EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Nuevo León

INSTITUTO TECNOLÓGICO DE NUEVO LEÓN

División de Estudios Profesionales

Trabajo de Titulación

Opción TI Informe Técnico de Residencia Profesional por Tesis

Proyecto: “Evaluación de algoritmo Prophet para el seguimiento de variables críticas en un proceso de manufactura.”

ALUMNO(S):	Jesús Alberto Reyes Hernández
No. CONTROL:	18480577
CARRERA:	Ingeniería en Sistemas Computacionales
ASESOR DE RESIDENCIA:	Dr. José Isidro Hernández Vega

Guadalupe, N.L. Enero, 2023

Departamento: **Sistemas y Computación**
Cd. Guadalupe, N.L., a 14 de Septiembre de 2022

Asunto: **Solicitud de registro de tesis**

Ing. Magaly Benítez Tamez
Jefa de Departamento de Sistemas y Computación del I.T de Nuevo León
Presente.

Por este conducto solicito a usted sea registrada la tesis que llevaré a cabo bajo la línea de investigación de Tecnologías de Información y Comunicación. La realización de esta tesis servirá como el producto académico para la obtención del grado de Ingeniero en Sistemas Computacionales en este Instituto Tecnológico de Nuevo León. Anexo a la solicitud presento a usted mi anteproyecto de investigación.
Los datos de registro de la tesis son:

a) Nombre del estudiante	**Jesús Alberto Reyes Hernández** No. de control: **18480577**
b) Programa Educativo	**Ingeniería en Sistemas Computacionales**
c) Línea de Generación y Aplicación del Conocimiento	**Tecnologías de la Información y Comunicación CLAVE: LGAC-2017-NLEO-ISCO-15**
d) Titulo de tesis:	**Evaluación de algoritmo Prophet para el seguimiento de variables críticas en un proceso de manufactura**
e) Periodo de realización:	**Del 01-Junio-2022 al 31-Enero-2023**

Así mismo, le informo que para este proyecto mi comité de seguimiento de tesis es:

Director de tesis: Dr. José Isidro Hernández Vega
Comité tutorial: M.C Elda Reyes Varela, Ing. Luis Alejandro Reynoso Guajardo

Agradezco de antemano su valioso apoyo para el registro de este proyecto académico.

Atentamente

C. Jesús Alberto Reyes Hernández

Estudiante de la carrera de Ingeniería en Sistemas Computacionales
No. de control: **18480577**

Vo.Bo.

Dr. José Isidro Hernández Vega
Director de Tesis

SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE NUEVO LEÓN
14 SEP 2022
DEPARTAMENTO DE INGENIERÍA EN SISTEMAS Y COMPUTACIÓN
RECIBIDO

c.c.p. Archivo
c.c.p Director de Tesis
c.c.p Estudiante

RESUMEN.

La razón por la que se realiza este proyecto es debido a que el machine learning es una de las herramientas que dentro de unos años será más habitual encontrarla en nuestra vida cotidiana por lo que es necesario investigar sus aplicaciones, los algoritmos de machine learning nos ofrecen analizar, estimar y predecir comportamientos de datos, una aplicación de estos algoritmos se da en los sistemas de producción.

El objetivo general de la investigación consistió en desarrollar una aplicación que analizó dentro de un periodo de tiempo variables dentro de un proceso, usando el algoritmo de predicción Prophet, evaluar la calidad del algoritmo probándolo con datos reales y comparándolo con el algoritmo de Bosques aleatorios, así como también se identificó su confiabilidad con ayuda de las métricas de errores mape (Error Absoluto Medio Porcentual), mae (Error Absoluto Medio) y mse (Error Cuadrático Medio), donde finalmente se evaluaron los resultados de los experimentos y de su evaluación.

Se trabajo bajo una metodología del proceso de desarrollo de Machine Learning, se trabajó con un proceso sistemático, en el cual primero se entendió el problema para preparar los datos, se limpiaron y filtraron los datos eliminando en los data sets valores en cero o el orden en el que se mostraba la fecha, una vez terminando con la depuración y limpieza de los datos, se siguió con la construcción del modelo en el cual funcionaron los datasets, se analizaron los errores con ayuda de las métricas para poder limpiar el código hasta llegar a una aplicación adecuada para predecir las variables y dar una salida que consistió en el pronóstico de las variables de entrada.

Una vez dados los resultados de los experimentos se procedió a evaluar el algoritmo tomando en cuenta las métricas y gráficas de los tres experimentos para el posterior análisis y comparación con el algoritmo de bosques aleatorios.

