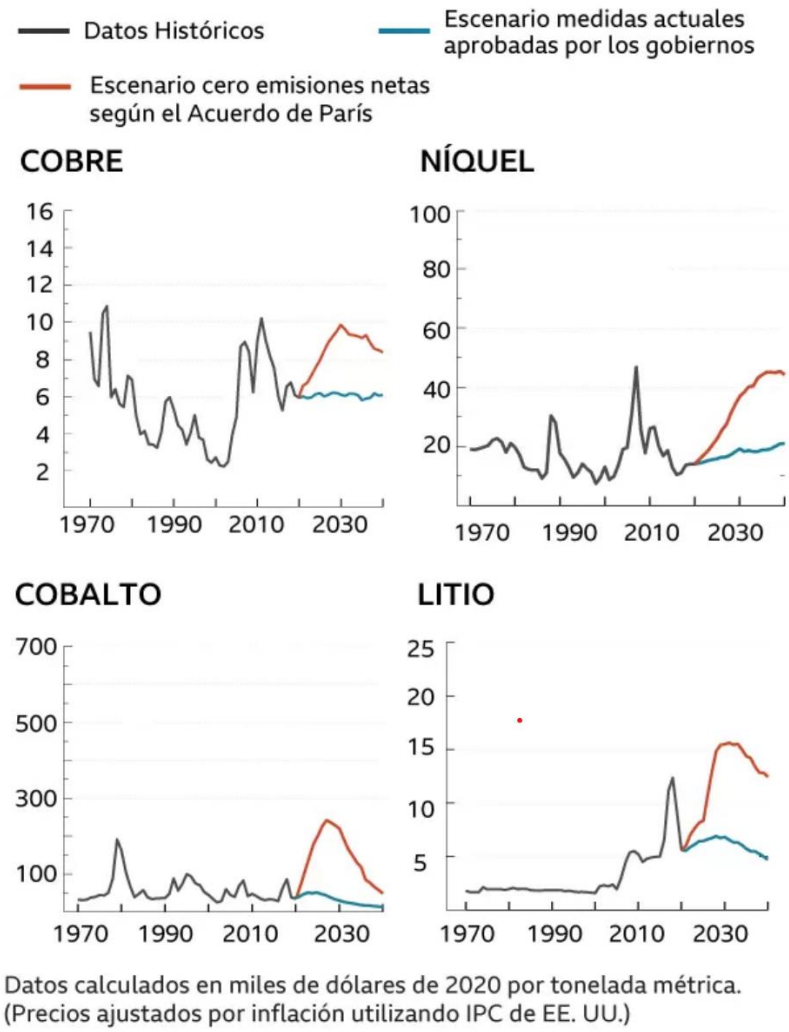
A inicios de año los precios de los metales del futuro y relacionados a las energías limpias y sostenibles ambientalmente presenciaron una impactante subida en sus precios y su demanda, metales como el Níquel, el Cobalto, el Lito y el grafito, según un informe del Grupo del Banco Mundial, se señala que la producción de minerales, como el grafito, el litio y el cobalto, podría experimentar un aumento de casi un 500 % de aquí a 2050, para satisfacer la creciente demanda de tecnologías de energía limpia. Se estima que se requerirán más de 3,000 millones de toneladas de minerales y metales para la implementación de la energía eólica, solar y geotérmica, así como el almacenamiento de energía, para lograr una reducción de la temperatura por debajo de los 2 °C en el futuro.

En el informe "Minerals for Climate Action: The Mineral Intensity of the Clean Energy Transition" también se indica que, si bien las tecnologías de energía limpia requerirán una mayor cantidad de minerales, la huella del carbono relacionada con su producción, desde la extracción hasta el consumo final, representará apenas el 6 % de las emisiones de gases de efecto invernadero generadas por las tecnologías basadas en combustibles fósiles. También se indica que incluso si se incrementaran en un 100 % las tasas de reciclado para minerales como el cobre y el aluminio, el reciclaje y la reutilización seguirán siendo insuficientes para satisfacer la demanda de tecnologías de energía renovable y almacenamiento de energía.

Esta necesidad de nuevas energías limpias y sostenibles también se vieron aceleradas gracias al contexto social y político que se vive actualmente en Europa con la invasión de Rusia a Ucrania, lo que ocasionó que la mayoría de países que eran socios comerciales de Rusia y grupos internacionales de países como la ONU, la OTAN y la Unión Europea impongan sanciones a Rusia tanto monetarias como en acuerdos comerciales, restringiendo la comercialización de sus recursos naturales con los países occidentales en contra de sus acciones en territorio ucraniano. Rusia como uno de los principales productores de gas y petróleo, demostró su gran influencia sobre muchos países gracias a la dependencia de estos del gas y petróleo ruso, que son fuente primordial de su energía. Por ejemplo, el último suceso sucedió hace poco cuando Rusia cortó el suministro de gas a Alemania, quien hasta ese momento se había mostrado a favor de todos los reclamos y advertencias hacia Rusia contra la invasión de Ucrania, pero sin mostrar un lado radical o de rivalidad, de todos modos, Rusia decidió tomar

medidas radicales para desestabilizar a uno de los principales países industriales y productivos del mundo.

Figura 1: Estimación del precio de Litio, Cobalto, Níquel y Cobre (20221-2040)



Fuente: Boer, Lukas y Pescatori, Energy Transition Metals (2021)

Sin embargo, todo este aumento exponencial en el precio relacionados a energías limpias no significa un aumento en los precios y demanda de los metales industriales como el estaño, aluminio, zinc, cadmio entre otros, que, a la fecha de hoy, han sufrido caídas históricas, mostrando un acercamiento brusco a una recesión. Materiales como el plomo o el zinc están en mínimos de 2022, pero todavía tienen margen para bajar de precio, al menos si lo comparamos con antes de la pandemia. El cobre actualmente cotiza en unos 8,500 dólares, lejos de los casi 11.000 dólares de principios de año, pero todavía por encima de los 6,000 dólares que tenía antes de la pandemia, y en el peor momento de esta logró bajar a 5,000 dólares.

El Perú y Latinoamérica no son ajenos a estos sucesos que se dan en el continente europeo, debido a que la escasez de combustible aumenta el precio de los transportes y fletes de los minerales, metales y otros recursos primarios provenientes de todas partes del mundo, derivando en un alza de los precios de los mismos. En Perú donde la mayoría de sus centrales eléctricas aún funcionan con combustibles fósiles no renovables como el petróleo, la escasez del mismo provoca aumento en los precios de transporte, aumento en los precios de la energía eléctrica, desabastecimiento de combustibles, debido a la oferta peruana no abastece la demanda, por lo tanto Perú es un gran dependiente de la importación de combustibles, aunque contamos con unas grandes reservas de gas, las cuales en su mayoría son para exportación debido a los mejores precios y cotizaciones en el extranjero que en territorio nacional.

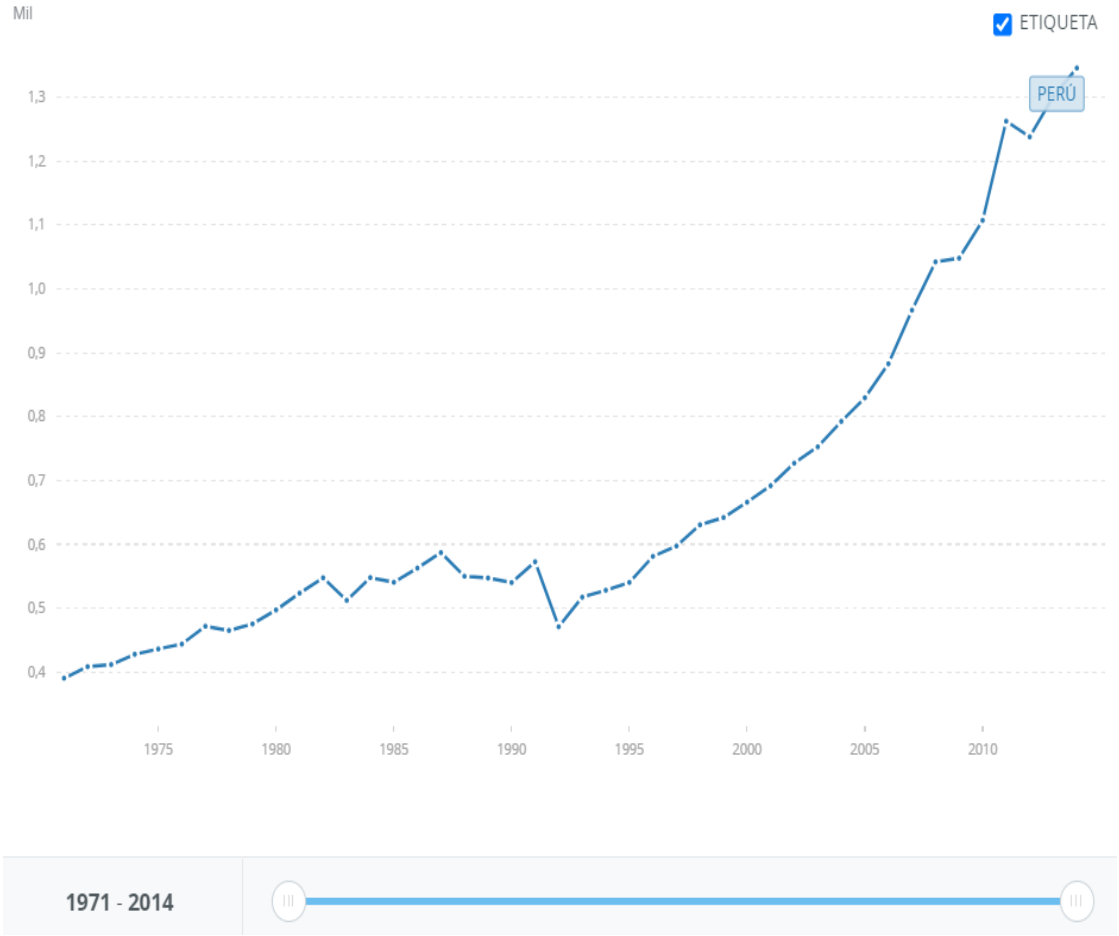
Toda esta cadena de sucesos y factores exteriores afecta a grandes compañías que laboran en nuestro país, como por ejemplo la unidad minera Nexa Cajamarquilla S.A., principal extractora y procesadora de zinc en el país, cuyos consumos de energía son considerables de acuerdo a su operación y cantidad de metal procesado. Por ejemplo uno de los grandes problemas que actualmente afectan considerablemente las finanzas y operaciones de Nexa es el consumo alto de energía en los picos altos de demanda de fabricación de los metales como zinc y cadmio, donde por razones obvias hay horas y momentos del día donde el abastecimiento de la EPS encargada de brindar el servicio de energía eléctrica en el distrito de Lurigancho, Lima, no sea suficiente el nivel de energía que le proporciona a la planta de Nexa Cajamarquilla ocasionando que utilicen su propia central de energía que aumenta los costos de producción, aumenta el consumo de la energía eléctrica y en muchas ocasiones al ser una generador muy potente descargue más energía que las máquinas que procesan los metales pueden soportar, generando paradas correctivas, las cuales ocasionan tiempos muertos, disminución de la producción, gasto en mantenimiento y otros problemas.

En la actualidad, muchas empresas tienen conciencia ambiental al momento de aprovechar los recursos renovables y no renovables pero muy pocos conocen el impacto que genera la saturación de estos en los equipos, producción y costos. La producción de cualquier empresa se mide bajo conceptos de eficiencia y eficacia, pero las empresas deben cuestionar que tan bien están aprovechando sus recursos orientados a una mejora en los flujos de producción.

El Perú viene con un incremento de consumo de energía eléctrica (Kwh per cápita) a lo largo de los años, lo cual indica que existe una mayor demanda. Para el año 2012 se tenía como consumo 1238 mil kwh, para el año 2013 esta cifra tuvo un incremento de 1301 kwh.

Finalmente, para el año 2014 la cifra obtenida fue de 1346 kWh. Esto puede estar arraigado a muchos factores, pero para fines de esta investigación veremos cómo repercute en las operaciones de la unidad de Cajamarquilla.

Figura 2: Kwh per Cápita - Perú.



Fuente: Banco Mundial (2014)

Nexa no es la excepción ya que conforme los años pasan, el consumo de energía viene siendo mayor.

Figura 3: Consumo de Energía dentro de la organización (2015-2017)

	2015	2016	2017
No renovables	**2.958.602**	**2.943.137**	**2.367.039**
Aceites (aceite y diésel)	2.132.655	2.297.137	1.604.598
Coque	577.459	416.021	413.286
Gas natural	182.528	168.931	292.628
Otros combustibles no renovables	65.959	61.048	56.526
Renovables	**11.773.036**	**12.277.147**	**12.479.865**
Energía eléctrica generada	1.990.768	2.478.583	2.355.143
Energía eléctrica comprada – otras fuentes	9.781.841	9.798.194	10.124.316
Otros combustibles renovables	427	370	406
Total	**14.731.637**	**15.220.284**	**14.846.904**

Fuente: Nexa (2022)

Se puede apreciar un aumento de energía eléctrica comprada a lo largo de los años. Si bien la energía renovable marcha en buenas condiciones, la energía eléctrica viene siendo mayor en adquisición. Existe un evento que sucede una vez al mes en donde se da la demanda máxima la cual se le conoce como consumo máximo de energía eléctrica. Durante este día los equipos se sobresaturan debido a la cantidad excesiva de carga que está entrando a la unidad. Esto trae consigo consecuencias como paradas de planta intempestivas para mantenimiento de equipos, bajar el flujo de producción, aumentos en costos, etc. Es de importancia poder administrar la energía, dosificarla a fin de obtener un mejor performance de las operaciones ya que existe una sinergia entre operaciones y mantenimiento.

1.2 Justificación de la Investigación

En las industrias mineras, existen diversas problemáticas de gran envergadura con respecto al consumo de energías renovables y no renovables, debido a la alta demanda de los equipos que trabajan en el sistema de producción de la materia prima, por lo cual, se ha tomado como enfoque el estudio del consumo de este recurso y del impacto positivo y negativo del mismo, a fin de poder tomar medidas correctivas que permitan establecer parámetros para el uso estratégico de la energía.

1.2.1 Justificación Teórica

Para el estudio del consumo de la energía eléctrica y el comportamiento de su demanda en la empresa Nexa, se ha determinado como herramienta principal la técnica de Machine

Learning, a fin de poder realizar un aprendizaje de datos relacionados y establecer patrones que ayuden a establecer una predicción con respecto al tiempo en que se suscitan los puntos máximos de consumo de energía, a partir de la metodología de regresión lineal.

La regresión lineal dentro del campo de Machine Learning, es un algoritmo de aprendizaje supervisado que permitirá modelar la relación entre una variable dependiente (consumo eléctrico) y una variable independiente (tiempo en horas), mediante el estudio de datos continuos de horario y consumo eléctrico, estableciendo una tendencia delimitada por una recta, la cual nos faculta de poder realizar predicciones ante el ingreso de nuevos datos de ambas variables.

A partir del lenguaje de programación desarrollado para nuestra base de datos, podremos cumplir con la etapa de entrenamiento y predicción de nuestra técnica de machine learning, permitiéndonos conocer los resultados como los horarios donde puede suceder la demanda máxima de una forma más detallada (Horario específico), además de a qué nivel de energía eléctrica (Kwh) se desarrolla este punto máximo, para poder aplicar y desarrollar estrategias de mejora que reduzcan los impactos negativos de esta sobrecarga de energía.

1.2.2 Justificación práctica

La finalidad de la aplicación de la técnica de machine learning en el estudio de la demanda máxima de energía eléctrica es encontrar los horarios más probables en los que se suscita este evento, a fin de poder establecer mecanismos de dosificación de energía en un horario previo al desenlace del consumo máximo, permitiendo así reducir la carga en los diferentes sistemas eléctrico con los que trabaja la maquinaria, concluyendo en lo siguiente:

- Reducción de costos por desgaste o mantenimiento de las máquinas de producción: Ante una reducción de carga eléctrica de los equipos de producción a partir de la dosificación anticipada, se podrá incrementar la vida útil de los mismos, incurriendo en menor proporción a costos por reemplazo de ciertas piezas por propio desgaste de funcionamiento.
- Reducción de costos por sobre incremento de consumo de energía de acuerdo con la tarifa: La empresa tiene una tarifa establecida de manera contractual con el proveedor de energía eléctrica, la cual tiene ciertas condiciones y limitaciones con respecto a los niveles de consumo del recurso. Por lo cual, superar el límite diario de consumo en Kwh, trae consigo pagar montos de más de S/.1,000,000, por lo que la dosificación de energía

anticipada va a permitir mantenerse al margen de la superación de estas limitaciones y, por ende, de tener que pagar estos montos significativos de dinero de manera mensual.

- Incrementar los niveles de producción reduciendo incidencias de no funcionamiento de las máquinas procesadoras: Al poder determinar la dosificación de energía en el horario propicio, va a permitir que la cadena de procesos no se vea interrumpida ante cualquier incidente por sobrecarga, manteniendo las cantidades pronosticadas en el PCP y, así, cumpliendo con la demanda solicitada por los diferentes clientes de la empresa, evitando pago de penalidades.

1.2.3 Justificación metodológica

De acuerdo con la cantidad de datos con respecto a la demanda de energía desde el año 2017 al año 2020, se propone utilizar la técnica de KDD (Knowledge discovery in databases), también denominado descubrimiento de conocimiento de base de datos.

Según la Universidad ESAN (2018) el KDD es un proceso de análisis de patrones que responden a tres causas: importancia, utilidad y comprensibilidad. Debido a la cantidad de datos considerable, el objetivo de agrupar y seguir los parámetros pasa a estar en segundo plano. El objetivo primordial de esta técnica es poder interpretar patrones, para así lograr tomar decisiones estratégicas en función al análisis de modelos y datos.

En el caso de demanda de energía, se utilizará la lógica y el análisis de un equipo de profesionales para visualizar, interpretar y modelar patrones que permitan a los datos transformarse en información sólida para encontrar y determinar los días de mayor demanda y poder tomar acciones de dosificación de producción de manera óptima y estratégica.

1.3 Delimitación de la Investigación

1.3.1 Delimitación Espacial

Este trabajo de investigación ha sido realizado en el distrito de Lurigancho en la unidad smelter de Nexa Cajamarquilla. La refinería actualmente cuenta con 4 plantas y áreas que van desde la sección 10 (área de concentrados) hasta la sección 90 (área de tratamiento de agua).

Figura 4: Delimitación Espacial.

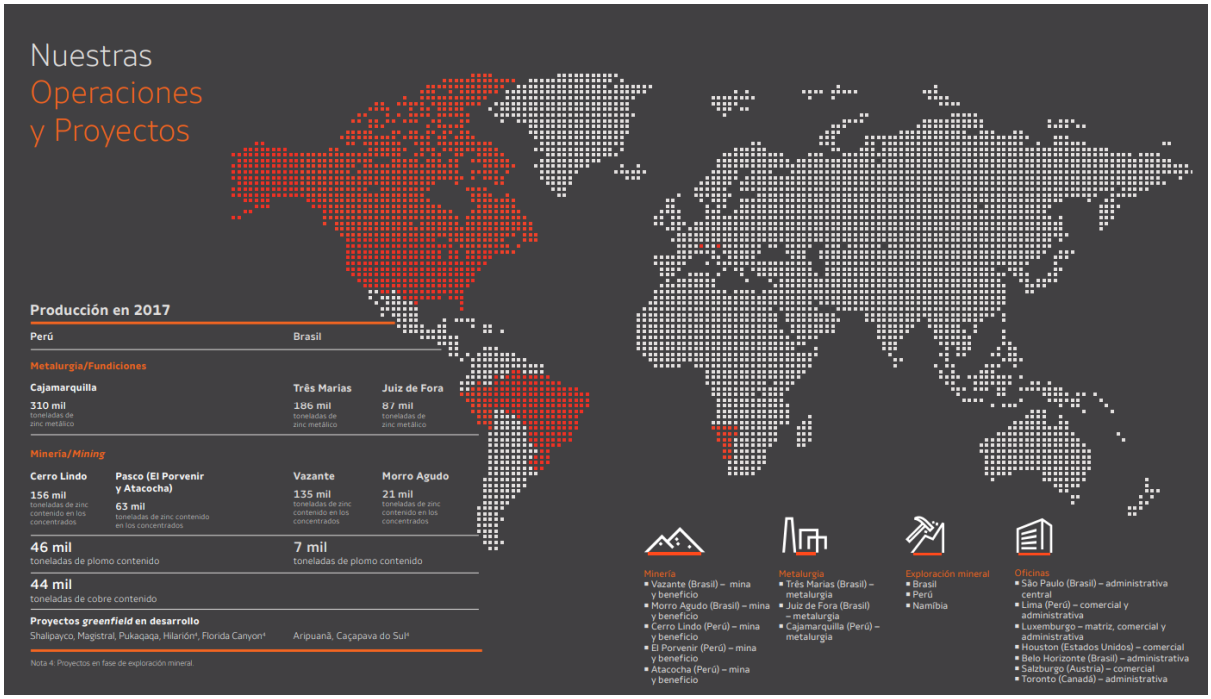
Fuente: NEXA CJM (2022)

En términos específicos, la investigación tomará lugar en el área de PCP (planeamiento y control de producción) considerando la energía como objeto de estudio y su aplicación en las diferentes áreas de la refinería. Este trabajo recolecta información de consumo de energía en intervalos de 30 minutos a fin de construir un modelo preciso y fiable que permita resultados óptimos.

1.3.2 Delimitación Temporal

La investigación usará información formal de la empresa Nexa CJM, una empresa de origen brasileño antes conocida como Grupo Votoramtin iniciando sus operaciones el 15 de febrero de 1995 siendo pioneros en la producción de zinc refinado a un nivel de pureza de 99,995% apuntando a clientes nacionales y extranjeros.

Figura 5: Mapa de operaciones y proyectos.



Fuente: Nexa (2022)

En términos específicos, el presente estudio toma un universo de 100,000 datos de consumo de energía calculados cada media hora desde el año 2017. Por políticas de la empresa no se puede acceder a una totalidad de datos. Dentro de la data obtenida se aprecian los picos de energía mensual los cuales suceden una vez al mes ocasionando costos elevados de uno a dos millones de soles adicionales.

Debido a que el modelo cuenta con variables que buscarán correlacionar el consumo de energía de otras empresas para hallar tendencias, se hizo necesario la adquisición de esta a fin de buscar tendencias en los días del evento. A fin de tener está comparación, se redujo a una base de datos de 60 000 datos en consumo a lo largo de los turnos de producción.

Capítulo II: Marco Teórico

2.1 Antecedentes de la Investigación

Ancco Y., Tatiana (2021) Análisis comparativo de series de tiempo para proyectar las ventas en las jerarquías de calzado en una empresa del sector retail.

Problema

La problemática se enfatiza en el área de compras en la empresa Grupo M&W, la cual tiene como principal objetivo garantizar la compra de productos que soliciten las diferentes áreas y subáreas, además de monitorear la venta de estos. Este proceso parte de la necesidad de compra de productos de acuerdo con la temporada, la cual, según las ventas pronosticadas, permitirá establecer una solicitud con cantidad determinada de productos (de acuerdo a jerarquía) para poder cumplir con la demanda de los clientes.

El problema yace en la falla del pronóstico de la demanda, lo cual trae consigo sobre stock de productos, a partir de la compra excesiva de ciertos productos que no cumplieron con la demanda predicha. A partir de ello, también se produce la reducción de rentabilidad en la venta de cada producto, ya que, al superar el tiempo de la temporada del producto, la empresa se ve obligada a reducir sus precios y, por ende, su margen de ganancia.

Objetivo

El desarrollo de este trabajo tiene como principal objetivo encontrar un modelo predictivo con mayor porcentaje de eficiencia en la obtención de resultados, los cuales deben estar ajustados a las ventas mensuales de acuerdo con la jerarquía de calzado.

Datos

Los datos que se van a utilizar para poder realizar el análisis predictivo es data de las ventas en unidades de todas las jerarquías de productos entre febrero del 2014 a junio de 2021, la cual será obtenida mediante SAP y Google Cloud

Metodología

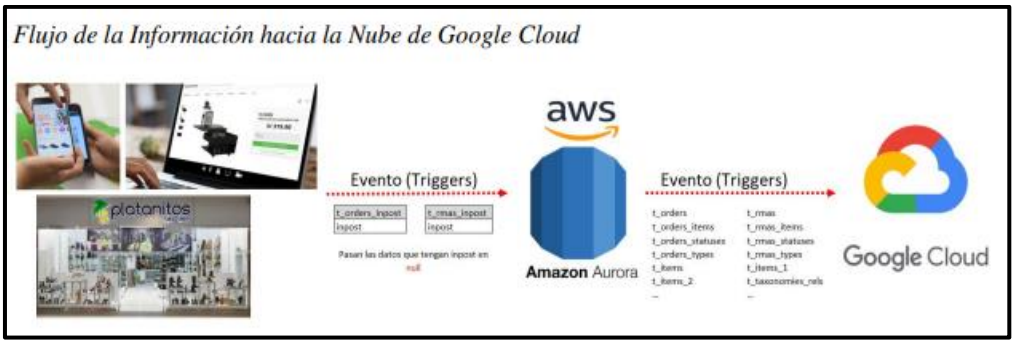
El trabajo presenta un enfoque cuantitativo, pues la variable en estudio es numérica.

- Procedimiento para la obtención de datos: Al tener dos fuentes de datos, Sap y *Google Cloud*, para el procesamiento es necesario la consolidación de data, por

lo cual fue necesario realizar el agrupamiento de los datos para una evaluación y análisis posterior.

Los datos que se almacenan en la nube provienen de distintas fuentes, ya sea de las aplicaciones web, cajas registradoras de venta física, entre otros. Por este motivo, los datos se guardan en 2 tablas temporales: t_orders_inpost y t_rmas_inpost. Después de haber completado las tablas en mención, se procede a llenar las tablas de órdenes y créditos. Por último, los datos viajan cada 5 minutos a la nube Google Cloud para ser almacenados.

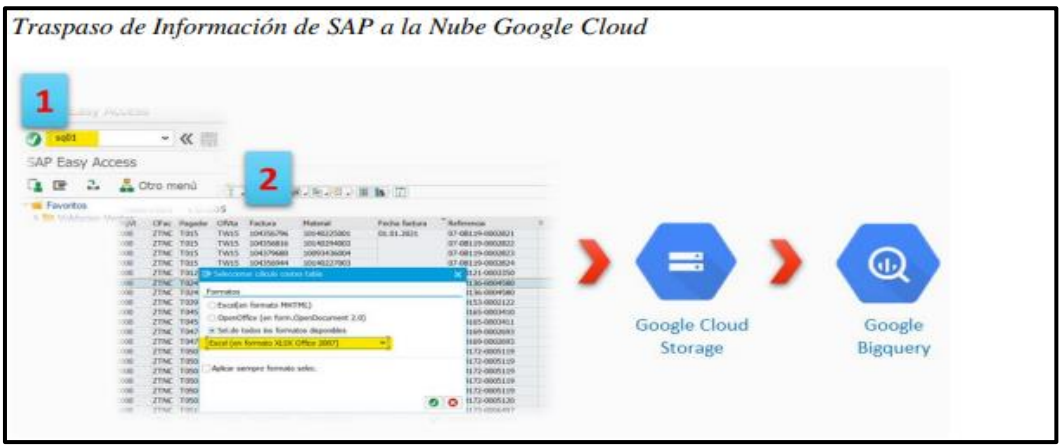
Figura 6: Flujo de información hacia la nube de Google Cloud.



Fuente: Ancco Y. (2021)

Para obtener datos del SAP, se tiene que utilizar la transacción “SQ01” y proceder a descargar la información de febrero a diciembre en las fechas establecidas, para luego ser cargados en la nube donde se almacenan los otros datos.

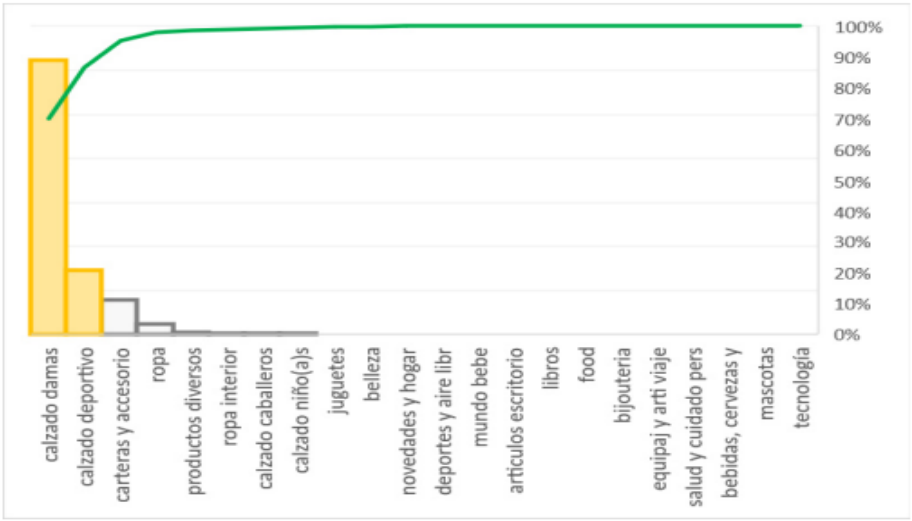
Figura 7: Flujo de información de SAP a *Google Cloud*.



Fuente: Ancco Y. (2021)

- Herramientas: Se utilizó el software R dentro de la plataforma Jupyter Notebook, donde se ingresó la información agrupada de la nube y SAP. Además, se utilizó paquetes como bigquery, tidyverse, dbplyr,DBI, lubridate, forecaste, car y nortest.
- Población: Ventas mensuales de productos según jerarquía en la temporada de verano entre febrero 2014 a junio del 2021.
- Muestra: Ventas mensuales de productos según jerarquía, ya sea calzado de dama y calzado deportivo de la temporada. Para efectos de estudio, las jerarquías seleccionadas representan el 80% de la data de ventas, la cual es dividida en muestra de entrenamiento (febrero 2014 a marzo 2021) y muestra de prueba (abril 2021 a junio 2021).

Figura 8: Pareto de Jerarquía 1.

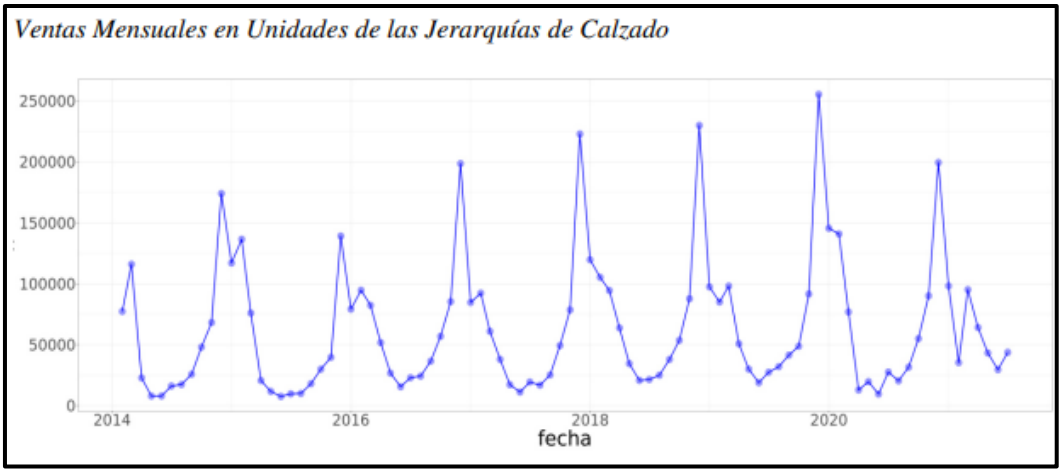


Fuente: Ancco Y. (2021)

Técnicas de Machine Learning.

Para la aplicación de series de tiempo, fue necesario elaborar gráficas de dispersión para la data de ventas entre las fechas seleccionadas, a fin de poder encontrar estacionalidad en las curvas resultantes cada cierto tiempo.

Figura 9: Gráfico de ventas mensuales según jerarquías de calzado del 2014 al 2020.



Fuente: Ancco Y. (2021)

Tal como se observa en la Figura, se demuestra los patrones repetitivos en la demanda de productos según avanza el tiempo, además de que la gráfica no denota un carácter de tendencia, por lo cual sustenta el grado de estacionalidad de la información.

Otra de las herramientas que ayudan a denotar que tipo de técnica se va a utilizar para el tipo de datos presentado, es el diagrama de cajas (mide la estacionalidad de acuerdo con la gráfica), correlogramas (tendencia y estacionalidad que presenta la serie con respecto a sus correlaciones tardadas simples y parciales), periodograma (verificación de ciclos repetitivos), entre otros.

Modelo ETS

El modelo exponencial tiene como objetivo ajustar el modelo exponencial que se adapta de una mejor forma a la serie de tiempo, comparándolo mediante la herramienta AIC, AICc y BIC (criterio de evaluación de modelos en términos de sus probabilidades posteriores).

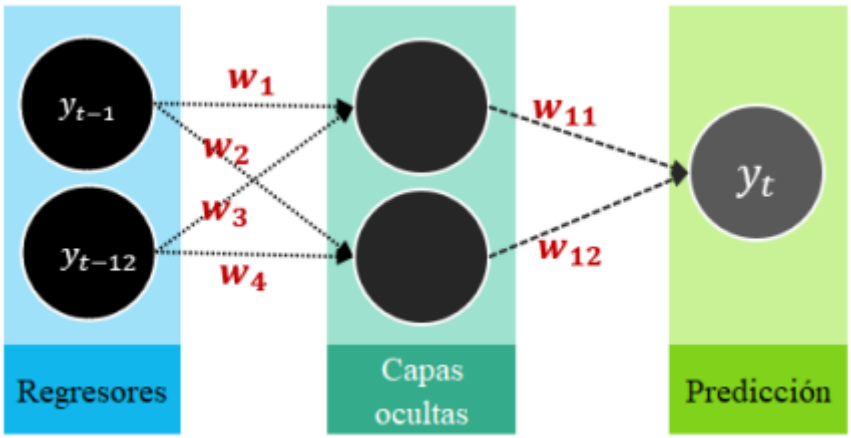
Por otro lado, para efectos de corroborar los errores de resultantes del modelo sean los óptimos, a partir de las herramientas de aleatoriedad, incorrelación y normalización de datos erróneos.

Modelo NAAR

La aplicación de redes neuronales en cada iteración se hizo con modelo NNAR (1,1,2)12, el primer y segundo valor corresponden a la parte no estacionaria y estacionaria de la serie, lo que nos indica que para el modelo solo se usa una variable retardada para modelar

la tendencia y una variable retardada estacional para modelar la estacionalidad. El tercer valor que se muestra corresponde a los nodos ocultos (2) en las capas intermedias.

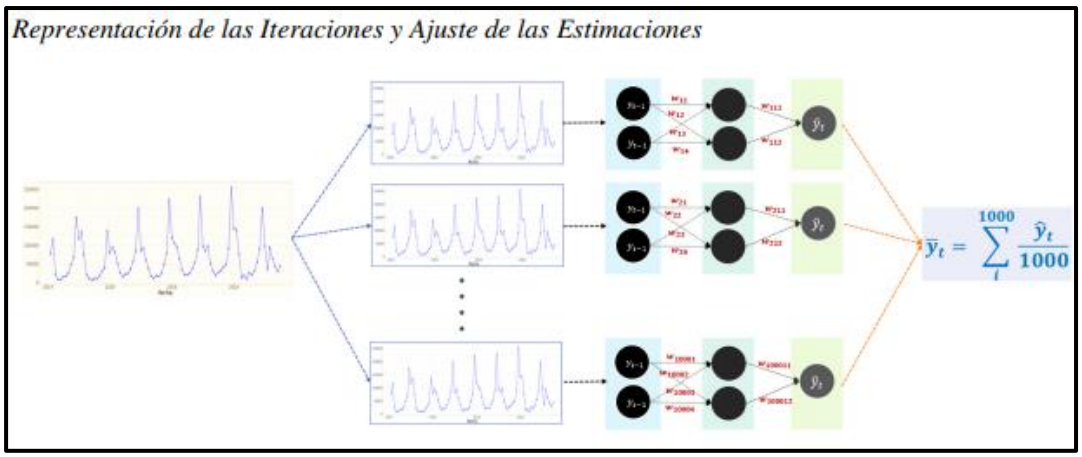
Figura 10: Modelo NNAR o redes neuronales.



Fuente: Ancco Y. (2021)

Para lograr una mejora en las predicciones en este modelo, es primordial hacer iteraciones en búsqueda de una corroboración interna para que se vea reflejada en proyecciones más consistentes.

Figura 11: Gráfica de datos y tratamiento con modelo NNAR.

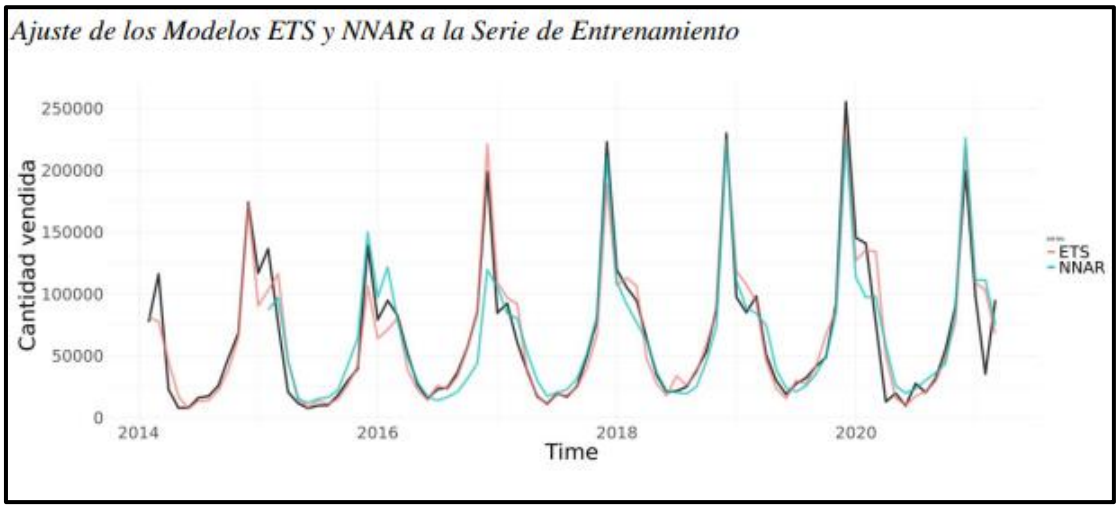


Fuente: Ancco Y. (2021)

Evaluación de errores

Para poder optimizar el desarrollo de predicciones, el análisis comparativo de métricas de errores va a permitir determinar el modelo con mejores resultados. Para ello, en la programación se trabaja con la serie de entrenamiento y prueba:

Figura 12: Gráfico de predicción con modelos NNAR y ETS.



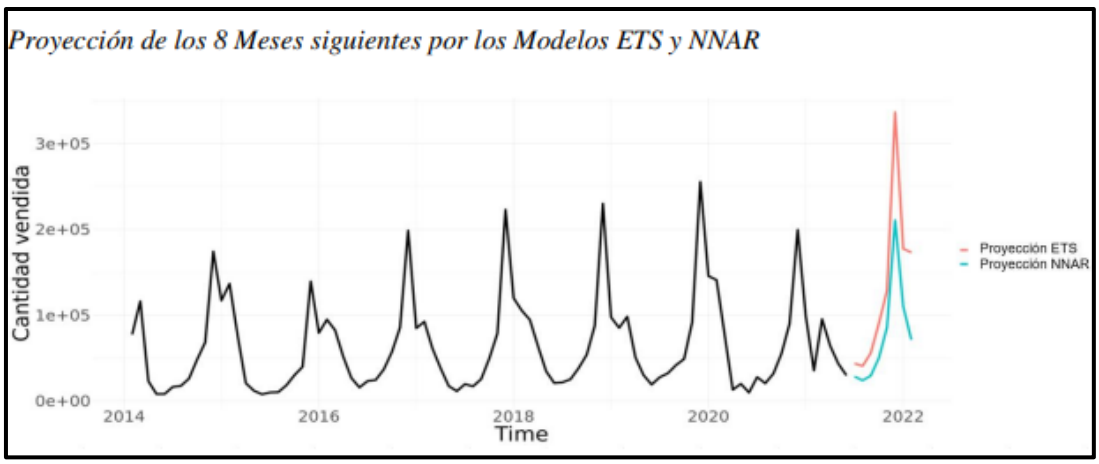
Fuente: Ancco Y. (2021)

El análisis gráfico permite determinar qué modelo se ajusta mejor a la gráfica original. En este caso, tanto el modelo ETS como el modelo NNAR en ciertos tramos de tiempo uno se ajusta mejor que el otro, teniendo en cuenta que en los meses de pandemia existe un error de pronóstico tanto para el modelo ETS y NNAR. Para saber elegir el modelo más apropiado, se evalúan las métricas de errores (MSE. RMSE, MAE, WAPE y SMAPE).

Resultados

Para efectos comparativos de ambos modelos, se requiere hacer la proyección superpuesta con la gráfica original a partir de la data de entrenamiento, obteniendo lo siguiente:

Figura 13: Gráfica de predicción según modelo ETS y NNAR.



Fuente: Ancco Y. (2021)

Como se puede observar, la proyección que se obtiene a partir de cada modelo permite diagnosticar lo siguiente:

- Mediante el modelo ETS se tiene un comportamiento con mayor cantidad de ventas con el pasar de los años, lo cual difiere con la data histórica de años anteriores (diciembre).
- Mediante el modelo NNAR se visualiza un comportamiento conservador con respecto a años anteriores, por lo cual se ajusta mejor al modelo original.
- Para modelos exponenciales, con el pasar de los meses la incertidumbre aumenta, por lo cual las proyecciones no van a estar tan acordes a la realidad.

Palloneto,F., Jin,C. y Mangina,E. (2022) Forecast electricity demand in commercial building with machine learning models to enable demand response programs. Energy and AI. 1-13.

Problema

En Europa existen regulaciones tanto para empresas como para personas con respecto a la emisión de carbono a partir del consumo de energía. Según datos estadísticos, la emisión por generación de electricidad representa el 25 % de todas las emisiones mundiales. Además, teniendo en cuenta la información con respecto a los edificios, estas edificaciones representan 41.1% del consumo de energía primaria y el 74% del consumo de electricidad.