Se consiguió adaptar el algoritmo en tres experimentos distintos que consistieron en un conjunto de sensores de temperatura, el caso de la temperatura de un horno y la temperatura promedio de dos motores en un proceso de manufactura en donde se expusieron sus respectivas gráficas y métricas, se discutió acerca de su efectividad tanto de manera individual como en comparación con el algoritmo de bosques aleatorios al menos en los primeros dos experimentos debido a que se usó el mismo dataset en dichos experimentos.

Una vez concluidos los experimentos se permitió analizar las capacidades que tienen el algoritmo Prophet en comparación del Random Forest así como de resaltar sus virtudes y sus casos de uso.

En donde se llegó a la conclusión de que el algoritmo Prophet, junto con un dataset adecuado para el uso de este algoritmo y sumado a una cantidad de datos proporcional, el algoritmo es el adecuado para hacer un pronostico de distintas variables el cual fue demostrado con las métricas de manera cuantitativa, sumado a la facilidad con la que el algoritmo puede ser manejado por su cualidad semiautomática de utilizarse .

PALABRAS CLAVE:

Machine Learning, pronóstico de fallas, Prophet, monitoreo de variables.

ABSTRACT.

The purpose for this project is that machine learning is one of the tools that in the next years will be more common in our daily lives, so it is necessary to investigate one of the advantages that these algorithms offer us as is the production sector.

The general objective of the research consisted of developing an application that analyzes in a period of time variables within a process using the Prophet prediction algorithm, as well as identifying the quality of the algorithm by testing it in different real cases and comparing it with the Random Forests algorithm to analyze it, as well as identifying its reliability with the help of the error metrics mape, mae and mse, where finally the results of the experiments and their evaluation were evaluated.

A systematic process was implemented where the problem was first understood to prepare the data, which were necessary since obstacles in the data sets had to be changed, such as zero values or the order in which the date was shown. Once the debugging and cleaning of the data were finished, the construction of the model in which the datasets worked was continued, the errors were analyzed with the help of the metrics to be able to clean the code until reaching an adequate application to predict the variables and give an output that consisted of the forecast of the input variables.

Once the results of the experiments were given by the process, the algorithm was evaluated considering the metrics and graphs of the three experiments for later analysis and comparison with the random forest algorithm.

The algorithm was adapted in three different cases consisting of a set of temperature sensors within a time, the case of the temperature of an oven and the average temperature of two engines in a manufacturing process where their respective graphs and metrics were exposed, where its effectiveness was discussed both individually and in comparison with the Random Forest algorithm at least in the first two experiments because the same dataset was used in these experiments but not in the case of the third one.

Once the experiments were concluded, it was possible to analyze the capabilities of the Prophet algorithm compared to the Random Forest algorithm, as well as to highlight its virtues and use cases.

In which it was concluded that the Prophet algorithm, together with a suitable dataset for the use of this algorithm and added to a proportional amount of data, the algorithm is adequate to make a forecast of different variables which was demonstrated with the metrics in a quantitative way, added to the ease with which the algorithm can be handled by its semi-automatic quality of handling.

KEYWORDS:

Machine learning, Failure forecasting, Prophet.

AGRADECIMIENTOS

Me gustaría darle las gracias a mi mamá Rosalía Alejandra Hernández González, que siempre me apoyo en todo momento, en los buenos y malos momentos desde mis primeros años en mi formación académica, no estuviera redactando este documento de no haber sido por su arduo trabajo y apoyo. Así como también quisiera agradecer a mi compañera Veronica Lucia Medina Rios, por el apoyo que me dio en todo el transcurso de realizar este proyecto.

También quisiera agradecer al doctor José Isidro Hernández Vega por haber confiado en mi trabajo, asesorarme constantemente e involucrarse en el proyecto que redactaré a continuación, además, de proponer los cursos impartidos por la CII.A que me ayudaron a mejorar mis técnicas en Python, con las librerías de pandas, Numpy, SciKitLearn, entre otras además de ser acreedor de un certificado que puede ser de mucha utilidad para mi desarrollo laboral.

INDICE

INDICE DE ILUSTRACIONES

INDICE DE ECUACIONES

INDICE DE TABLAS

CAPITULO I. DESCRIPCIÓN DEL PROYECTO

Introducción

Hoy en día, situados en la industria 4.0, en la cual el uso de tecnologías es factor clave para el desarrollo de componentes dependiendo del sector industrial, se han implementado diversas herramientas como el internet de las cosas, así como la big data y la nube, dichas herramientas tiene como objetivo la optimización de la producción en aspectos como la manufacturación, distribución o incluso revisión de los productos, así como también del mantenimiento de las herramientas físicas que realizan dichas actividades dentro de la producción.

Como ayuda dentro de la línea de producción es posible el uso de algoritmos de predicción para el mantenimiento de las herramientas que realizan dicho proceso, un ejemplo de algoritmo es Prophet, el cual es uno que se encarga del análisis de series temporales, lo cual indica que puede servir de ayuda para casos en donde se análisis tomando en cuenta un periodo de tiempo extenso, con observaciones históricas donde eventos irregulares determinen la predicción de los acontecimientos (Amazon Web Services, Inc., 2021).

Como motivación, se busca ampliar el conocimiento acerca de algoritmo de análisis de series temporales Prophet y poder así aplicar el algoritmo Prophet en una línea de producción, analizando el mantenimiento de la maquinaria haciendo ahorro de gastos en reparación o reajustes que muchas veces en mantenimientos preventivos se realizan por mera precaución e incluso cuando no se presenta una real necesidad sino por un protocolo, lo cual la implementación del algoritmo a mediano plazo implicarían ahorros multimillonarios para la empresa.

Se debe de conocer más a detalle las capacitades y limitantes que se podrían tener con el algoritmo Prophet, así como el estudio de casos de uso que este algoritmo pude tener y adecuar nuestro caso con los valores que el algoritmo necesité en caso de ser posible. Es necesario también la recolección de los datos de las maquinarias para poder poner en práctica el algoritmo.

Poner en práctica el algoritmo de análisis de series temporales Prophet en un caso de un proceso de manufactura en el uso de un mantenimiento predictivo, así como también la evaluación de los experimentos con ayuda de la validación cruzada así como de las métricas para el análisis de sus confiabilidad.

Antecedentes

En el trabajo de Aplicaciones de machine learning en el mantenimiento predictivo industrial con herramientas de código abierto de los autores Romero Gelvez y Rincón Quintero (2020) pusieron en práctica el algoritmo Prophet para el análisis de unos datos de rodamiento de un repositorio del centro de investigación Ames de la NASA en Moffett, California (Conjunto de datos de rodamiento) para la manipulación y predicción de sus resultados además de calcular los errores del modelo, Porcentual Absoluto Medio, Porcentual medio, Absoluto Medio, Cuadrático Medio. Como conclusiones llegaron a definir que el uso del algoritmo Prophet en una estrategia de mantenimiento predictivo puede llegar a reducir los costos de operación.

En otro documento Extracción y predicción de datos de series temporales de reservas de vuelo de Mirete Blanco (2019) se extrajeron varios datos de reservas de vuelos de diversos destinos para su análisis y predicción usando como modelo a Prophet basado en el lenguaje R usando Rmarkdown y flexdashboard para la visualización estadística en formato HTML, así como también del paquete leaflet para la visualización de un mapa de los destinos.

También en un estudio de Galmés Mifsud (2019) se compararon varios algoritmos de series temporales para el análisis de series en base a días, meses y cada cinco minutos en donde Prophet destacó de entre todos los otros algoritmos especialmente en el de series diaria, no así en las series mensuales ni cinco-minutales en donde se tuvo que modificar los datos y valoraciones del análisis para reducir el nivel de error, lo cual desmiente en este caso la predicción automática (automatic forecasting) que se presume como una de las ventajas de Prophet by Facebook.

Planteamiento del problema a resolver

Dentro del área de producción en los procesos de manufactura de cualquier sector industrial, es necesario el mantenimiento dentro de un tiempo de las herramientas con las que se realizan las actividades, estas pueden ser del tipo de mantenimiento predictivo y preventivo, en los cuales el mantenimiento predictivo normalmente recae en que no siempre se hacen por necesidad latente o evidente, sino más bien por un protocolo establecido dentro de la empresa.

Mientras tanto, el mantenimiento predictivo se encarga de la antelación de desastres o errores por medio de un análisis de datos basados en hechos previo en donde se recopilan datos que podrían afectar al deterioro de las máquinas de una línea de producción. Para esto, se ha pensado en enfocarse primordialmente en el mantenimiento predictivo, usando en el primer algoritmo de machine learning que puedan predecir mediante variables un aproximado del momento en el que la maquina requerirá del mantenimiento.

Uno de estos algoritmos es el de Prophet, un algoritmo que se encarga de analizar variables dentro de un periodo de tiempo, el cual puede que analice el comportamiento y rendimiento de una maquina dentro de un periodo de tiempo e incluso graficar los resultados para el análisis de la empresa.