De acuerdo con este contexto, el siguiente estudio pretende analizar la problemática con respecto al consumo de energía eléctrica en este tipo de edificaciones y las emisiones de carbono

excesivo que se producen ante el uso desmedido de la electricidad, debido a los picos altos de consumo de los edificios comerciales y no comerciales. Se utilizará como muestra un edificio comercial en Dublín-Irlanda, que comprende espacios donde se requiere gran consumo de energía, tales como centros deportivos, centros de entretenimiento, piscinas, sistemas de ventilación, entre otros.

Objetivo

El objetivo principal de este proyecto es poder pronosticar la demanda eléctrica de este edificio a fin de estabilizar la oferta y la demanda para poder proporcionar de manera eficiente a cada una de las edificaciones un porcentaje apropiado de Kw de electricidad, evitando generar picos altos o demandas máximas de energía y así, mayor porcentaje de emisión de carbono al medio ambiente.

Por lo tanto, la predicción de la demanda máxima y las previsiones a corto plazo podrían admitir sistemas de gestión de energía para poder ejecutar estrategias óptimas durante los eventos de picos altos de energía.

Datos

Los datos están centralizados en la lectura de los Kw. consumidos en el edificio comercial ubicado en Dublín, Irlanda. Este edificio de entretenimiento cuenta con diversos espacios que operan con diferentes sistemas eléctricos de diferentes capacidades, establecidos de acuerdo con la afluencia de personas durante los días de semana.

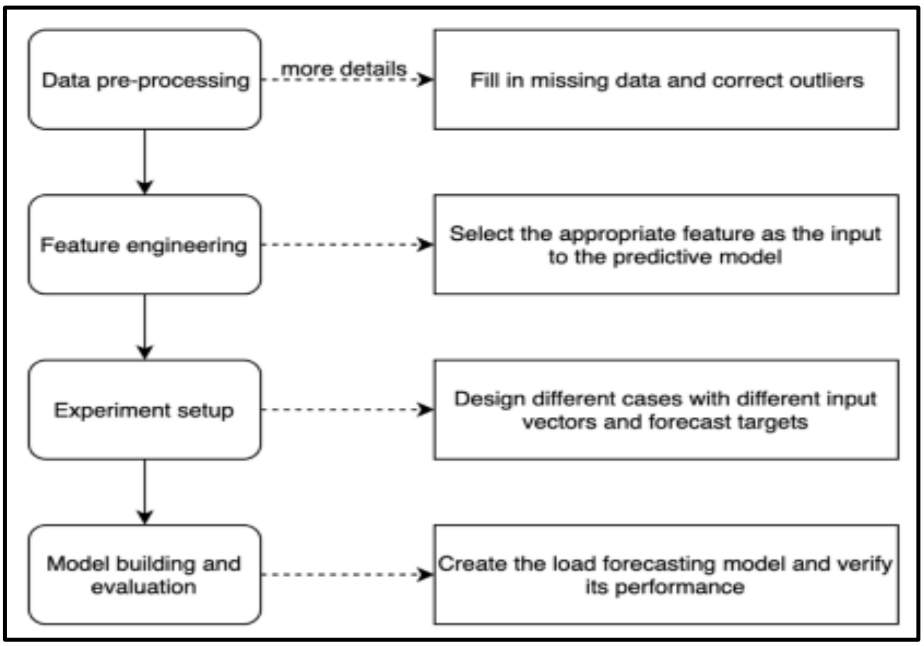
Para efectos de estudios, se han tomado datos históricos de carga eléctrica y datos de factores meteorológicos desde el 1 de enero del 2013 hasta 31 de diciembre del 2018.

Metodología

La novedad del presente trabajo es el uso de los modelos de pronóstico para predecir tanto la demanda de electricidad, pico diario de carga y carga de valle para optimizar dinámicamente el generador local, el almacenamiento térmico y la demanda de un nuevo edificio comercial altamente eficiente y equipado con sistemas de control avanzados, lo que también es una unidad de respuesta a la demanda. Además, el trabajo explora el modelo adaptabilidad a través de múltiples casos: en el pronóstico de carga con una hora de anticipación y el pronóstico de carga pico y valle con un día de anticipación que podría ser utilizado para programar medidas de respuesta a la demanda en respuesta a las señales de la red.

La metodología aplicada es experimental, con diferentes fases como se presentan en la siguiente Figura:

Figura 14: Metodología del proyecto.

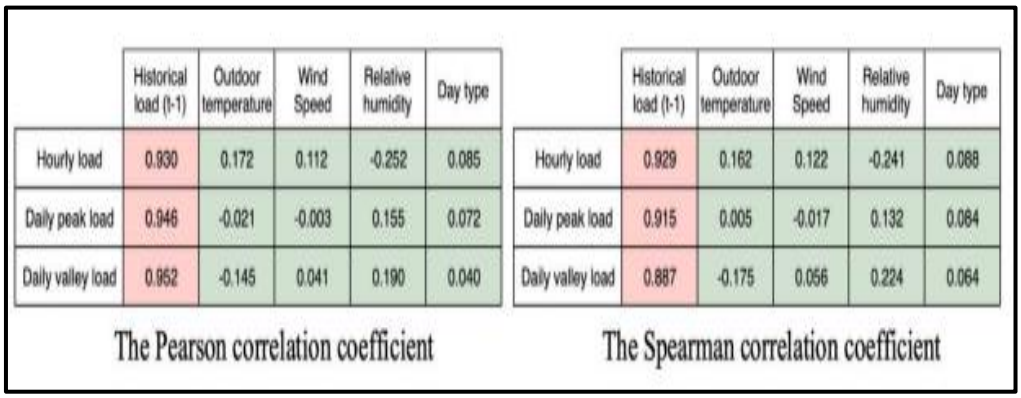


Fuente: Palloneto, Jin & Mangina (2022)

- **Preprocesamiento de datos:** Incluye el procesamiento de datos faltantes, valores atípicos y normalización de datos. Para ello, se realiza el procedimiento de llenado de datos de manera adecuada, a fin de garantizar la integridad de la secuencia de datos de carga de potencia. Luego, la identificación de los valores atípicos para su posterior modificación de los errores aleatorios a fin de no perjudicar el tratamiento de datos y tener resultados desfavorables. Por último, la normalización de datos busca eliminar los efectos del ordenamiento de datos a partir de su magnitud.
- **Selección de características:** Con el fin de tratar los datos de manera óptima, se ve necesario aplicar el análisis de correlación de datos, a partir de los coeficientes de Pearson y Spearman, para poder medir la linealidad de correlación y relación monótona de las variables en estudio. De acuerdo al estudio, hay tres factores que afectan a la predicción de carga de electricidad comercial, tales como el clima, el tipo de día (días laborables y días no laborables) y el factor económico.

Por ejemplo, para el caso del tipo de día, se asigna valor de 1 cuando es de lunes a viernes y 0 cuando se trata de sábado y domingo, a fin de poder hacer el análisis de correlación con las otras variables afectantes en mención, tal como se muestra en la siguiente Figura:

Figura 15: Cuadro de correlación entre variables.



	Historical load (t-1)	Outdoor temperature	Wind Speed	Relative humidity	Day type
Hourly load	0.930	0.172	0.112	-0.252	0.085
Daily peak load	0.946	-0.021	-0.003	0.155	0.072
Daily valley load	0.952	-0.145	0.041	0.190	0.040

The Pearson correlation coefficient

	Historical load (t-1)	Outdoor temperature	Wind Speed	Relative humidity	Day type
Hourly load	0.929	0.162	0.122	-0.241	0.088
Daily peak load	0.915	0.005	-0.017	0.132	0.084
Daily valley load	0.887	-0.175	0.056	0.224	0.064

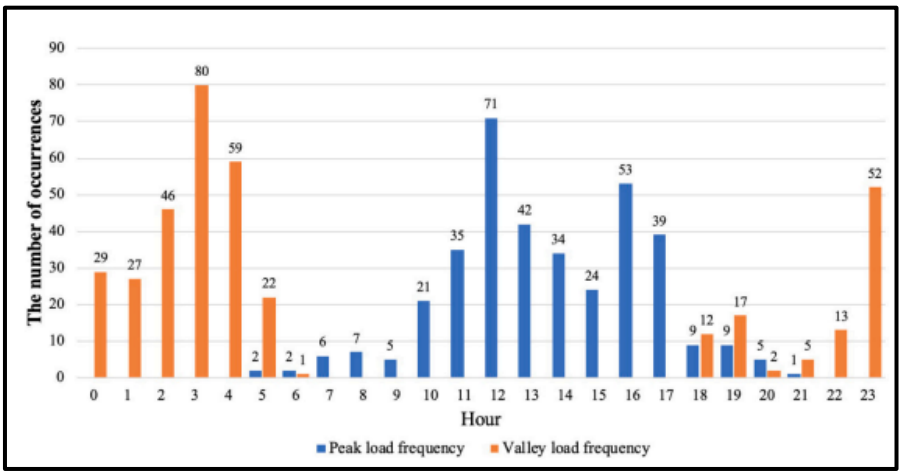
The Spearman correlation coefficient

Fuente: Palloneto, Jin & Mangina (2022)

Nota: Los coeficientes de correlación en color rojo explican que los valores obtenidos superan el umbral y los que no, son rellenados de color verde.

- **Análisis de carga pico y valle diario:** Establecer pronósticos de las demandas máximas y demandas mínimas podría facilitar la implementación de programas de respuesta ante este tipo de demandas. A partir de la gráfica se puede determinar los siguiente:

Figura 16: Gráfico de barras para obtener picos y valles de carga eléctrica.



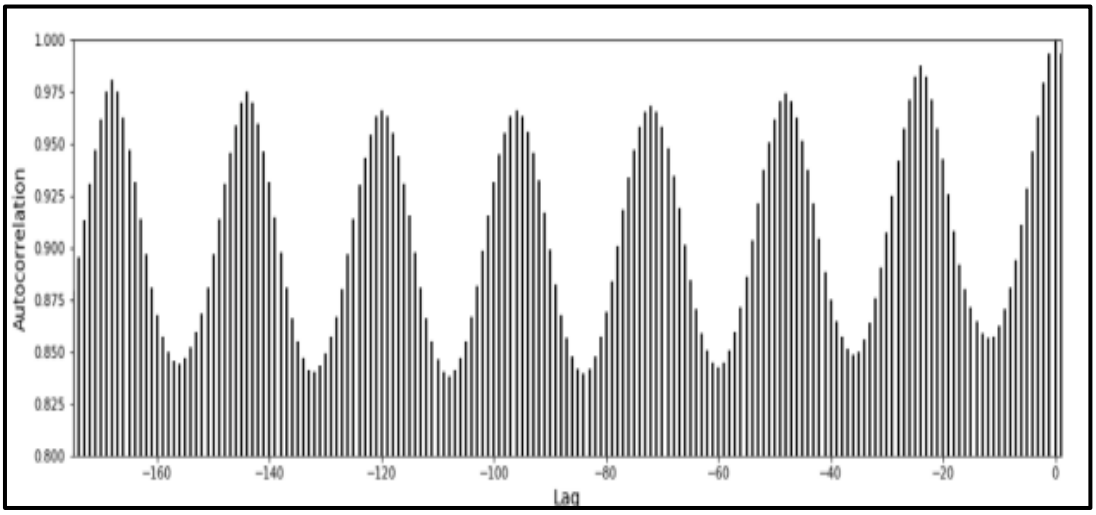
Fuente: Palloneto, Jin & Mangina (2022)

Debido a que los picos y valles no son fijos en tiempos establecidos, la distribución de sus tiempos de ocurrencia está concentrados en una zona del gráfico, por lo cual estudio del gráfico se facilita, obteniendo datos referentes con respecto a la hora en que se desarrollan estos eventos, además de los días (laborables o no laborables) en lo que se suscitan, determinando en primera instancia dónde se va a centralizar los esfuerzos con respecto a la dosificación de energía mediante los sistemas que se pretenden implementar.

- **Configuración del experimento:** Para este proyecto se van a utilizar dos modelos (LTSM y SVM) a fin de efectuar una comparación entre el rendimiento de ambas propuestas. Para ello, el caso se dividió en dos partes: pronóstico de carga con una hora de anticipación y la otra para el pronóstico de carga con un día de anticipación. El primero se lleva a cabo utilizando la historia datos de carga horaria de 2013 a 2018, con un total de 52.584 datos. El último se basa en la carga diaria máxima y mínima desde 2013 hasta 2018, con 2191 datos. Todos los datos se vuelven a muestrear a partir del conjunto de datos original.

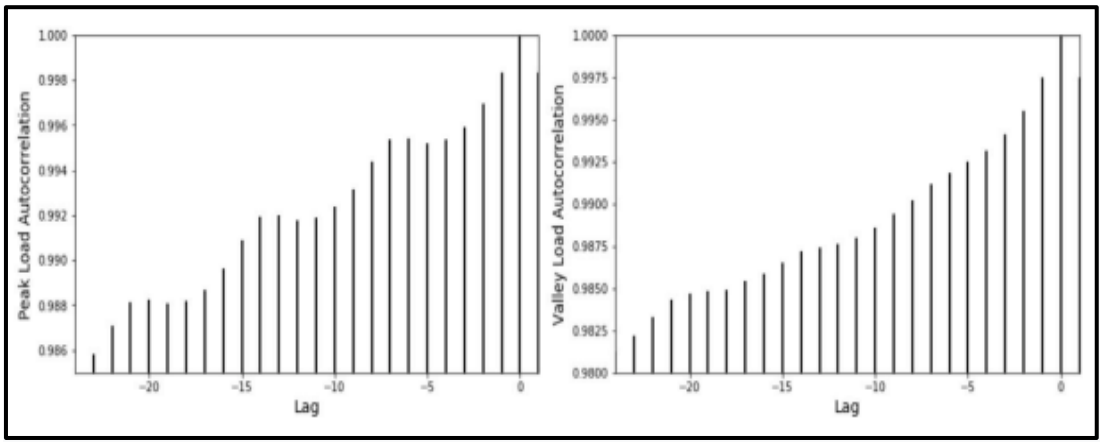
  En las siguientes Figuras se muestra el grado de correlación diferenciado a partir del tipo de pronóstico aplicado, ya sea con una hora o día de anticipación.

Figura 17: Gráficas de grado de correlación de datos con una hora de anticipación.



Fuente: Palloneto, Jin & Mangina (2022)

Figura 18: Gráficas de grado de correlación de datos con un día de anticipación de picos altos y valles.



Fuente: Palloneto, Jin & Mangina (2022)

Los coeficientes de autocorrelación entre las secuencias de datos de carga máxima diaria son periódicos, mientras que la autocorrelación entre las secuencias de datos de carga valle diaria no es significativa. En general, la autocorrelación entre el pico diario y la carga del valle disminuye a medida que el intervalo de tiempo aumenta.

Con base en las conclusiones anteriores, este trabajo crea 4 casos con diferentes características de entrada:

- La entrada del Caso 1 es un vector de 8 dimensiones que usa la carga eléctrica con hora h-1, h-24, h-48, h-72, h-96, h-120, h-144, h-168 para pronosticar la carga eléctrica de la hora h.
- La entrada del Caso 2 es un vector de 168 dimensiones que usa la carga eléctrica de la hora h-1 a la h-168, la carga horaria de la semana anterior, para pronosticar la carga eléctrica de la hora h.
- La entrada del Caso 3 es un vector tridimensional que usa la carga eléctrica del pico y valle del día d-1, d-2, d-7 para pronosticar la carga eléctrica del pico y valle del día d.
- La entrada del Caso 4 es un vector de 7 dimensiones que usa la carga eléctrica del pico y valle del día d-1 a d-7 y la carga eléctrica del pico o valle diaria de la semana anterior, para pronosticar la carga eléctrica del pico y valle del día d.

Técnica de Machine Learning

- **Configuración del modelo:** Como se mencionó anteriormente, los pronósticos están basados en dos modelos:
    - Modelo basado en LSTM:

        Consta de dos LSTM, una capa contiene 100 unidades y la segunda capa 50 unidades. Se crea una capa de abandono de 20% de tasa, la cual se agrega entre las dos primeras capas. La función de activación es la regla y el MSE se usa como función de pérdida y AMDA como optimizador del modelo.
    - Modelo basado en SVM:

        La función lineal se utiliza como principal recurso del modelo de regresión vectorial y la tolerancia para el criterio de parada se establece en 0.001, donde el parámetro de penalización C del término de error es 1.
- **Validación cruzada:** Para efectos de evitar los sobre ajustes, se utiliza la técnica de cross value de series temporales, con el objetivo de que todo el conjunto de datos recabados entre el 2013 y 2017 se utiliza como un grupo de datos de entrenamiento para dar garantía de que se cumpla con un modelo de entrenamiento de datos suficiente para realizar los pronósticos de la demanda eléctrica. Para este tipo de validación cruzada, se realiza el entrenamiento con datos anteriores a la secuencia del conjunto de prueba.

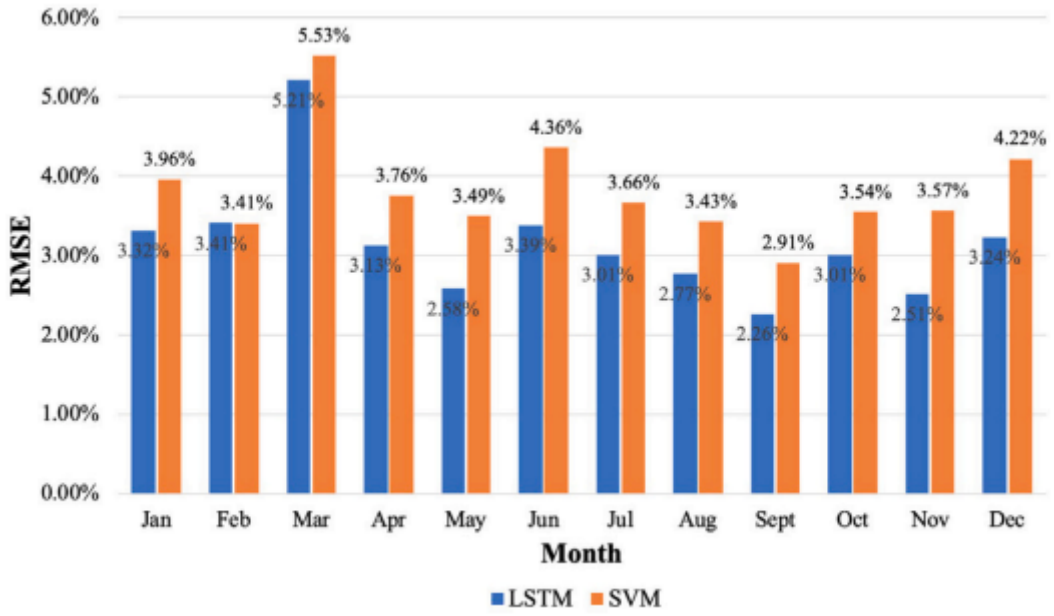
Resultados

Los resultados presentados a continuación han sido validados y corroborados con datos extraídos del sistema eléctrico del edificio. Por este motivo, las métricas y validaciones obtenidas se desarrollan en referencia a data real del edificio, por lo cual, se presenta el performance de los modelos aplicados a los datos obtenidos.

Performance de los modelos de pronóstico de energía eléctrica con una hora de anticipación.

El análisis de rendimiento de los 2 modelos propuestos con el detalle de pronóstico con una hora de anticipación, para el caso 1, con respecto al RMSE mensual del 2018 se presenta en el siguiente gráfico:

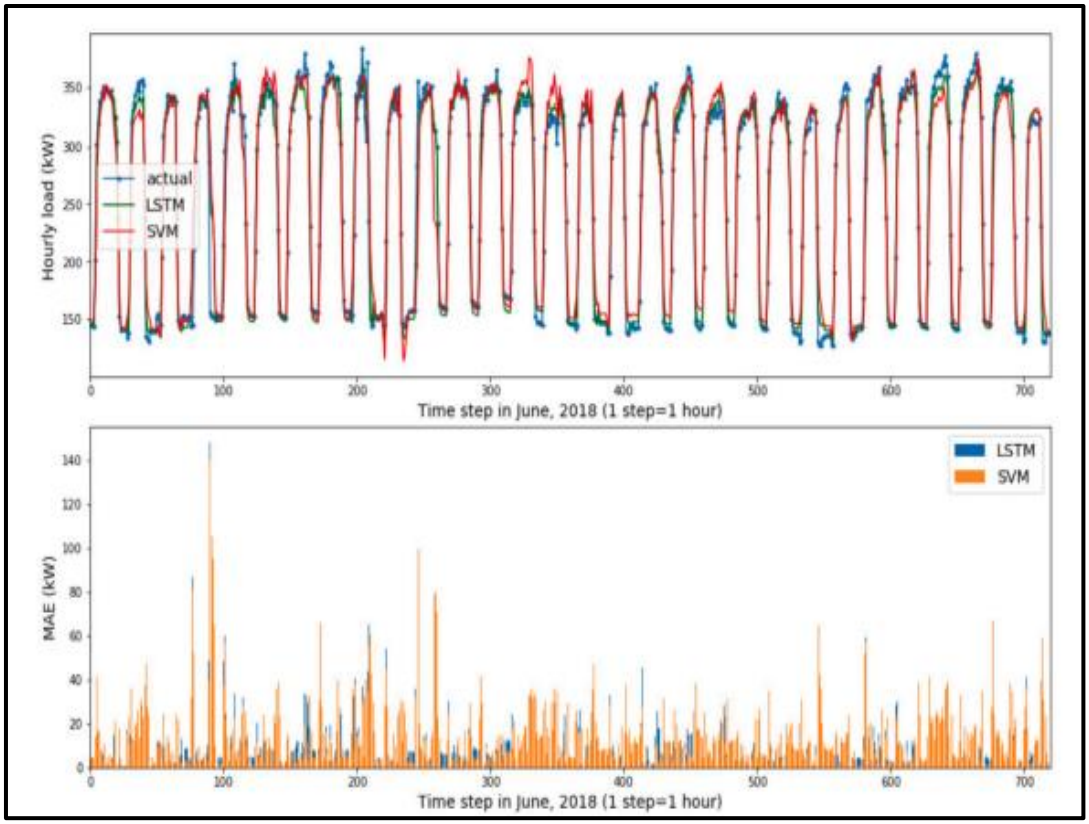
Figura 19: El RMSE mensual del año 2018 para el pronóstico de carga eléctrica por hora en el Caso 1.



Fuente: Palloneto, Jin & Mangina (2022)

Las tendencias mensuales de la métrica RMSE de los dos modelos en cierto grado son similares y la variación de cambio es equilibrada entre ambos modelos. Para análisis del comportamiento de la predicción con ambos modelos para la carga eléctrica real y la carga eléctrica predicha por hora en junio de dicho año, se muestra la siguiente gráfica:

Figura 20: Proyección de datos con modelos SVM y LSTM.



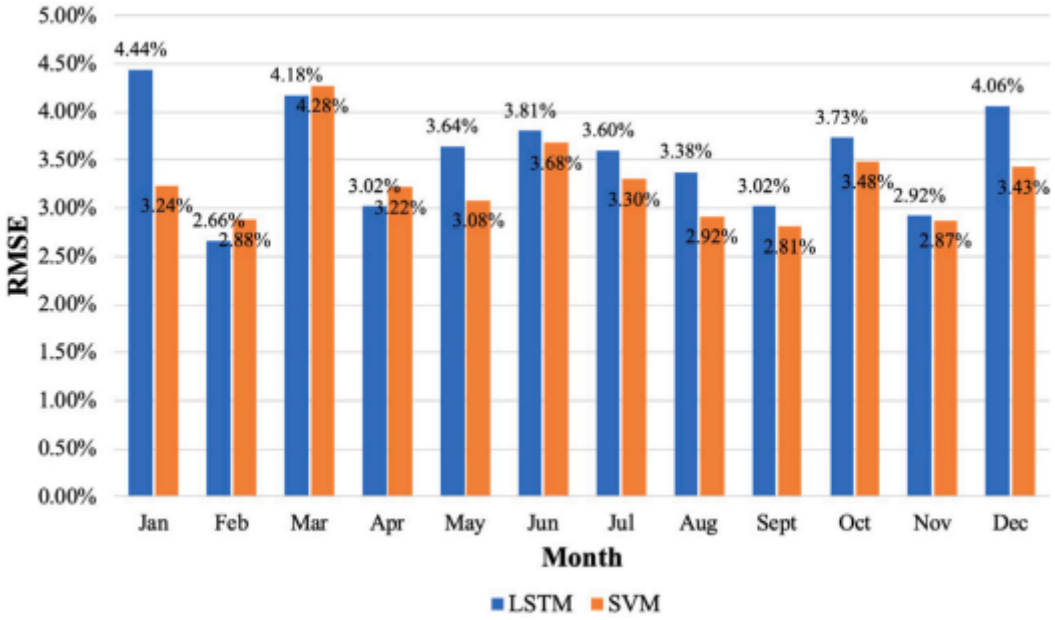
Fuente: Palloneto, Jin & Mangina (2022)

En el primer gráfico (Figura 20) de la Figura mostrada anteriormente, la línea azul describe la potencia eléctrica real, la línea verde muestra el pronóstico de carga eléctrica basado en el modelo LTSM y la línea roja denota los datos predichos por el modelo SVM.

Para efectos comparativos, se desarrolla el siguiente gráfico con los valores absolutos de la carga eléctrica real y pronosticada, donde la barra azul representa el modelo LTSM y la barra naranja representa el modelo SVM.

Para estudio del comportamiento en el Caso 2, se realiza el mismo procedimiento con diferente toma de datos, obteniendo la siguiente gráfica:

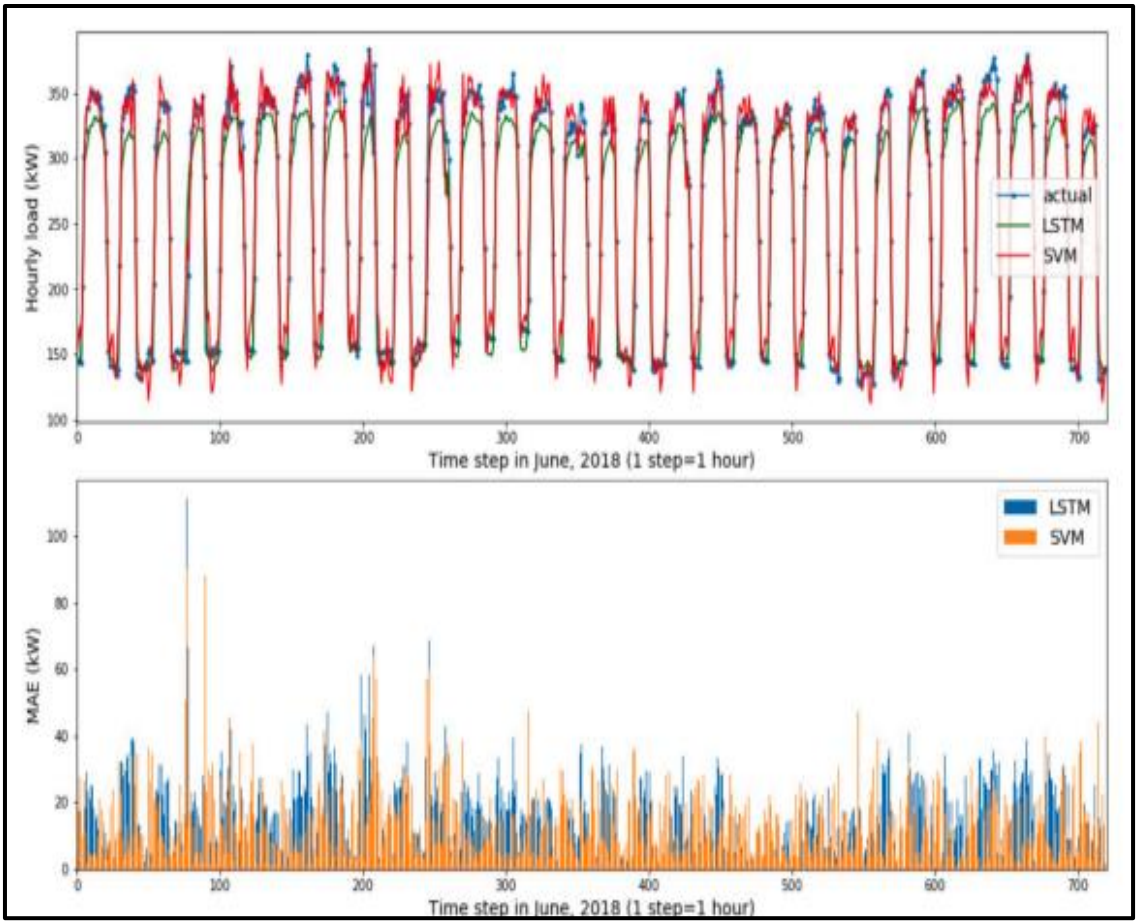
Figura 21: Gráfico de barras de datos con modelos LSTM y SVM Caso 2.



Fuente: Palloneto, Jin & Mangina (2022)

Con este gráfico (Figura 21) se puede concluir en un comportamiento distinto mes a mes con respecto al Caso 1, donde el RMSE mensual del modelo SVM es más bajo que el modelo LTSM, por lo que se podría determinar que el modelo SVM se ajusta mejor a la gráfica de la carga eléctrica real, tal y como se muestra en este gráfico:

Figura 22: Proyección de datos con modelos SVM y LSTM Caso 2.



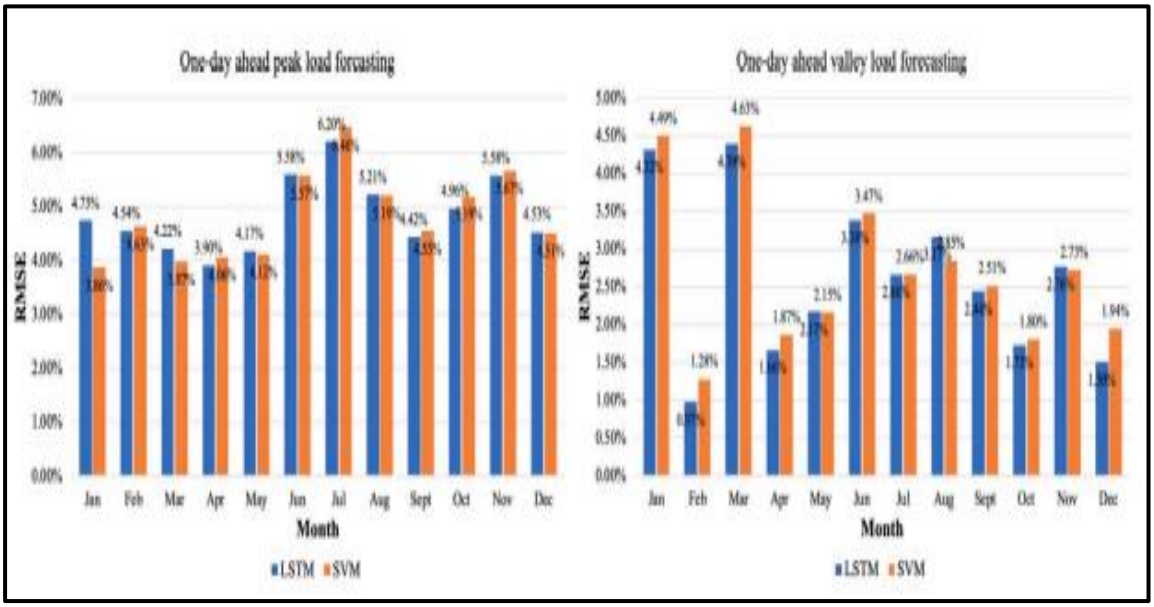
Fuente: Palloneto, Jin & Mangina (2022)

En resumen, teniendo en cuenta tanto la selección de funciones como la carga eléctrica horaria, se puede concluir que el rendimiento del modelo basado en LSTM es mejor que el del modelo basado en SVM.

Desempeño de los modelos de predicción con respecto a la demanda máxima y demanda mínima

De la misma manera que el análisis anterior, es primordial evaluar el RMSE para determinar cuál es el mejor modelo para pronosticar demanda máximas y mínimas que permitan hacer los ajustes necesarios. Por ese motivo, se presenta la siguiente gráfica:

Figura 23: El RMSE mensual de 2018 para el pronóstico diario de carga pico y valle en el Caso 3.



Fuente: Palloneto, Jin & Mangina (2022)

Además, es de vital importancia mantener el análisis de error de los modelos, a fin de encontrar el modelo que mejor se adapte a la data histórica (data de entrenamiento):

Figura 24: Tabla de comparación de casos y sus respectivas métricas de error.

Table 2
Comparison of models performances.

	Forecast	Model	MAE (kW)	RMSE (%)	MAPE (%)	Runtime
Case 1	1-h ahead load	LSTM	9.258	3.15	4.05	497.70
		SVM	11.940	3.82	5.30	4.21
Case 2	1-h ahead load	LSTM	12.642	3.54	5.54	706.96
		SVM	11.539	3.26	5.37	23.94
Case 3	1-day ahead peak load	LSTM	9.957	4.84	3.04	67.93
		SVM	9.873	4.81	3.01	0.05
	1-day ahead valley load	LSTM	3.603	2.60	2.57	66.96
		SVM	3.758	2.70	2.71	0.05
Case 4	1-day ahead peak load	LSTM	9.800	4.73	3.00	66.82
		SVM	9.709	4.70	2.96	0.06
	1-day ahead valley load	LSTM	3.683	2.63	2.63	69.60
		SVM	3.665	2.66	2.63	0.05

Fuente: Palloneto, Jin & Mangina (2022)

A partir de este cuadro, se puede determinar que para la determinación de pronósticos de picos de carga eléctrica en el caso 3 y el caso 4 el mejor modelo es el SVM. Por otro lado,

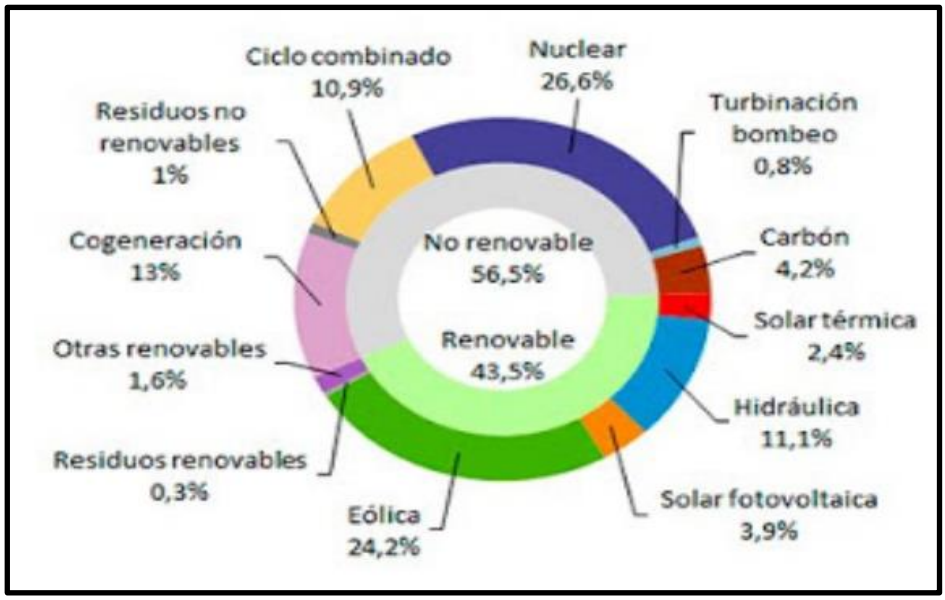
para los valles o demanda mínimas, el modelo LSTM tiene menor % de RMSE, por lo que se convertiría en un mejor modelo de pronóstico que el SVM.

Del Campo, Rubén (2020). Técnicas de Machine Learning aplicadas a la predicción de los desvíos del Mercado Eléctrico.

Problema

Del Campo, Rubén (2020) en su Trabajo de Fin de Grado analiza la tendencia creciente de las empresas pertenecientes al sistema eléctrico español de predecir con exactitud la demanda de energía eléctrica y de esta manera evitar las sanciones gubernamentales ante una mala gestión en la producción de energía eléctrica poco beneficiosa para el usuario. En el trabajo también se señala como realidad problemática el auge de energías renovables, las cuales, aunque actualmente se acerquen al 40% de oferta en el mercado español, también son fuentes energéticas muy dependientes de factores meteorológicos, los cuales tienen una probabilidad de predicción muy baja.

Figura 25: Desglose de las tecnologías de generación eléctrica en España en marzo del 2019.



Fuente: REE (2019)

Objetivo

Debido a esa problemática, Del Campo, Rubén (2020) planteó el objetivo de aplicar técnicas de Machine Learning, como modelos de clasificación en los desvíos de demanda

eléctrica y modelos de regresión para el comportamiento del precio de la energía eléctrica, para identificar el comportamiento del consumo eléctrico español y de este modo entender sus variaciones.

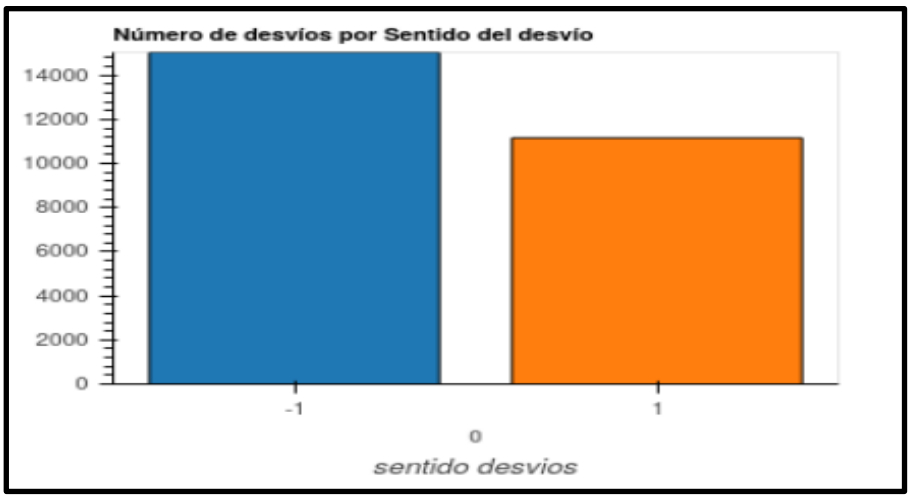
Datos

Las fuentes de datos del presente trabajo son varias, pero como fuente principal tenemos los datos obtenidos de ESIOS (Sistema de Información del operador del Sistema), perteneciente a la Red Eléctrica de España, de esta fuente cualquier usuario puede recabar información en tiempo real acerca de la producción de energía eléctrica y la demanda, los cuales son utilizados en este estudio. También, se usó como fuente secundaria una API REST, obtenida de Nager.Date, con la cual se obtiene información sobre una variable en un rango de tiempo específico. Otro método de obtención de datos fue el INE (Instituto Nacional de Estadística), el cual nos proporcionó la cantidad de personas que viven en una localidad específica.

Metodología.

El modelo de investigación es experimental, el autor en primer lugar realizó un análisis exploratorio y estadístico, como se busca predecir el sentido de desvíos en el consumo eléctrico español, se empleó la siguiente gráfica.

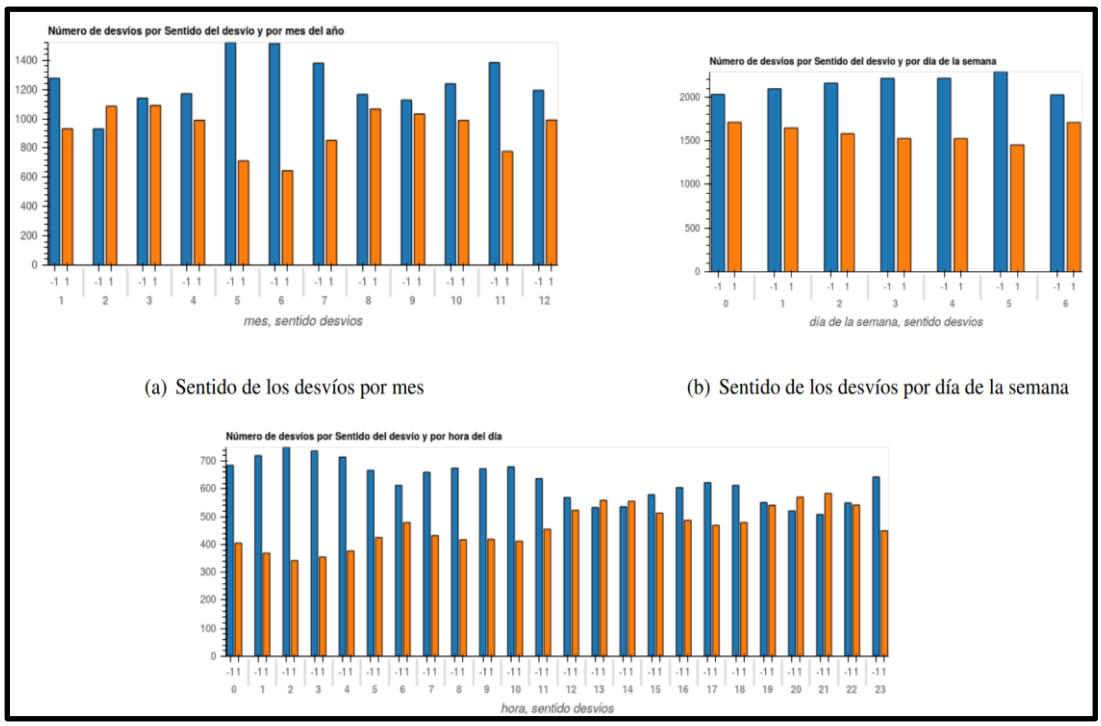
Figura 26: Distribución del sentido de los desvíos.



Fuente: Del Campo, Rubén (2020)

Posteriormente, se aplicó agrupaciones de datos utilizando diferentes intervalos temporales.

Figura 27: Distribución de los desvíos (1 para desvíos a subir y -1 para desvíos a bajar) agrupados en diferentes resoluciones temporales.



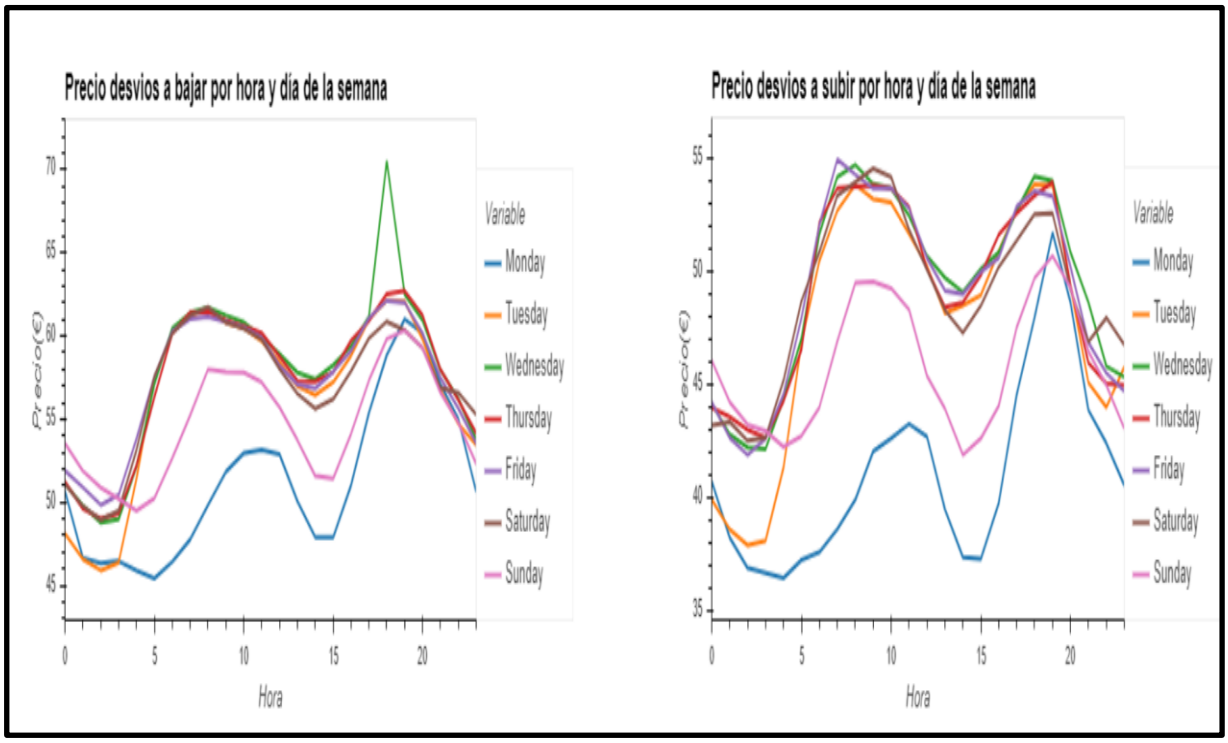
(a) Sentido de los desvíos por mes

(b) Sentido de los desvíos por día de la semana

Fuente: Del Campo, Rubén (2020).

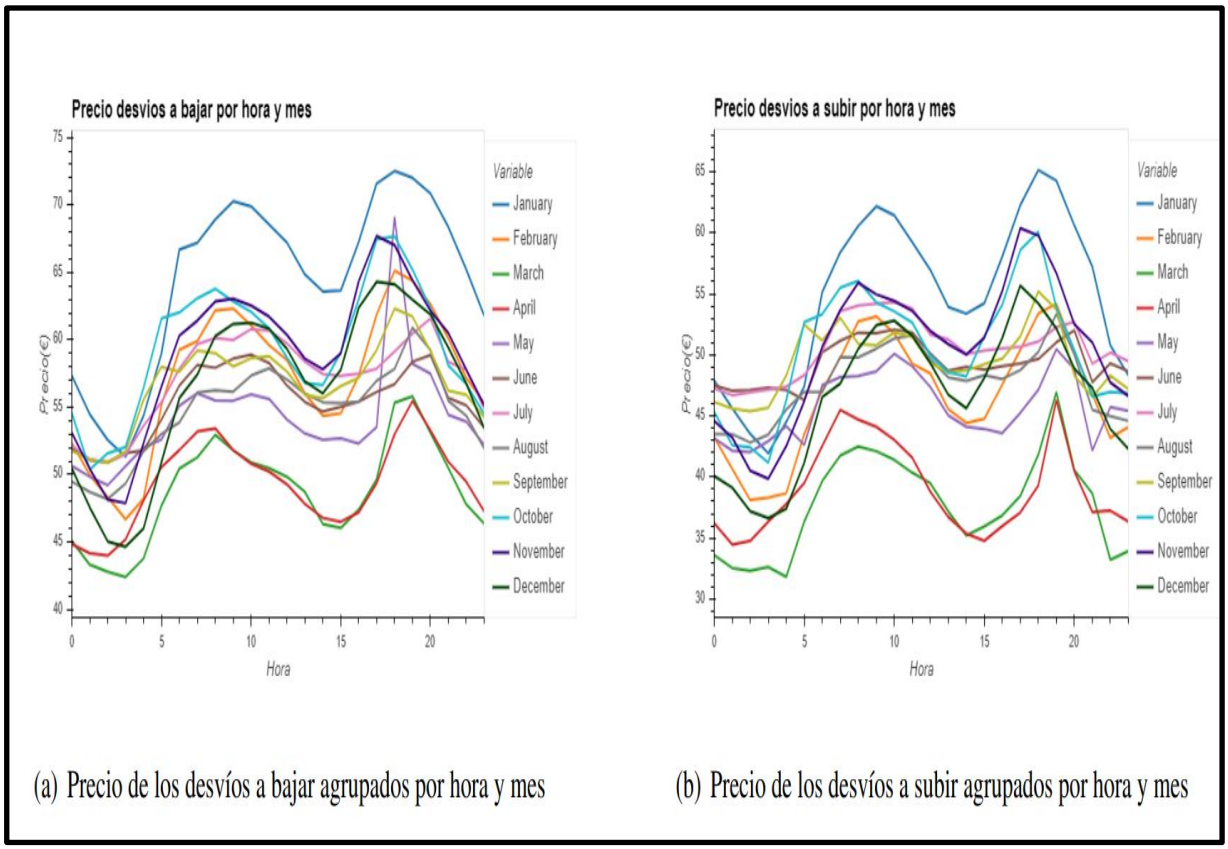
En relación con los precios que ocasionan estos desvíos se agruparon estos datos con un horario y fecha específico, intervalo de tiempos, todo con el objetivo de conocer la influencia de la variable temporal en los desvíos y precios de la energía eléctrica.

Figura 28: Precio de los desvíos a subir y bajar agrupados por horas y días de la semana.



Fuente: Del Campo, Rubén (2020).

Figura 29: Precio de los desvíos a subir y a bajar agrupados por hora y mes.



(a) Precio de los desvíos a bajar agrupados por hora y mes

(b) Precio de los desvíos a subir agrupados por hora y mes

Fuente: Del Campo, Rubén (2020).

El autor recabó 49 variables de su fuente principal, las cuales se dividieron en **"variables de tiempo real"**, son las que se miden y se proporcionan para la última hora acontecida, y **"variables a futuro"**, son variables que se generan a partir de predicciones por REE. Debido a que las variables temporales usadas en el estudio presentan un comportamiento cíclico, se empleó las siguientes fórmulas:

- $Variable\ Seno = \text{sen}\frac{2*\pi*variable}{n\ posibles\ valores}$
- $Variable\ Coseno = \cos\frac{2*\pi*variable}{n\ posibles\ valores}$

Técnicas de Machine Learning

El autor decidió implementar el lenguaje Python para su elaboración, aplicando Modelos de Predicción como Clasificación y Regresión Lineal.

En el caso de Clasificación el autor decidió basarse en los siguientes conceptos:

- True Positive (TP) cuando se ha predicho A para un elemento de clase A.
- False Positive (FP) cuando se ha predicho A para un elemento de clase B.
- False Negative (FN) cuando se ha predicho B para un elemento de clase A.
- True Negative (TN) cuando se ha predicho B para un elemento de clase B.

Una vez el autor sentó las bases, se implementó las siguientes métricas:

- **Precisión**, consiste en efectuar la ratio de TP / (TP + FP). Intuitivamente podemos considerarla como la capacidad del clasificador de no etiquetar como A un elemento de clase B.
- **Recall**, se calcula con el ratio TP / (TP + FN). Podemos decir que intuitivamente supone la capacidad del clasificador de etiquetar correctamente todos los elementos A.
- **F1,** es la media armónica de precisión y recall. De este modo, podemos evaluar que nuestro modelo se comporte bien en ambas métricas simultáneamente.

Los valores de las tres métricas presentadas oscilan entre 0 y 1, donde 1 es el mejor resultado y 0 el peor.

De acuerdo con las aplicaciones de Regresión Lineal, el autor vio por conveniente aplicar dos tipos de métricas diferentes a las de clasificación con el objetivo de mejorar el accuracy de la predicción en el modelo. Las métricas empleadas son:

- MAE (Mean Absolute Error) que se calcula como: 1 n Pn i=1(yi − xi), donde los yi son los valores predichos por nuestro modelo, los xi los valores de nuestro conjunto de test con los que comparamos la calidad de nuestras predicciones y n el número de patrones que se quieren testear. Intuitivamente podemos decir que nos proporciona cuánto error tienen de media nuestras predicciones.
- MSE (Mean Squared Error), calculado como 1 n Pn i=1(yi − xi) 2. Se puede decir que nos indica cuánto error medio tienen nuestras predicciones acentuando más los errores más grandes.

Resultados

Los resultados obtenidos en referencia a los desvíos por el autor son que el modelo de predicción desarrollado dio como resultado mejores predicciones con mucha diferencia a las anteriores.

Tabla 1: Tabla de resultados de predicción del sentido de los desvíos aplicando modelos de clasificación para multi-output.

Model	Precision	Recall	F1
LR con MultioutputClassifier	0.728	0.631	0.645
SVM con MultioutputClassifier	0.731	0.633	0.648
RF con MultioutputClassifier	0.712	0.627	0.635
RF	0.713	0.643	0.645
MLP con MultioutputClassifier	0.704	0.604	0.627
MLP	0.707	0.595	0.616
Persistencia dia-hora anterior	0.604	0.605	0.606
Persistencia semana-día-hora anterior	0.546	0.546	0.548
Persistencia siempre -1	0.287	0.364	0.5

Fuente: Del Campo, Rubén (2020).

En relación con los resultados obtenidos por el autor en relación con los precios de los desvíos, se obtuvo que las predicciones obtenidas tienen mejoras en el MAE y en el MSE.

Tabla 2: Tabla de resultados de predicción del precio de los desvíos aplicando modelos de regresión para multi-output.

Model	MAE	MSE
SVM con MultioutputRegressor	5.321	64.553
RF con MultioutputRegressor	6.202	84.008
RF	6.344	95.110
MLP con MultioutputRegressor	6.725	98.065
MLP	6.894	98.645
Persistencia con día-hora anterior	8.558	155.429
Persistencia con semana-día-hora anterior	10.792	230.917

Fuente: Del Campo, Rubén (2020)

Ejaz Ui, Haq & Xue, Lyu & Youwei, Jia & Mengyuan, Hua & Fiaz, Ahmad (2020). Forecasting household electric appliances consumption and peak demand based on hybrid machine learning approach.

Problema

Actualmente las redes inteligentes se encuentran en un escenario prometedor, mediante las cuales en el presente artículo se busca demostrar su aplicación en el consumo de energía en los hogares, administrando y controlando la energía de manera más confiable, económica e inteligente. Se sabe que la demanda de energía eléctrica por parte de los hogares representa entre el 30% y 40% de la demanda mundial, uno de los factores de esta participación y su progresivo crecimiento es el constante crecimiento de la población mundial.

La gestión correcta de la energía eléctrica puede ayudar a una mejor performance entre demanda y producción de energía eléctrica, por lo tanto, la predicción correcta y asertiva de la demanda es muy importante para todos los actores a lo largo de su ciclo de vida, como los consumidores finales, administradores de energía, recurso de generación, empresas de servicio de electricidad, entre otros.

Objetivo

El objetivo del presente artículo es pronosticar o predecir la demanda máxima de energía en los hogares por el uso de electrodomésticos, aplicando Machine Learning. De esta manera se podrá identificar a los clientes más idóneos para un programa de respuesta de demanda máxima.

Datos

El artículo indica que los datos obtenidos provienen de contadores inteligentes, este dispositivo permite conocer los datos de consumo de electricidad a los proveedores y también el costo del consumo de energía. Los datos recabados son los tiempos de consumo, ya sea en segundos, minutos u horas, también se obtiene la carga de energía de cada consumo.

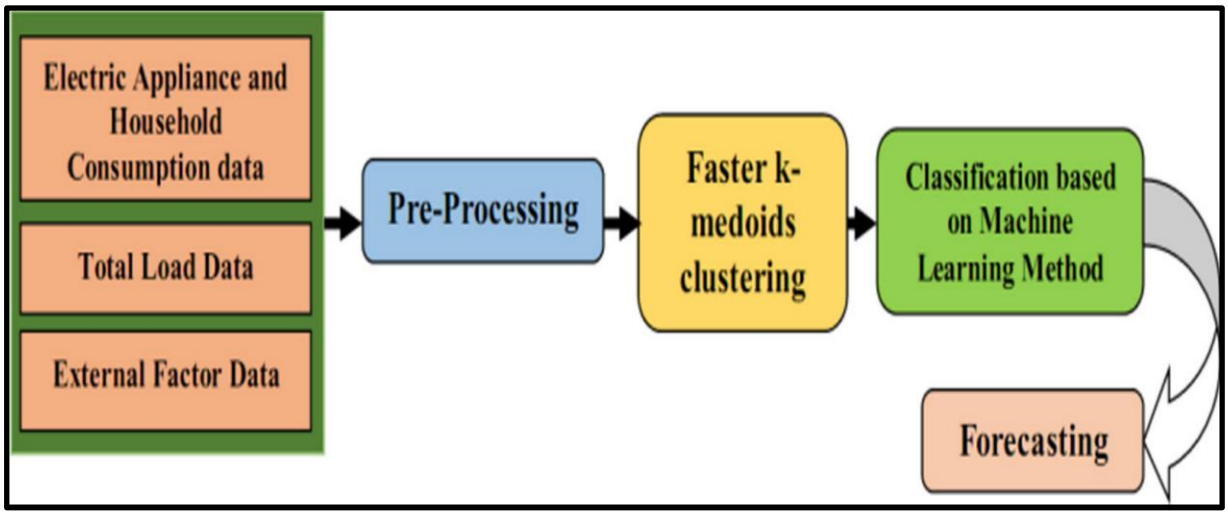
Metodología

El artículo implementa un método de aprendizaje automático híbrido, aplicando técnicas de clasificación y agrupación para la predicción de la demanda máxima de los electrodomésticos en los hogares.

Se utilizó un agrupamiento de k-medoides más rápido, buscando tres grupos diferentes del conjunto de datos experimentales, esto abarcando una técnica de clasificación. A continuación, se adjuntará el modelo de flujo de trabajo establecido en el artículo de estudio.

Los datos de consumo de electricidad de un año desde junio de 2017 hasta mayo de 2018. El conjunto de datos está compuesto por un total de 550 hogares que tienen 30 minutos de datos de consumo eléctrico que se obtienen de los medidores inteligentes . El conjunto de datos consiste en una identificación única de cada hogar, atributos de género y edad de los clientes y 30 minutos de datos de consumo en kWh.

Figura 30: Flujo de trabajo del modelo propuesto.



Fuente: Ejaz Ui, Haq & Xue, Lyu & Youwei, Jia & Mengyuan, Hua & Fiaz, Ahmad (2020)

Técnicas de Machine Learning.

- Preparación de datos:

Primero se seleccionaron los datos de consumo eléctrico de 10 electrodomésticos. Las medidas de consumo eléctrico se toman en kilovatios (kW) con un intervalo de tiempo de 3 min entre dos muestras.

Posteriormente, se pronostica la demanda máxima de los clientes. Utilizando un agrupamiento de k-medoids más rápido para procesar el conjunto de datos para pronosticar la demanda máxima.

- Procesamiento y selección de las mejores características:

La detección de información perdida, la limpieza de datos, la eliminación de ruido, el filtrado de datos y otros métodos se aplicaron como pasos de pre procesamiento. Se tomaron las características más básicas de los electrodomésticos esenciales, filtraron los datos diarios, semanales, mensuales y anuales de consumo de electrodomésticos con la resolución de tiempo 3 min.

- Agrupamiento de K-medoides más rápidos.

Se utilizan algoritmos de agrupamientos que funcionan con la base de agrupamiento K-mean, calculando la matriz de distancia solo una vez y usa la matriz de distancia calculada en cada paso iterativo para encontrar nuevos medoides. Necesita la distancia entre dos objetos sólo por una vez. El algoritmo Faster k-medoids funciona en los siguientes tres pasos:

- En el primer paso se calcula la distancia entre dos objetos. El medoide inicial se predice seleccionando n objetos que tienen menos distancia
- Los medoides se actualizan.
- Los objetos se asignan a su medoid correspondiente y dan como resultado la formación de un grupo.

- Clasificación basada en algoritmos de aprendizaje automáticos:

Se emplea el Support Vector Machine (SVM), como algoritmo de aprendizaje supervisado, aplicando Hyperplane, un parámetro de mejor conjunto en SVM que clasifica un conjunto de datos en varias clases. El hiperplano se selecciona como una línea recta si solo hay dos clases que necesitan clasificación. La distancia desde el hiperplano es la idea de trabajo clave del clasificador SVM. Cuanto mayor sea la distancia de un punto al hiperplano, el clasificador SVM puede clasificar mejor un punto específico en su clase correspondiente.

También se empleó Red Neuronal Artificial (RNA), esta herramienta es una serie de neuronas interconectadas que procesan datos como lo haría el sistema de neuronas de un ser humano desde la entrada hasta la salida. RNA consiste en la capa de entrada, capa oculta y capa de salida.

Resultados.

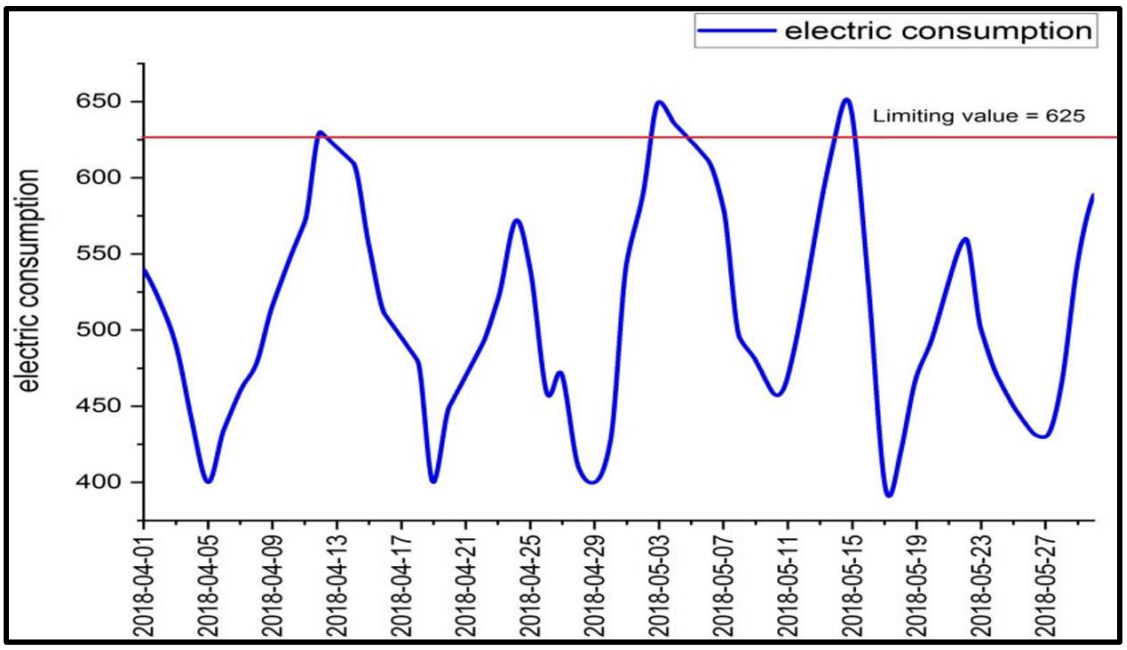
Se analizó los datos de 10 electrodomésticos con mayor relevancia en un hogar, luego se predijo el consumo de energía de estos y el consumo singular de cada uno. Las siguientes tablas y Figuras son el resultado de la predicción de la demanda máxima de energía eléctrica de los electrodomésticos de los hogares.

Tabla 3: Descripción del valor medio de las siete características por aparato eléctrico.

Electric appliance	Daily	Weekly	Monthly	Spring	Summer	Fall	Winter	Total
Fridge	0.0832	0.087654	0.0976543	1.43212	1.50678	1.031245	0.754327	2.35276
Oven	0.010632	0.005127	0.009823	0.034176	0.030987	0.025209	0.45825	0.198041
Hair Dryer	0.009365	0.009801	0.013863	0.048163	0.049512	0.048740	0.0490786	0.179241
Room1 lighting	0.116003	0.378002	0.384482	0.430086	0.445012	0.35983	0.35103	1.606804
Air conditioner 1	0.310323	0.814326	0.957234	1.342156	1.402341	0.875198	0.715678	5.098721
Laptop	0.00632	0.008867	0.009899	0.017243	0.020987	0.024207	0.028018	0.186509
Water heater	0.12897	0.159082	0.189563	0.401253	0.370976	0.403487	0.417409	2.26793
Television	0.003345	0.005083	0.008863	0.049972	0.056542	0.074356	0.065628	0.145698
Iron	0.003412	0.0044562	0.008053	0.049765	0.050432	0.046543	0.049712	0.187324
Clothes dryer	0.0682	0.088763	0.099753	1.20982	1.39802	1.198623	1.443207	2.10982

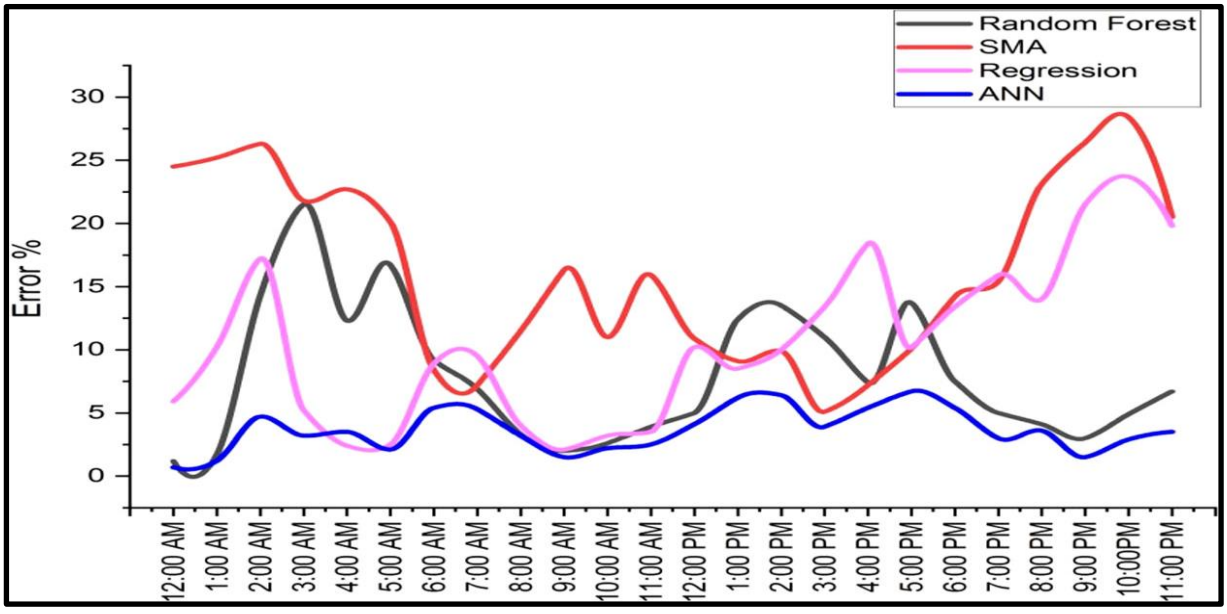
Fuente: Ejaz Ui, Haq & Xue, Lyu & Youwei, Jia & Mengyuan, Hua & Fiaz, Ahmad (2020).

Figura 31: Demanda máxima de energía eléctrica de los electrodomésticos.



Fuente: Ejaz Ui, Haq & Xue, Lyu & Youwei, Jia & Mengyuan, Hua & Fiaz, Ahmad (2020)

Figura 32: % de error de predicción de diferentes modelos predictivos.



Fuente: Ejaz Ui, Haq & Xue, Lyu & Youwei, Jia & Mengyuan, Hua & Fiaz, Ahmad (2020)

Finalmente, se obtuvo que el modelo de predicción propuesto por los autores del artículo logró un 99,2% de precisión en la predicción del consumo de energía de los electrodomésticos y un MAPE muy reducido.

Namir, Labriji y Ben (2021). Decision Support Tool for Dynamic Inventory Management using Machine Learning, Time Series and Combinatorial Optimization.

Problema

La problemática de este estudio se centra en los stocks de inventarios que son valores operativos que se deben administrar, sin embargo, existen costos asociados a su existencia. Cuando los suministros son menos importantes, la empresa se ve amenazada por un desabastecimiento que puede interrumpir el proceso de producción. Esta interrupción crea costos adicionales y puede dañar la imagen de marca de la empresa. Por otro lado, cuando los suministros son demasiado importantes, constituyen activos fijos que inflan el precio de costo y perturban el equilibrio del flujo de caja. Esto es lo que hace que la gestión de inventarios sea un problema desafiante en la gestión de la cadena de suministro. Asimismo, Los ciclos de demanda de los clientes son diferentes y la tasa de demanda de cada cliente varía, por lo que es difícil predecir la cantidad exacta de inventario que necesitan las empresas.

Objetivo

Desarrollar un modelo que combine series de tiempo, algoritmo de machine learning y optimización combinatoria para identificar las oportunidades de comprar stock a un costo menor y vender una parte del stock no utilizado para generar más utilidades para la organización. El propósito de la gestión de inventarios es satisfacer la demanda, generar más ganancias y también garantizar el buen funcionamiento de la empresa ya que es una función crucial para la competitividad.

Datos

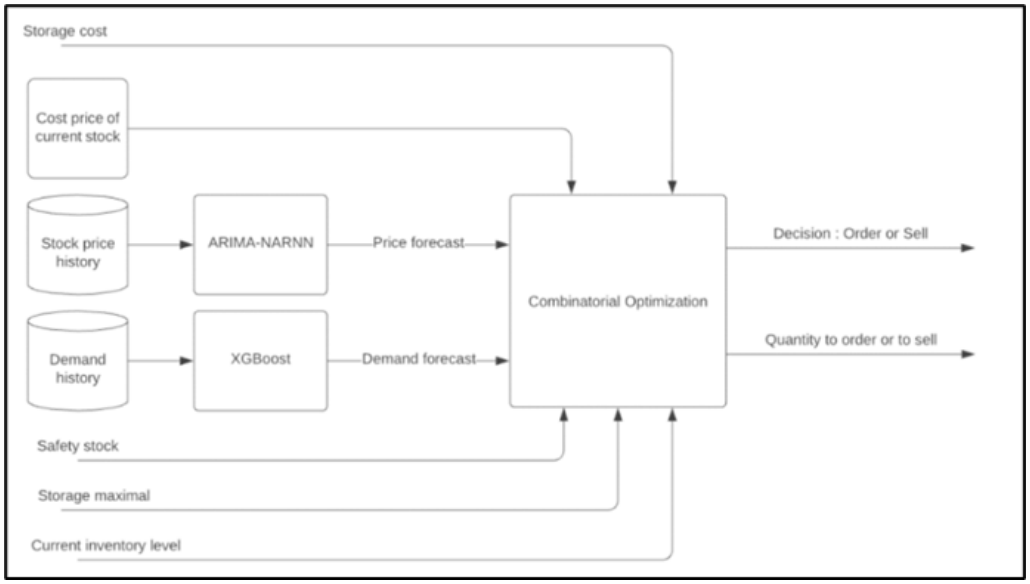
El estudio utilizó datos de stocks de dos años de una empresa privada que produce y distribuye aceites para motores de automóviles. La materia prima utilizada para su elaboración es el aceite base. El historial de precios del aceite base se usó en ARIMA-NARNN y el historial de demanda se usó para XGboost. Python se utilizó para desarrollar los modelos.

Metodología

La metodología usada es experimental combinando todos los submodelos e introduciendo las diferentes variables en el modelo de optimización combinatoria

Para realizar una mejor gestión de inventarios, se utilizó el modelo ARIMA (promedio móvil integrado autorregresivo) para pronosticar el precio de las materias primas en función de los datos históricos. Luego, se usó el modelo de regresión XGBoost (eXtreme Gradient Boosting) para realizar el pronóstico de la demanda. Finalmente, se integraron diferentes variables en un modelo de optimización combinatoria para ayudar a tomar la mejor decisión.

Figura 33: Arquitectura de la Herramienta de Apoyo a la Decisión para la Gestión Dinámica de Inventario.



Fuente: Namir, Labriji y Ben (2021)

ARIMA solo se puede aplicar en una serie de tiempo estacionaria y univariante. Para ello, fue fundamental hacer estacionarios los datos utilizados con una diferencia finita "*d*" y determinar el número de términos autorregresivos "*p*" y el número de errores de pronóstico rezagados en la ecuación de predicción "*q*".

Técnicas de Machine Learning

Se utiliza modelos de machine learning, series de tiempo para predecir los precios del stock y el pronóstico de la demanda del stock para garantizar la producción y luego se integra en un algoritmo de optimización combinatoria para ayudar a tomar la mejor decisión (Comprar, Vender o Mantener), y las cantidades exactas para comprar o vender con el fin de maximizar las ganancias sin riesgos.

Para la optimización Combinatoria, las decisiones previstas se toman en función de la maximización del beneficio y la minimización del riesgo ya que el objetivo es asegurar, durante un período determinado, la mejor decisión posible para la gestión de los stocks. El modelo da una de las tres salidas: "Vender" o "Comprar" o "Retener". Para ello, se tuvieron en cuenta las siguientes variables:

- P = Precio del stock en el mercado
- K = Precio del costo de Inventario
- C = Costo de almacenamiento
- S_0 = Stock de seguridad
- D = demanda
- S_{max} = Capacidad máxima de almacenamiento
- Q = Nivel de inventario actual

Si $Q \geq D$, son posibles tres decisiones y la mejor opción es según el beneficio neto. Sin embargo, si $Q < S$, tenemos dos decisiones posibles: "Comprar" o "Retener". El problema se puede modelar matemáticamente mediante la siguiente ecuación (1):

Figura 34: Ecuación 1: Modelamiento

$$\begin{cases} \text{Max}[(P-K-C)x_1 + (K-P-C)x_2 - Cx_3] \\ x_1 + x_2 + x_3 = 1 \quad \text{(Decision constraint)} \\ (D-Q)x_1 + (S_0-Q)x_3 \leq 0 \quad \text{(Quantity constraint)} \\ x_1, x_2 \text{ et } x_3 \in \{0,1\} \end{cases} \quad (1)$$

$$x_1 = \begin{cases} 1 & \text{if the decision is to "}Sell\text{"} \\ 0 & \text{if not} \end{cases} \qquad x_2 = \begin{cases} 1 & \text{if the decision is to "}Buy\text{"} \\ 0 & \text{if not} \end{cases}$$

$$x_3 = \begin{cases} 1 & \text{if the decision is to "}Hold\text{"} \\ 0 & \text{if not} \end{cases}$$

Fuente: Namir, Labriji y Ben (2021)

Una vez identificada la mejor decisión a tomar, el siguiente problema a resolver consiste en determinar la cantidad adecuada a comprar o vender. Para ello, q_1 es la cantidad máxima a vender y q_2 es la cantidad máxima a comprar de acuerdo a la decisión tomada para la gestión de inventario. q_1 y q_2 son soluciones del siguiente programa lineal:

Figura 35: Ecuación 2 Programa Lineal

$$\begin{cases} \text{Max}[(P-K-C)q_1+(K-P-C)q_2] \\ 0 \le q_1 \le \text{Max}(0, Q-S) \quad \text{(Constraint of quantity to sell)} \\ \text{Max}(0, S_0-Q) \le q_2 \le S_{max}-Q \quad \text{(Constraint of quantity to buy)} \\ q_1 \text{ et } q_2 \in \mathbb{N} \end{cases} \quad (2)$$

Fuente: Namir, Labriji y Ben (2021)

La ecuación (2) determina la cantidad óptima de stock para pedir o vender.

El modelo de pronóstico ARIMA incluye el proceso de promedio móvil (MA) y el proceso autorregresivo (AR). Un proceso autorregresivo de orden p: AR(p) se expresa de la siguiente manera:

Figura 36: Ecuación 3 Proceso autoregresivo

$$M_t = c + v_t + \sum_{i=1}^{p} \emptyset_i M_{t-i} \quad (3)$$

Fuente: Namir, Labriji y Ben (2021)

Un proceso de promedio móvil de q-ésimo orden: MA(q) se expresa de la siguiente manera:

Figura 37: Ecuación 4 Proceso de promedio móvil

$$M_t = u + \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} \quad (4)$$

Fuente: Namir, Labriji y Ben (2021)

ARMA (p,q) se obtiene fusionando (3) y (4):

Figura 38: Ecuación 5 Fusión

$$M_t = c + \varepsilon_t + \sum_{i=1}^{p} \emptyset_i M_{t-i} + \sum_{i=1}^{q} \varepsilon_i M_{t-i} \quad (5)$$

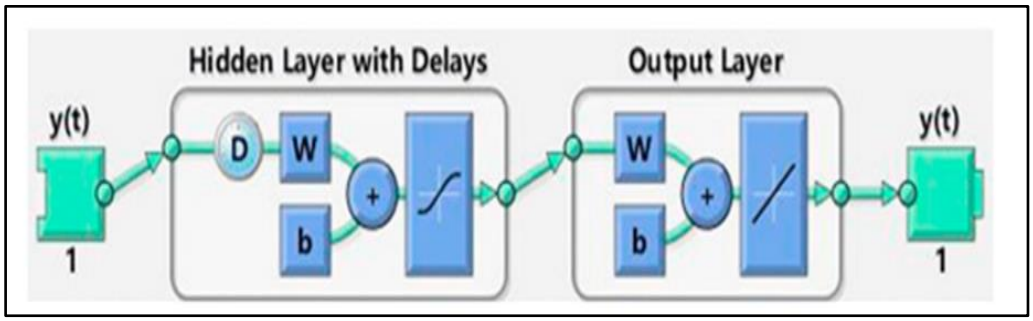
Fuente: Namir, Labriji y Ben (2021)

Para estimar el parámetro del modelo ARIMA se utilizó la función auto_arima () en Python. Esta función proporciona los valores *p, q y d* óptimos adecuados para el conjunto de datos.

El modelo NARNN se utiliza para predecir la misma serie temporal del precio del aceite base Y_t como Arima.

El algoritmo incorpora la salida a la entrada de la red (a través de retardos) como se muestra en la Figura:

Figura 39: Configuración del modelo NARNN para la previsión de la demanda.



Fuente: Namir, Labriji y Ben (2021)

El historial de datos se dividió aleatoriamente en dos partes: el 90 % se usó para entrenar la red y el 10 % restante se usó para probar la red.

La serie de tiempo del precio del aceite base se pudo modelar combinando la autocorrelación lineal M_t y Y_t. El precio del aceite base P_t se puede expresar usando ARIMA-NARNN como:

Figura 40: Ecuación 6 Precio

$$P_t = M_t + Y_t \ (6)$$

Fuente: Namir, Labriji y Ben (2021)

El modelo XGBoost estará asociado a un residual inicial. Luego, se ajusta un nuevo modelo a los residuos del paso anterior. Después de eso, el algoritmo combina los dos primeros modelos para impulsar el modelo inicial definido para predecir la demanda. Este nuevo modelo disminuye el error cuadrático. Estos tres pasos se repiten hasta que se mejora el error. El resultado de la predicción es la suma de las puntuaciones predichas por los árboles, como se muestra en la siguiente fórmula:

Figura 41: Ecuación 7 Resultado de predicción

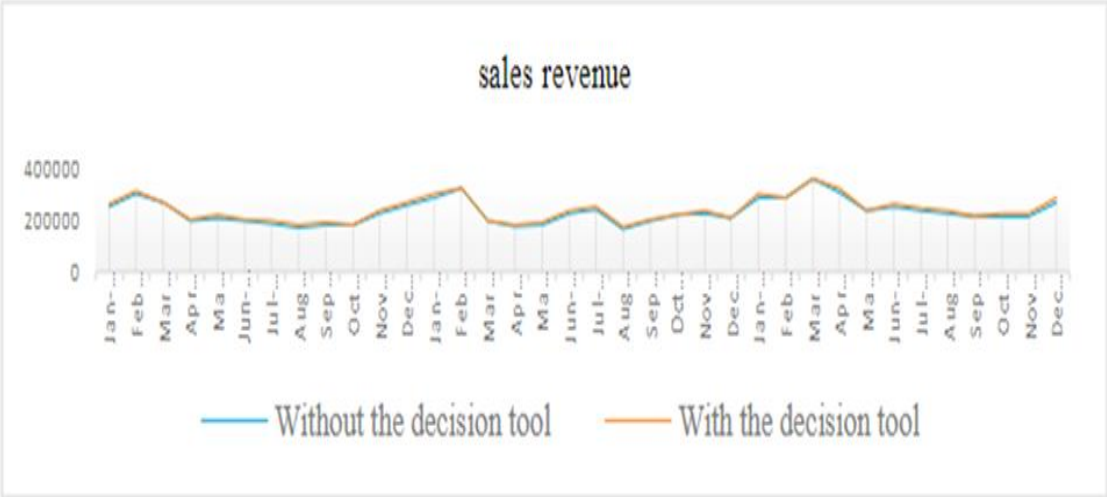
$$P_t = \sum_{k=1}^{K} f_k(x_i), f_k \epsilon F \ \ (7)$$

Fuente: Namir, Labriji y Ben (2021)

Resultados

Como resultado de esta investigación combinando los submodelos e introduciendo las diferentes variables en el modelo de optimización combinatoria, se obtuvo una herramienta de decisión para la gestión de inventarios que puede utilizarse para optimizar el ciclo del stock maximizando el beneficio y evitando al máximo la escasez de stock. Asimismo, estos resultados ayudarán a otras empresas que gestionen inventarios a poder pronosticar la demanda y a minimizar el riesgo de abastecimiento.

Figura 42: Comparación entre la gestión de inventarios con y sin la herramienta de decisión propuesta.



Fuente: Namir, Labriji y Ben (2021)

Gutiérrez (2020). Predicción de múltiples series de tiempo univariadas a través de diversos modelos predictivos y meta-learning aplicado en la industria del retail.

Problema

La problemática se enfoca en una empresa retail que tiene como función la distribución de sus artículos desde su centro de distribución. Para ello, se debe distribuir a cada local para tener un stock de los productos y pueda reponerse cada vez que haya un desabastecimiento. El autor menciona que existen dos problemas, primero que no existe una buena predicción de la demanda lo que genera algún quiebre en el stock, no teniendo suficientes artículos para ventas y perdiendo ingresos y segundo; si se subestima la demanda se tendrá demasiado stock de productos generando altos costos de almacenamiento. Lo que se intenta es disminuir estos costos asociados a cada problema descrito anteriormente.

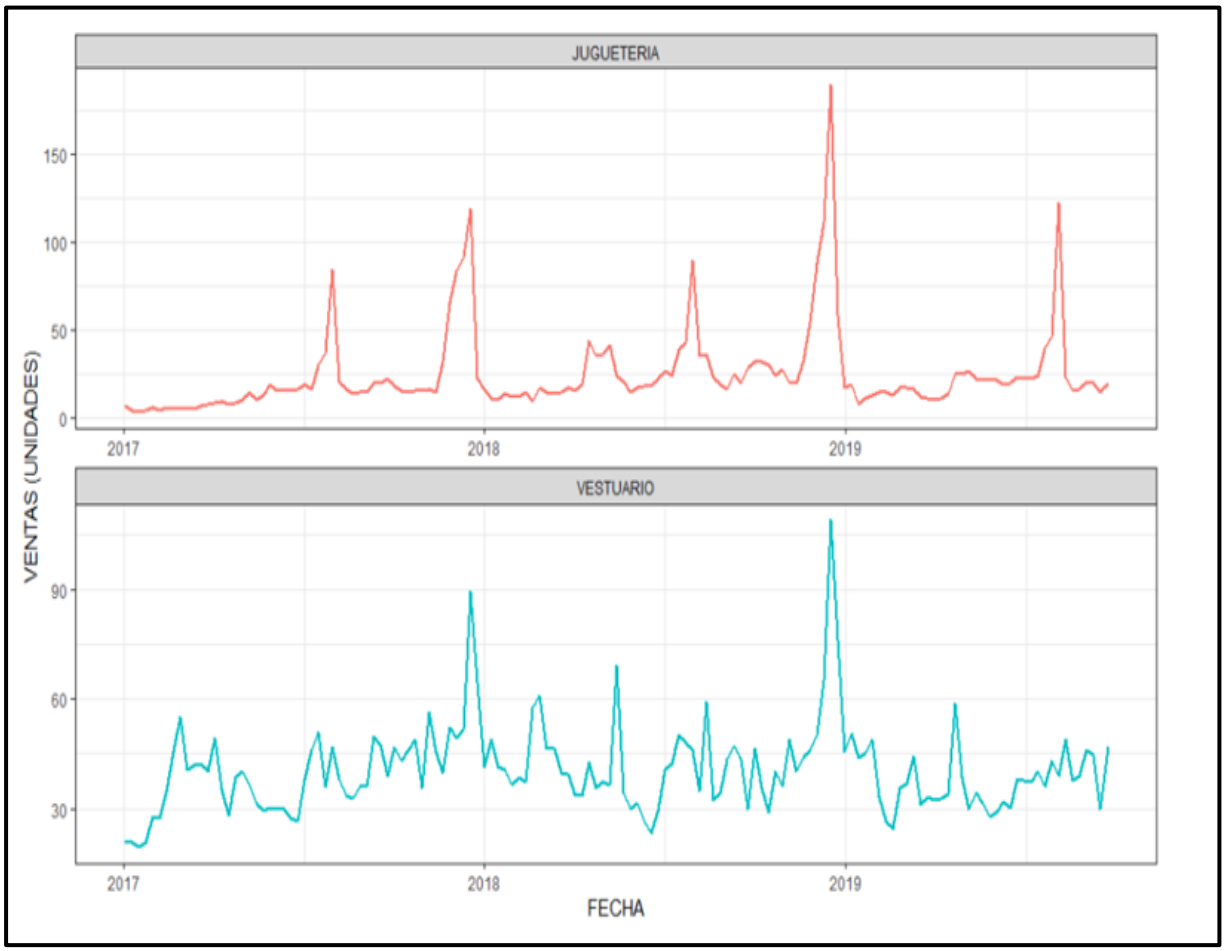
Objetivo

Buscar el desarrollo de un modelo predictivo para tener el pronóstico de la media móvil de los artículos y/o productos de vestuario y juguetería del retail disminuyendo los costos logísticos y obteniendo una mejor precisión del pronóstico. Asimismo, otros de los objetivos específicos son a través de meta-learning poder validar un modelo que se ajuste mejor a la predicción lo cual permite reducir costos y eficiencia de tiempos.