El objetivo es el de desarrollar una aplicación que analice en un periodo de tiempo las variables identificadas como críticas de un proceso de manufactura mediante el algoritmo Prophet.

Objetivos

Objetivo general

Evaluar el algoritmo prophet mediante el desarrollar una aplicación que analice en un periodo de tiempo las variables identificadas como críticas de un proceso de manufactura y su comparación con el algoritmo de bosques aleatorios.

Objetivos específicos

Los objetivos específicos son los siguientes:

- Representar la solución mediante un algoritmo.
- Identificar la función de calidad
- Implementar el algoritmo en una herramienta de programación
- Probar el algoritmo
- Evaluar sus resultados y compararlo con el algoritmo de bosques aleatorios.

Justificación

Los algoritmos de machine learning son una tecnología que eventualmente se usará en muchos sectores de la población por lo que se quiere investigar los beneficios que tendría una empresa o línea de producción con el uso de estos para la detección de fallas y su nivel de error hoy en la actualidad, especialmente la factibilidad que estos algoritmos tienen hoy en día con ese sector industrial y su posible implementación a corto o mediano plazo.

Actualmente el algoritmo Prophet se puede utilizar para la predicción de datos dentro de una serie de tiempo, lo que se busca con esa información es la aplicación de este algoritmo dentro de una línea de producción para la detección de errores relacionados con la temperatura, presión, humedad y otros datos que se puedan analizar mediante el uso de sensores.

Se puede construir metodológicamente la solución debido a que el proceso del análisis de los datos dentro del programa en Python requiere que los pasos sean en ese orden, ya que antes de analizar los datos primero hay que extraerlos en formato CSV, de lo cual los sensores se harán cargo de la recolección de estos datos que serán después transformados en CSV para después la extracción y predicción de los datos.

El uso de este tipo de algoritmos incentivaría a las empresas que se desempeñan en producción al uso de estos algoritmos para la detección de errores y el posible uso de estos dentro de un mantenimiento predictivo, además de que podemos ver como investigadores el potencial que estos tienen en un principio para este tipo de escenarios.

Alcances y limitaciones

El proyecto requerirá de componentes de hardware que en este caso serán de una computadora de escritorio o laptop para el desarrollo del documento de texto, también se ocupara del Google Colab para el desarrollo del programa en lenguaje Python y de disposición de internet para que el programa se ejecute de manera óptima.

El proyecto se situará en el Instituto Tecnológico de Nuevo León en la sección de postgrado, en laboratorios y aulas que se asignan para desarrollo de los programas y el proyecto.

Beneficios esperados

Este proyecto tiene como fin el investigar la factibilidad que tiene el algoritmo prophet para la implementación dentro del área de producción para aplicarlo en el mantenimiento predictivo con respecto al tiempo, analizando variables que podrían deteriorar una maquina en una línea de producción lo cual puede ser aplicable dentro de muchas empresas y que podría dentro de estas generar un ahorro en gastos de reparación o de mantenimiento.

CAPITULO II. MARCO TEÓRICO

¿Qué es machine learning?

El machine learning es una rama de los algoritmos computacionales que están diseñados para emular la inteligencia humana mediante el aprendizaje del ambiente a su alrededor.

Según IBM Cloud Education (2020) “El machine learning es una rama de la inteligencia artificial (IA) y la ciencia de computación que se centra en el uso de datos y algoritmos para imitar la forma en que los seres humanos aprenden, con una mejora gradual de su precisión.”

Diversas técnicas que han sido basadas mediante machine learning han sido utilizadas en diversos sectores industriales, desde patrones de reconocimientos, ingeniería aeroespacial, finanzas, entretenimiento entre otras.

Big data en el contexto de ML.

En el artículo de Gandomi y Haider (2014) ,"Forrester define Big Data como las técnicas y tecnologías que hacen que sea económico hacer frente a los datos a una escala extrema. Big Data trata de tres cosas:

1) Las técnicas y la tecnología, lo que significa que la empresa tenga personal, el cual tenga gran representación y análisis de datos para tener un valor agregado con información que no ha sido manejada.

2) Escala extrema de datos que supera a la tecnología actual debido a su volumen, velocidad y variedad.

3) El valor económico, haciendo que las soluciones sean asequibles y ayuden a la inversión de los negocios".

La Big Data se ha presentado ante los algoritmos de machine learning como un reto debido a como su nombre lo menciona, se maneja una gran cantidad de datos para el aprendizaje y análisis de los algoritmos en los cuales se pueden generar valores predictivos y más acertados que con una cantidad no tan grande de datos reduciendo el nivel de error en estos, aunque varios algoritmos más simples se ven en la dificultad del análisis de tantos datos en donde la información de preprocesa , se aprende en base a ella y se evalúa, por lo que el machine learning se ve en la necesidad de seguir creciendo y evolucionando para el beneficio de todos en los sectores en los que se ponen en práctica.