Datos

Se consideró para el estudio un universo de 5 000 series de tiempo univariadas, teniendo las variables de ventas y fechas. Los datos de ventas son desde enero 2017 hasta septiembre de 2019. Los artículos seleccionados fueron de las categorías vestuario y juguetes. El primero representa un 90%. Con esta información se pudo obtener series de tiempo con diferentes casuísticas.

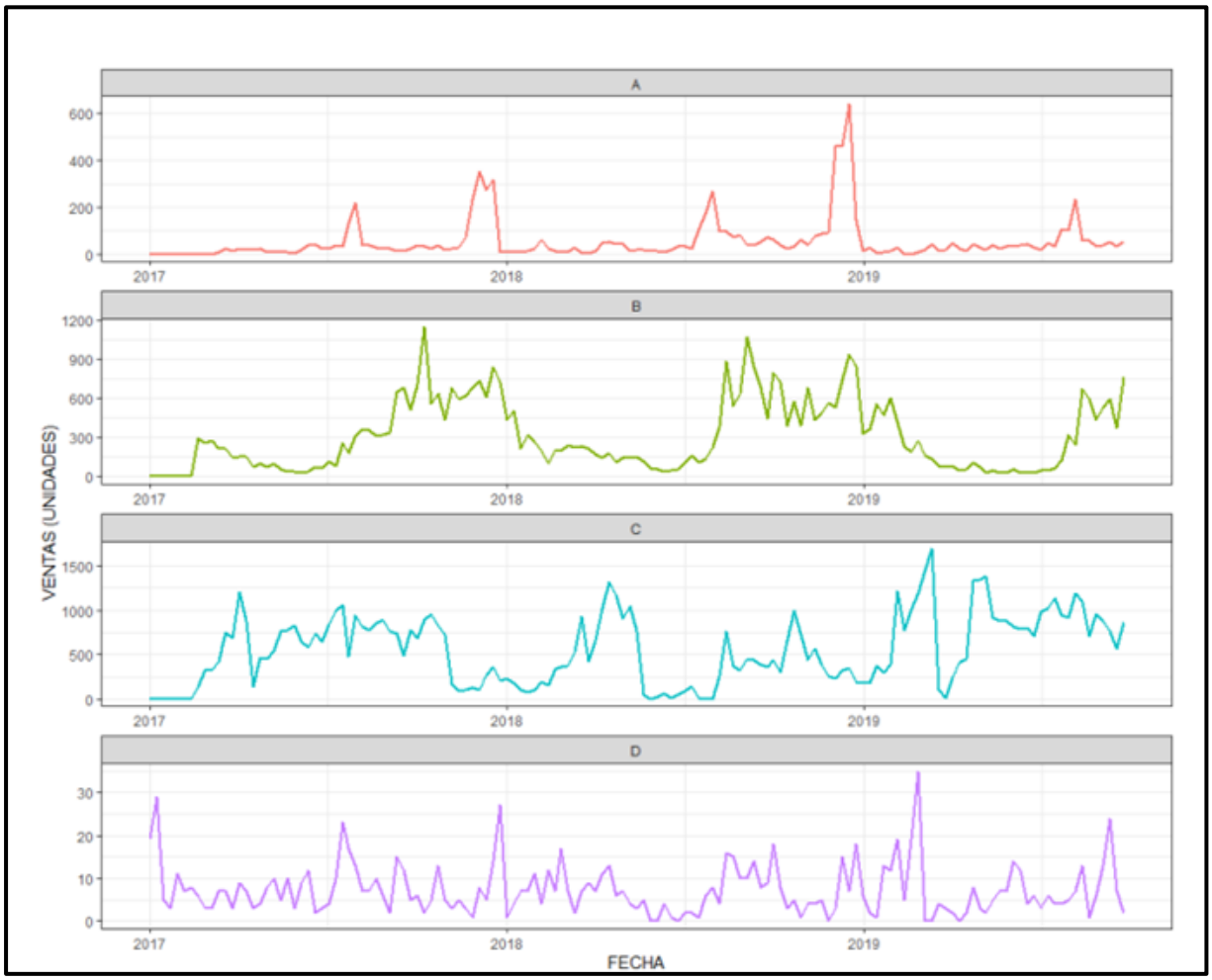
Figura 43: Promedio de ventas agregadas por sección.



Fuente: Gutierrez (2020)

Se muestra a continuación el comportamiento de productos:

Figura 44: Ejemplos de comportamientos de productos.



Fuente: Gutiérrez (2020)

Metodología

En primer lugar, se recopilan los datos para poder comprender las series de tiempo, se eliminan valores nulos, negativos o que presentan algún error que puedan perjudicar los resultados del estudio. Luego, con la limpieza de la data se podrá entender mejor cuál es el comportamiento que siguen los datos presentados.

Para la investigación, se presentan 7 modelos de pronóstico: media móvil, Holt, Arima, Stlf, NNetAR, TBATS y un ensamble, las 5000 series de tiempo serán pronosticadas. Esto permitirá evaluar los mejores candidatos a un mejor pronóstico.

Para pronosticar estos modelos, se consideró un horizonte de cuatro semanas discernido por el criterio MAE y WMAPE. para considerar el peso de WMAPE lo definieron por las

ventas, Asimismo, los dos criterios MAE y WMAPE llevan a iguales conclusiones, El primero, permite visualizar la diferencia en las unidades promedio, y la segunda muestra un error porcentual ajustado a las ventas. Además, se tomó una métrica que permite analizar la mejor predicción, que se ajuste mejor a la realidad. Posteriormente, a través del modelo meta-learning podrá clasificar las series de tiempo en su ingreso, analizándolas, considerando sus características, luego, selecciona el mejor modelo con mayor rendimiento.

Para este estudio se usa un 80% para el entrenamiento y el 20% será destinado para realizar el testeo. También, se muestra el modelo clasificador random forest para el entrenamiento.

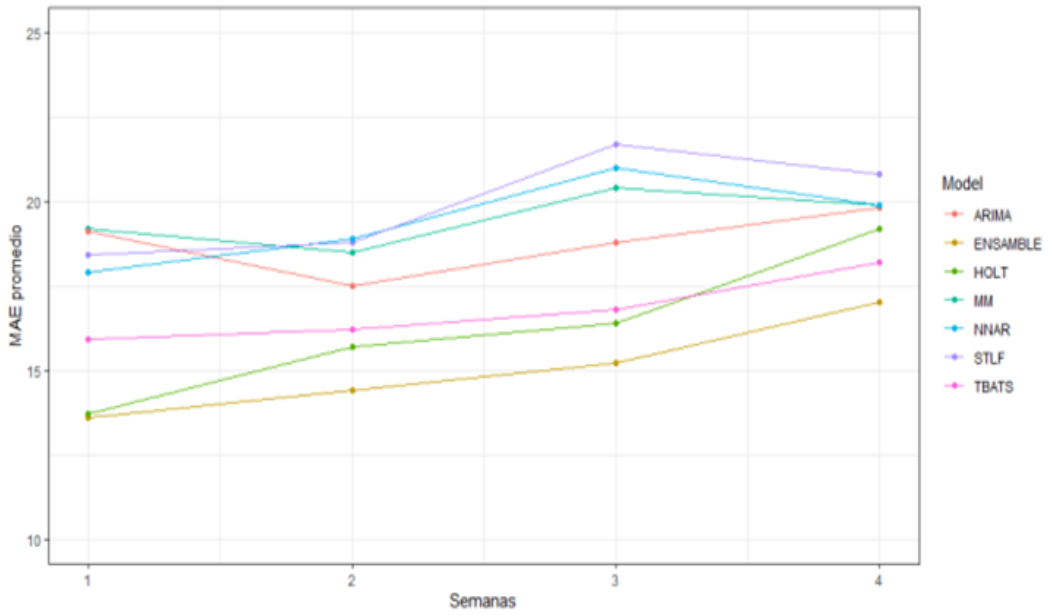
Técnicas de Machine Learning

Con las 5 000 series de tiempo se podrá separar en set tanto de entrenamiento como de testeo, según lo mencionado anteriormente para evaluar el más preciso.

Análisis por criterio MAE

En la tesis, se pudo evaluar que el modelo con menor MAE es el de Ensamblado. A continuación, se muestra la imagen del análisis por semanas.

Figura 45: MAE por modelo a distintos horizontes.



Fuente: Gutiérrez (2020)

Se utiliza el paquete “tsfeatures” para atributos características. El modelo random forest permite utilizar variedad y multiplicidad de árboles de decisión para poder obtener un mejor resultado. Para el estudio, se usó 1 000 árboles donde se pudo observar que los resultados obtenidos no tienen una variación significativa con cantidades mayores.

Se entrena el 80% y se predice el restante, el 20%.

En la tesis se consideraron 4 casos a estudiar: Los tres modelos con mayor MAE, Dos mejores modelos según MAE, Todos los modelos según MAE y todos los modelos entrenando solo con los dos mejores

Todos los modelos entrenando solo con los dos mejores

Para el estudio, y en particular el caso, se usa como input únicamente los modelos de Holt y Ensamble, ya que estos modelos tienen mejor rendimiento a nivel de MAE/WMAPE lo que quiere decir que el clasificador predice los dos modelos a pesar de realizar la comparación con todos los demás.

Tabla 4: Comparación entre modelo de clasificación a una semana con modelos, entrenado con los mejores dos.

MODEL	MAE	WMAPE	WIN_RATE
CLAS	**11**	**27.1%**	**24.9%**
ENSAMBLE	12.7	31.3%	9.2%
HOLT	12.9	31.8%	19.3%
TBATS	14.8	36.5%	12.8%
STLF	17	41.9%	14.0%
NNAR	17.2	42.4%	16.5%
MM	18.3	45.1%	14.2%
ARIMA	18.5	45.6%	14.1%

Fuente: Gutiérrez (2020)

Se presenta según el autor que al entrenar con los dos mejores modelos se tiene un MAE con un buen y mejor resultado. Además, comenta que la predicción de ARIMA, si es que no es el correcto solo te acerca al 18.5 de MAE promedio que tiene ese modelo por sí solo, Por el contrario, la predicción de un ensamble y lograr equivocarse, acercará a un MAE de 12.7, que viene a ser un mejor resultado. En el testeo realizado se obtuvo una venta de 40.6 unidades en

promedio, aduciendo que el modelo clasificador tiene un WMape de 27.1%, mientras que la media móvil tiene WMape de 45.1%, generando una predicción un 18% más precisa. Los resultados obtenidos son de mucha importancia ya que se puede evidenciar que hay oportunidad de mejora en la predicción de las ventas.

Resultados

Se obtiene como resultado que al entrenar al clasificador con algunas etiquetas (pocas) este clasificador podrá mejorar la predicción de manera satisfactoria a nivel de MAE y Win_rate. Por lo que el autor menciona que a nivel de series de tiempo este es el modelo con mayor precisión.

Luego de las pruebas, el autor concluye que el modelo clasificador de meta-learning fue realmente exitosa en MAE, WMAPE y win_rate, por lo que se puede deducir que servirá para el pronóstico de varias series de tiempo en retails, pero debe tenerse mucho cuidado en su implementación para no alcanzar resultados no deseados o deficientes.

Además, con el clasificador de meta-learning si este se entrena de forma correcta el modelo clasificador random forest puede elegir el modelo aceptable a predecir y el que debe implementarse en las ventas del caso de estudio del retail.

Asimismo, debe ser entrenado de forma adecuada y con toda la información que se tenga disponible ya que en el estudio el mejor clasificador fue entrenado solo con 2 modelos de los 7 que había ya que con todos aumentaba el error promedio absoluto en la etapa de testeo.

2.2 Bases Teóricas

2.2.1 Inteligencia Artificial

La Inteligencia Artificial se inicia en teorías existentes en otras áreas del conocimiento. Para mencionar algunas de ellas que sirvieron de base para este conocimiento, tenemos: las ciencias de computación, filosofía, matemática, psicología y lingüística que ayudaron como inspiración para lograr lo que hoy en día se conoce como IA. Con las ciencias antes descritas ayudaron y contribuyeron con las diferentes herramientas y la experiencia del estudio de cada una de ellas. De esta manera se pudo obtener el desarrollo de esa área de conocimiento. (Ponce, Torres, Quezada, Silva, Martinez, Casali, Scheihing, Túpac, Torres, Ornelas, Hernandez, Zavala, Vakhnia, y Pedreno, 2014)

Asimismo, Ponce et al. (2014) comenta que en la actualidad la Inteligencia Artificial es un área de la ciencia de mucho interés ya que es un área multidisciplinaria donde se realizan sistemas que intentan hacer tareas y resolver problemas como lo haría un humano, se trata de una simulación artificial del pensamiento de un cerebro y de cómo este trabaja para poder tomar decisiones. Todavía no se ha podido realizar todo lo que nos podemos imaginar o ver en ciencia ficción, pero ya se encuentra presente en muchas aplicaciones, dispositivos y aparatos que el ser humano utiliza de manera cotidiana.

Por otro lado, Sosa, M. (2007) define a la inteligencia artificial como la capacidad que tienen los programas de computador para operar en la misma forma en que el pensamiento. Asimismo, comenta que es la simulación de una inteligencia humana en una máquina que ayudará a solucionar problemas a través de identificaciones y usando piezas de conocimiento.

2.2.2 Machine Learning

Según Bobadilla (2020), Machine learning viene a ser una ciencia que permite que los ordenadores o computadoras aprendan a partir de datos. También, comenta que el área de Machine Learning se dedica a desarrollar algoritmos que permitan extraer patrones de diferentes tipos de datos.

Asimismo, el aprendizaje automático para Ponce et al. (2014) usa una teoría estadística para construir modelos matemáticos ya que de esta forma puede hacer inferencias a partir de muestras. Además, la ciencia de la computación es requerida en la fase de entrenamiento para la implementación de algoritmos de optimización eficientes ya que es necesaria en las tareas de almacenamiento y procesamiento de datos a altos volúmenes. Cuando el modelo es ajustado, es requerido la eficiencia en su representación y solución algorítmica para la fase de inferencia.

Los algoritmos de machine learning se pueden clasificar en aprendizaje supervisado y en aprendizaje no supervisado.

En el primer caso, el aprendizaje supervisado corresponde a la situación cuando se tiene una variable de salida, ya sea cuantitativa o cualitativa, que se desea predecir basándose en un conjunto de características. Para ello se establece un modelo que permitirá relacionar las características con la variable de salida. Ponce et al. (2014)

En cambio, para el aprendizaje no supervisado, Ponce et al. (2014) indica que este corresponde a una situación donde existan un conjunto de datos con diversidad de

características de determinados individuos. En ninguna de ellas se puede considerar una variable de salida para poder predecirla. Por ello, el objetivo es poder describir cómo están organizados los dados para asociarlos entre ellos o agruparlos.

La subclasificación corresponde a lo siguiente:

- Aprendizaje supervisado
    - Regresión
    - Clasificación
    - Series de Tiempo
- Aprendizaje no supervisado
    - Clustering (agrupamiento)
    - Reducción de dimensiones

2.2.3 Aprendizaje Supervisado

"El aprendizaje supervisado en machine learning se aplica cuando cada dato, o conjunto de datos de entrada (muestra) tiene asociada una etiqueta." (Bobadilla, 2020, p.14)
Para poder predecir mediante un aprendizaje supervisado se requiere de dos variables: Una variable de entrada "*X*" y una variable de salida "*Y*".

Figura 46: Variables cuantitativas o cualitativas.

Fuente: Ponce et al. (2014)

VanderPlas (2017) comenta que el aprendizaje supervisado es la relación que existe entre los datos con sus diferentes características y los datos que tengan una etiqueta. Después de definir el modelo, comenta que se puede aplicar etiquetas a los datos que se desconocen o algunos datos nuevos.

Por lo que comenta que se puede dividir en clasificación y regresión. Para la clasificación, se trata de categorías discretas y, por otro lado, para la regresión se consideran cantidades continuas.

- Clasificación

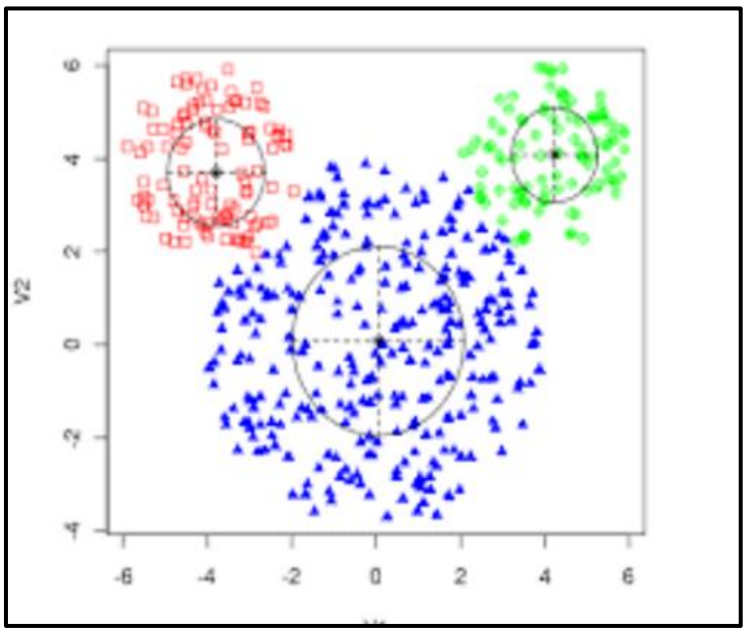
Según Sandoval (2018) mediante este algoritmo se conoce el grupo al que pertenece el elemento a estudiar. El algoritmo encuentra una relación y patrones en los datos brindados para posteriormente agruparlos.

Después, realiza la comparación de los nuevos datos y son ubicados en los grupos descritos anteriormente. De esta manera se puede predecir. A continuación, se muestran las variables que se pueden predecir tanto discretos como categóricos.

Pueden ser:

- ➢ Binaria: Sí, No; Azul, Rojo; Fuga, No Fuga; etc.
- ➢ Múltiple: Comprará, Artículo 1, Artículo 2…, etc.
- ➢ Ordenada: Por riesgo, Bajo, medio, alto, etc.

Figura 47: Clasificación.

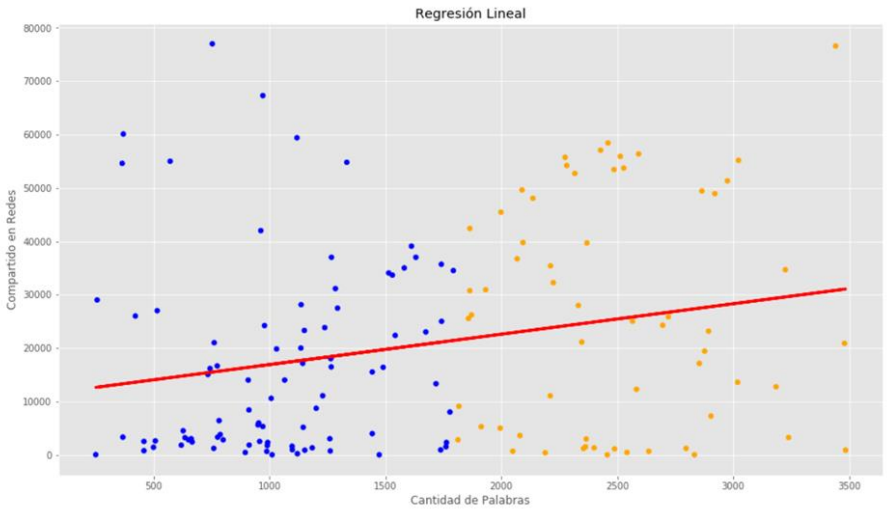


Fuente: Sandoval (2018)

- Regresión.

Para este método se requieren y esperan datos numéricos. A diferencia del anterior, el algoritmo no logra clasificarlos en grupos, sino que devuelve un valor específico. Bagnato (2018) menciona que para hacer este método se mide el error con respecto a los puntos de entrada y el valor "Y" de salida real. El algoritmo deberá minimizar el coste de una función de error cuadrático y esos coeficientes corresponden con la recta óptima.

Figura 48: Regresión Lineal.



Fuente: Bagnato (2018)

2.2.4 Series Tiempo

La finalidad de todo método de serie de tiempo es encontrar un patrón o comportamiento en la data histórica para estimarla y luego sacarle el máximo provecho a futuro, la predicción se basa únicamente en valores anteriores de la variable que buscamos predecir o en errores históricos ajustada por influencia estacional.

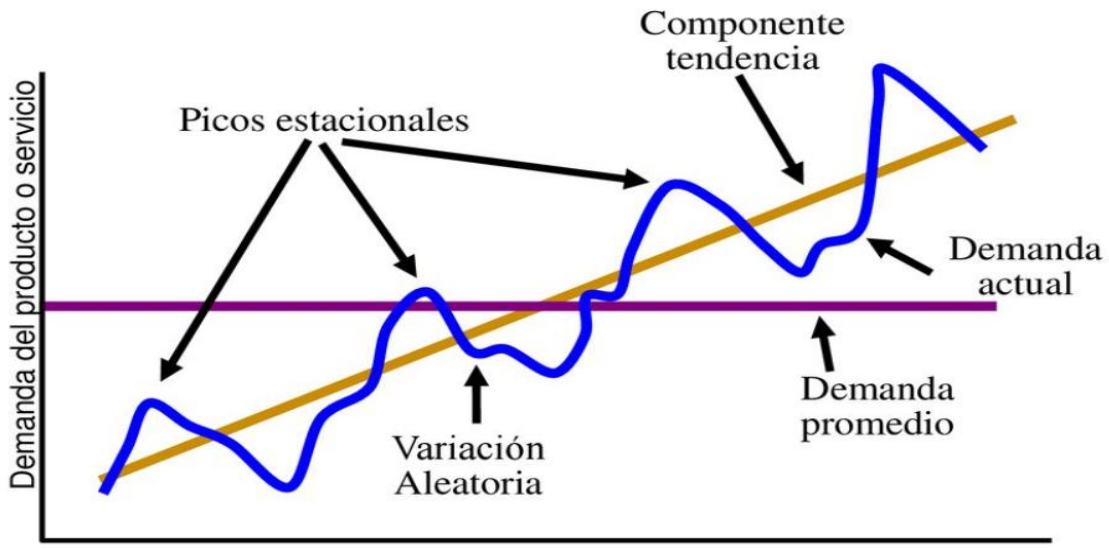
Según Cerna (2021) las series de tiempo también llamadas series cronológicas o históricas, se definen como la sucesión de observaciones tomadas para una variable en distintos momentos de tiempo separados de manera regular (horas, días, meses, trimestres, años).

Cerca (2021) menciona que los componentes de esta técnica:

- Tendencia: En una serie cronológica en largos periodos puede crecer o decrecer

- Estacionalidad: La oscilación periódica en la cronología de datos dentro de un lapso generando un comportamiento que tiende a repetirse después de un periodo estacional consecutivo.
- Ciclos: Se crean durante intervalos de tiempos largos, y los tiempos que transitan entre los valles consecutivos no son necesariamente idénticos.
- Movimiento Irregular: Error en una serie de tiempo.

Figura 49: Componentes de una serie de tiempo.



Fuente: Cerna Ramirez (2021)

2.2.4.1 Regularización

Estas estrategias funcionan bajo la incorporación de penalizaciones con la finalidad de reducir la varianza. En general, la regularización incrementa un mayor poder predictivo de modelos. Para la aplicación de esta estrategia, es de suma importancia normalizar los predictores previo al entrenamiento del modelo.

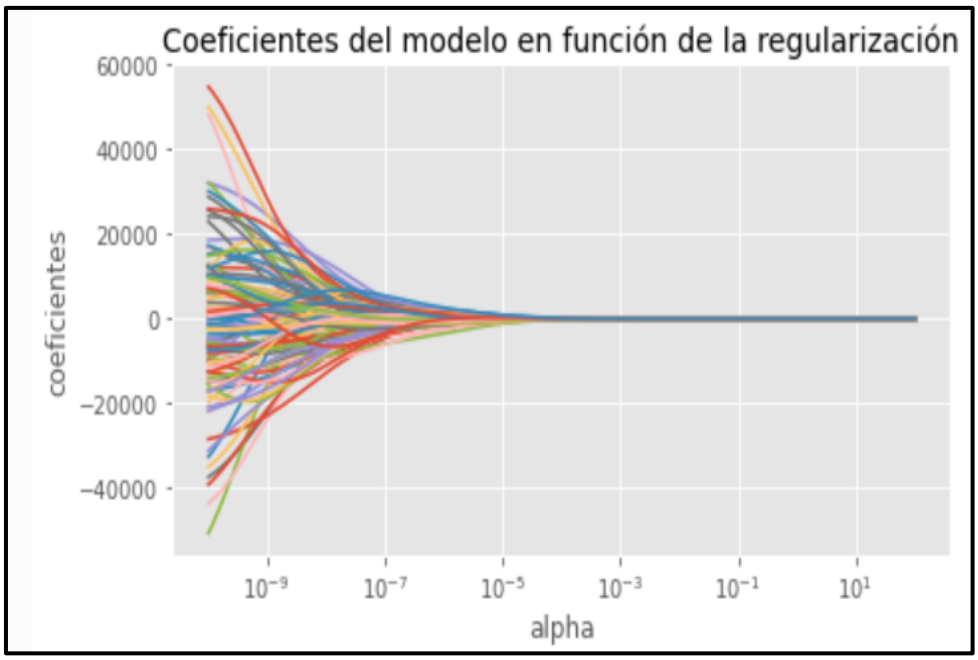
2.2.4.2 Ridge

Esta regularización multa la adición de los coeficientes elevados al cuadrado y tiene como consecuencia según Joaquín Amat (2020) reducir de forma proporcional el valor de todos los coeficientes del modelo, pero sin que estos lleguen a cero.

Este método tiene una serie de ventajas teniendo como principal la reducción de varianza y la principal desventaja es que modelo final toma a todos los valores. Esto se debe a que la penalización orienta a los coeficientes a ser cero más nunca llegan a ser ceros.

En otras palabras, Joaquín Amat (2022) menciona que “Este método consigue minimizar la influencia sobre el modelo de los predictores menos relacionados con la variable respuesta, pero, en el modelo final, van a seguir apareciendo. Aunque esto no supone un problema para la precisión del modelo, sí lo es para su interpretación.”

Figura 50: Coeficiente del modelo en función de la regularización.



Fuente: Joaquín, Amat (2022)

Se puede apreciar que conforme mayor es el valor de alpha, la regularización es mayor y el valor de los coeficientes se reduce.

2.2.4.3 Métricas de error

Una vez ya abordado el tema y conceptos de regresión, es de suma importancia conocer el desempeño del modelo, esto se hace o se informa como un error en tales predicciones calculadas mediante la función obtenida en el modelamiento de la regresión. En otras palabras, al momento de predecir una cifra numérica como espesor

o un monto en soles, no se quiere saber si en la práctica el modelo predijo el valor exactamente (esto sería ideal en la práctica), lo que se busca es saber qué tan cercano están las predicciones con los valores esperados.

Las métricas de error envuelven este análisis y en promedio muestra que tan cerca están las predicciones de los valores esperados. Según D. Ramón (2019) hay 3 métricas que se emplean con frecuencia para reportar el desempeño de un modelo de regresión; y estos son:

- Error cuadrático medio (MSE)
- Error cuadrático medio (RMSE)
- Error absoluto medio (MAE)

2.2.4.3.1 Error Medio Cuadrado. Llamado en inglés como mean squared error (MSE) es una métrica común de error para ejercicios de regresión. Según D. Ramón (2019) "También es una función de pérdida importante para los algoritmos ajustados u optimizados utilizando el marco de mínimos cuadrados de un problema de regresión. Aquí «mínimos cuadrados" se refiere a minimizar el error cuadrático medio entre las predicciones y los valores esperados."

Este se calcula mediante el promedio de las diferencias del cuadrado entre los valores esperados y predichos en un historial de datos.

Figura 51: Ecuación de MSE.

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

Fuente: D. Ramón (2019)

2.2.4.3.1 La raíz del error cuadrático medio. También se le conoce como RMSE. Esta métrica de error es una extensión del error cuadrático medio (MSE). Este se calcula como la raíz cuadrada del error. Posteriormente, se afirma que las unidades originales del valor a predecir son idénticas a las unidades del RMSE. Por ejemplo, si tu variable a predecir tiene las unidades de "soles", entonces el cálculo del error RMSE también tendrá como unidad "soles" y no "soles al cuadrado" como el MSE. Este se calcula de la siguiente manera:

Figura 52: Ecuación RMSE.

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

Fuente: D. Ramón (2019)

***2.2.4.3.1 Error absoluto medio*.** Según D. Ramón (2019) se calcula como el promedio de los valores de error absoluto. Absoluto o abdominales () es una función matemática que simplemente hace que un número sea positivo. Por lo tanto, la diferencia entre un valor esperado y predicho puede ser positiva o negativa y está obligada a ser positiva al calcular el MAE.

El cálculo del MAE se da bajo la siguiente fórmula:

Figura 53: Ecuación MAE.

$$\mathrm{MAE} = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i|$$

Fuente: D. Ramón (2019)

2.2.4.4 Scaler model

La mayoría de los modelos de aprendizaje autónomo se desempeñan de una mejor manera cuando las variables se escalan a un nivel estándar. La estandarización identifica las observaciones se ajusten a una distribución en forma de campana con una media y una desviación estándar de buen comportamiento.

La estandarización es buena cuando los datos tienen valores de entrada con diferentes escalas. La puntuación estándar de una muestra x se calcula como:

$$z = \frac{(x - u)}{s}$$

donde u es la media de las muestras de entrenamiento o cero y S es la desviación estándar de las muestras de un entrenamiento.

2.2.4.5 ARIMA

Según Gonzáles, M (2009) son modelos paramétricos que tratan de obtener una Figura de la serie de tiempo en evaluación, a fin de tener la interrelación temporal de los elementos que la componen. Este tipo de modelo tiene como instrumento fundamental el coeficiente de autocorrelación, el cual mide el grado de asociación lineal entre observaciones separadas en "n" periodos.

El coeficiente de autocorrelación permite conocer el tipo y grado de correlación que tienen las observaciones, a fin de poder establecer un modelo ajustado a la lectura de datos actuales e históricos. Para poder evaluar la relación entre la variable dependiente e independiente, es necesario aplicar la siguiente fórmula de autocorrelación:

$$\rho_{xy} = \frac{cov(x,y)}{\sqrt{V(x)V(y)}}$$

Donde P_{xy} tiene como rango de obtención de valores entre -1 y 1, la cual se explica de la siguiente manera:

- Si el valor de P_{xy} es 0, no existe relación alguna entre ambas variables en estudio.
- Si el valor de P_{xy} es 1, existe una relación perfecta entre ambas variables de manera positiva
- Si el valor de P_{xy} es -1, existe también una relación perfecta entre ambas variables de manera negativa.

2.2.4.5.1 Procesos Estocásticos. Según Gonzáles, M (2009) un proceso estocástico se define con una serie de variables aleatorias familiarizadas, las cuales siguen una distribución conjunta.

Este proceso tiene una serie de característica que lo identifican, las cuales son las siguientes:

- Función de distribución: Es primordial para entender esta función conocer las funciones de distribución univariantes, bivariantes y trivariantes de cada una de las variables aleatorias.

- Momentos del proceso estocástico: Al ser complicado caracterizar el proceso mediante la función de distribución, se procede a utilizar para la caracterización los dos primeros momentos, los cuales refieren a momentos donde el conjunto de medias y varianzas de las variables aleatorias del proceso son usados para caracterizar el proceso, teniendo en cuenta que la distribución del proceso tiene que ser normal.

Los procesos estocásticos se dividen en 2 grupos: estacionarios y no estacionarios. Los procesos estocásticos estacionarios refieren a procesos donde la serie de tiempo presenta un grado de estacionalidad o patrón con respecto al incremento de años, lo que permite desarrollar predicciones más consistentes. Si no fuese el caso del comportamiento estacionario, no sería factible desarrollar el modelo de series de tiempo para el estudio de las variables.

A partir de estos procesos, se puede definir a los procesos estocásticos como agentes de estudio para la modelización ARIMA, la cual, trabaja con modelos estacionarios y no estacionarios con referencia a la varianza, covarianza, medias y otros datos estadísticos, los cuales deben tener un comportamiento y valorización específica con respecto al tiempo a fin de poder tener modelos de predicción más acertados. Para ello, se realiza una cadena de procesos a fin de aplicar este modelo, ya sea el proceso de identificación, el cual se enfoca en validar que la variable en estudio tiene un comportamiento estacionario. Luego, se realiza el proceso de estimación de parámetros, los cuales ayudan a que el modelo pueda completar un procesamiento de datos de predicción con respecto a la data histórica que se le ha brindado, utilizando lenguaje de programación ARIMA (p,d,q) y ARMA (p,q). A continuación, se procede a ingresar al proceso de validación del modelo, donde se busca realizar diversos análisis para verificar el ajuste del modelo predicho con el modelo real. Entre esos análisis están el de parámetros, el de residuos, el de bondad de ajuste y el de estabilidad. Por último, se realiza el proceso de predicción de acuerdo con el modelo ARIMA, donde las predicciones tienen como característica principal lo siguiente:

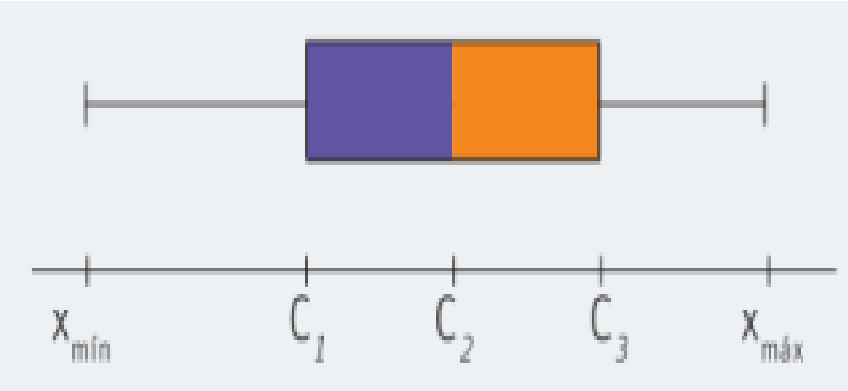
- Modelos AR(p): A medida que aumenta el tiempo, la predicción se basa en el promedio del proceso.
- Modelos MA(q): Si el tiempo es mayor que el orden del proceso, la predicción va a tener el mismo valor que el promedio de datos.

- Modelos ARMA (*p,q*): Para periodos de tiempo mayores a la orden del proceso, las predicciones tienen un comportamiento como el Modelo AR, por lo cual, la predicción también se basa en el promedio del proceso.
- Modelos ARIMA (*p,q,d*): La predicción en este modelo ya no se basa en el promedio del proceso, sino que, a partir de la transformación de datos para darle el carácter de estacionalidad, se basa en una línea recta con el mismo valor de pendiente que la media del proceso.

2.2.4.6 Gráficos de análisis

2.2.4.6.1 Diagrama de cajas. Según Videla, X. (2017) los gráficos de caja visualizan las medias, las medianas, cuartiles de diverso grado y también los valores atípicos del conjunto de datos. Este gráfico consiste en un rectángulo denominado caja, donde los lados con mayor longitud muestran el recorrido entre los cuartiles. La caja está dividida por una recta, la cual hace referencia a la mediana y el grado de relación con los demás cuartiles. Asimismo, la información que proporciona el diagrama de cajas es el nivel de dispersión y simetría de datos, a partir de la ubicación de la recta en la caja y la distancia con los cuartiles, además de la longitud de los bigotes de la caja.

Figura 54: Diagrama de cajas y bigotes.

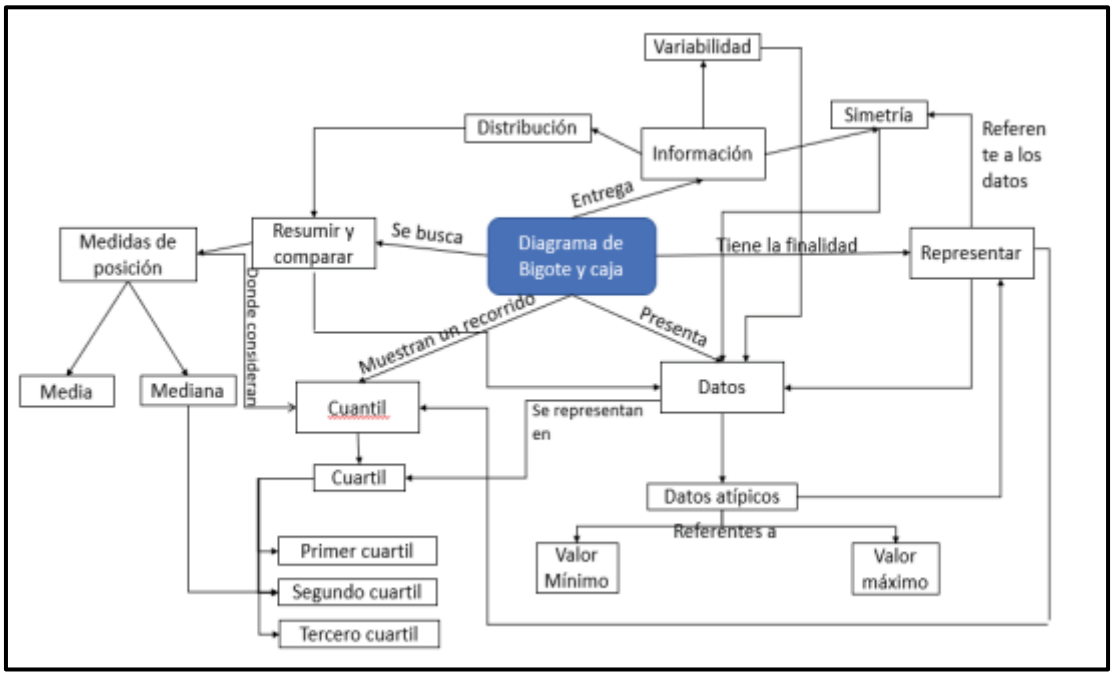


Fuente: Videla, X. (2017)

Donde C_x, son los valores de los cuartiles y el X_{min} y X_{max} son los valores mínimos y máximos de la muestra, respectivamente.

Para efectos de una mejor explicación del diagrama de cajas, se adjunta el siguiente mapa:

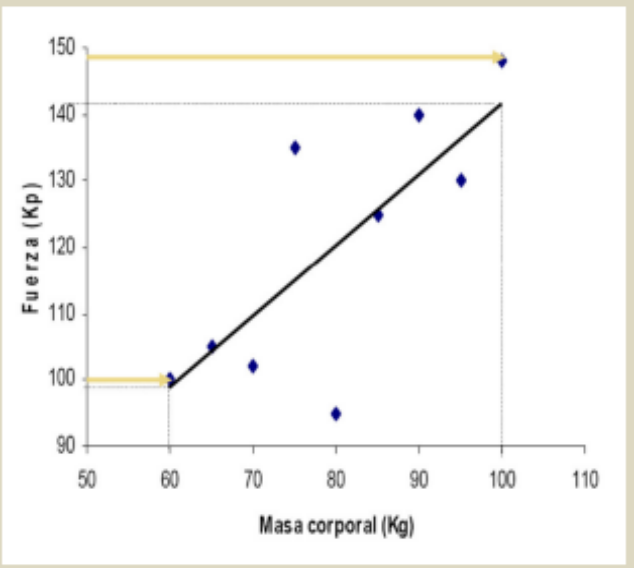
Figura 55: Mapa conceptual de diagrama de caja y bigote



Fuente: Videla, X. (2017)

2.2.4.6.2 Diagrama de correlación. El diagrama de dispersión se usa para mostrar el grado de relación entre diferentes variables, para así poder estudiar los tipos de relaciones que existen entre dichos factores. Su objetivo principal es determinar a partir del análisis de correlación la forma en que se relacionan y el grado de dependencia entre una y otra variable. (Diagrama de dispersión, 2019).

Figura 56: Diagrama de dispersión

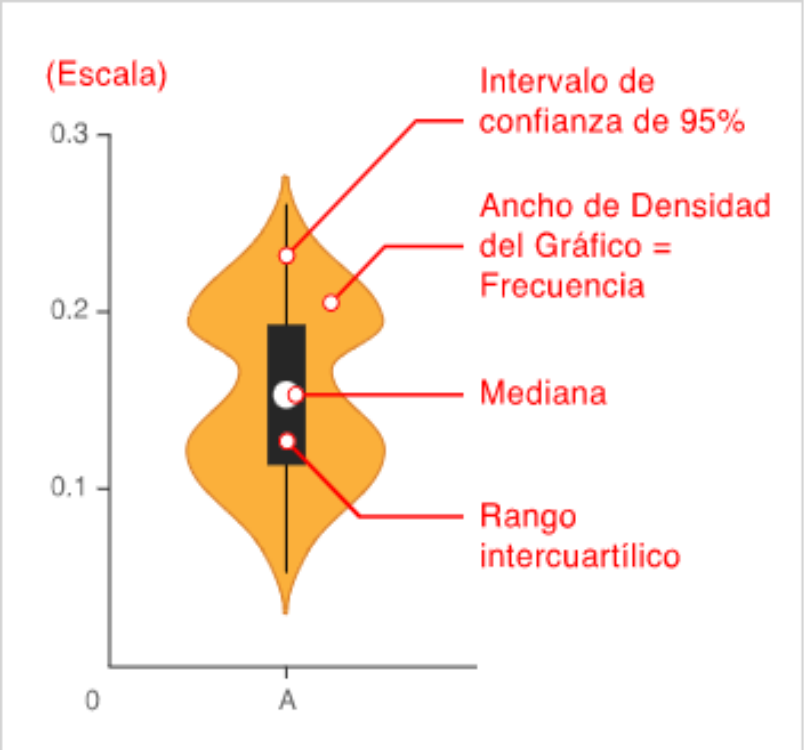


Fuente: Ramón, G. (2000)

***2.2.4.6.3 Diagrama de violín*.** Los diagramas de violín tienen la composición similar con los diagramas de caja, con la única diferencia que este gráfico incluye el diagrama de densidad de Kernel rotado en cada lado. Incluye los mismos datos que un diagrama de caja, ya sea la mediana, los cuartiles y la media, teniendo como agente diferenciador la representación de la distribución completa de los datos. Este tipo de diagrama se utiliza cuando la distribución de datos es multimodal, es decir, muestra diferentes picos o puntos máximos, además de su posición y amplitud relativa. (Gráfico de violín, 2022).

El diagrama de violín está compuesto por varias capas, donde la forma exterior representa todos los resultados posibles y la capa interior representa todos los valores que se suscitan en cierto porcentaje de tiempo. Hay una capa intermedia donde se pueden ubicar todos los datos que ocurren en la mitad del porcentaje total con respecto al tiempo.

Figura 57: Diagrama de Violín.

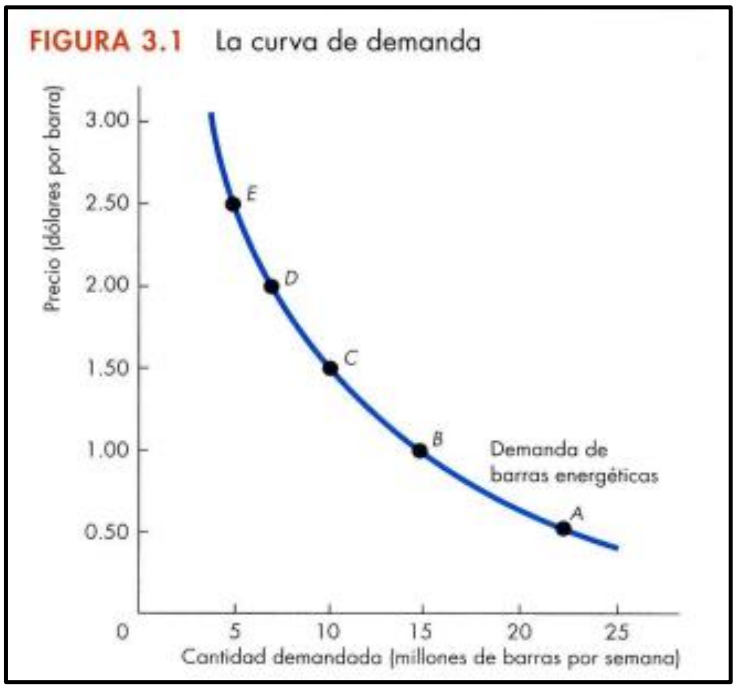


Fuente: Catálogo de visualización de datos (2018)

2.2.5 Demanda

La demanda se puede explicar de distintas maneras de acuerdo con el enfoque que se le pretende dar, por ejemplo, Coca A. (2011) define a la demanda de mercado de un producto como el volumen total que adquiere un grupo de personas bajo cierto contexto en específico. Es decir, se puede determinar la demanda como una cantidad específica, ya sea un bien o servicio, los cuales una o más personas desean adquirir a fin de satisfacer una necesidad.

Figura 58: Curva de demanda



Fuente: Parkin, M. (2009)

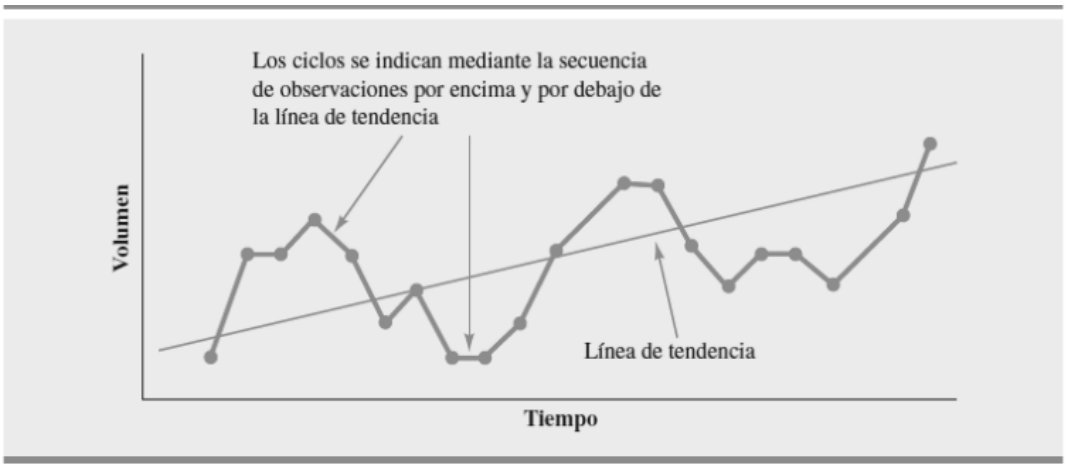
2.2.6 Pronóstico

Según Villarreal F. (2016) el pronóstico es una estimación cuantitativa y cualitativa de una o más variables que componen un hecho a futuro, utilizando como base datos actuales o históricos con referencia al tema a pronosticar.

Para pronosticar se puede utilizar diversos métodos estadísticos según el tipo de datos:

- Modelo cuantitativo: Para poder utilizar este modelo, es necesario contar con datos históricos de la variable en estudio que permitan generar predicciones, dicha data tiene que ser cuantificable y adaptable a modelos que permitan analizar patrones, tales como series de tiempo.
- Modelo cualitativo: Para poder desarrollar pronósticos de este tipo de variables, es primordial el juicio experto, los cuales ayudan a guiar sobre el tratamiento de datos, teniendo en cuenta que la información que se tiene de la variable no es cuantificable y no es abundante.

Figura 59: Modelo de pronóstico cuantitativo.



Fuente: Villarreal, F. (2016)

2.2.7 Energía Eléctrica

Según Orza Couto (2021) la energía eléctrica es un tipo de energía el cual interviene el desplazamiento de electrones entre 2 puntos cuando se manifiesta un potencial diferencial entre ellos, esto genera la llamada corriente eléctrica. Esta energía es importante para nuestra vida y para nuestro día a día debido a su alta conveniencia y versatilidad.

2.2.7.1 Ventajas de la energía eléctrica

- Accesible y fácil de producir: Se genera a partir de múltiples fuentes de energía tanto renovable como no renovable.
- Manejable y de fácil transportar: Esta energía puede ser transportada a gran escala y distancias mediante redes de transmisión y distribución.
- Versátil y fácil de transformar: Esta energía se puede transformar en otro tipo de energía como luz, calor o movimiento.

2.2.7.2 Coste de la Energía

El costo de la energía es establecido por la empresa eléctrica que suministra la energía teniendo como factores el consumo de la energía eléctrica y el precio unitario de KWH en sole. Podemos calcular el costo de energía consumida de la siguiente manera:

$$\text{Costo } (S/.) \;=\; \text{Energía consumida } (kw.h) \;*\; \text{precio } \frac{S./}{kw.h})$$

Capítulo III: Entorno Empresarial

3.1 Descripción de la empresa

3.1.1 Reseña histórica y actividad económica

Nexa Resources S.A. es una empresa metalúrgica con orígenes en Brasil cuyo propietario es el grupo Votorantim S.A., en 1956 se funda la "Companhia Mineira de Matais" (CMM). donde se inicia la investigación de un yacimiento de zinc en el Municipio de Vazante (Minas Gerais).

En el año 2004 Votorantim Metais inicia un proceso de expansión y posicionamiento en América Latina, adquiriendo la concesión de la unidad metalúrgica de zinc de Cajamarquilla en Perú, dicha unidad cuenta con una capacidad de producción de 160,000 toneladas de zinc por año. Posteriormente, en el año 2005 Votorantim S.A. aumenta su participación en el mercado peruano de zinc adquiriendo el 24.9% de las acciones de MILPO, la cuarta mayor minera del Perú, gracias a ello en 2006 comenzó las operaciones de Chapi, mina cuprífera localizada en Moquegua, 2007 comienzan las operaciones de Cerro Lindo, mina polimetálica ubicada en Chincha-Ica, 2008 nace Atacocha, otra mina polimetálica ubicada en Pasco. Los planes no quedaron ahí, ya que en el año 2010 Votorantim Metais se hace cargo del control mayoritario de MILPO, que en ese entonces era la tercera minera de zinc más grande del Perú, todo esto mediante su unidad en Cajamarquilla que aumenta su producción a 330,000 toneladas de zinc por año.

En el año 2014 se inicia un proceso de reestructuración, se crea el Holding Votorantim S.A. que asume el rol de orientadora y gestora de portafolios, mientras que Votorantim Metais gana mayor autonomía. Luego, en 2016 Votorantim Metais con sus nuevas atribuciones autónomas amplía su participación y control de MIPLO, adquiriendo el 80.24% de sus acciones, enfocándose en la producción de zinc en Brasil y Perú.

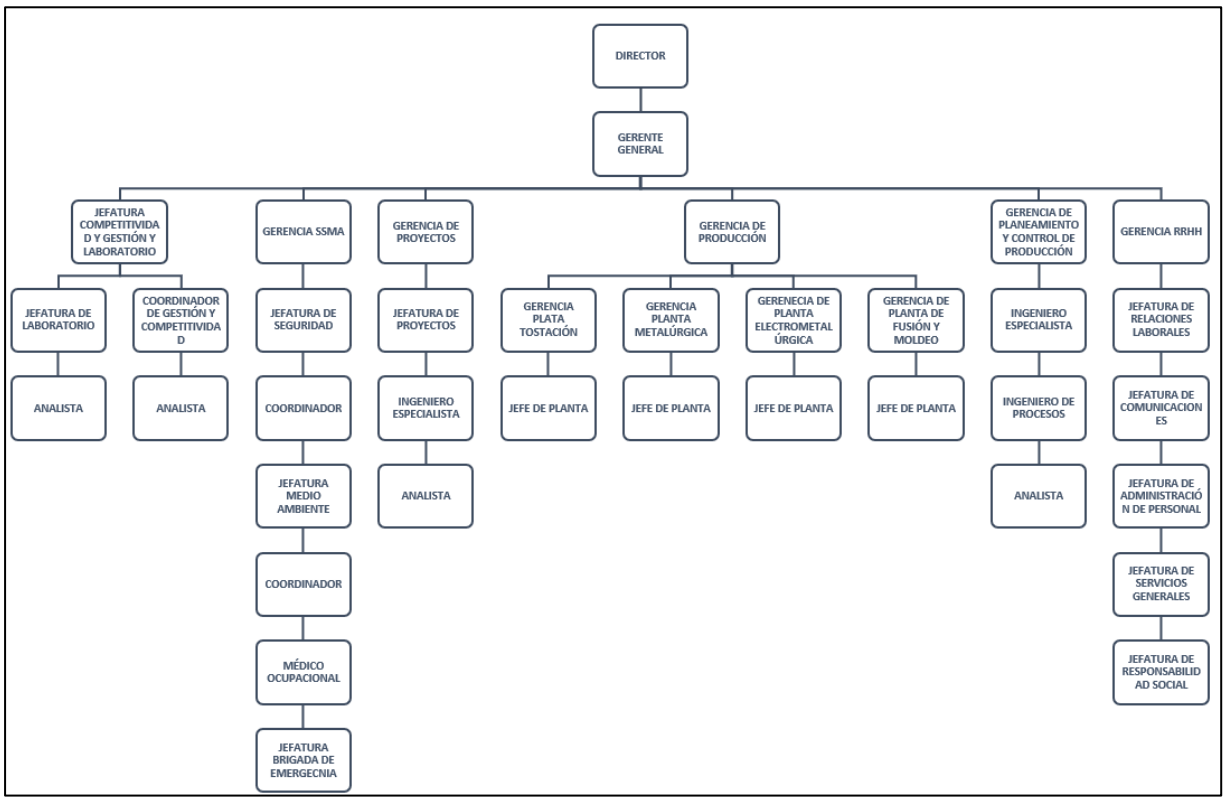
Finalmente, en el año 2017 Votorantim Metais y MIPLO se fusionan y se crea una nueva entidad llamada Nexa Resources S.A., como se mencionó en el 2004 Nexa Resources adquiere la refinería de Cajamarquilla, creándose Nexa Resources Cajamarquilla S.A., que según (WOOD Mackenzie, 2017), un grupo de investigación y consultoría global, la refinería de Cajamarquilla es la única operación de fundición de zinc en el Perú y la séptima más grande del mundo contando con altos estándares de calidad y tecnología.

3.1.2 Descripción de la organización

3.1.2.1 Organigrama

Nexa Resources S.A. cuenta con una cantidad amplia de personal directo como personal de empresas terceras. En el presente organigrama consideramos los puestos del personal de trabajo directo y resaltamos las áreas más importantes de la empresa.

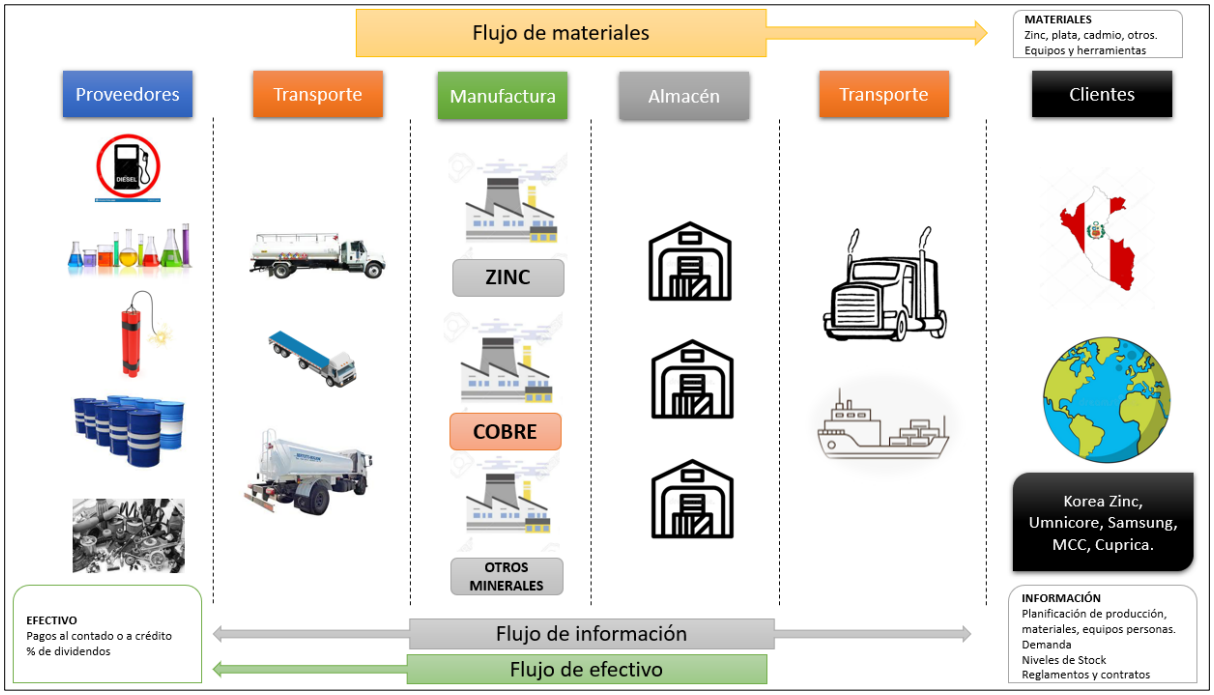
Figura 60: Organigrama



Fuente: Nexa Resources (2022)

3.1.2.1 Cadena de suministros

Figura 61: Cadena de Suministro Nexa Resources.



Fuente: Elaboración propia (2022)

3.1.3 Datos generales estratégicos de la empresa

3.1.3.1 Visión, misión y valores o principios

Visión: Ser una empresa reconocida a nivel mundial como inteligente y confiable, que crece en minería de zinc y cobre.

Misión: Somos una compañía minera global que produce zinc cobre, plomo y otros minerales esenciales presentes en nuestra vida.

Valores:

- Integridad: La empresa se preocupa por ser éticos en cualquier parte del mundo, honrar su historia y construir el futuro con respeto. La confianza y el respeto por las individualidades y las diferencias es lo que los acerca y les permite evolucionar. La integridad recompensa, impulsa y sostiene el éxito de su negocio.
- Colaboración: En Nexa, creen en el diálogo constante y constructivo entre las personas, el mercado y la sociedad. Se guían por la marca que sus acciones tendrán para la sociedad. Su continuidad sólo tiene sentido porque generan valor

compartido de forma sostenible. Un ecosistema de colaboración y crecimiento, que valora a las personas, permite la divergencia de ideas y los acerca cada vez más a sus clientes.

- Valor: La compañía asume la responsabilidad e impulsa los resultados para construir el futuro. Saben que la prosperidad de sus negocios proviene de sus más nobles intenciones y los resultados que logran. Por eso, inspiran con el ejemplo y trabajan con dedicación, buscando continuamente llegar más lejos. Actúan, innovan y son valientes para pensar y actuar de manera diferente siempre que sea necesario.

3.1.3.2 Objetivos estratégicos

- Ser una empresa innovadora, conectada, que asigna sus recursos de manera eficiente y gestiona los riesgos inherentes a nuestro negocio.
- Operar de manera sostenible y segura, dejando un legado relevante en los lugares donde operan.
- Incrementar la rentabilidad económica y financiera.
- Aumentar la utilización de la capacidad instalada mediante mejoras en la recuperación de zinc y eliminación de cuellos de botella.
- Aumentar la productividad y optimización de costos.
- Racionalizar el capital.
- Lograr estabilidad operativa mediante el mantenimiento de prácticas de gestión, automatización, reducción de medidas de emergencia y mejora del mantenimiento preventivo.
- Reducir el consumo de agua y promulgar el aumento de la recirculación.
- Reducir la huella de residuos.
- Reducir las emisiones (gases de efecto invernadero CO_2 equiv.).
- Reducir la tasa de frecuencia de lesiones. Cero muertes.
- Incrementar la estabilidad operacional con calidad.
- Maximizar la producción de Zinc a 327,000 Toneladas por año.
- Disminuir costos de producción.

3.1.3.3 Evaluación interna y externa. FODA cuantitativo

Tabla 5: Matriz FODA

	Fortalezas		**Oportunidades**
F1	Sólida situación económica y financiera: alta liquidez y bajo endeudamiento.	**O1**	Tendencia creciente favorable del precio de los metales, de Zinc, Plomo y Plata.
F2	Entre los 5 productores mundiales de concentrados de zinc, e importante participación en plomo, plata y cobre.	**O2**	Fomento del gobierno peruano en inversiones mineras
F3	Experiencia, respaldo y estrategia del grupo económico Votorantim (Ahora Nexa Resources S.A.A.) además de ser la única operación de fundición de Zinc en el Perú y la 7ma más grande en el mundo	**O3**	Disponibilidad de innovaciones tecnológicas en la industria minera
F4	Mindset de costos como estrategia de valor para el negocio (Manera Nexa)	**O4**	Incremento de la demanda de metales en el mundo como consecuencia del crecimiento mundial de la población
F5	Grado de inversión, otorgado por Fitch Rating (BBB-)	**O5**	Diversificada cartera de proveedores
F6	Implementación de las normas OHSAS 18 001, ISO 1 4001 e ISO 9 001.	**O6**	Presencia en zonas geográficas altamente mineras
F7	Integración vertical de la cadena productiva (minería y refinería).	**O7**	A partir de 2025, autorización legal para elevar la capacidad de extracción

			de minerales a superficie
F8	Cadena logística integrada (cercanía a minas y puertos).	**O8**	Minería como socio estratégico en la recuperación de la economía antes los efectos de la pandemia
	Debilidades		**Amenazas**
D1	Gran responsabilidad sobre procesos críticos a manos de terceros (contratistas)	**A1**	Posibles crisis económicas y desaceleración de economías importadoras de metales
D2	Requerimiento de altas inversiones para la exploración, desarrollo y proyectos mineros.	**A2**	Alta dependencia a variaciones en los precios de metales. Así como exposición en cambios en el Costo de tratamiento en las refinerías
D3	Estrategias de sinergias en procesos internos en las áreas y terceros en proceso, así como implementación de programas de compliance.	**A3**	Regulaciones y normativas cambiantes, y más exigentes en el país en el mediano plazo.
D4	Cambio de clima laboral	**A4**	Riesgo político - social del sector minero.
D5	Exposición a un menor treatment charge (TC).	**A5**	Cambios en leyes y marco regulatorio en el sector minero.
D6	Alta concentración de proyectos en un solo país (Perú)	**A6**	Conflictos sociales. Riesgos ambientales
D7	Alta concentración en explotación de zinc y plomo	**A7**	Aumento constante en los precios de los combustibles

D8	Falta de cobertura por riesgo en el tipo de cambio	**A8**	Escasez de concentrados a nivel global.

Fuente: Elaboración propia (2022)

Para obtener un resultado cuantitativo debemos identificar los factores endógenos (fortalezas y debilidades) y los factores exógenos (oportunidades y amenazas). Se debe puntuar en una escala del 1 al 7.

Se debe identificar si la fortaleza permite aprovechar la oportunidad. El número 1 es el puntaje más bajo que representa un bajo aprovechamiento y el número 7 es el puntaje más alto que representa un buen aprovechamiento. Asimismo, se debe identificar si las fortalezas permiten enfrentar las amenazas puntuando con un 7 un buen aporte y un 1 un bajo aporte.

Además, se debe identificar si la debilidad no permite aprovechar la oportunidad. Un puntaje 7 permite que la debilidad interfiera en el aprovechamiento de la oportunidad y un puntaje de 1, por el contrario, no interfiere en nada para aprovechar la oportunidad presente.

Enfocándonos en la debilidad, si esta permite que se active la amenaza se puntuará con un máximo de 7, mientras que un puntaje de 1 quiere decir que la debilidad no permite que se active la amenaza.

Para esta investigación, los integrantes del grupo realizaron un análisis cuantitativo al FODA para determinar las conclusiones.

Tabla 6: FODA cuantitativo

		OPORTUNIDADES									AMENAZAS								
		O1	O2	O3	O4	O5	O6	O7	O8	Promedio	A1	A2	A3	A4	A5	A6	A7	A8	Promedio
FORTALEZAS	F1	6.0	6.0	5.7	5.3	5.0	3.7	4.7	4.7	**5.1**	6.0	5.3	4.3	4.3	3.0	3.3	4.7	4.3	**4.4**
	F2	5.7	5.3	4.0	5.3	4.3	5.3	5.7	5.0	**5.1**	3.7	3.7	3.3	4.0	3.0	2.3	2.3	4.0	**3.3**
	F3	4.3	5.3	4.3	4.3	4.3	3.7	4.7	4.3	**4.4**	4.3	4.3	3.7	2.7	3.0	3.3	2.3	3.7	**3.4**
	F4	4.0	4.0	3.7	4.0	4.0	3.0	3.0	3.7	**3.7**	4.0	4.3	3.3	3.0	2.7	2.7	3.3	3.7	**3.4**
	F5	6.3	5.7	5.7	5.0	4.3	4.3	4.3	3.3	**4.9**	5.3	5.0	4.0	3.3	4.0	3.3	3.0	3.7	**4.0**
	F6	2.7	5.3	3.3	3.0	3.7	4.3	4.7	5.0	**4.0**	2.7	2.7	4.7	3.0	4.3	4.7	2.0	2.0	**3.3**
	F7	3.0	4.0	3.7	4.0	3.7	3.0	3.7	4.0	**3.6**	3.3	3.7	2.3	2.7	2.0	2.7	2.7	3.0	**2.8**
	F8	4.7	5.3	4.0	5.3	4.7	5.7	5.3	5.3	**5.0**	2.3	3.7	2.0	2.0	2.3	3.0	4.3	2.3	**2.8**
	Promedio	**4.6**	**5.1**	**4.3**	**4.5**	**4.3**	**4.1**	**4.5**	**4.4**		**4.0**	**4.1**	**3.5**	**3.1**	**3.0**	**3.2**	**3.1**	**3.3**	
DEBILIDADES	D1	5.7	5.3	5.3	5.0	3.7	4.0	5.3	4.3	**4.8**	6.0	5.7	5.7	5.3	5.0	5.0	5.3	5.0	**5.4**
	D2	3.0	2.7	3.7	3.7	3.7	2.7	3.7	3.0	**3.3**	3.3	4.7	5.0	4.3	4.0	4.0	3.7	3.0	**4.0**
	D3	2.0	3.3	3.3	3.3	3.0	3.0	3.0	3.0	**3.0**	4.3	4.3	5.0	4.3	5.0	4.0	3.3	4.3	**4.3**
	D4	5.7	4.0	3.3	5.3	4.3	3.3	4.3	4.7	**4.4**	5.7	6.0	4.0	4.7	3.7	3.3	5.7	4.3	**4.7**
	D5	3.7	3.7	2.7	3.3	3.7	2.3	3.0	3.3	**3.2**	4.0	4.0	4.3	4.0	4.3	3.7	3.0	3.3	**3.8**
	D6	4.0	3.0	4.0	4.7	3.0	4.0	3.3	3.7	**3.7**	6.0	5.3	6.0	5.3	5.0	5.7	6.0	5.3	**5.6**
	D7	3.7	4.0	3.3	4.0	3.0	3.7	3.3	3.3	**3.5**	5.7	4.7	4.7	5.0	5.0	4.0	4.0	5.0	**4.8**
	D8	5.0	4.0	4.0	4.0	3.3	3.7	3.7	3.7	**3.9**	5.3	4.7	4.0	3.7	4.0	3.7	4.7	4.7	**4.3**
	Promedio	**4.1**	**3.8**	**3.7**	**4.2**	**3.5**	**3.3**	**3.7**	**3.6**		**5.0**	**4.9**	**4.8**	**4.6**	**4.5**	**4.2**	**4.5**	**4.4**	

Fuente: Elaboración propia (2022)

Como conclusiones del análisis de FODA cuantitativo se deduce que las fortalezas 1: Sólida situación económica y financiera y fortaleza 2; Entre los 5 productores mundiales de concentrados de zinc, e importante participación en plomo, plata y cobre son los más importantes para aprovechar las oportunidades mencionadas. Por el contrario, la fortaleza 7: Integración vertical de la cadena productiva (minería y refinería) es la que menos impacta en lograr aprovechar las oportunidades.

Asimismo, se debe aprovechar la oportunidad 2: Fomento del gobierno peruano en inversiones mineras ya que ha obtenido el mayor puntaje frente a las otras oportunidades descritas y se pueden obtener grandes beneficios.

También, podemos observar que la fortaleza 1: Sólida situación económica y financiera permite enfrentar las amenazas con mayor éxito y la fortaleza 5: Grado de inversión, otorgado por Fitch Rating (BBB-) también debería considerarse ya que obtuvo una puntuación en segundo lugar.

Respecto a las debilidades, la debilidad 1: Gran responsabilidad sobre procesos críticos a manos de terceros (contratistas) no nos permite aprovechar las oportunidades y es en la que se debe poner más el foco. Asimismo, la debilidad 6: Alta concentración de proyectos en un

solo país (Perú) puede permitir que se activen las amenazas por lo que debemos elaborar estrategias que eviten consecuencias para la empresa Nexa Resources.

3.2 Modelo de negocio actual (CANVAS)

A través del modelo CANVAS podemos tener una mejor visión del modelo de negocio de Nexa Resources:

Tabla 7: Modelo de Negocio Actual.

Fuente: Elaboración Propia (2022)

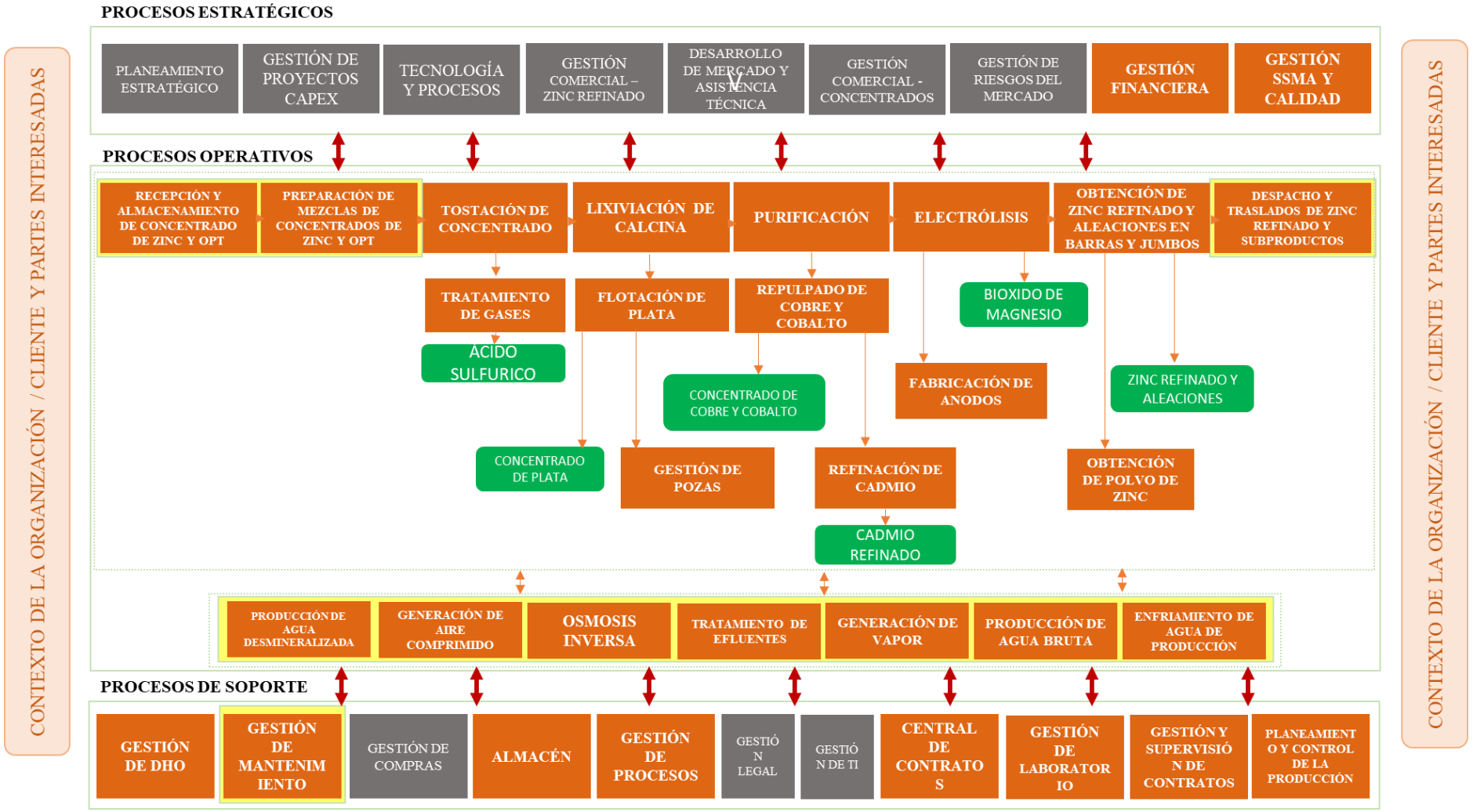
3.3 Mapa de procesos actual

Se presenta gráficamente los procesos de la empresa Nexa Resources Cajamarquilla clasificados por procesos estratégicos, operativos y de soporte donde se observan las diferentes áreas de la empresa, la interrelación entre ellas, los procesos tercerizados y los productos para el cliente externo.

Figura 62: Leyenda de Mapa de procesos

Fuente: Nexa Resources Cajamarquilla (2022)

Figura 63: Mapa de procesos de Nexa Resources Cajamarquilla



Fuente: Nexa Resources Cajamarquilla (2022)

Capítulo IV: Metodología de la Investigación

4.1 Diseño de la Investigación

4.1.1 Enfoque de la investigación

El enfoque de la presente investigación es cuantitativo, ya que se busca desarrollar herramientas que permitan identificar y realizar la medición de atributos del objeto de estudio, en nuestro caso, la demanda máxima de energía eléctrica, donde también se utilizan fórmulas y evaluaciones estadísticas que permitan optimizar el modelo propuesto para el tratamiento de datos.

4.1.2 Alcance de la investigación

El alcance de la investigación se enfoca en un análisis correlacional entre 2 o más variables, en nuestro caso, la variable dependiente es la demanda eléctrica y las variables independientes son el tiempo en tipo de día, día y hora de estudio. Además, se realiza el análisis de correlación de las predicciones obtenidas a partir del modelo, considerando los algoritmos de series de tiempo para dicho análisis.

4.1.3 Tipo de investigación

Se trabaja con un diseño experimental, debido al tratamiento y estudio de una o más variables, basado en el análisis relacional entre las variables seleccionadas con respecto a los aumentos o disminuciones de consumo para poder determinar las demandas máximas de acuerdo con el día, la hora y el tipo de día.

4.1.4 Población y muestra

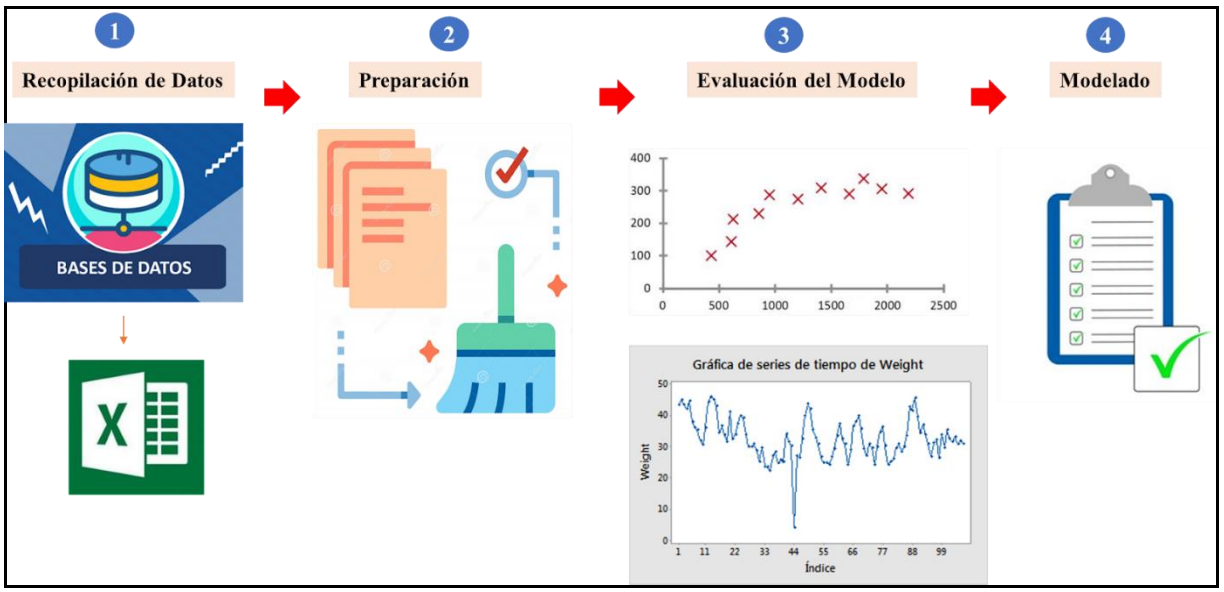
Tabla 8: Población y muestra de la investigación.

Población	**Muestra**
Consumo de energía eléctrica desde enero del 2017 a diciembre del 2021	Consumo de energía eléctrica desde el 01 enero del 2017 al 15 de marzo del 2020

Fuente: Elaboración propia (2022)

4.2 Metodología de implementación de la solución

Figura 64: Metodología de implementación de la solución.



Fuente: Elaboración propia (2022)

4.2.1 Recopilación de datos

Los datos utilizados para el estudio fueron solicitados a la empresa Nexa Resources Cajamarquilla. Estos datos son relevantes para nuestra investigación ya que presentan información histórica acerca del consumo de energía por fecha (año, mes, día) tomados en intervalos de cada 30 minutos. Además, según lo estudiado en el curso de Machine Learning y lo consultado en diversas literaturas, se podrá obtener un resultado más preciso al contar con mayor data, por lo que se está considerando un universo desde el año 2017 hasta 2021.

Para facilitar el análisis de esta información se ha unificado los datos en un solo archivo que nos servirá de base para la siguiente etapa.

4.2.2 Pre-Procesamiento

En esta fase realizamos el análisis de la base de datos obtenida y se procede a realizar una limpieza, por ejemplo: eliminar datos ausentes, outliers y/o valores incorrectos que puedan originar un resultado ajeno a la realidad.

4.2.3 Modelado

En esta fase se realiza el entrenamiento de los algoritmos de Machine Learning, buscando obtener el modelo que mejor se ajuste a obtener el resultado de la problemática presentada en los capítulos anteriores. Asimismo, la programación se realizará con Phyton para poder encontrar los días de demanda máxima de energía.

- Series de Tiempo: En el capítulo II, se detallan los conceptos de esta técnica. Además, es importante resaltar que se utilizará la librería: numpy, pandas.

4.2.4 Evaluación del Modelo

En esta fase, se valida que el modelo elegido se ajuste a la realidad comparando los datos ya existentes. Se realizan varias pruebas a fin de comprobar que el modelo sea de utilidad para poder pronosticar los días con altos consumos de energía.

4.3 Metodología para la medición de resultados de la implementación

En el presente trabajo de investigación, implementaremos y evaluaremos las siguientes métricas, buscando el mejor modelo de predicción que se ajuste a la demanda máxima de energía en Nexa Cajamarquilla. Se adjunta las tablas de los dos modelos empleados, Regresión Lineal y series de Tiempo.

Tabla 9: Metodología para la medición de resultados (REGRESIÓN LINEAL).

REGRESIÓN LINEAL				
VARIABLE		DESCRIPCIÓN	RESPONSABLE DE VARIABLE	MÉTRICAS
INDEPENDIENTE	X: FECHA Y HORA	VARIABLE QUE INDICA LA HORA Y FECHA CADA MEDIA HORA.	ÁREA DE PLANEAMIENTO Y CONTROL DE LA PRODUCCIÓN	MAE: MEAN ABSOLUTE ERROR. R2: VARIANZA. R: DESVIACIÓN ESTÁNDAR.
DEPENDIENTE	Y: DEMANDA DE CONSUMO DE ENERGÍA	CONSUMO DE ENERGÍA ELÉCTRICA	ÁREA DE PLANEAMIENTO Y CONTROL DE LA PRODUCCIÓN	DEMANDA DE ENERGÍA EN KW.

Fuente: Elaboración propia (2022)

Tabla 10: Metodología para la medición de resultados (SERIES DE TIEMPO)

SERIES DE TIEMPO				
VARIABLE		DESCRIPCIÓN	RESPONSABLE DE VARIABLE	MÉTRICAS
INDEPENDIENTE	X: FECHA Y HORA	VARIABLE QUE INDICA LA HORA Y FECHA CADA MEDIA HORA.	ÁREA DE PLANEAMIENTO Y CONTROL DE LA PRODUCCIÓN	ERROR DEL BACKTEST
DEPENDIENTE	Y: DEMANDA DE CONSUMO DE ENERGÍA	CONSUMO DE ENERGÍA ELÉCTRICA	ÁREA DE PLANEAMIENTO Y CONTROL DE LA PRODUCCIÓN	DEMANDA DE ENERGÍA EN KW.

Fuente: Elaboración propia (2022)

4.4 Cronograma de actividades y presupuesto

La presente investigación tendrá una duración de 4.5 meses de los cuales, un mes y medio ya ha sido ejecutado. A continuación, se detalla el cronograma que fue empleado para llevar a cabo el presente trabajo.

Tabla 11: Cronograma de actividades

TRABAJO DE SUFICIENCIA PROFESIONAL						JUNIO				JULIO					AGOSTO				SEPTIEMBRE				OCTUBRE			
						4-Jun	11-Jun	18-Jun	25-Jun	2-Jul	9-Jul	16-Jul	23-Jul	30-Jul	6-Ago	13-Ago	20-Ago	27-Ago	3-Set	10-Set	17-Set	24-Set	1-Oct	8-Oct	15-Oct	22-Oct
Actividades	Status	Inicio Previsto	Inicio Real	Término Previsto	Término Real	S 1	S 2	S 3	S 4	S 1	S 2	S 3	S 4	S 5	S 1	S 2	S 3	S 4	S 1	S 2	S 3	S 4	S 1	S 2	S 3	S 4
CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA	COMPLETADO	25/06/2022	25/06/2022	2/07/2022	2/07/2022																					
Descripción de la Realidad Problemática	COMPLETADO	25/06/2022	25/06/2022	2/07/2022	2/07/2022																					
Justificación de la Investigación	COMPLETADO	25/06/2022	25/06/2022	2/07/2022	2/07/2022																					
Delimitación de la Investigación	COMPLETADO	25/06/2022	25/06/2022	2/07/2022	2/07/2022																					
CAPÍTULO II: MARCO TEÓRICO	COMPLETADO	2/07/2022	23/07/2022	9/07/2022	30/07/2022																					
Antecedentes de la Investigación	COMPLETADO	2/07/2022	23/07/2022	9/07/2022	30/07/2022																					
Bases Teóricas	COMPLETADO	2/07/2022	23/07/2022	9/07/2022	30/07/2022																					
CAPÍTULO III: ENTORNO EMPRESARIAL	COMPLETADO	9/07/2022	2/07/2022	16/07/2022	9/07/2022																					
Descripción de la empresa	COMPLETADO	9/07/2022	2/07/2022	16/07/2022	9/07/2022																					
Modelo de negocio actual	COMPLETADO	9/07/2022	2/07/2022	16/07/2022	9/07/2022																					
Mapra de procesos actual	COMPLETADO	9/07/2022	2/07/2022	16/07/2022	9/07/2022																					
CAPÍTULO IV: METODOLOGÍA DE LA INVESTIGACIÓN	COMPLETADO	16/07/2022	23/07/2022	23/07/2022	30/07/2022																					
Diseño de la Investigación	COMPLETADO	16/07/2022	23/07/2022	23/07/2022	30/07/2022																					
Metodología de implementación de la solución	COMPLETADO	16/07/2022	23/07/2022	23/07/2022	30/07/2022																					
Metodología para la medición de resultados de la implementación	COMPLETADO	16/07/2022	23/07/2022	23/07/2022	30/07/2022																					
Cronograma de actividades y presupuesto	COMPLETADO	16/07/2022	23/07/2022	23/07/2022	30/07/2022																					
CAPÍTULO V: DESARROLLO DE LAS SOLUCIÓN	COMPLETADO	23/07/2022	23/07/2022	30/07/2022	30/07/2022																					
Propuesta solución	COMPLETADO	23/07/2022	23/07/2022	30/07/2022	30/07/2022																					
Medición de la solución	COMPLETADO	23/07/2022	23/07/2022	30/07/2022	30/07/2022																					
CIERRE	COMPLETADO	23/07/2022	23/07/2022	30/07/2022	30/07/2022																					
Correción del informe	COMPLETADO	6/08/2022	6/08/2022	27/08/2022	20/08/2022																					
Sustentación	COMPLETADO	22/10/2022	22/10/2022	22/10/2022	22/10/2022																					

Fuente: Elaboración propia (2022)

Para el desarrollo de esta investigación se realizó un presupuesto estimado de gastos de acuerdo con lo que cada integrante empleó como herramienta de trabajo. Cabe mencionar, que los montos son aproximados y que no necesariamente se realizaron los gastos exclusivamente para este trabajo, es decir, algunos gastos, como las laptops, por ejemplo, no fueron realizados antes de la ejecución de la investigación, sino que estos fueron adquiridos mucho tiempo antes, pero se ha considerado como si fueran gastos recientes para estimar lo que costaría realizar esta investigación sin ninguna herramienta previa.