Aprendizaje automático

El aprendizaje automático dentro del Machine Learning es el proceso mediante el cual se usan modelos matemáticos sobre datos para ayudar a un equipo a aprender sin instrucciones directas. Normalmente se le considera como una de las ramas de la inteligencia artificial. El aprendizaje automático usa algoritmos para identificar patrones dentro del conjunto de datos, y esos patrones luego se usan para crear un modelo de datos que puede hacer predicciones. Entre más datos, los resultados del aprendizaje automático son más precisos, ya que la predicción se fía de varios eventos, así como también la experiencia forma parte importante dentro del aprendizaje automático, de forma similar a cómo los humanos mejoran con la práctica.

En el aprendizaje automático se tienden a presentar dos fases, la primera es la selección, en esta el sistema elige los aspectos más relevantes dentro de un suceso, compara estos aspectos con otros que ya se conocen mediante un cotejamiento, en esta comparación verá las diferencias y si estas son muy significativas, procede a adaptar el modelo en base a estos aspectos del suceso.
Tal como los humanos, los sistemas también pueden aprender por diferentes sistemas. En el caso de los sistemas estos pueden llegar a emplear complicados procesos matemáticos, así como la recolección de la Big Data para la recolección de eventos relevantes.

Análisis predictivo

El análisis predictivo utiliza técnicas estadísticas de modelización, big data y machine learning para extraer datos históricos y realizar predicciones. En el mundo empresarial es una técnica muy cotizada por los beneficios que puede reportar a la hora de, por ejemplo, identificar riesgos y oportunidades (Brea Guzmán, 2022).

El análisis predictivo es una forma de análisis avanzado que examina datos o contenidos para responder a la pregunta: ¿qué es probable que ocurra en el futuro? Gracias al big data, los datos obtenidos a través de todos los sistemas conectados pueden interpretarse para obtener predicciones sobre cómo se va a comportar una persona o un grupo de población, algo también aplicable a negocios o procesos según Iberdrola, S.A. (2022)

Análisis descriptivo

En este tipo de análisis lo que se estudia es lo que ha pasado, y finalmente describirlo, las formas en las que este se puede describir varían entre el uso de estadísticas como el promedio, el uso de gráficos como los de barras o los de pastel, así como también el uso de tablas según Martínez (2020).

Técnicas del aprendizaje automático

- Aprendizaje supervisado: Abordar los conjuntos de datos con etiquetas o estructura sirve como un profesor y “entrena” al equipo, lo que aumenta su capacidad para realizar una predicción o tomar una decisión. En esta técnica se dan los ejemplos y se le especifica el concepto.
- Aprendizaje no supervisado: Aborda los conjuntos de datos sin etiquetas ni estructuras, buscar patrones y relaciones mediante la agrupación de datos. Esta técnica no está dirigida por metas sino al descubrimiento de datos, está diseñada para el descubrimiento de nuevos conocimientos.

- Refuerzo de aprendizaje: Es una combinación de las otras dos técnicas anteriores. En esta técnica se le proponen al sistema problemas que debe resolver. Un programa reemplaza al humano como ‘profesor’, ayuda a determinar el resultado en función señales en caso de que se realice correctamente el programa.

Paradigmas del aprendizaje automático

- Aprendizaje deductivo: Se realiza mediante inferencias ante unos hechos o fenómenos previamente conocidos de los cuales se provienen nuevos hechos.

- Aprendizaje analítico: El aprendizaje analítico requiere que se proporcione al sistema un amplio conocimiento del dominio. Este conocimiento es usado para guiar las cadenas deductivas que se utilizan para resolver nuevos problemas.

- Aprendizaje analógico: Este método intenta emular las capacidades humanas como el entendimiento de situaciones conocidas, la adecuación de temas resueltos en el pasado con nuevos temas parecidos por lo que este tipo de aprendizaje requiere de una gran cantidad de conocimiento.

- Aprendizaje inductivo: En estos sistemas se desconocen los eventos que se tratan desde antes de ser tratados, así como de la cantidad de eventos, averigua la descripción de un evento haciendo uso de algunos ejemplos, así como de contraejemplos de este evento.

- Aprendizaje mediante el descubrimiento: Se adquiere el conocimiento sin necesidad de ayuda, normalmente se usa en los casos en los que no se encuentran fuentes de conocimiento que puedan ayudar al aprendizaje del sujeto.