Tabla 12: Presupuesto de la investigación

Recurso	Cantidad	P.U.	Total
Gastos Fijos			
Equipo: Laptop	5.00	S/ 2,500.00	S/ 12,500.00
Software: Jupyter Notebook	5.00	S/ 0.00	S/ 0.00
Zoom	5.00	S/ 0.00	S/ 0.00
MS Teams	5.00	S/ 0.00	S/ 0.00
Gastos Variables			
Electricidad	5.00	S/ 60.00	S/ 300.00
Internet	5.00	S/ 100.00	S/ 500.00
		TOTAL	**S/ 13,300.00**

Fuente: Elaboración propia (2022)

Capítulo V: Desarrollo de la Solución

5.1 Propuesta solución

El presente trabajo tiene como objetivo desarrollar un modelo que se ajuste lo máximo posible al comportamiento que ha tenido la demanda de energía de la planta Nexa Cajamarquilla a lo largo del año 2017 al 2021, con la finalidad de predecir los picos máximos y determinar no solo los días sino también a qué hora específica se incrementará el uso de la energía y, de esta forma, anticiparnos en el tiempo permitiéndonos dosificar la energía empleada y re-distribuirla en días anteriores.

Dentro del alcance contemplado en este trabajo, una ligera porción del ordenamiento de datos se llevará a cabo en Excel, sin embargo, el desarrollo de la solución se realizará estricta y completamente en el lenguaje de programación Python debido a la polivalencia de este para emplearse en distintos propósitos, ya sea graficar, codificar o modelar datos.

5.1.1 Planteamiento y descripción de Actividades

De acuerdo a lo mencionado en el capítulo 4.2, la primera etapa que debe de realizar es la de adquisición o recolección de data; seguido de la preparación de datos, etapa la cual se divide en dos fases: exploración y pre-procesamiento; luego se realiza el modelado de la información; y, por último, se emiten reportes del análisis y se determina la acción a realizarse para dar una solución al problema.

5.1.1.1 Adquisición de datos

La primera etapa de la metodología empleada consiste en recolectar datos que sean relevantes para el estudio. La recolección de datos no es más que el enfoque sistemático de reunir y medir información de diversas fuentes a fin de obtener un panorama completo y preciso de una zona de interés. Para la presente investigación, será necesario recolectar datos sobre nuestras variables, en este caso, información sobre la demanda de energía eléctrica y las fechas de dichas demandas. Así como también será necesario adquirir información sobre qué días fueron festivos durante ese periodo.

5.1.1.2 Preparación

5.1.1.2.1 Exploración. Esta etapa es importante para entender la naturaleza de los datos, entender el comportamiento de los mismos y hacer un análisis preliminar de la información

resultante. Para el presente trabajo se deberá analizar el comportamiento de la demanda a fin de determinar qué modelo será el más indicado para predecir la demanda máxima de energía. Se deberá hacer un análisis anual, mensual, semanal y diario para entender el comportamiento en cada escala de tiempo.

5.1.1.2.2 Pre-procesamiento. La etapa de pre-procesamiento consiste en hacer la limpieza, integración de la data, adaptación de valores y construir el dataset final que será empleado en la siguiente etapa de modelamiento. Para el presente análisis, se deberá acomodar la data de las fechas en un formato que el programador logre reconocer. Además, se deberá de hacer verificaciones para validar que la data esté completa, caso contrario, se deberá rellenar los campos vacíos.

5.1.1.3 Modelamiento

Esta fase de la metodología es la más importante debido a que es en esta parte donde se define el modelo final que nos permitirá predecir de la manera más precisa posible la demanda energética. Escoger una metodología que no se adecue correctamente podría inducir en un error considerable y por lo tanto, a un modelo que no agregue valor a la investigación. Es en esta etapa donde decidimos qué técnicas usar, cómo se va a diseñar, construir y evaluar el modelo. Para la presente investigación, se deberá evaluar si la metodología de regresión lineal será un modelo correcto y que se ajuste al comportamiento de la demanda o, si por el contrario, se deberá de emplear otra metodología como series de tiempo.

5.1.1.4 Evaluación del modelo.

En esta última etapa del proceso se evalúan los resultados obtenidos y se determina si será un modelo útil para implementarlo en la empresa y que este será relevante en la toma de decisiones de la organización. Se deberá analizar cuál modelo se ajusta mejor al comportamiento de la demanda y con cuál método permite predecir de manera más precisa los picos de demanda energética, ya sea por el método de regresión lineal o por el método de series de tiempo.

5.1.2 Desarrollo de actividades. Aplicación de herramientas de solución

5.1.2.1 Adquisición de datos

La recopilación de datos permite a un individuo o empresa responder a preguntas relevantes, evaluar los resultados y anticipar mejor las probabilidades y tendencias futuras.

Se obtuvo la data histórica del consumo de energía eléctrica empleada en la planta con un intervalo de 30 minutos por registro a partir del año 2017 hasta el presente año. Esta data fue proporcionada por el Comité de Operación Económica del Sistema Interconectado Nacional (COES). De igual forma, se logró adquirir data complementaria sobre los días que fueron festivos durante ese período, esto con la finalidad de ajustar el modelo e incrementar el grado de precisión.

En total se cuentan con 87648 datos sobre la demanda de la energía eléctrica de la planta Nexa Cajamarquilla y 87648 datos sobre las fechas y horas en las cuales ocurrió la cantidad de demanda. Por otro lado, también se logró obtener 103 datos correspondientes a los días que fueron festivos en el intervalo de tiempo mencionado.

Figura 65: Formato de data.

	A	B
1		
2	COES	
3		
4	FECHA	EJECUTAD
5	01/01/2017 00:30	5297.58565
6	01/01/2017 01:00	5204.42518
7	01/01/2017 01:30	5130.042
8	01/01/2017 02:00	5024.78366
9	01/01/2017 02:30	4906.16097
10	01/01/2017 03:00	4825.88566
11	01/01/2017 03:30	4796.79196
12	01/01/2017 04:00	4703.76515
13	01/01/2017 04:30	4669.9345
14	01/01/2017 05:00	4654.33931
15	01/01/2017 05:30	4590.42285
16	01/01/2017 06:00	4391.89862
17	01/01/2017 06:30	4194.41017
18	01/01/2017 07:00	4141.22966
19	01/01/2017 07:30	4143.53818
20	01/01/2017 08:00	4267.38769

DEMANDA FERIADOS

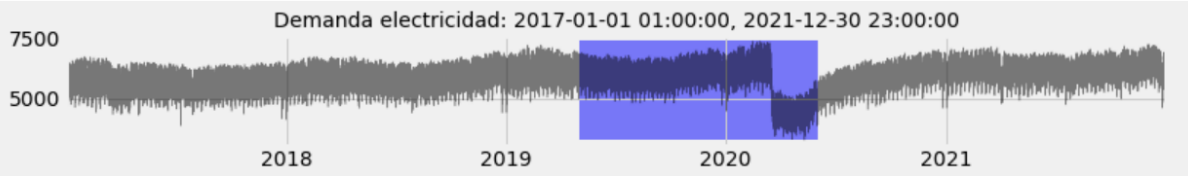
Fuente: Nexa Cajamarquilla S.A. (2022)

5.1.2.2 Preparación

5.1.2.2.1. Exploración. Para esta etapa del proceso de la data Science es necesario entender la naturaleza de los datos. Como se puede observar en la siguiente imagen, el comportamiento de la demanda eléctrica en la planta de Nexa Cajamarquilla ha tenido un

comportamiento estacional, exceptuando el año 2020 donde se evidencia una marcada disminución en la demanda producto por la pandemia.

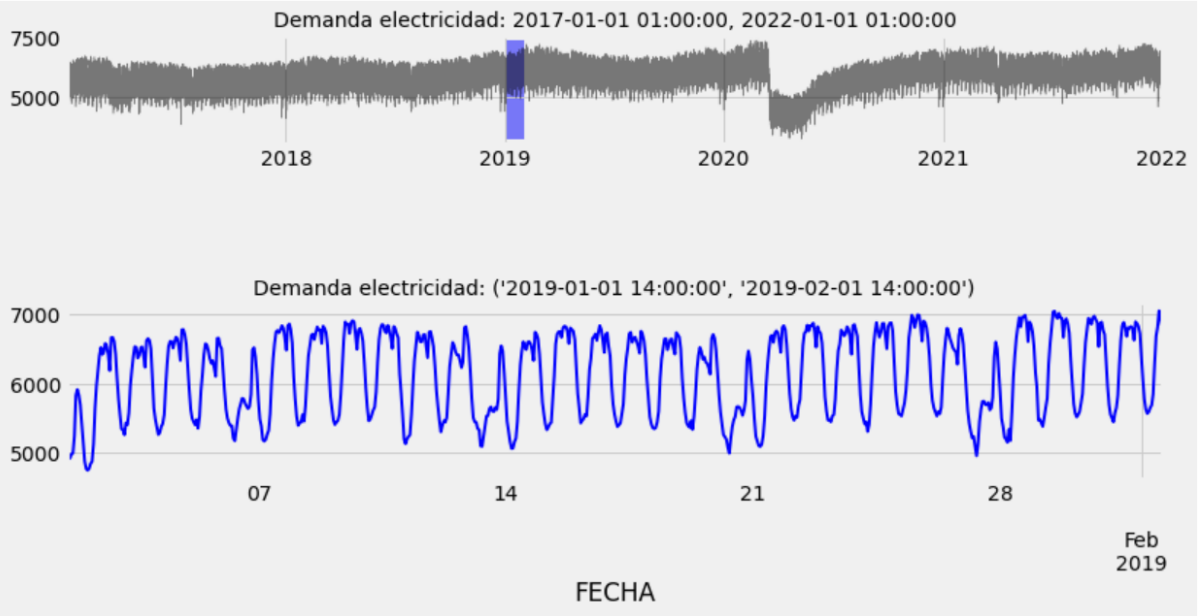
Figura 66: Comportamiento de la demanda eléctrica periodo 2017-2022



Fuente: Elaboración en Python con la librería matplotlib.pyplot (2022)

Se buscó analizar un mes en el cual la demanda haya tenido un comportamiento habitual. En la siguiente gráfica podemos observar la información del mes de febrero del 2019 en la cual se puede concluir que los días de menor demanda eléctrica se dan en los fines de semana.

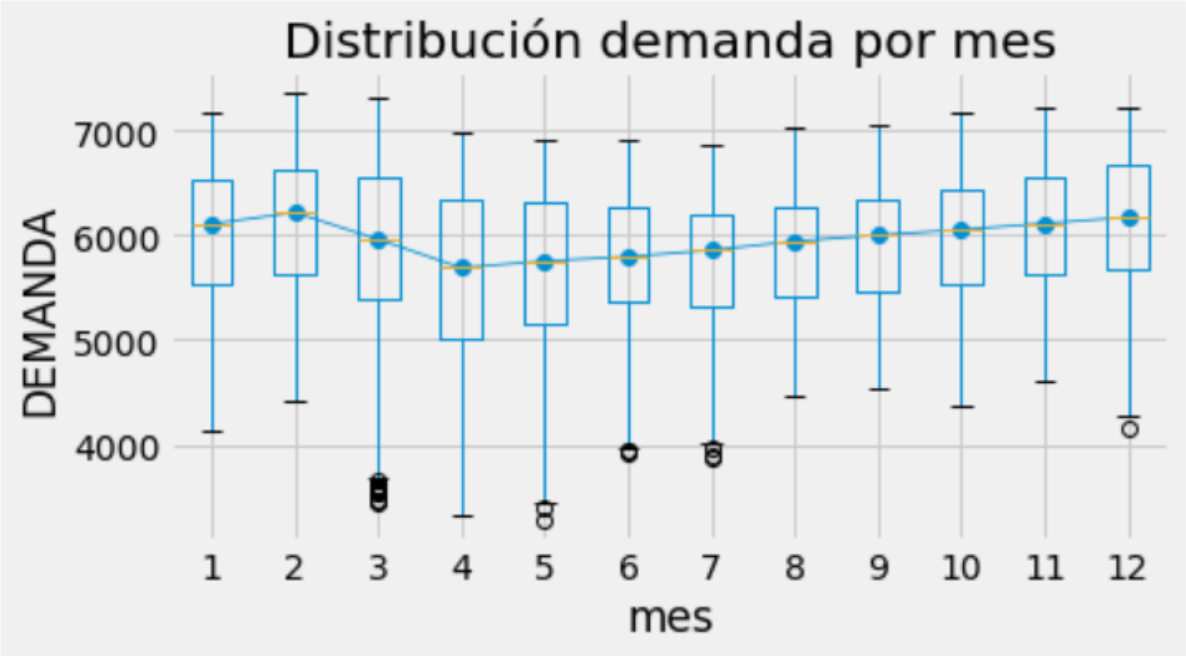
Figura 67: Zoom comportamiento de la demanda mes de febrero 2019.



Fuente: Elaboración en Python con la librería matplotlib.pyplot (2022)

Se realizó un análisis anual de la demanda, en la cual se pudo determinar que los meses de mayor demanda energética eran los meses de febrero y diciembre seguido por los meses de enero y noviembre.

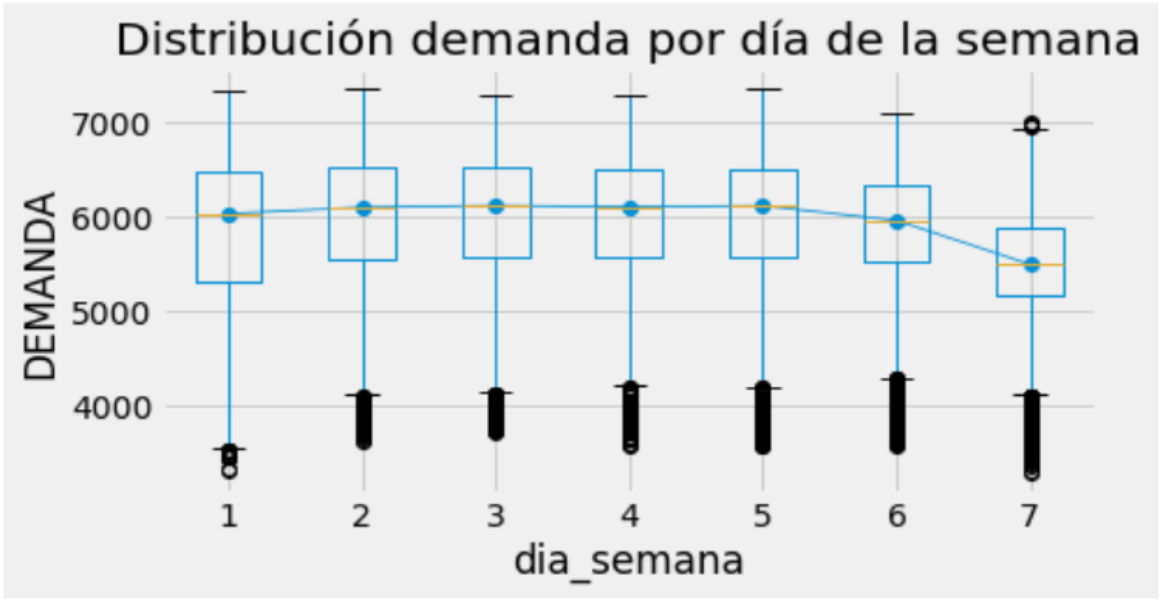
Figura 68: Distribución de la demanda por mes



Fuente: Elaboración en Python con la librería matplotlib.pyplot (2022)

Como mencionamos anteriormente, la demanda de energía disminuye los fines de semana, principalmente en los domingos donde en toda la data registrada nunca se ha tenido un pico de demanda. En la siguiente gráfica se puede observar claramente como en el día 7 la demanda disminuye considerablemente.

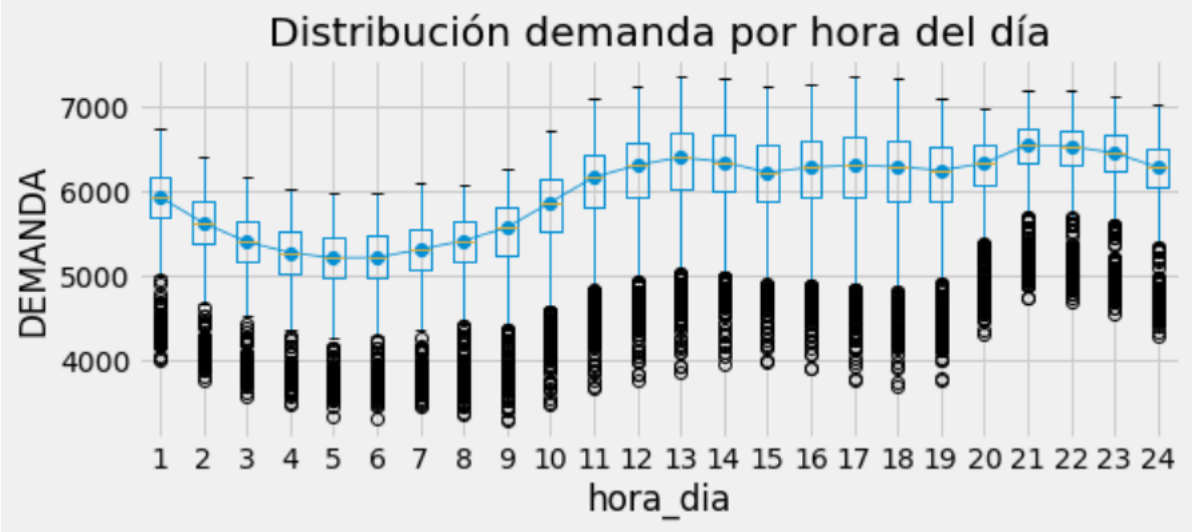
Figura 69: Distribución de la demanda por semana



Fuente: Elaboración en Python con la librería matplotlib.pyplot (2022)

Por otro lado, el objetivo del presente trabajo es predecir la demanda máxima no solo por el día sino también por la hora. En ese sentido, se analizó la información de un solo día y se concluyó que los picos de demanda se dan al medio día y a las 9 de la noche, tal y como se refleja en la siguiente gráfica.

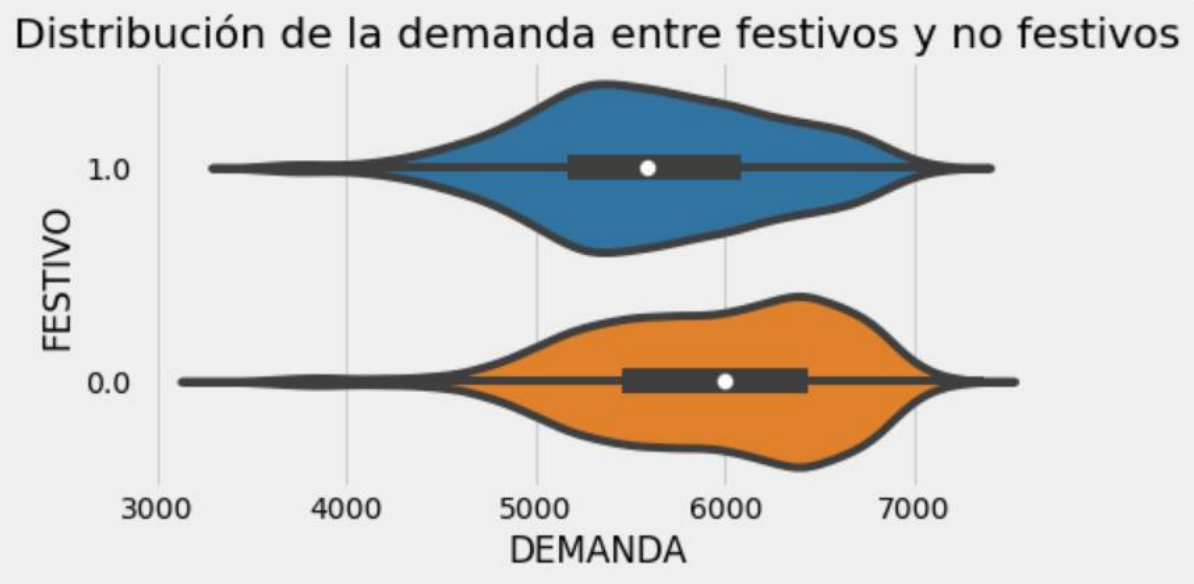
Figura 70: Distribución de la demanda por hora del día



Fuente: Elaboración en Python con la librería matplotlib.pyplot (2022)

Es necesario entender la diferencia entre el comportamiento de la demanda de un día normal y un día festivo, según la entrevista que pudimos obtener del experto metalurgista de la empresa Nexa Cajamarquilla, en un día festivo la demanda de energía disminuye considerablemente debido a que gran parte del personal no labora y por tanto el nivel de producción tiene que reducirse, y, en consecuencia, las máquinas que procesan el mineral no se sobrecargan de energía. A continuación, una gráfica de violín donde se muestra la diferencia entre el comportamiento de ambos días, donde 1 corresponde a los días festivos y 0 a los días no festivos.

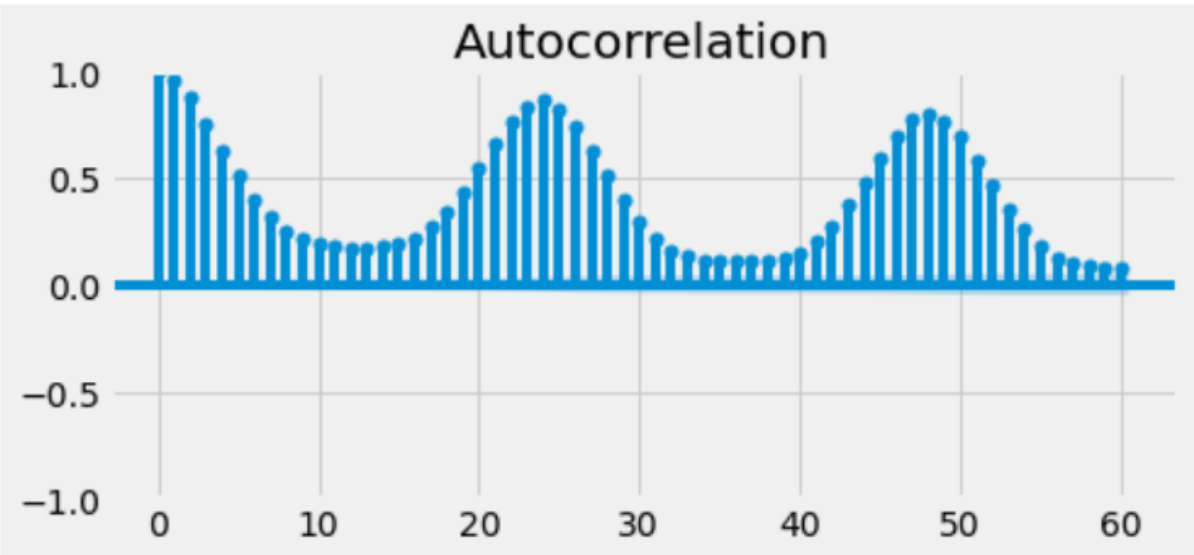
Figura 71: Distribución de la demanda entre días festivos y no festivos



Fuente: Elaboración en Python con la librería matplotlib.pyplot (2022)

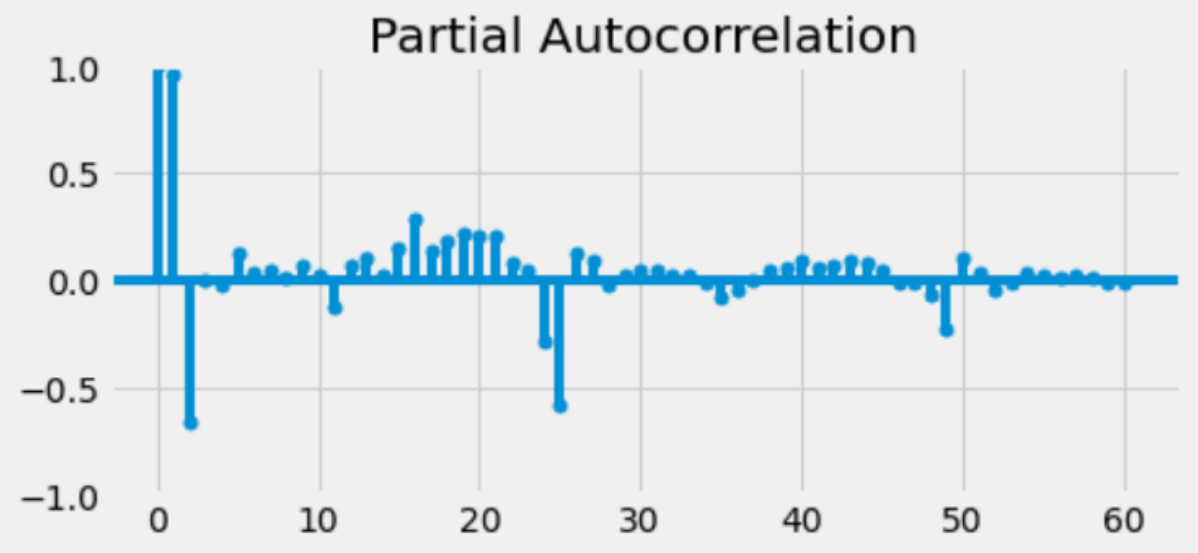
Los gráficos de autocorrelación y autocorrelación parcial muestran una clara asociación entre la demanda de una hora y las horas anteriores, así como entre la demanda de una hora y la demanda de esa misma hora los días anteriores. Este tipo de correlación es un indicativo de que los modelos autorregresivos pueden funcionar bien.

Figura 72: Gráfica de autocorrelación



Fuente: Elaboración en Python con la librería matplotlib.pyplot (2022)

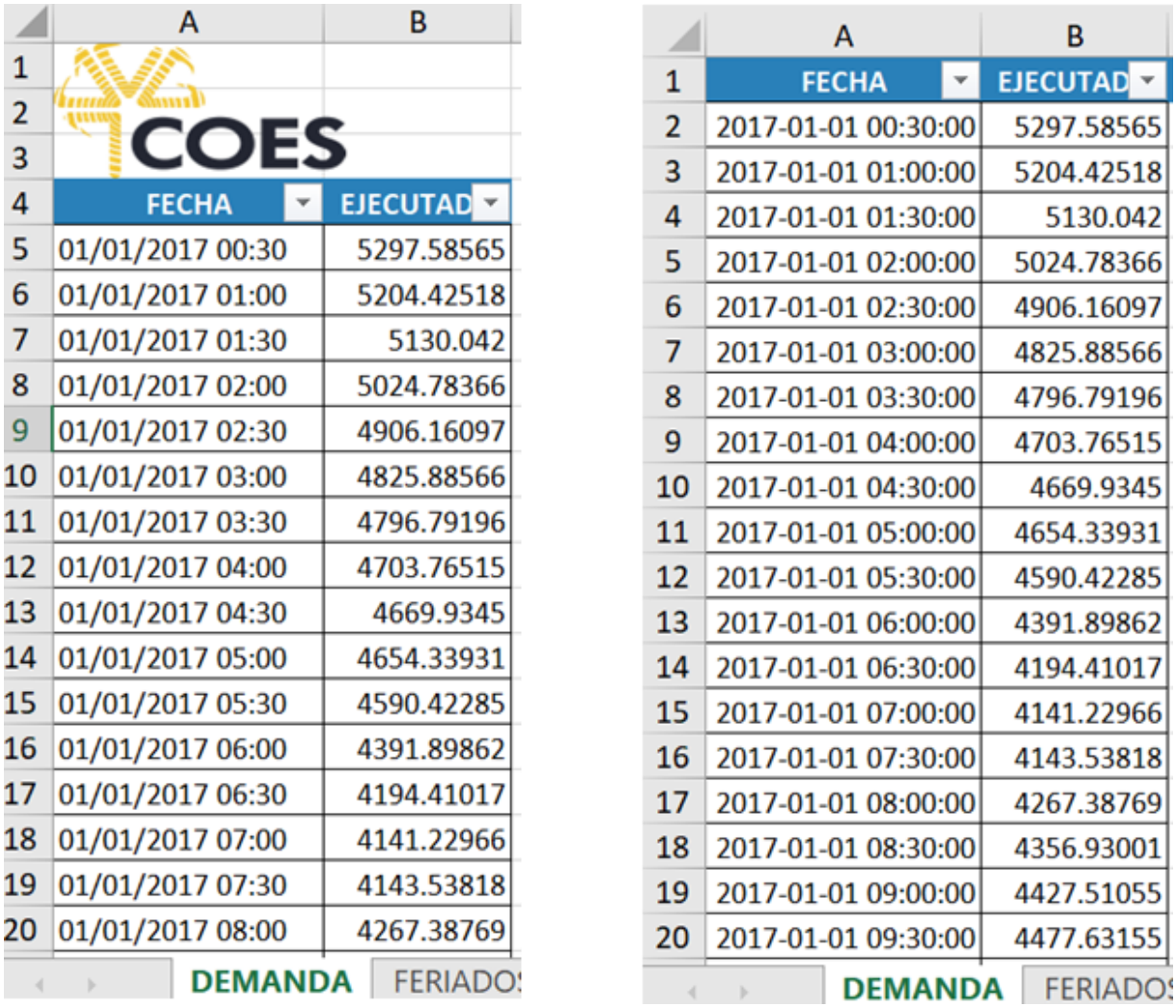
Figura 73: Gráfica de autocorrelación parcial



Fuente: Elaboración en Python con la librería matplotlib.pyplot (2022)

***5.1.2.2.2. Preprocesamiento*.** Como se mencionó anteriormente la base de datos que se cuenta consta de un total de 87648 filas. En esta fase de la metodología se buscó realizar la limpieza e integración de la data, así como la adaptación de valores. El primer paso que realizamos fue eliminar las filas de la 1 a la 3, las cuales no aportan ningún valor al modelo. Seguidamente se le dio un formato de preferencia a la fecha con la finalidad de retransmitir este formato a la programación. En la siguiente figura podemos observar cómo quedaron nuestros datos procesados.

Figura 74: Base de datos de demanda eléctrica de Nexa Resources



	A	B
1		
2		
3		
4	FECHA	EJECUTAD
5	01/01/2017 00:30	5297.58565
6	01/01/2017 01:00	5204.42518
7	01/01/2017 01:30	5130.042
8	01/01/2017 02:00	5024.78366
9	01/01/2017 02:30	4906.16097
10	01/01/2017 03:00	4825.88566
11	01/01/2017 03:30	4796.79196
12	01/01/2017 04:00	4703.76515
13	01/01/2017 04:30	4669.9345
14	01/01/2017 05:00	4654.33931
15	01/01/2017 05:30	4590.42285
16	01/01/2017 06:00	4391.89862
17	01/01/2017 06:30	4194.41017
18	01/01/2017 07:00	4141.22966
19	01/01/2017 07:30	4143.53818
20	01/01/2017 08:00	4267.38769

DEMANDA FERIADO

	A	B
1	FECHA	EJECUTAD
2	2017-01-01 00:30:00	5297.58565
3	2017-01-01 01:00:00	5204.42518
4	2017-01-01 01:30:00	5130.042
5	2017-01-01 02:00:00	5024.78366
6	2017-01-01 02:30:00	4906.16097
7	2017-01-01 03:00:00	4825.88566
8	2017-01-01 03:30:00	4796.79196
9	2017-01-01 04:00:00	4703.76515
10	2017-01-01 04:30:00	4669.9345
11	2017-01-01 05:00:00	4654.33931
12	2017-01-01 05:30:00	4590.42285
13	2017-01-01 06:00:00	4391.89862
14	2017-01-01 06:30:00	4194.41017
15	2017-01-01 07:00:00	4141.22966
16	2017-01-01 07:30:00	4143.53818
17	2017-01-01 08:00:00	4267.38769
18	2017-01-01 08:30:00	4356.93001
19	2017-01-01 09:00:00	4427.51055
20	2017-01-01 09:30:00	4477.63155

DEMANDA FERIADO

Fuente: Elaboración propia (2022)

De igual manera, se colocó una columna con título HORA en la cual se simplificó la hora que existe en la columna de FECHA asignándole un valor del 1 al 48 debido a la frecuencia de 30 minutos de los datos que hay al día. En la siguiente imagen se puede observar cómo quedaron los datos.

Figura 75: Base de datos procesada

	A	B	C	D	E
1	FECHA	EJECUTAD	HO	HOLIDAY	Holida
2	2017-01-01 00:30:00	5297.58565	1	TRUE	1
3	2017-01-01 01:00:00	5204.42518	2	TRUE	1
4	2017-01-01 01:30:00	5130.042	3	TRUE	1
5	2017-01-01 02:00:00	5024.78366	4	TRUE	1
6	2017-01-01 02:30:00	4906.16097	5	TRUE	1
7	2017-01-01 03:00:00	4825.88566	6	TRUE	1
8	2017-01-01 03:30:00	4796.79196	7	TRUE	1
9	2017-01-01 04:00:00	4703.76515	8	TRUE	1
10	2017-01-01 04:30:00	4669.9345	9	TRUE	1
11	2017-01-01 05:00:00	4654.33931	10	TRUE	1
12	2017-01-01 05:30:00	4590.42285	11	TRUE	1
13	2017-01-01 06:00:00	4391.89862	12	TRUE	1
14	2017-01-01 06:30:00	4194.41017	13	TRUE	1
15	2017-01-01 07:00:00	4141.22966	14	TRUE	1
16	2017-01-01 07:30:00	4143.53818	15	TRUE	1
17	2017-01-01 08:00:00	4267.38769	16	TRUE	1
18	2017-01-01 08:30:00	4356.93001	17	TRUE	1
19	2017-01-01 09:00:00	4427.51055	18	TRUE	1
20	2017-01-01 09:30:00	4477.63155	19	TRUE	1

Fuente: Nexa Cajamarquilla S.A. (2022)

Como se mencionó anteriormente, también se pudo recopilar datos sobre los días festivos, por ello, se creó una columna dentro de la misma hoja de cálculo cuyo título fue HOLIDAYS. Tomamos la hoja del excel “HOLIDAYS” con el listado de todos los días que fueron festivos, y con ayuda de la función CONTARSI, comparamos la lista de feriados con las fechas de demanda que tenemos en la columna A de la hoja “DEMANDA” y de esta forma validamos con un 1 si la fecha fue un día feriado 0 si fue un día no festivo.

Una vez los datos fueron importados al programador, se realizaron verificaciones para corroborar que todos los datos estaban completos y no hubiera filas vacías.

Figura 76: Verificación de data completa

```
In [40]:
# Verificar que un índice temporal está completo
# ============================================================================
(datos.index == pd.date_range(start=datos.index.min(),
                              end=datos.index.max(),
                              freq=datos.index.freq)).all()
```

```
Out[40]: True
```

Fuente: Elaboración en Python con la librería pandas (2022)

Como se puede observar en la Figura anterior, el programador no encontró celdas vacías pues arrojó un valor de TRUE corroborando de esta forma que todas las celdas están completas.

Para el caso en el cual se hubieran encontrado datos incompletos y Python hubiera arrojado un valor de FALSE se hubiera podido usar el siguiente código para completar los vacíos. Sin embargo, como no fue el caso, no hubo necesidad de emplearlo.

Figura 77: Completar datos incompletos o vacíos

```
In [41]:
# Completar huecos en un índice temporal
# ============================================================================
# datos.asfreq(freq='30min', fill_value=np.nan)
```

Fuente: Elaboración en Python con la librería pandas (2022)

Debido a que Python está considerando la variable fecha como tipo String se hizo empleo de la función pd.to_datetime() para convertirla a formato de fecha además de establecer que los datos tienen una frecuencia de 30 minutos.

Figura 78: Conversión al formato fecha

```
In [39]:
# Conversión del formato fecha
# ============================================================================
datos['FECHA'] = pd.to_datetime(datos['FECHA'], format='%Y-%m-%dT%H:%M:%SZ')
datos = datos.set_index('FECHA')
datos = datos.asfreq('30min')
datos = datos.sort_index()
```

Fuente: Elaboración en Python con la librería pandas (2022)

Por último, debido al análisis realizado en la etapa previa en el cual se observó un comportamiento fuera de lo común en el año 2020, esto debido a la pandemia por el COVID-19 que ocurrió en ese año, se decidió retirar gran parte de la data referente a ese año para evitar que el modelo se distorsione.

En ese sentido, se tuvo dos opciones, usar la data del 01/01/2021 hasta 01/01/2022, lo cual nos permitiría usar la data más reciente; sin embargo, por otro lado, se tenía la opción de usar la data del 01/01/2017 hasta el 15/03/2020 (día previo al inicio de la pandemia) y tener una mayor cantidad de data. Cabe recalcar que se realizó el análisis y se evaluó la opción de usar la data del 01/01/2017 hasta el 15/03/2020 y del 01/01/2021 hasta 01/01/2022 juntamente con la finalidad de tener la mayor cantidad de data posible, sin embargo, esto llega a ser contraproducente por el hecho de que el programa entendería que la demanda en el año 2020 fue cero y que este efecto se replica cada 3 años, lo cual no sería correcto.

Finalmente, la opción que se tomó fue la primera, ya que como dijimos anteriormente, el comportamiento de la demanda es estacional y casi idéntica, por lo cual, la variable proximidad de la data no sería tan trascendente, y lo que se busca priorizar es tener la mayor cantidad data para poder tener un modelo predictivo lo más preciso y ajustado posible a la realidad.

Figura 79: Delimitación de la data

```
In [ ]: datos = datos.loc['2017-01-01 00:30:00': '2020-03-15 23:00:00']
```

Fuente: Elaboración en Python con la librería pandas (2022)

5.1.2.3 Modelamiento

Para poder realizar el modelado de los datos se usó el lenguaje de programación Python bajo la plataforma de Anaconda y usando la interfaz de Jupyter Notebook.

Para el presente trabajo se emplearon dos modelos, regresión lineal y series de tiempo. En la siguiente etapa analizaremos qué modelo se ajusta mejor al comportamiento de la demanda.

5.1.2.3.1 Regresión lineal. La primera metodología que empleamos fue la de regresión lineal. Para la aplicación de esta metodología se usó sklearn, técnica extraída de la biblioteca de scikit-learn o sklearn, biblioteca que contiene varias herramientas o algoritmos para el análisis de datos, tales como, svm, knn, random forest, kmeans, entre otros.

A partir del dataframe ya pre procesado, se realizó la separación en dos dataframe: "X", para la variable independiente "HORA" y "Y" para la variable dependiente "EJECUTADO" tal y como se muestra en la Figura.

Figura 80: Asignación de variables

```
In [19]:  1  X=datos[['HORA.1']]
          2  Y=datos[['EJECUTADO']]
```

Fuente: Elaboración en Python con la librería pandas (2022)

Luego ambas variables fueron divididas en una proporción de 80:20 entre los datasets de entrenamiento y de test. Como se observa en la siguiente imagen se obtuvo un total de 44926 datos para el entrenamiento y 11232 datos para el test.

Figura 81: Separación de datos en train y test

```
In [32]: import sklearn
         from sklearn.model_selection import train_test_split
         X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=10)

In [33]: X_train.shape, X_test.shape

Out[33]: ((44926, 1), (11232, 1))
```

Fuente: Elaboración en Python con la librería sklearn (2022)

Como se muestra en la Figura 83, a partir del dataset de train se aplicó la técnica de machine learning de regresión lineal. De esta forma, se logró obtener el coeficiente "m", la constante independiente "b" para así poder determinar la ecuación de la recta (y=mx+b).

Figura 82: Modelamiento de la regresión lineal

```
In [ ]: from sklearn.linear_model import LinearRegression

        alumnoR=LinearRegression()
        alumnoR.fit(X_train,y_train)
        res=alumnoR.predict(X_test)
        print('Coefficients: \n', alumnoR.coef_)
        independiente=float(print('Independent term: \n', alumnoR.intercept_)
```

Fuente: Elaboración propia (2022)

5.1.2.3.2 Series de Tiempo. La segunda metodología que empleamos fue las series de tiempo. Para la aplicación de esta metodología también se usó sklearn, y skforecast, una librería que te permite usar fórmulas de predicción.

Figura 83: Sklearn y Skforecast

```
In [171]: from skforecast.ForecasterAutoreg import ForecasterAutoreg
          from skforecast.ForecasterAutoregMultiOutput import ForecasterAutoregMultiOutput
          from skforecast.model_selection import grid_search_forecaster
          from skforecast.model_selection import backtesting_forecaster
```

Fuente: Elaboración en Python con la librería sklearn (2022)

Se creó y se entrenó un modelo autorregresivo ForecasterAutoreg que forma parte de un modelo de regresión lineal pero que a su vez emplea Ridge, una ventana temporal de 24 lags, lo cual significa que, por cada predicción se emplearán las 24 horas anteriores como predictores de la demanda.

Figura 84: Entrenamiento del forecaster

In [172]:
```
# Crear y entrenar forecaster
# ==============================================================================
forecaster = ForecasterAutoreg(
                regressor = make_pipeline(StandardScaler(), Ridge()),
                lags      = 24
             )

forecaster.fit(y=datos.loc[:fin_validacion, 'EJECUTADO'])
forecaster
```

Fuente: Elaboración en Python con la librería skforecast (2022)

Con la finalidad de optimizar aún más los hiperparámetros, lo que se conoce como *tunning*, se hizo empleo de Grid Search. Esta herramienta permite identificar la mejor combinación de lags e hiperparámetros, y evaluar su capacidad predictiva mediante backtesting.

Figura 85: Grid Search

In [15]:
```
# Grid search de hiperparámetros
# ==============================================================================
forecaster = ForecasterAutoreg(
                    regressor = make_pipeline(StandardScaler(), Ridge()),
                    lags      = 24 # Este valor será remplazado en el grid search
             )

# Lags utilizados como predictores
lags_grid = [5, 24, [1, 2, 3, 23, 24, 25, 47, 48, 49]]

# Hiperparámetros del regresor
param_grid = {'ridge__alpha': np.logspace(-3, 5, 10)}

resultados_grid = grid_search_forecaster(
                        forecaster  = forecaster,
                        y           = datos.loc[:fin_validacion, 'EJECUTADO'],
                        param_grid  = param_grid,
                        lags_grid   = lags_grid,
                        steps       = 24,
                        metric      = 'mean_absolute_error',
                        refit       = False,
                        initial_train_size = len(datos[:fin_train]),
                        return_best = True,
                        verbose     = False
                    )
```

Fuente: Elaboración en Python con la librería skforecast (2022)

En este punto, el modelo está asumiendo que las predicciones del día siguiente se ejecutan justo al final del día anterior. Lo cual no sería correcto ya que, en las primeras horas del día, apenas se dispone de anticipación.

Supóngase ahora que, para poder tener suficiente margen de acción, a las 11:00 horas de cada día se tienen que generar las predicciones del día siguiente. De esta forma, a las 11:00h de cada día, el modelo tiene acceso a los valores reales de demanda registrados hasta ese momento.

Para realizar este proceso se hace empleo del método predict() de un objeto ForecasterAutoreg. Donde se le especifica los valores como punto de partida mediante el argumento last_window.

Figura 86: Predicción Diaria Anticipada

```
In [23]: def backtest_predict_next_24h(forecaster, y, hour_init_prediction, exog=None,
                                       verbose=False):

             y = y.sort_index()
             if exog is not None:
                 exog = exog.sort_index()

             dummy_steps = 24 - (hour_init_prediction + 1)
             steps = dummy_steps + 24

             for datetime in y.index[y.index.hour == hour_init_prediction]:
                 if len(y[:datetime]) >= len(forecaster.last_window):
                     datetime_init_backtest = datetime
                     print(f"Backtesting starts at day: {datetime_init_backtest}")
                     break

             days_backtest = np.unique(y[datetime_init_backtest:].index.date)
             days_backtest = pd.to_datetime(days_backtest)
             days_backtest = days_backtest[1:]
             print(f"Days predicted in the backtesting: {days_backtest.strftime('%Y-%m-%d').values}")
             print('')
             backtest_predicciones = []

             for i, day in enumerate(days_backtest):

                 end_window = (day - pd.Timedelta(1, unit='day')).replace(hour=hour_init_prediction)
                 start_window = end_window - pd.Timedelta(forecaster.max_lag, unit='hour')
                 last_window = y.loc[start_window:end_window]

                 if exog is None:
                     if verbose:
                         print(f"Forecasting day {day.strftime('%Y-%m-%d')}")
                         print(f"Using window from {start_window} to {end_window}")

                     pred = forecaster.predict(steps=steps, last_window=last_window)
```

```
        else:
            start_exog_window = end_window + pd.Timedelta(1, unit='hour')
            end_exog_window   = end_window + pd.Timedelta(steps, unit='hour')
            exog_window = exog.loc[start_exog_window:end_exog_window]
            exog_window = exog_window

            if verbose:
                print(f"Forecasting day {day.strftime('%Y-%m-%d')}")
                print(f"    Using window from {start_window} to {end_window}")
                print(f"    Using exogen variable from {start_exog_window} to {end_exog_window}")

            pres = forecaster.predict(steps=steps, last_window=last_window, exog=exog_window)

        pred = pred[dummy_steps:]
        backtest_predicciones.append(pred)

    backtest_predicciones = np.concatenate(backtest_predicciones)
    backtest_predicciones = pd.Series(
                            data  = backtest_predicciones,
                            index = pd.date_range(
                                        start = days_backtest[0],
                                        end   = days_backtest[-1].replace(hour=23),
                                        freq  = 'h'
                                    )
                        )

    return backtest_predicciones
```

Fuente: Elaboración en Python con la librería skforecast (2022)

Al momento de realizar la predicción diaria anticipada se le añade una nueva variable, que será la variable exógena “HOLIDAYS”. De esta forma, llegada las 11:00 horas del día, tendremos que la predicción del día siguiente incluya si es un día festivo o no.

Figura 87: Variable Exógena Feriados

```
In [19]: # Se convierte la columna Holiday de boolean a integer
         datos.loc[:,'HOLIDAY'] = datos['HOLIDAY'].astype(int)
         datos_train.loc[:,'HOLIDAY'] = datos_train['HOLIDAY'].astype(int)
         datos_test.loc[:,'HOLIDAY'] = datos_test['HOLIDAY'].astype(int)

In [20]: # Crear y entrenar forecaster
         # ==============================================================================
         forecaster = ForecasterAutoreg(
                         regressor = make_pipeline(StandardScaler(), Ridge(alpha=215.44)),
                         lags      = [1, 2, 3, 23, 24, 25, 47, 48, 49],
                     )

         forecaster.fit(y=datos.EJECUTADO[:fin_validacion], exog=datos.HOLIDAY[:fin_validacion])
         forecaster
```

Fuente: Elaboración en Python con la librería skforecast (2022)

Con la finalidad de ajustar aún más el modelo, se añadió a la programación información adicional sobre si el día anterior y siguiente son festivos, el día de la semana y hora del día.

Figura 868: Variable Exógena Feriados

```
In [29]: # Creación de nuevas variables exógenas
         # ======================================================================
         # Al ser datos  horarios y empezar a las 00:00 se tienen que desplazar los valores
         # de 24 en 24.
         datos.loc[:,'HOLIDAY_day_before'] = datos['HOLIDAY'].shift(24)
         datos.loc[:,'HOLIDAY_next_day']   = datos['HOLIDAY'].shift(-24)
         datos=datos.dropna()

         # One hot encoding del día de la semana y la hora del día
         datos=pd.get_dummies(datos, columns=['dia_semana', 'hora_dia'])
         datos.head(3)
```

Fuente: Elaboración en Python con la librería skforecast (2022)

Cabe mencionar que en la búsqueda de variables exógenas que permitan un mejor ajuste al modelo, se encontró la variable temperatura, que, si bien existe cierta correlación con el consumo de energía, debido al uso de aires condicionados y demás sistemas de refrigeración, se concluyó que esta variable no generaba aporte al modelo ya que predecir esta variable es imposible. Sí es posible utilizar la variable temperatura como un predictor del modelo, pero, en tal caso, durante el entrenamiento habría que utilizar la temperatura que había en ese momento, no la temperatura de ahora y la recopilación de dicha data es una tarea bastante ardua y tediosa.

5.1.2.4 Evaluación del modelo

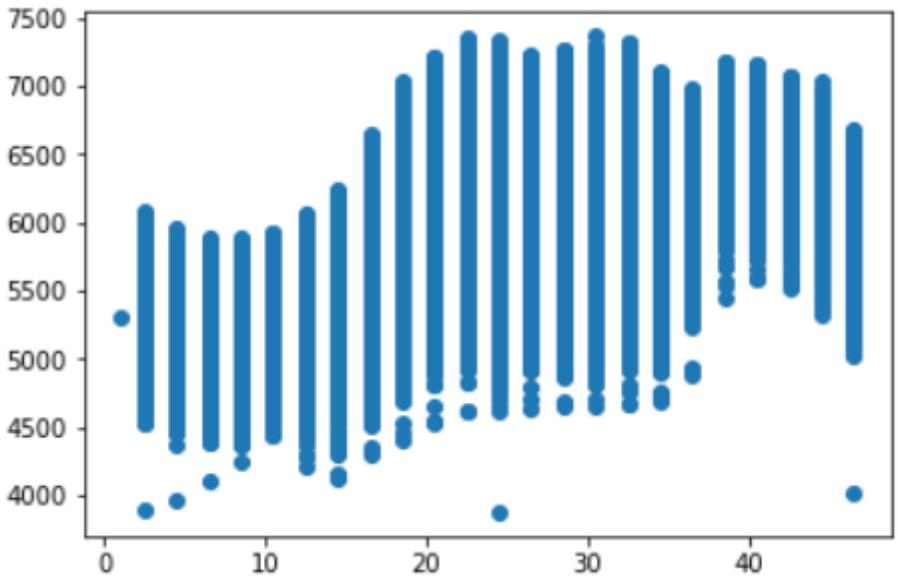
5.1.2.4.1 Regresión lineal.

Analizando los resultados podemos concluir que este modelo no se ajusta al comportamiento de la demanda de energía de la planta ya que el modelo arroja un valor de varianza del 0.42 y para ser un modelo aceptable, este debería de ser cercano al 1.

Por otro lado, el error cuadrático medio es demasiado alto, reflejando nuevamente que el modelo no es el más adecuado para poder predecir la demanda.

A continuación, se muestra una gráfica de dispersión donde se aprecia el comportamiento irregular de la data y cómo un modelo de regresión lineal no es aplicable.

Figura 879: Gráfica de dispersión regresión lineal



Fuente: Elaboración en Python con la librería matplotlib.pyplot (2022)

5.1.2.4.2 Series de Tiempo.

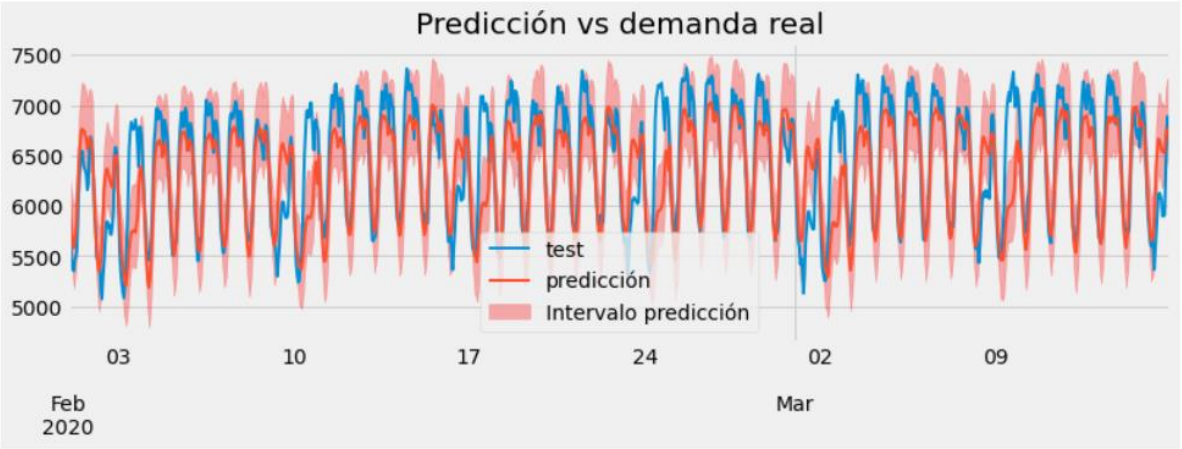
Podemos observar que los resultados obtenidos en este modelo inicialmente no son los mejores, aunque si son más aceptables que el modelo de regresión lineal. Antes de afinar el modelo, el error del backtest que nos arroja la programación es de un valor de 260,47. Si bien es un error todavía elevado, el modelo se ajusta con mayor precisión al comportamiento de la demanda.

Figura 90: Gráfica de predicción vs demanda real primer forecast

Fuente: Elaboración en Python con la librería matplotlib.pyplot (2022)

Del mismo modelo pudimos obtener gráficamente el intervalo de cobertura predicho, concluyendo que es un modelo que se ajusta bastante bien al comportamiento de la demanda de energía.

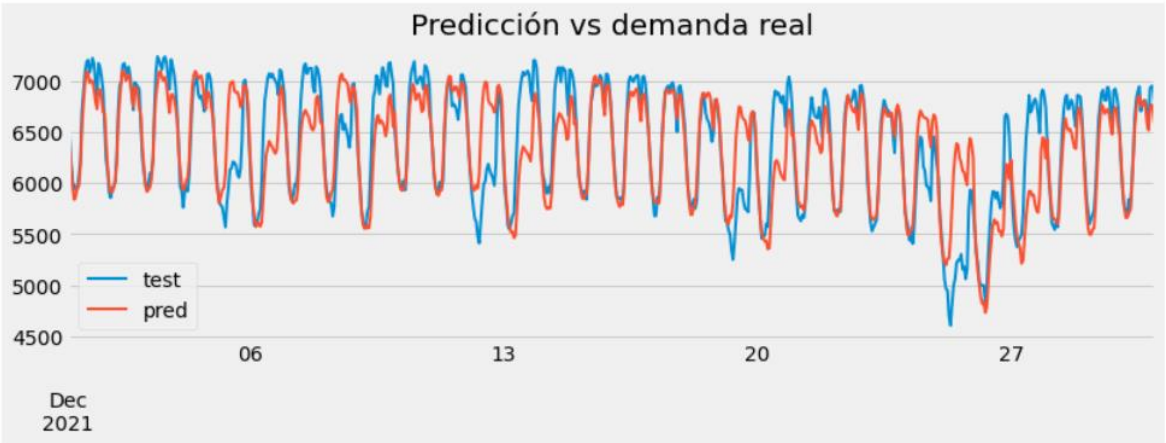
Figura 9188: Gráfica de predicción vs demanda real primer forecast intervalo predicción



Fuente: Elaboración en Python con la librería matplotlib.pyplot (2022)

Podemos observar cómo al agregar al modelo la herramienta de optimización de hiperparámetros, Grid Search el modelo logra ajustarse mejor, obteniéndose un error del backtest de 233,66. De manera gráfica, se puede ver cómo el modelo predictivo se ajusta de manera más precisa al comportamiento de la demanda de energía.

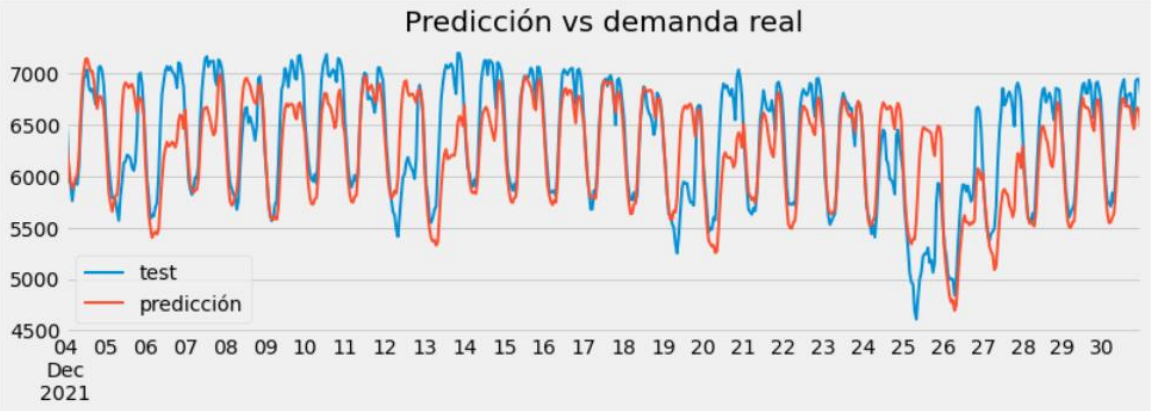
Figura 9289: Gráfica de predicción vs demanda real Grid Search



Fuente: Elaboración en Python con la librería matplotlib.pyplot (2022)

Al agregar nuevos parámetros al modelo y definir horarios para que el modelo pueda hacer las predicciones diarias de manera anticipada, podemos observar que el error del backtest aumenta a 289,47 y la gráfica se ve alterada, esto se debe a que el comportamiento de un día a otro de la muestra tomada para hacer el test no es del todo constante, sin embargo, el modelo sigue ajustándose de mejor manera que el método de regresión lineal.

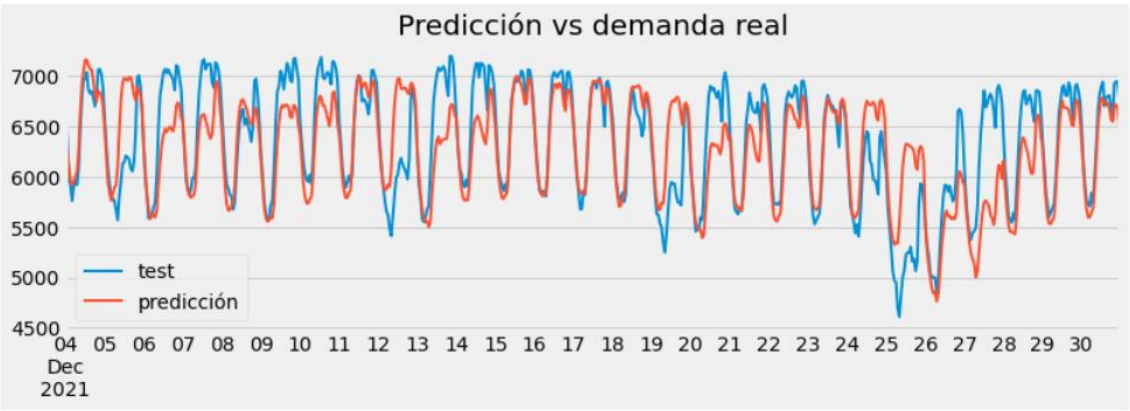
Figura 9390: Gráfica de predicción vs demanda real predicción diaria anticipada



Fuente: Elaboración en Python con la librería matplotlib.pyplot (2022)

En ese sentido, se añadió una variable exógena que ajuste al modelo considerando los parámetros ya colocados, se optó por incluir la variable “Feriados”, la cual servía como indicador de los días en los que la demanda no podía ser máxima. Esta inclusión al modelo nos permitió reducir nuevamente el error del backtesting, obteniendo un valor de 275,30 por debajo del valor obtenido anteriormente.

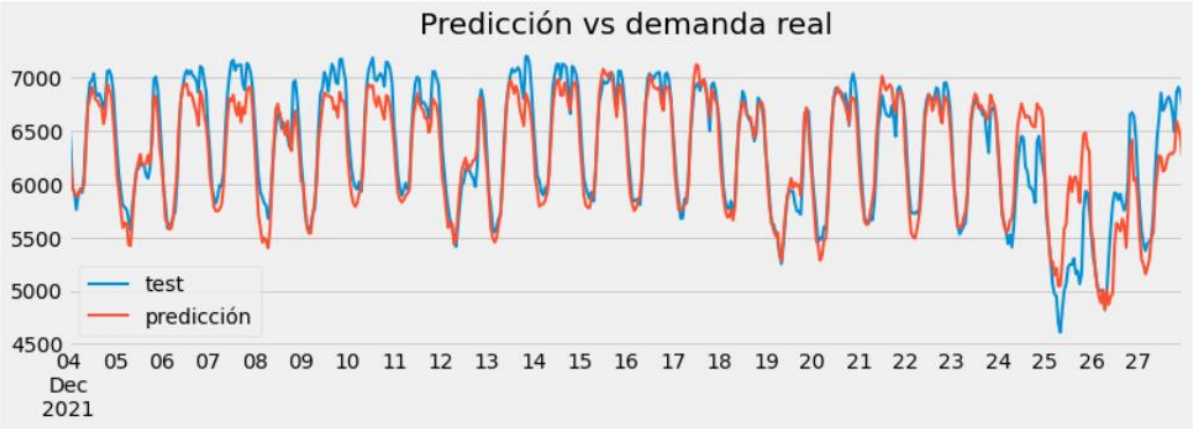
Figura 9491: Gráfica de predicción vs demanda real variable exógena



Fuente: Elaboración en Python con la librería matplotlib.pyplot (2022)

En primera instancia parecería que el mejor modelo fue cuando se realizó empleo de la herramienta de Grid Search, el optimizador de parámetros. Sin embargo, si se ajusta el modelo con la variable exógena y se añade la predicción anticipada vemos cómo se logra obtener un resultado del backtest del 169,57 convirtiéndolo así en nuestro mejor modelo predictorio a ser elegido. De esta forma se refleja gráficamente como este modelo es el que se ajusta con mayor precisión al comportamiento de la demanda de energía de la empresa Nexa Resources S.A.

Figura 9592: Gráfica de predicción vs demanda real variable exógena con predicción anticipada



Fuente: Elaboración en Python con la librería matplotlib.pyplot (2022)

5.2 Medición de la solución

Para medir la solución del modelo de regresión lineal se usaron las métricas MAE, MSE y RMSE, que son el error absoluto medio, error cuadrático medio y la raíz del error cuadrático medio respectivamente:

- MAE, tiene como objetivo evaluar la calidad del modelo empleado o, en otras palabras, refleja la diferencia entre el valor predicho y lo real
- MSE, tiene como objetivo mostrar la cercanía de la línea de regresión con un grupo de puntos.
- RMSE, su objetivo es medir la dispersión de los datos con respecto a la línea de regresión.

Además, se empleó como medida de medición para el modelo de regresión lineal a la varianza.

Para medir la solución del modelo de serie de tiempo se usó el error del backtest, que refleja que tan cercana es la predicción con lo real. Por otro lado, también se usó como medida de solución la cobertura del intervalo predicho.

5.2.1 Análisis de Indicadores cuantitativo y/o cualitativo

5.2.1.1 Regresión Lineal

Tras aplicar el modelo de regresión lineal, se obtuvieron los siguientes resultados:

Tabla 13: Resultado del modelo de regresión lineal

	Resultado
MAE	357.26
MSE	203511.05
RMSE	451.12
VARIANZA	0.42

Fuente: Elaboración propia (2022)

Mean absolute error (MAE)

Podemos concluir que la calidad del modelo es bastante baja debido al alto resultado del MAE.

Figura 9693: MAE

```
In [62]: #MAE
         mean_absolute_error(y_test, res)
Out[62]: 357.25891077872694
```

Fuente: Elaboración en Python con la librería sklearn (2022)

Mean squared error (MSE)

Observamos de la tabla ## que el error cuadrático medio es excesivamente alto lo cual nos indica que la recta de la ecuación lineal está muy alejada a los valores reales de la demanda.

Figura 9794: MSE

```
In [61]: #MSE
         mean_squared_error(y_test, res)

Out[61]: 203511.04777895726
```

Fuente: Elaboración en Python con la librería sklearn (2022)

Root Mean Squared Error (RMSE)

De igual forma, vemos que la dispersión de los datos con respecto a la línea de regresión es alta.

Figura 9895: RMSE

```
In [63]: #RMSE
         mean_squared_error(y_test, res)**0.5

Out[63]: 451.121987691752
```

Fuente: Elaboración en Python con la librería sklearn (2022)

Varianza

El resultado de la varianza es de 0.42, lo cual nos indica que el modelo no se ajusta lo suficiente al comportamiento de la demanda eléctrica de la planta. Para que el modelo sea considerado aceptable tiene que ser cercano a 1.

Figura 9996: Varianza

```
In [67]: print('Variance score: %.2f' % r2_score(y_test, res))
         Variance score: 0.42
```

Fuente: Elaboración en Python con la librería sklearn (2022)

5.2.1.2 Series de tiempo

Error backtest.

En el primer ejercicio del modelo predictorio por series de tiempo, los resultados del error backtest arrojaron un valor de 260.47, el cual, como mencionamos en el capítulo 5.1.2.4.2, es un error relativamente alto, sin embargo, el modelo se ajusta correctamente al comportamiento de la demanda.

Figura 10097: Error backtest forecast inicial

```
In [218]: # Error backtest
          # ======================================
          print(f'Error backtest: {metrica}')

          Error backtest: [260.46514909]
```

Fuente: Elaboración en Python con la librería skforecast (2022)

Cobertura del intervalo predicho

De este primer ejercicio también pudimos obtener el intervalo de cobertura del modelo predictorio, el cual como se puede observar es de 81.34%.

Figura 10198: Cobertura del intervalo predicho forecast inicial

```
In [223]: # Cobertura del intervalo predicho
          # ======================================================================
          dentro_intervalo = np.where(
                               (datos.loc[fin_validacion:, 'EJECUTADO'] >= predicciones['lower_bound']) & \
                               (datos.loc[fin_validacion:, 'EJECUTADO'] <= predicciones['upper_bound']),
                               True,
                               False
                             )

          cobertura = dentro_intervalo.mean()
          print(f"Cobertura del intervalo predicho: {round(100*cobertura, 2)} %")

          Cobertura del intervalo predicho: 81.34 %
```

Fuente: Elaboración en Python con la librería skforecast (2022)

Error backtest.

Con la incorporación de la herramienta Grid Search, se logró reducir el error a un valor de backtest de 233,66.

Figura 10299: Error backtest Grid Search

```
In [19]: # Error backtest
         # ==================================
         print(f'Error backtest: {metrica}')

         Error backtest: [233.65687989]
```

Fuente: Elaboración en Python con la librería skforecast (2022)

Posteriormente, agregando nuevos parámetros al modelo que permitan realizar una predicción diaria anticipada, se encontró un error de backtest de 289,47.

Figura 103100: Error backtest predicción diaria anticipada

```
In [26]: # Error backtest
         # ==============================================================
         error = mean_absolute_error(
                     y_true = datos.loc[predicciones.index, 'EJECUTADO'],
                     y_pred = predicciones
                 )
         print(f"Error de backtest: {error}")

         Error de backtest: 289.46852642807875
```

Fuente: Elaboración en Python con la librería skforecast (2022)

Luego de la incorporación de la variable exógena feriados al modelo se logró volver a reducir el valor del backtest a 275.30, sin embargo, este aún seguía siendo alto, comparado con el que obtuvimos con el optimizador de hiper-parámetros, Grid Search.

Figura 104101: Error backtest variable exógena

```
In [24]: # Error backtest
         # ==============================================================
         error = mean_absolute_error(
                     y_true = datos.loc[predicciones.index, 'EJECUTADO'],
                     y_pred = predicciones
                 )

         print(f"Error de backtest: {error}")

         Error de backtest: 275.29841121517325
```

Fuente: Elaboración en Python con la librería skforecast (2022)

Finalmente, luego de ajustar el modelo añadiendo información respecto a la predicción del día siguiente. Se consiguió disminuir el error del backtest a 169.57, es decir se logró encontrar un modelo más preciso y que aporte mayor valor a la investigación.

Figura 105102: Error backtest variable exógena con predicción diaria anticipada

```
In [35]: # Error backtest
         # ==============================================================
         error = mean_absolute_error(
                     y_true = datos.loc[predicciones.index, 'EJECUTADO'],
                     y_pred = predicciones
                 )

         print(f"Error de backtest: {error}")

         Error de backtest: 169.5695889579889
```

Fuente: Elaboración en Python con la librería skforecast (2022)

5.2.2 Simulación de solución. Aplicación de Software

5.2.2.1 Regresión lineal

5.2.2.1.1 Importación de información

Figura 106103: Importación de librerías y algoritmos a utilizar

```
In [53]: import pandas as pd
         import sklearn
         from sklearn.model_selection import train_test_split
         from sklearn.linear_model import LinearRegression
         from sklearn.metrics import mean_squared_error
         from sklearn.metrics import mean_absolute_error
         from sklearn.metrics import r2_score
         from matplotlib import pyplot
```

Fuente: Elaboración propia (2022)

Figura 107104: Importación de datos

```
In [28]: # IMPORTAMOS LOS DATOS
         datos=pd.read_excel('DEMANDA 2017-2021.xlsx')
         datos
```

Out[28]:

	FECHA	EJECUTADO	HORA	HOLIDAY	Holiday
0	2017-01-01 00:30:00	5297.58565	1	True	1
1	2017-01-01 01:00:00	5204.42518	2	True	1
2	2017-01-01 01:30:00	5130.04200	3	True	1
3	2017-01-01 02:00:00	5024.78366	4	True	1
4	2017-01-01 02:30:00	4906.16097	5	True	1
...	...	...	...	...	...
87643	2021-12-31 22:00:00	6434.73100	44	False	0
87644	2021-12-31 22:30:00	6358.67950	45	False	0
87645	2021-12-31 23:00:00	6301.94050	46	False	0
87646	2021-12-31 23:30:00	6189.40500	47	False	0
87647	2022-01-01 00:00:00	5973.43200	48	True	1

87648 rows × 5 columns

Fuente: Elaboración en Python con la librería panda (2022)

5.2.2.1.2 Pre-procesamiento de información

Figura 108105: Conversión del formato fecha y delimitación de data

```
In [29]: # Conversión del formato fecha
         # ============================================================================
         datos['FECHA'] = pd.to_datetime(datos['FECHA'], format='%Y-%m-%dT%H:%M:%SZ')
         datos = datos.set_index('FECHA')
         datos = datos.asfreq('30min')
         datos = datos.sort_index()
```

```
In [30]: datos = datos.loc['2017-01-01 00:30:00': '2020-03-15 23:00:00']
```

Fuente: Elaboración en Python con la librería pandas (2022)

5.2.2.1.3 Definición de variables independientes y variable dependiente

Figura 109106: Variable independiente

```
In [54]: X=datos[['HORA']]
         X
```

Out[54]:

FECHA	HORA
2017-01-01 00:30:00	1
2017-01-01 01:00:00	2
2017-01-01 01:30:00	3
2017-01-01 02:00:00	4
2017-01-01 02:30:00	5
...	...
2020-03-15 21:00:00	42
2020-03-15 21:30:00	43
2020-03-15 22:00:00	44
2020-03-15 22:30:00	45
2020-03-15 23:00:00	46

56158 rows × 1 columns

Fuente: Elaboración en Python con la librería pandas (2022)

Figura 110107: Variable dependiente

```
In [55]: Y=datos[['EJECUTADO']]
         Y
```

Out[55]:

FECHA	EJECUTADO
2017-01-01 00:30:00	5297.58565
2017-01-01 01:00:00	5204.42518
2017-01-01 01:30:00	5130.04200
2017-01-01 02:00:00	5024.78366
2017-01-01 02:30:00	4906.16097
...	...
2020-03-15 21:00:00	6847.36808
2020-03-15 21:30:00	6765.06195
2020-03-15 22:00:00	6631.42358
2020-03-15 22:30:00	6463.85280
2020-03-15 23:00:00	6211.84267

56158 rows × 1 columns

Fuente: Elaboración en Python con la librería pandas (2022)

Figura 111108: Definición de set de entrenamiento y set de prueba

```
In [32]: X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=10)

In [33]: X_train.shape, X_test.shape

Out[33]: ((44926, 1), (11232, 1))
```

Fuente: Elaboración en Python con la librería sklearn (2022)

5.2.2.1.3 Aplicación de algoritmos

Figura 112109: Elegir el modelo - alumno

```
In [34]: alumnoR=LinearRegression()
```

Fuente: Elaboración en Python con la librería sklearn (2022)

Figura 113110: Entrenar al alumno

```
In [35]: alumnoR.fit(X_train,y_train)

Out[35]: LinearRegression()
```

Fuente: Elaboración en Python con la librería sklearn (2022)

Figura 111: Examen - predicción

```
In [36]: res=alumnoR.predict(X_test)
```

Fuente: Elaboración en Python con la librería sklearn (2022)

5.2.2.1.4 Medición

Figura 11215: MSE, MAE, RMSE, coeficientes de la recta y varianza

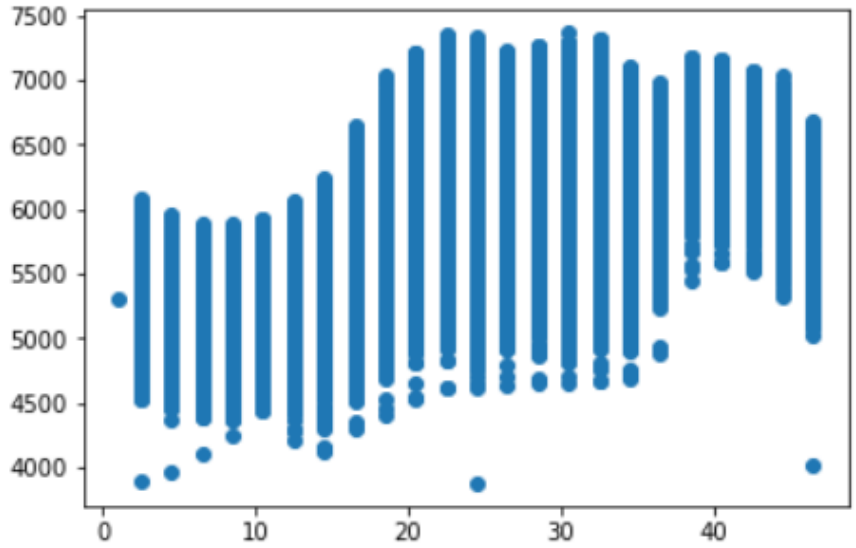
```
In [61]: #MSE
         mean_squared_error(y_test, res)

Out[61]: 203511.04777895726

In [62]: #MAE
         mean_absolute_error(y_test, res)

Out[62]: 357.25891077872694

In [63]: #RMSE
         mean_squared_error(y_test, res)**0.5

Out[63]: 451.121987691752

In [64]: print('Coefficients: \n', alumnoR.coef_)

         Coefficients:
          [[28.59130444]]

In [65]: print('Independen term: \n', alumnoR.coef_)

         Independen term:
          [[28.59130444]]

In [67]: print('Variance score: %.2f' % r2_score(y_test, res))

         Variance score: 0.42
```

Fuente: Elaboración en Python con la librería sklearn (2022)

5.2.2.1.5 Gráficos

Figura 11316: Gráfico de dispersión

```
In [69]: from matplotlib import pyplot
         fig, ax=pyplot.subplots()
         ax.scatter(X,Y)
         pyplot.show()
```



Fuente: Elaboración en Python con la librería matplotlib.pyplot (2022)

5.2.2.2 Series de Tiempo

5.2.2.2.1 Importación de información

Figura 11417: Importación de librerías y algoritmos a utilizar

In [1]:
```
# Tratamiento de datos
# ==============================================================================
import numpy as np
import pandas as pd

# Gráficos
# ==============================================================================
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from statsmodels.graphics.tsaplots import plot_acf
from statsmodels.graphics.tsaplots import plot_pacf
plt.style.use('fivethirtyeight')

# Modelado y Forecasting
# ==============================================================================
from sklearn.linear_model import Ridge
from lightgbm import LGBMRegressor
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_absolute_error
from skforecast.ForecasterAutoreg import ForecasterAutoreg
from skforecast.ForecasterAutoregMultiOutput import ForecasterAutoregMultiOutput
from skforecast.model_selection import grid_search_forecaster
from skforecast.model_selection import backtesting_forecaster

# Configuración warnings
# ==============================================================================
import warnings
warnings.filterwarnings('ignore')
```

Fuente: Elaboración propia (2022)

Figura 11518: Importación de datos

```
In [208]: datos=pd.read_excel('DEMANDA 2017-2021.xlsx')
          datos.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 87648 entries, 0 to 87647
Data columns (total 5 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   FECHA      87648 non-null  datetime64[ns]
 1   EJECUTADO  87648 non-null  float64
 2   HORA       87648 non-null  int64
 3   HOLIDAY    87648 non-null  bool
 4   Holiday    87648 non-null  int64
dtypes: bool(1), datetime64[ns](1), float64(1), int64(2)
memory usage: 2.8 MB
```

Fuente: Elaboración en Python con la librería pandas (2022)

5.2.2.2.2 Pre-procesamiento de información

5.2.2.2.2.1 Pre- procesamiento series de tiempo

Figura 11619: Conversión del formato fecha y la verificación de la data completa

```
In [209]: # Conversión del formato fecha
          # ============================================================================
          datos['FECHA'] = pd.to_datetime(datos['FECHA'], format='%Y-%m-%dT%H:%M:%SZ')
          datos = datos.set_index('FECHA')
          datos = datos.asfreq('30min')
          datos = datos.sort_index()

In [210]: # Verificar que un índice temporal está completo
          # ============================================================================
          (datos.index == pd.date_range(start=datos.index.min(),
                                        end=datos.index.max(),
                                        freq=datos.index.freq)).all()

Out[210]: True

In [211]: # Completar huecos en un índice temporal
          # ============================================================================
          # datos.asfreq(freq='30min', fill_value=np.nan)
```

Fuente: Elaboración en Python con la librería pandas (2022)

Figura 11720: Convirtiendo data a intervalos de 1 hora

```
In [212]: # Agregado en intervalos de 1H
          datos = datos.resample(rule='H', closed='left', label ='right').mean()
          datos
```

Out[212]:

FECHA	EJECUTADO	HORA	HOLIDAY	Holiday
2017-01-01 01:00:00	5297.585650	1.0	1.0	1.0
2017-01-01 02:00:00	5167.233590	2.5	1.0	1.0
2017-01-01 03:00:00	4965.472315	4.5	1.0	1.0
2017-01-01 04:00:00	4811.338810	6.5	1.0	1.0
2017-01-01 05:00:00	4686.849825	8.5	1.0	1.0
...	...	...	...	...
2021-12-31 21:00:00	6621.195500	40.5	0.0	0.0
2021-12-31 22:00:00	6498.185000	42.5	0.0	0.0
2021-12-31 23:00:00	6396.705250	44.5	0.0	0.0
2022-01-01 00:00:00	6245.672750	46.5	0.0	0.0
2022-01-01 01:00:00	5973.432000	48.0	1.0	1.0

43825 rows × 4 columns

Fuente: Elaboración en Python con la librería pandas (2022)

5.2.2.2.2.2 Pre- procesamiento variable exógena

Figura 11821: Convirtiendo variable exógena a entero

```
In [19]: # Se convierte la columna Holiday de boolean a integer
         datos.loc[:,'HOLIDAY'] = datos['HOLIDAY'].astype(int)
         datos_train.loc[:,'HOLIDAY'] = datos_train['HOLIDAY'].astype(int)
         datos_test.loc[:,'HOLIDAY'] = datos_test['HOLIDAY'].astype(int)
```

Fuente: Elaboración en Python con la librería pandas (2022)

5.2.2.2.2.3 Pre- procesamiento variable exógena con predicción anticipada

Figura 11922: Adición de nuevos parámetros

```
In [29]: # Creación de nuevas variables exógenas
         # ======================================================================
         # Al ser datos  horarios y empezar a las 00:00 se tienen que desplazar los valores
         # de 24 en 24.
         datos.loc[:,'HOLIDAY_day_before'] = datos['HOLIDAY'].shift(24)
         datos.loc[:,'HOLIDAY_next_day']   = datos['HOLIDAY'].shift(-24)
         datos=datos.dropna()

         # One hot encoding del día de la semana y la hora del día
         datos=pd.get_dummies(datos, columns=['dia_semana', 'hora_dia'])
         datos.head(3)
```

Fuente: Elaboración en Python con la librería pandas (2022)

5.2.2.2.3 Aplicación de algoritmos

Figura 12023: Paso 1: Separando los datos en train y test

```
In [5]: # Separación datos train-val-test
        # ======================================================================
        datos = datos.loc['2017-01-01 00:30:00': '2021-12-30 23:00:00']
        fin_train = '2020-12-31 23:59:00'
        fin_validacion = '2021-11-30 23:59:00'
        datos_train = datos.loc[: fin_train, :]
        datos_val   = datos.loc[fin_train:fin_validacion, :]
        datos_test  = datos.loc[fin_validacion:, :]

        print(f"Fechas train      : {datos_train.index.min()} --- {datos_train.index.max()}  (n={len(datos_train)})")
        print(f"Fechas validacion : {datos_val.index.min()} --- {datos_val.index.max()}  (n={len(datos_val)})")
        print(f"Fechas test       : {datos_test.index.min()} --- {datos_test.index.max()}  (n={len(datos_test)})")
```

```
Fechas train      : 2017-01-01 01:00:00 --- 2020-12-31 23:00:00  (n=35063)
Fechas validacion : 2021-01-01 00:00:00 --- 2021-11-30 23:00:00  (n=8016)
Fechas test       : 2021-12-01 00:00:00 --- 2021-12-30 23:00:00  (n=720)
```

Fuente: Elaboración en Python con la librería pandas (2022)

5.2.2.2.3.1 Forecast inicial

Figura 12124: Paso 2 y 3: Eligiendo el modelo y entrenando al alumno

```
In [215]: # Crear y entrenar forecaster
          # ======================================================================
          forecaster = ForecasterAutoreg(
                         regressor = make_pipeline(StandardScaler(), Ridge()),
                         lags      = 24
                       )

          forecaster.fit(y=datos.loc[:fin_validacion, 'EJECUTADO'])
          forecaster
```

```
Out[215]: ================
          ForecasterAutoreg
          ================
          Regressor: Pipeline(steps=[('standardscaler', StandardScaler()), ('ridge', Ridge())])
          Lags: [ 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24]
          Window size: 24
          Included exogenous: False
          Type of exogenous variable: None
          Exogenous variables names: None
          Training range: [Timestamp('2017-01-01 01:00:00'), Timestamp('2020-01-31 23:00:00')]
          Training index type: DatetimeIndex
          Training index frequency: H
          Regressor parameters: {'standardscaler__copy': True, 'standardscaler__with_mean': True, 'standardscaler__with_std': True, 'ridg
          e__alpha': 1.0, 'ridge__copy_X': True, 'ridge__fit_intercept': True, 'ridge__max_iter': None, 'ridge__normalize': 'deprecated',
          'ridge__positive': False, 'ridge__random_state': None, 'ridge__solver': 'auto', 'ridge__tol': 0.001}
          Creation date: 2022-07-29 19:15:53
          Last fit date: 2022-07-29 19:15:53
          Skforecast version: 0.4.2
```

Fuente: Elaboración en Python con la librería skforecast (2022)

Figura 12225: Paso 4: Examen - predicción

```
In [216]: # Backtest
          # ======================================================================
          metrica, predicciones = backtesting_forecaster(
                                      forecaster = forecaster,
                                      y          = datos.EJECUTADO,
                                      initial_train_size = len(datos.loc[:fin_validacion]),
                                      steps      = 24,
                                      metric     = 'mean_absolute_error',
                                      refit      = False,
                                      verbose    = True
                                  )
```

```
Data partition in fold: 38
    Training:   2017-01-01 01:00:00 -- 2020-01-31 23:00:00
    Validation: 2020-03-10 00:00:00 -- 2020-03-10 23:00:00
Data partition in fold: 39
    Training:   2017-01-01 01:00:00 -- 2020-01-31 23:00:00
    Validation: 2020-03-11 00:00:00 -- 2020-03-11 23:00:00
Data partition in fold: 40
    Training:   2017-01-01 01:00:00 -- 2020-01-31 23:00:00
    Validation: 2020-03-12 00:00:00 -- 2020-03-12 23:00:00
Data partition in fold: 41
    Training:   2017-01-01 01:00:00 -- 2020-01-31 23:00:00
    Validation: 2020-03-13 00:00:00 -- 2020-03-13 23:00:00
Data partition in fold: 42
    Training:   2017-01-01 01:00:00 -- 2020-01-31 23:00:00
    Validation: 2020-03-14 00:00:00 -- 2020-03-14 23:00:00
Data partition in fold: 43
    Training:   2017-01-01 01:00:00 -- 2020-01-31 23:00:00
    Validation: 2020-03-15 00:00:00 -- 2020-03-15 23:00:00
```