Modelo de datos para el machine learning

El modelado de datos según el artículo de Shin (2021) se menciona la categorización de los modelos que se dividen en supervisado y no supervisado, el cual dentro de la primera categoría se subcategoriza en modelos de clasificación y de regresión.

Dentro de la categoría de modelos de regresión se encuentran algunos como la regresión linear el cual su propósito es el de encontrar una línea que pase entre los datos que más se acerque a estos, también pueden hacerse soluciones más complejas donde se pueden hacer múltiples regresiones lineales o polinomiales como curvas. También se encuentra el árbol de decisiones donde se hace uso de nodos los cuales serán decisiones que se tomarán dependiendo de lo que se haga. Luego se encuentran los random forest el cual crea varios árboles de decisión en base a los datos de un inicio y selecciona un conjunto de variables en cada paso de los árboles de decisión y se confía en los resultados de la mayoría de los árboles de decisión. El último ejemplo de la subclasificación de regresión es el de redes neuronales el cual se compone de una red de ecuaciones matemáticas, el cual puede resultar en uno o más resultados.

Después se encuentra la subcategoría de los modelos de clasificación donde se pueden mencionar los modelos de regresión logística el cual es parecido a la antes mencionada regresión linear solo que esta esta normalmente utilizado para observar probabilidades ente una cantidad de números determinada, después está la máquina de modelos de soporte el cual en términos sencillos encuentra un margen o frontera entre dos tipos distintos de datos, y finalmente se encuentra el modelo Naïve Bayes el cual está basado en el Teorema de Bayes, el cual el propósito en términos generales es el de saber la probabilidad de ‘y’ dada ‘x’. A su vez , los árboles de decisión, random forest y redes neurales pueden entrar dentro de esta subclasificación con un ajuste dando resultados discretos.

Serie de tiempo

Una serie de tiempo es una secuencia de datos u observaciones, medidos en determinados momentos y ordenados cronológicamente. Visualmente, es una curva que evoluciona en el tiempo.

En el caso de las industrias pueden observar la actividad del negocio dentro del periodo de tiempo, actividades como el crecimiento de clientes dentro de un lapso de un año, así como también de la predicción del comportamiento en fechas futuras de estos mismos datos como el visualizar el crecimiento de los clientes dentro del siguiente año.

Algoritmo prophet

El un algoritmo desarrollado por Facebook el cual se encarga de pronosticar series de tiempo, en este se basa en recolectar datos variados dentro de un periodo de tiempo y predecir datos futuros, pudiendo soportar datos atípicos, así como amplios márgenes de variables y datos faltantes.

El procedimiento es un modelo de regresión aditivo de tres funciones de tiempo (crecimiento, estacionalidad y eventos) más un término de error (Taylor et al., 2017).

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t \tag{1}$$

Donde g(t) es el crecimiento, s(t) es la estacionalidad, h(t) son los holidays y ε_t es el término de error. Prophet permite al analista proporcionar una lista de eventos pasados y futuros identificados en un único nombre: holidays (del Rosso, 2021; Galmés Mifsud, 2019).

Actualmente Prophet solo se encuentra disponible en Python y R y una de sus ventajas es que está diseñado para ser fácil de usar sin un conocimiento experto en la predicción de series temporales o estadística.

Según un estudio que comparó varios logaritmos en series de tiempo en parámetros diarios, mensuales y cinco-minutales de los cuales se sacó la conclusión de que el algoritmo Prophet es de buena confiabilidad debido a que su margen de error es el menor entre los comparados, sin embargo no es el mismo caso con las series mensuales y cinco-minutales ya que el nivel de error es demasiado alto en ambos casos además de que comparado con los otros algoritmos estaba con un nivel de confiabilidad muy bajo y se tuvo que realizar un ajuste en las variables, desmintiendo en gran medida que el algoritmo prophet es un forecasting automático en estos dos casos por lo que tienes que tener amplios conocimientos en este algoritmo y en temas como matemáticas para poder implementarlos en series mensuales y cinco-minutales teniendo en cuenta que aun así el nivel de error es algo elevado por lo que no es recomendable aun así en estos dos casos ya que prophet está diseñado más para datos diarios.