Fuente: Elaboración en Python con la librería skforecast (2022)

5.2.2.2.3.2 Grid Search

Figura 12326: Paso 2 y 3: Eligiendo el modelo y entrenando al alumno Grid Search

In [15]:
```
# Grid search de hiperparámetros
# ==============================================================================
forecaster = ForecasterAutoreg(
                 regressor = make_pipeline(StandardScaler(), Ridge()),
                 lags      = 24 # Este valor será remplazado en el grid search
             )

# Lags utilizados como predictores
lags_grid = [5, 24, [1, 2, 3, 23, 24, 25, 47, 48, 49]]

# Hiperparámetros del regresor
param_grid = {'ridge__alpha': np.logspace(-3, 5, 10)}

resultados_grid = grid_search_forecaster(
                      forecaster  = forecaster,
                      y           = datos.loc[:fin_validacion, 'EJECUTADO'],
                      param_grid  = param_grid,
                      lags_grid   = lags_grid,
                      steps       = 24,
                      metric      = 'mean_absolute_error',
                      refit       = False,
                      initial_train_size = len(datos[:fin_train]),
                      return_best = True,
                      verbose     = False
                  )
```

```
Number of models compared: 30
loop lags_grid:   0%|                                                    | 0/3 [00:00<?, ?it/s]
loop param_grid:   0%|                                                   | 0/10 [00:00<?, ?it/s]
loop param_grid:  10%|███                                              | 1/10 [00:03<00:35,  3.95s/it]
loop param_grid:  20%|██████                                           | 2/10 [00:07<00:30,  3.76s/it]
loop param_grid:  30%|█████████                                        | 3/10 [00:13<00:31,  4.56s/it]
loop param_grid:  40%|████████████                                     | 4/10 [00:16<00:25,  4.25s/it]
loop param_grid:  50%|███████████████                                  | 5/10 [00:21<00:21,  4.29s/it]
loop param_grid:  60%|██████████████████                               | 6/10 [00:25<00:16,  4.16s/it]
loop param_grid:  70%|█████████████████████                            | 7/10 [00:29<00:12,  4.14s/it]
loop param_grid:  80%|████████████████████████                         | 8/10 [00:33<00:08,  4.03s/it]
loop param_grid:  90%|███████████████████████████                      | 9/10 [00:36<00:03,  3.96s/it]
loop param_grid: 100%|██████████████████████████████| 10/10 [00:40<00:00,  3.94s/it]
loop lags_grid:  33%|██████████                                       | 1/3 [00:40<01:21, 40.74s/it]
loop param_grid:   0%|                                                   | 0/10 [00:00<?, ?it/s]
loop param_grid:  10%|███                                              | 1/10 [00:04<00:41,  4.59s/it]
loop param_grid:  20%|██████                                           | 2/10 [00:08<00:35,  4.38s/it]
loop param_grid:  30%|█████████                                        | 3/10 [00:12<00:28,  4.11s/it]
loop param_grid:  40%|████████████                                     | 4/10 [00:18<00:27,  4.65s/it]
loop param_grid:  50%|███████████████                                  | 5/10 [00:23<00:24,  4.93s/it]
loop param_grid:  60%|██████████████████                               | 6/10 [00:28<00:20,  5.03s/it]
loop param_grid:  70%|█████████████████████                            | 7/10 [00:33<00:15,  5.07s/it]
loop param_grid:  80%|████████████████████████                         | 8/10 [00:39<00:10,  5.28s/it]
loop param_grid:  90%|███████████████████████████                      | 9/10 [00:44<00:05,  5.16s/it]
loop param_grid: 100%|██████████████████████████████| 10/10 [00:48<00:00,  4.94s/it]
loop lags_grid:  67%|████████████████████                             | 2/3 [01:29<00:45, 45.58s/it]
loop param_grid:   0%|                                                   | 0/10 [00:00<?, ?it/s]
loop param_grid:  10%|███                                              | 1/10 [00:04<00:37,  4.22s/it]
loop param_grid:  20%|██████                                           | 2/10 [00:08<00:32,  4.07s/it]
loop param_grid:  30%|█████████                                        | 3/10 [00:12<00:28,  4.00s/it]
loop param_grid:  40%|████████████                                     | 4/10 [00:16<00:24,  4.16s/it]
loop param_grid:  50%|███████████████                                  | 5/10 [00:20<00:21,  4.25s/it]
loop param_grid:  60%|██████████████████                               | 6/10 [00:25<00:16,  4.21s/it]
loop param_grid:  70%|█████████████████████                            | 7/10 [00:29<00:12,  4.25s/it]
loop param_grid:  80%|████████████████████████                         | 8/10 [00:34<00:08,  4.47s/it]
loop param_grid:  90%|███████████████████████████                      | 9/10 [00:39<00:04,  4.68s/it]
loop param_grid: 100%|██████████████████████████████| 10/10 [00:43<00:00,  4.52s/it]
loop lags_grid: 100%|██████████████████████████████| 3/3 [02:13<00:00, 44.45s/it]
`Forecaster` refitted using the best-found lags and parameters, and the whole data set:
  Lags: [ 1  2  3 23 24 25 47 48 49]
  Parameters: {'ridge__alpha': 27.825594022071257}
  Backtesting metric: 204.1287605923872
```

Fuente: Elaboración en Python con la librería skforecast (2022)

Figura 12427: Paso 4: Examen – predicción Grid Search

In [18]:
```
# Backtest
# ==============================================================================
metrica, predicciones = backtesting_forecaster(
                            forecaster = forecaster,
                            y          = datos.EJECUTADO,
                            initial_train_size = len(datos[:fin_validacion]),
                            steps      = 24,
                            metric     = 'mean_absolute_error',
                            refit      = False,
                            verbose    = False
                        )

fig, ax = plt.subplots(figsize=(12, 3.5))
datos.loc[predicciones.index, 'EJECUTADO'].plot(linewidth=2, label='test', ax=ax)
predicciones.plot(linewidth=2, label='predicción', ax=ax)
ax.set_title('Predicción vs demanda real')
ax.legend();
```

Fuente: Elaboración en Python con la librería skforecast (2022)

5.2.2.2.3.3 Predicción diaria anticipada

Figura 12528: Paso 2 y 3: Eligiendo el modelo y entrenando al alumno predicción anticipada

In [23]:
```
def backtest_predict_next_24h(forecaster, y, hour_init_prediction, exog=None,
                              verbose=False):

    y = y.sort_index()
    if exog is not None:
        exog = exog.sort_index()

    dummy_steps = 24 - (hour_init_prediction + 1)
    steps = dummy_steps + 24

    for datetime in y.index[y.index.hour == hour_init_prediction]:
        if len(y[:datetime]) >= len(forecaster.last_window):
            datetime_init_backtest = datetime
            print(f"Backtesting starts at day: {datetime_init_backtest}")
            break

    days_backtest = np.unique(y[datetime_init_backtest:].index.date)
    days_backtest = pd.to_datetime(days_backtest)
    days_backtest = days_backtest[1:]
    print(f"Days predicted in the backtesting: {days_backtest.strftime('%Y-%m-%d').values}")
    print('')
    backtest_predicciones = []

    for i, day in enumerate(days_backtest):

        end_window = (day - pd.Timedelta(1, unit='day')).replace(hour=hour_init_prediction)
        start_window = end_window - pd.Timedelta(forecaster.max_lag, unit='hour')
        last_window = y.loc[start_window:end_window]

        if exog is None:
            if verbose:
                print(f"Forecasting day {day.strftime('%Y-%m-%d')}")
                print(f"Using window from {start_window} to {end_window}")

            pred = forecaster.predict(steps=steps, last_window=last_window)
```

```
        else:
            start_exog_window = end_window + pd.Timedelta(1, unit='hour')
            end_exog_window   = end_window + pd.Timedelta(steps, unit='hour')
            exog_window = exog.loc[start_exog_window:end_exog_window]
            exog_window = exog_window

            if verbose:
                print(f"Forecasting day {day.strftime('%Y-%m-%d')}")
                print(f"    Using window from {start_window} to {end_window}")
                print(f"    Using exogen variable from {start_exog_window} to {end_exog_window}")

            pres = forecaster.predict(steps=steps, last_window=last_window, exog=exog_window)

        pred = pred[dummy_steps:]
        backtest_predicciones.append(pred)

    backtest_predicciones = np.concatenate(backtest_predicciones)
    backtest_predicciones = pd.Series(
                            data  = backtest_predicciones,
                            index = pd.date_range(
                                        start = days_backtest[0],
                                        end   = days_backtest[-1].replace(hour=23),
                                        freq  = 'h'
                                    )
                        )

    return backtest_predicciones
```

Fuente: Elaboración en Python con la librería skforecast (2022)

Figura 12629: Paso 4: Examen – predicción diaria anticipada

```
In [24]: # Backtest
         # ==============================================================================
         predicciones = backtest_predict_next_24h(
                            forecaster = forecaster,
                            y          = datos.loc[fin_validacion:, 'EJECUTADO'],
                            hour_init_prediction = 11,
                            verbose    = False
                        )

         Backtesting starts at day: 2021-12-03 11:00:00
         Days predicted in the backtesting: ['2021-12-04' '2021-12-05' '2021-12-06' '2021-12-07' '2021-12-08'
          '2021-12-09' '2021-12-10' '2021-12-11' '2021-12-12' '2021-12-13'
          '2021-12-14' '2021-12-15' '2021-12-16' '2021-12-17' '2021-12-18'
          '2021-12-19' '2021-12-20' '2021-12-21' '2021-12-22' '2021-12-23'
          '2021-12-24' '2021-12-25' '2021-12-26' '2021-12-27' '2021-12-28'
          '2021-12-29' '2021-12-30']
```

Fuente: Elaboración en Python con la librería skforecast (2022)

5.2.2.2.3.4 Variable exógena

Figura 12730: Paso 2 y 3: Eligiendo el modelo y entrenando al alumno variable exógena

```
In [20]: # Crear y entrenar forecaster
         # ======================================================================
         forecaster = ForecasterAutoreg(
                        regressor = make_pipeline(StandardScaler(), Ridge(alpha=215.44)),
                        lags      = [1, 2, 3, 23, 24, 25, 47, 48, 49],
                      )

         forecaster.fit(y=datos.EJECUTADO[:fin_validacion], exog=datos.HOLIDAY[:fin_validacion])
         forecaster

Out[20]: ================
         ForecasterAutoreg
         ================
         Regressor: Pipeline(steps=[('standardscaler', StandardScaler()),
                        ('ridge', Ridge(alpha=215.44))])
         Lags: [ 1  2  3 23 24 25 47 48 49]
         Window size: 49
         Included exogenous: True
         Type of exogenous variable: <class 'pandas.core.series.Series'>
         Exogenous variables names: HOLIDAY
         Training range: [Timestamp('2017-01-01 01:00:00'), Timestamp('2021-11-30 23:00:00')]
         Training index type: DatetimeIndex
         Training index frequency: H
         Regressor parameters: {'standardscaler__copy': True, 'standardscaler__with_mean': True, 'standardscaler__with_std': True, 'ridg
         e__alpha': 215.44, 'ridge__copy_X': True, 'ridge__fit_intercept': True, 'ridge__max_iter': None, 'ridge__normalize': 'deprecate
         d', 'ridge__positive': False, 'ridge__random_state': None, 'ridge__solver': 'auto', 'ridge__tol': 0.001}
         Creation date: 2022-07-27 16:16:09
         Last fit date: 2022-07-27 16:16:09
         Skforecast version: 0.4.2
```

Fuente: Elaboración en Python con la librería skforecast (2022)

Figura 12831: Paso 4: Examen – variable exógena

```
In [22]: # Backtest
         # ======================================================================
         predicciones = backtest_predict_next_24h(
                            forecaster = forecaster,
                            y          = datos.loc[fin_validacion:, 'EJECUTADO'],
                            exog       = datos.loc[fin_validacion:, 'HOLIDAY'],
                            hour_init_prediction = 11,
                            verbose    = False
                        )

         Backtesting starts at day: 2021-12-03 11:00:00
         Days predicted in the backtesting: ['2021-12-04' '2021-12-05' '2021-12-06' '2021-12-07' '2021-12-08'
          '2021-12-09' '2021-12-10' '2021-12-11' '2021-12-12' '2021-12-13'
          '2021-12-14' '2021-12-15' '2021-12-16' '2021-12-17' '2021-12-18'
          '2021-12-19' '2021-12-20' '2021-12-21' '2021-12-22' '2021-12-23'
          '2021-12-24' '2021-12-25' '2021-12-26' '2021-12-27' '2021-12-28'
          '2021-12-29' '2021-12-30']
```

Fuente: Elaboración en Python con la librería skforecast (2022)

5.2.2.2.3.5 Variable exógena con predicción diaria anticipada

Figura 12932: Paso 2 y 3: Eligiendo el modelo y entrenando al alumno variable exógena con predicción diaria anticipada

```
In [30]: # Crear y entrenar forecaster
         # ==============================================================================
         forecaster = ForecasterAutoreg(
                         regressor = make_pipeline(StandardScaler(), Ridge(alpha=215.44)),
                         lags      = [1, 2, 3, 23, 24, 25, 47, 48, 49],
                     )

         exog = [column for column in datos.columns if column.startswith(('dia', 'hora', 'HOLIDAY'))]
         forecaster.fit(y=datos.EJECUTADO[: fin_validacion], exog=datos[exog][: fin_validacion])
```

Fuente: Elaboración en Python con la librería skforecast (2022)

Figura 13033: Paso 4: Examen – variable exógena con predicción diaria anticipada

```
In [33]: # Backtest
         # ==============================================================================
         predicciones = backtest_predict_next_24h(
                             forecaster = forecaster,
                             y          = datos.loc[fin_validacion:, 'EJECUTADO'],
                             exog       = datos.loc[fin_validacion:, exog],
                             hour_init_prediction = 11,
                             verbose    = False
                         )

         Backtesting starts at day: 2021-12-03 11:00:00
         Days predicted in the backtesting: ['2021-12-04' '2021-12-05' '2021-12-06' '2021-12-07' '2021-12-08'
          '2021-12-09' '2021-12-10' '2021-12-11' '2021-12-12' '2021-12-13'
          '2021-12-14' '2021-12-15' '2021-12-16' '2021-12-17' '2021-12-18'
          '2021-12-19' '2021-12-20' '2021-12-21' '2021-12-22' '2021-12-23'
          '2021-12-24' '2021-12-25' '2021-12-26' '2021-12-27']
```

Fuente: Elaboración en Python con la librería skforecast (2022)

5.2.2.2.4 Medición

5.2.2.2.4.1 Forecast inicial

Figura 13134: Paso 5: Backtest e intervalo de cobertura forecast inicial

```
In [218]: # Error backtest
          # ==============================================================================
          print(f'Error backtest: {metrica}')

          Error backtest: [260.46514909]
```

```
In [223]: # Cobertura del intervalo predicho
          # ==============================================================================
          dentro_intervalo = np.where(
                               (datos.loc[fin_validacion:, 'EJECUTADO'] >= predicciones['lower_bound']) & \
                               (datos.loc[fin_validacion:, 'EJECUTADO'] <= predicciones['upper_bound']),
                               True,
                               False
                             )

          cobertura = dentro_intervalo.mean()
          print(f"Cobertura del intervalo predicho: {round(100*cobertura, 2)} %")

          Cobertura del intervalo predicho: 81.34 %
```

Fuente: Elaboración en Python con la librería skforecast (2022)

5.2.2.2.4.2 Grid Search

Figura 13235: Paso 5: Backtest Grid Search

```
In [19]: # Error backtest
         # ==================================
         print(f'Error backtest: {metrica}')

         Error backtest: [233.65687989]
```

Fuente: Elaboración en Python con la librería skforecast (2022)

5.2.2.2.4.3 Predicción anticipada diaria

Figura 13336: Paso 5: Backtest predicción anticipada diaria

```
In [26]: # Error backtest
         # ==============================================================================
         error = mean_absolute_error(
                     y_true = datos.loc[predicciones.index, 'EJECUTADO'],
                     y_pred = predicciones
                 )
         print(f"Error de backtest: {error}")

         Error de backtest: 289.46852642807875
```

Fuente: Elaboración en Python con la librería skforecast (2022)

5.2.2.2.4.4 Variable exógena

Figura 13437: Paso 5: Backtest variable exógena

```
In [24]: # Error backtest
         # ======================================================================
         error = mean_absolute_error(
                     y_true = datos.loc[predicciones.index, 'EJECUTADO'],
                     y_pred = predicciones
                 )

         print(f"Error de backtest: {error}")

         Error de backtest: 275.29841121517325
```

Fuente: Elaboración en Python con la librería skforecast (2022)

5.2.2.2.4.5 Variable exógena con predicción diaria anticipada

Figura 13538: Paso 5: Backtest variable exógena con predicción diaria anticipada

```
In [35]: # Error backtest
         # ======================================================================
         error = mean_absolute_error(
                     y_true = datos.loc[predicciones.index, 'EJECUTADO'],
                     y_pred = predicciones
                 )

         print(f"Error de backtest: {error}")

         Error de backtest: 169.5695889579889
```

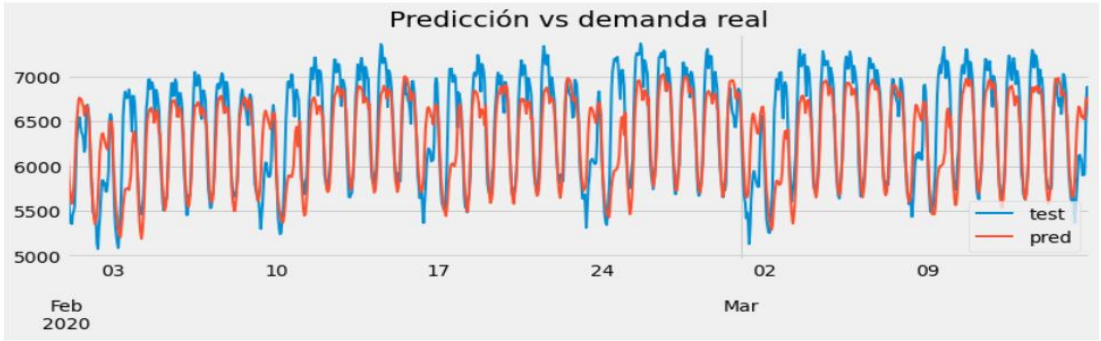
Fuente: Elaboración en Python con la librería skforecast (2022)

5.2.2.2.5 Gráficos

5.2.2.2.5.1 Forecast inicial

Figura 13639: Paso 6: Programación y gráfica del forecast inicial

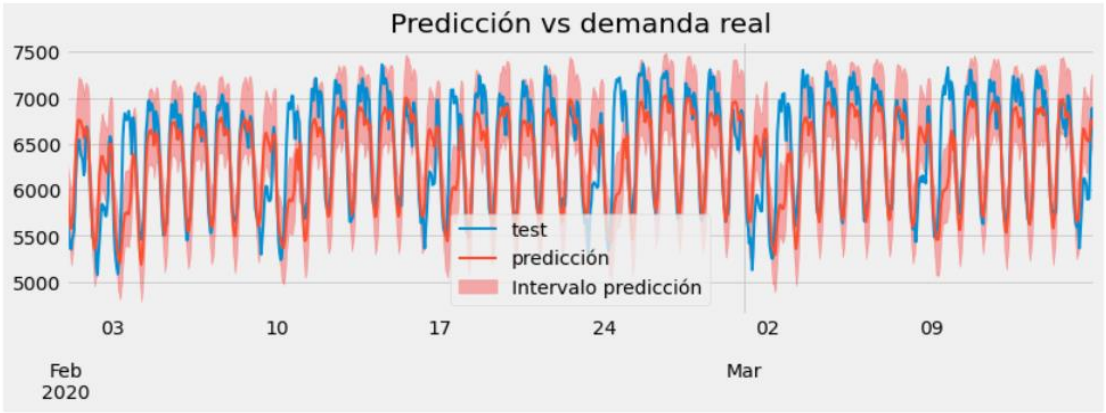
```
In [217]: # Gráfico
          # ======================================================================================
          fig, ax = plt.subplots(figsize=(12, 3.5))
          datos.loc[predicciones.index, 'EJECUTADO'].plot(ax=ax, linewidth=2, label='test')
          predicciones.plot(linewidth=2, label='predicción', ax=ax)
          ax.set_title('Predicción vs demanda real')
          ax.legend();
```



Fuente: Elaboración en Python con la librería matplotlib.pyplot (2022)

Figura 13740: Gráfico intervalo de predicción del forecast inicial

```
In [221]: # Gráfico
          # ======================================================================================
          fig, ax = plt.subplots(figsize=(12, 3.5))
          datos.loc[predicciones.index, 'EJECUTADO'].plot(linewidth=2, label='test', ax=ax)
          predicciones['pred'].plot(linewidth=2, label='predicción', ax=ax)
          ax.set_title('Predicción vs demanda real')
          ax.fill_between(
              predicciones.index,
              predicciones['lower_bound'],
              predicciones['upper_bound'],
              alpha = 0.3,
              color = 'red',
              label = 'Intervalo predicción'
          )
          ax.legend();
```

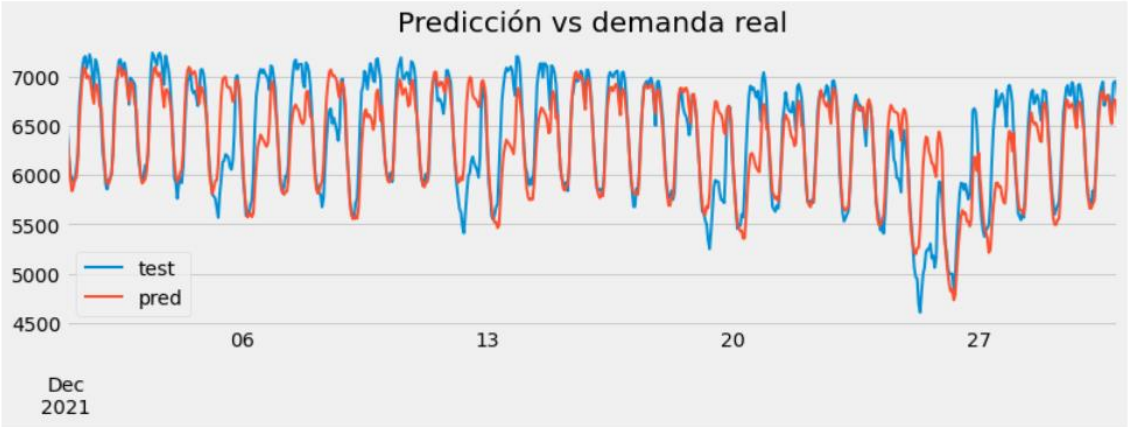


Fuente: Elaboración en Python con la librería matplotlib.pyplot (2022)

5.2.2.2.5.2 Grid Search

Figura 13841: Programación y gráfica del Grid Search

In [18]:
```
# Backtest
# ==============================================================================
metrica, predicciones = backtesting_forecaster(
                            forecaster = forecaster,
                            y          = datos.EJECUTADO,
                            initial_train_size = len(datos[:fin_validacion]),
                            steps      = 24,
                            metric     = 'mean_absolute_error',
                            refit      = False,
                            verbose    = False
                        )

fig, ax = plt.subplots(figsize=(12, 3.5))
datos.loc[predicciones.index, 'EJECUTADO'].plot(linewidth=2, label='test', ax=ax)
predicciones.plot(linewidth=2, label='predicción', ax=ax)
ax.set_title('Predicción vs demanda real')
ax.legend();
```

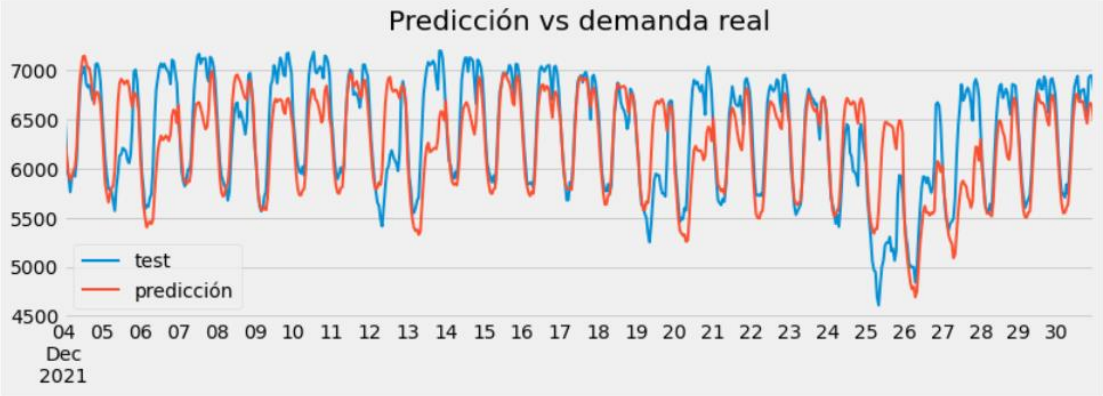


Fuente: Elaboración en Python con la librería matplotlib.pyplot (2022)

5.2.2.2.5.3 Predicción diaria anticipada

Figura 13942: Programación y gráfica del predicción diaria anticipada

In [25]:
```
# Gráfico
# ==============================================================================
fig, ax = plt.subplots(figsize=(12, 3.5))
datos.loc[predicciones.index, 'EJECUTADO'].plot(linewidth=2, label='test', ax=ax)
predicciones.plot(linewidth=2, label='predicción', ax=ax)
ax.set_title('Predicción vs demanda real')
ax.legend();
```

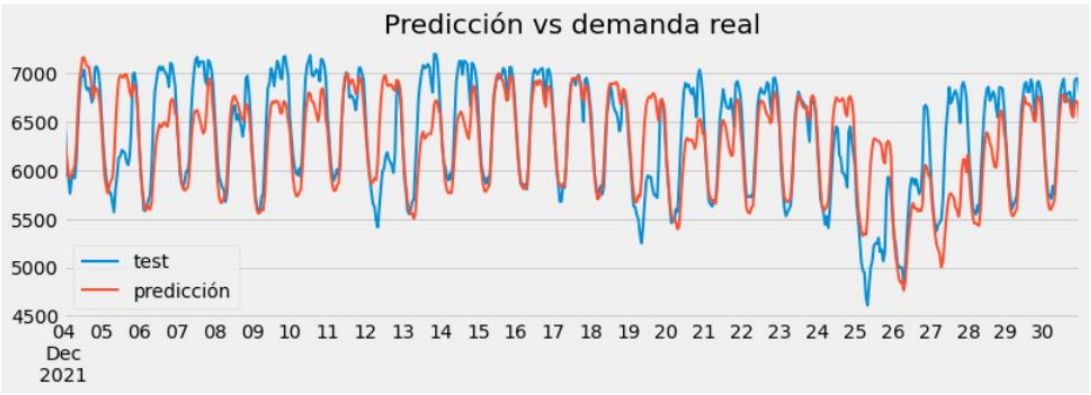


Fuente: Elaboración en Python con la librería matplotlib.pyplot (2022)

5.2.2.2.5.4 Variable exógena

Figura 14043: Programación y gráfica de la variable exógena

In [23]:
```
# Gráfico
# ==============================================================================
fig, ax = plt.subplots(figsize=(12, 3.5))
datos.loc[predicciones.index, 'EJECUTADO'].plot(linewidth=2, label='test', ax=ax)
predicciones.plot(linewidth=2, label='predicción', ax=ax)
ax.set_title('Predicción vs demanda real')
ax.legend();
```

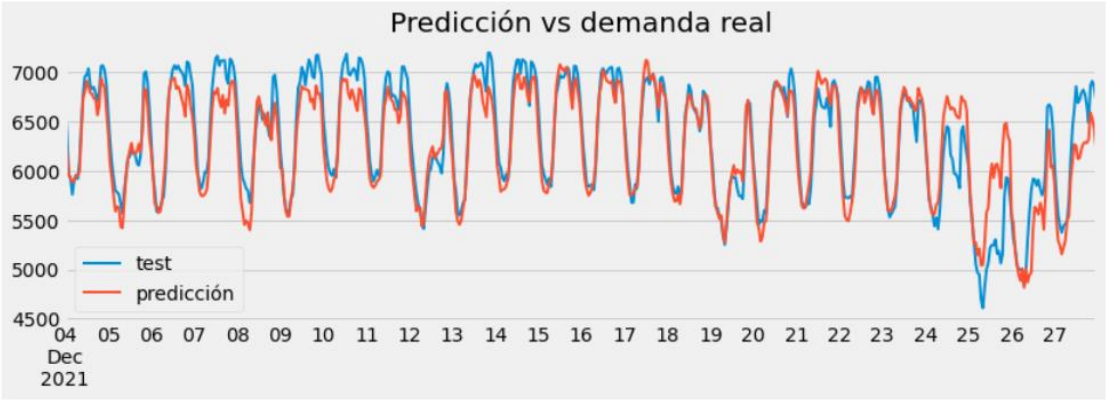


Fuente: Elaboración en Python con la librería matplotlib.pyplot (2022)

5.2.2.2.5.5 Variable exógena con predicción diaria anticipada

Figura 14144: Programación y gráfica de la variable exógena con predicción diaria anticipada

In [34]:
```
# Gráfico
# ======================================================================
fig, ax = plt.subplots(figsize=(12, 3.5))
datos.loc[predicciones.index, 'EJECUTADO'].plot(linewidth=2, label='test', ax=ax)
predicciones.plot(linewidth=2, label='predicción', ax=ax)
ax.set_title('Predicción vs demanda real')
ax.legend();
```



Fuente: Elaboración en Python con la librería matplotlib.pyplot (2022)

Capítulo VI: Conclusiones y Recomendaciones

6.1 Conclusiones

La demanda eléctrica es un tema de vital importancia a nivel mundial, debido a su alto grado porcentual de afectación con respecto al cambio climático que se suscita en todos los continentes, a partir de la emisión de gases por efecto invernadero. En el mundo empresarial, debido al crecimiento de modelos de negocio que trabajan en función a la digitalización y nuevas tecnologías, el consumo de energía eléctrica viene representando un gran valor con respecto a la estructura de costos en cada una de las áreas que implica cada empresa, ya sea económico o intangible, por lo cual, muchas de ellas vienen aplicando estrategias operacionales y culturales que permitan concientizar sobre el impacto de cada actividad que realizan dentro y fuera de la empresa con respecto al medio ambiente.

En dicho contexto, se ha encontrado en la empresa Nexa Cajamarquilla, a partir de una investigación exhaustiva de sus operaciones, una problemática constante con respecto al consumo de energía eléctrica, lo cual no se ha podido solucionar porque existen factores externos que no son controlables como el consumo de energía eléctrica de casas, empresas, fábricas, entre otros, que influyen en el comportamiento del consumo de electricidad en la zona. Por ello, en este trabajo se realizó una propuesta de modelo para pronosticar o predecir mediante la técnica de series de tiempo los valores de la demanda máxima de energía eléctrica, teniendo como datos los consumos de Kwh desde el año 2020 al año 2022, sin tomar en cuenta los datos en la etapa de la pandemia Covid 19. A partir del lenguaje de programación de Python, mediante la técnica de lenguaje supervisado, se pudo obtener a partir de datos reales, valores pronosticados con cierto grado de variación con respecto a la data real, pero muy similares en términos cuantitativos y gráficos, lo cual estuvo demostrado a partir de la evaluación de métricas del error (MSE,MAE,RMSE).

Finalmente, a partir de los valores obtenidos mediante el lenguaje de backtest, se pudo determinar que el modelo aún mantenía un alto porcentaje de error en base a la comparación de los datos reales y los datos pronosticados. Para ello, se utilizaron variables exógenas que permitan ajustar aún más el modelo, ya que se le brinda mayor cantidad de información al modelo en estudio. Asimismo, se utilizaron técnicas como Ridge y Grid, los cuales ayudan a eliminar o reducir el valor de los coeficientes de cada una de las variables y la optimización del

modelo mediante la inclusión de lags e hiperparámetros, respectivamente. Estas dos herramientas permitieron ajustar aún más el modelo, lo cual se pudo demostrar y visualizar en la prueba de backtest, en la gráfica de predicción y los indicadores del error del modelo. Con ello, se pudo demostrar que el modelo estaba alineado al comportamiento de los datos y a los objetivos de la investigación, con lo cual ya se podía obtener data importante para poder tomar decisiones con respecto a la dosificación de la producción y, por ende, de la energía, con tiempos y fechas más acordes a los suscitado en años anteriores.

6.2 Recomendaciones

De acuerdo a los resultados obtenidos y a las investigaciones realizadas en base a otros trabajos similares, además de comentarios de expertos con respecto al estudio de la problemática de la investigación, es factible poder desarrollar aún mejoras en el modelo de programación que permitan tener a cabalidad datos más exactos con respecto al mes, fecha y hora tentativa en que se suscite la demanda máxima de energía en la refinería de Nexa. Las mejoras se basan en lo siguiente:

- Añadir mayor cantidad de variables exógenas y endógenas que permitan ajustar aún más el modelo de pronóstico en función a la programación aplicada. Variables como temperatura de la maquinaria, KwH promedio por zona aledaña, entre otros. Esto permitiría tener una media cuadrática de error más cercana a 0, lo que indicaría que el modelo se ajusta en un mayor porcentaje al comportamiento de los datos que se le ha brindado como aprendizaje.
- Añadir el estudio de la estacionalidad de los datos mediante gráfico de Siles o histograma que permitan tener lectura de los residuos, los cuales se determinan a partir de la comparación de la demanda real con la demanda proyectada. Esto es otro argumento que sostiene la optimización del modelo de acuerdo a las características de los datos, por lo cual, es una nueva manera de explicar la aplicación de series de tiempo en función a otras técnicas de pronóstico que se podrían plantear utilizar.
- La utilización de cortes verticales en la gráfica de tiempos-KwH mediante un lenguaje de programación que dictamine el estudio del modelo de acuerdo a un intervalo de tiempo, va a permitir reconocer y realizar los ajustes necesarios para observar el comportamiento del pronóstico, además de evaluar los picos de demanda de acuerdo a intervalos más pequeños, lo que ayudaría a encontrar una mejor manera de determinar el mes, la fecha y la hora potencial del suceso en estudio.

- Por último, realizar gráficas de Pareto que permitan tener una imagen visual y porcentual del mes, la fecha y la hora potencial en la que va a ocurrir la demanda máxima, a partir del estudio de comportamiento de los datos reales y pronosticados, da soporte al modelo con respecto a los resultados buscados en el estudio, teniendo con un alto grado de probabilidad la fecha en que se va a suscitar la demanda máxima. Con ello, ya se podría trabajar el programa de dosificación de producción para evitar esos picos de energía y, por ende, el sobre costo generado.

Referencia Bibliográfica

- Academia Lab (s.f) Gráfico de Violín. 2022 Academia Lab CC BY-NC-ND. [Texto en un blog] Recuperado de: https://academia-lab.com/enciclopedia/grafico-de-violin/

- Ancco Y., Tatiana (2021) Análisis comparativo de series de tiempo para proyectar las ventas en las jerarquías de calzado en una empresa del sector retail. Universidad Nacional Mayor de San Marcos, 63, 14-57. Recuperado de: https://cybertesis.unmsm.edu.pe/bitstream/handle/20.500.12672/17282/Ancco_yt.pdf?sequence=1&isAllowed=y

- Bagnato, J. (2018). Regresión Lineal en español con Python. Obtenido de: https://www.aprendemachinelearning.com/tag/regresion-lineal/

- Banco Mundial (2020). La producción minera se dispara con el aumento de la demanda de energía limpia. Obtenido de: https://www.bancomundial.org/es/news/press-release/2020/05/11/mineral-production-to-soar-as-demand-for-clean-energy-increases

- Barría, C (2022). La carrera por los codiciados "minerales del futuro" que pueden crear gigantescas fortunas e influir en la seguridad nacional de los países. Obtenido de: https://www.bbc.com/mundo/noticias-61144362

- Bobadilla, J. (2020). Machine Learning y Deep Learning. Madrid, España: Ra-ma

- Cancino, C. (2012). Matriz de Análisis FODA cuantitativo. Obtenido de: https://christiancancino.cl/wp-content/uploads/2016/09/MATRIZ-DCS-FODA-CUANTITATIVA.pdf

- Cerna Ramirez, C. (11 de febrero de 2021). Series de tiempo - periodos homogéneos. Universidad de Católica del norte, Chile. Obtenido de: https://slideplayer.es/slide/13782716/

- Coca C, Andrés (2011) La demanda. Una perspectiva de marketing: reflexiones conceptuales y aplicaciones. Universidad Católica Boliviana San Pablo-Bolivia. Recuperado de: https://www.redalyc.org/pdf/4259/425941257008.pdf

- Conexión ESAN (2019) Diagrama de Dispersión: ¿cómo usar esta herramienta de control de calidad? [Texto en un blog] Recuperado de:

https://www.esan.edu.pe/conexion-esan/diagrama-de-dispersion-como-usar-esta-herramienta-de-control-de-calidad

- Conexión Esan (08 de agosto de 2018). Minería de datos: ¿en qué consiste el knowledge discovery in databases?. [Texto en un blog]. Recuperado de: https://www.esan.edu.pe/conexion-esan/mineria-de-datos-en-que-consiste-el-knowledge-discovery-in-databases

- Del Campo, Rubén (2020). Técnicas de Machine Learning aplicadas a la predicción de los desvíos del Mercado Eléctrico. Obtenido de: https://repositorio.uam.es/bitstream/handle/10486/693446/del_campo_hernando_ruben_tfg.pdf?sequence=1

- D. Ramón (2019). Métricas de regresión para el aprendizaje automático. Universidad Católica. Chile. Recuperado de: https://topbigdata.es/metricas-de-regresion-para-el-aprendizaje-automatico/

- Ejaz Ui, Haq & Xue, Lyu & Youwei, Jia & Mengyuan, Hua & Fiaz, Ahmad (2020). Forecasting household electric appliances consumption and peak demand based on hybrid machine learning approach. Obtenido de: https://www.sciencedirect.com/science/article/pii/S2352484720314967

- Gonzáles C, Maria (2009) Análisis de series temporales: Modelos ARIMA. Universidad del País Vasco. Departamento de Economía Aplicada III. SARRIKO ON. Recuperado de: https://addi.ehu.es/bitstream/handle/10810/12492/04-09gon.pdf

- Gutierrez, L. (2020). Predicción de múltiples series de tiempo univariadas a través de diversos modelos predictivos y meta-learning aplicado en la industria del retail. Obtenido de: https://repositorio.uchile.cl/bitstream/handle/2250/177531/Prediccion-de-multiples-series-de-tiempo-univariadas-a-traves-de-diversos-modelos-predictivos-y-meta-learning-aplicado-en-la-industria-del-retail.pdf?sequence=1&isAllowed=y

- Joaquin Amat, Rodirgo (Noviembre 2020) Regularización de Ridge, Lasso, Elastic Net Con Python .Obtenido de https://www.cienciadedatos.net/documentos/py14-ridge-lasso-elastic-net-python.html

- Namir, K. Labriji,L y Ben,E. (2021). Decision Support Tool for Dynamic Inventory Management using Machine Learning, Time Series and Combinatorial Optimization. Obtenido de: https://www.sciencedirect.com/science/article/pii/S1877050921025035

- Orza Cauto, Antonio (2021). La electricidad: conceptos, fenómenos y magnitudes eléctricas. Obtenido de https://www.edu.xunta.gal/centros/cpiantonioorzacouto/system/files/TEMA%202%20LA%20ELECTRICIDAD%20I.pdf

- Palloneto,F., Jin,C. y Mangina,E. (2022) Forecast electricity demand in commercial building with machine learning models to enable demand response programs. Energy and AI. 1-13. Recuperado de : https://reader.elsevier.com/reader/sd/pii/S2666546821000690?token=ACE48646DED4F80CBB5EAD3A77A1E361020E0521577DB8B49527F55F31C2B9A2DBE7CC213FBC0536F1C85ECB018F9E72&originRegion=us-east-1&originCreation=20220728201804

- Parkin, M. (2009) Economía. University of Western Ontario. Recuperado de: http://recursosbiblio.url.edu.gt/publimjrh/ECONOMIA%20-%20MICHAEL%20PARKIN/cap/03.pdf

- Pérez, E. (2022). El precio de los metales se derrumba como no se había visto desde 2008: un primer aviso del riesgo de recesión. Obtenido de: https://www.xataka.com/empresas-y-economia/precio-metales-se-derrumba-como-no-se-habia-visto-crisis-2008-primer-aviso-riesgo-recesion

- Ponce, J. Torres,A. Quezada, F. Silva, A. Martinez, E. Casali, A. Scheihing,E. Túpac, I. Torres, M. Oernelas, F. Hernandez, J. Zavala, C. Vakhnia, N. Pedreno O., (2014). Inteligencia Artificial.

- Ramón S, Gustavo (2000) Correlación entre variables. Universidad de Antioquía Colombia. Recuperado de: http://viref.udea.edu.co/contenido/menu_alterno/apuntes/ac36-correlacion-variables.pdf

- Ric Energy (2022). El Impacto de la relación de consumo y demanda al costo de energía de tu negocio. Obtenido de: https://www.ric.mx/cultura/energia/mexico/impacto-de-relacion-de-consumo-y-demanda-al-costo-energia-de-negocio/

- Ric Energy (2022). El futuro de la energía. Obtenido de: https://www.ric.mx/cultura/energia/el-futuro-de-la-energia/

- Sosa, M. (2007). Inteligencia artificial en la gestión financiera empresarial. Pensamiento y Gestión.

- VanderPlas, J. (2017). Python Data Science Handbook. USA: O'Reilly Media.

- Videla C, Ximena (2017) Medidas de posición y gráfico de caja y bigote: una propuesta didáctica. Pontificia Universidad Católica de Valparaíso. Facultad de Ciencias Instituto de Matemáticas. Recuperado de: http://opac.pucv.cl/pucv_txt/txt-2500/UCC2616_01.pdf

- Villarreal, Fernanda (2016) Introducción a los modelos predictivos. Universidad Nacioanl del Sur - Departamento de Matemáticas. Recuperado de: https://www.matematica.uns.edu.ar/uma2016/material/Introduccion_a_los_Modelos_de_Pronosticos.pdf

Anexos

Anexo 1: Reunión con especialista de procesos electrometalúrgicos