Sin embargo, según el estudio de Romero Gelvez y Rincón Quintero (2020) demostró que si se puede realizar un análisis de serie de tiempo en base a minutos con el algoritmo de Prophet ya que ellos analizaron el mantenimiento predictivo de herramientas para la manufacturación basado en datos del repositorio de la NASA “Conjunto de Datos de Rodamiento”

Herramientas para el proceso de resultados

Para hacer las actividades para el análisis de los datos y el proceso de machine learning se hace uso de diversas herramientas recolectadas de distintas librerías como lo son las siguientes:

Numpy

Numpy es usado para el trabajo con arreglos en datos, numpy también sirve para otras funciones matemáticas como lo es la algebra lineal, las transformaciones de Fourier y matrices. El significado de NumPy es Numerical Python, o Python numérico en español.

Una de las ventajas de usar numpy es que este es más rápido para hacer las tareas que sin una librería a la hora de compilar.

Además, los arreglos son muy usados dentro de las áreas de la ciencia de datos y el machine learning. En actividades como guardar, usar y analizar datos con los cuales se realiza un análisis de datos de una forma más matemática donde los recursos que se tienen en la computadora son importantes, además de que, dentro del área de la programación, uno de los múltiples objetivos es el buscar el mejor rendimiento del hardware y software haciendo un código óptimo, ósea, más rápido y menos líneas de código.

Pandas

Panda es otra librería de Python que sirve para trabajar para datos tabulados llamados dataframe, lo que va a ayudar a que los datos sean más accesibles en los procesos de limpieza eliminando los datos que se identificaron como impuros o que pueden llegar a afectar los resultados de manera negativa, la exploración la cual nos puede permitir observar información bajo ciertos criterios determinados como en ciertas fechas, así como y el proceso de los datos aplicándolo en algoritmos de machine learning, además de que en caso del algoritmo Prophet, se requiere de un formato especial con nombres de columnas específicos para que el proceso.

Otra función importante que tiene la librería es la lectura de los datos en distintos formatos, algo muy importante en sectores como el análisis de datos, ya que permite una flexibilidad a la hora de recibir datos de distintos formatos, con ayuda de pandas, uno es capaz de hacer lectura de data sets de tipo CSV, XLS, JSON y SQL por mencionar los datos más conocidos, así como también la capacidad de escribir nuevos datos en dichos formatos.

Matplotlib

Esta librería tiene como propósito visualizaciones de tipo estática, animadas o interactivas de los datos que recuperemos y transformemos con las librerías de panda, numpy y el algoritmo.

En este caso, Matplotlib nos permite poder ver mediante gráficas distintas visualizaciones de los datos, donde podemos seleccionar las columnas que se desplegaran en la gráfica, la posición ‘x’ o ‘y’ en donde estas se encuentren.

Esto servirá para el análisis gráfico de los datos, una visualización más dinámica para nuestro proceso de comprensión de los datos nos puede hacer observar anomalías y datos que deben de ser limpiados.

Seaborn

Seaborn es una librería basada en la librería Matplotlib, que nos permite observar los datos en base a su distribución mediante elementos básicos, esta es una tarea importante en el análisis de datos al igual que la librería antes mencionada, sin embargo, las gráficas en comparación a las generadas con matplotlib, estas pueden generarse diversos tipos de gráficos como los siguientes:

- Dispersión
- Densidad
- Histograma
- Boxplot
- Violín

SciKitLearn

Librería de machine learning la cual contienen otras librerías como las mencionadas Pandas, numpy y matplotlib.

Otra característica que posee esta librería es que contiene diversos algoritmos de aprendizaje como los supervisados como el de máquinas de vectores de soportes, árboles de decisiones y métodos bayesianos, y los no supervisados como el análisis factorial, y redes neuronales no supervisadas, con los que puedes aplicar el aprendizaje de máquina dentro de un programa.

A su vez, podemos hacer uso de esta librería para aplicar métodos de precisión para los modelos supervisados como lo es la validación cruzada, así como para separar datos de entrenamiento y de prueba para aplicarlos a nuestro algoritmo.

Prophet

Prophet es la librería de Python y R requerida para poder aplicar el modelo Prophet, el cual no se encuentra dentro de la librería SciKitLearn.

El uso de este algoritmo es para el pronóstico de datos en base a una serie de tiempo.

CAPITULO III. METODOLOGÍA

Metodología de experimento

Para la implementación de un modelo de machine learning se siguió el proceso como se menciona en la ilustración 2 basado en una sistematización y un color basado en el esfuerzo que requerirán las acciones. Donde siempre se debe:

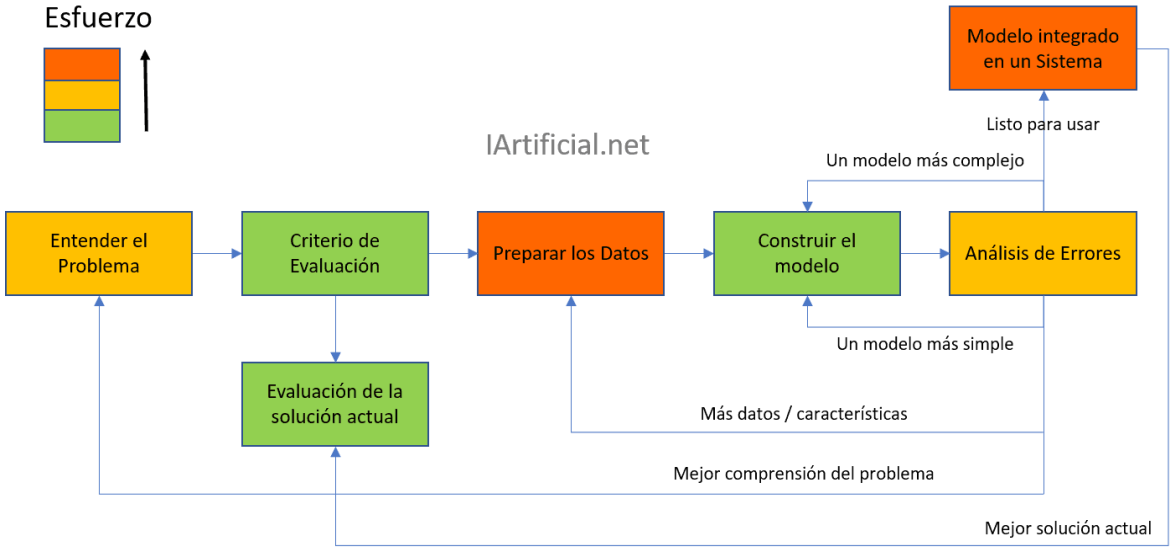


Ilustración 2. Fases del proceso de machine learning. Recuperado de Heras (2020)

empezar por entender el problema, saber lo que se hará y analizar los objetivos previamente para la planificación así como del criterio de evaluación, saber que se va a evaluar para preparar los datos, esto mediante una limpieza de estos, quitando anomalías o valores impuros, aplicar normalizaciones en caso de ser necesarias y/ a su vez separar los datos de prueba con los de entrenamiento, una vez preparados los datos, se construirá el modelo aplicarle los parámetros adecuados que se planificaron en los dos primeros pasos para el análisis de los resultados y seguir con el análisis de errores, en este paso se usaron librerías para medir las métricas con ayuda de la validación cruzada, donde primordialmente en el modelo a manipular que será el prophet nos interesará en especial el error absoluto medio (mae) y error absoluto porcentual medio (mape) para saber en dado caso de tener métricas aceptadas, si el algoritmo es apto para su uso en un sistema.

Para el uso de algoritmos, se hizo en base sobre la plataforma Google Colab para una mayor flexibilidad a la hora de compartir los resultados en otros dispositivos de escritorio a otros usuarios, la facilidad de ejecutar los códigos sin necesidad de descargar algún elemento en las computadoras.

Como primer paso importamos las librerías siguientes:

- Prophet desde prophet, que será la librería con la que efectuaremos nuestra predicción de datos.
- Numpy, para la manipulación de datos numéricos.
- Pandas, para la manipulación de los data frames.
- Matplotlib, para graficar los data frames.
- Seaborn, para los mapas de calor.
- Train_test_split desde sklearn.model_selection, para el train y el test
- Cross_validation y performance_metrics desde prophet.diagnostics para el análisis de confiabilidad de los datos.
- Plot_cross_validation_metric desde prophet.plot para graficar las métricas.

Se hizo uso de Google Drive para la facilitación de la extracción de datos en formato CSV. Para el análisis de los datos hicimos uso de métodos de pandas para leer un CSV, para ver información, una cantidad delimitada de datos, así como valores como medias, mínimos y máximos de los datos. Y hacer nuestra primera gráfica para visualizar los datos y comenzar con la limpieza.

Para el limpiado de los datos se verá por medio del análisis de datos que hicimos, donde principalmente se eliminarán datos que se encuentren nulos debido a la desconexión del sensor y se hará una segunda gráfica para volver a visualizar los datos. Después, se hará una separación de datos haciendo uso de la técnica de test-train, en base a la data frame, esto con ayuda de la biblioteca sklearn.model_selection , con un tamaño de test del 0.1, se tuvo en consideración este valor debido a que la cantidad de datos no es tan grande y no afectar a los datos ubicados en el train que serán los que aplicaremos en el algoritmo y un random state de 10.

Después, de hizo uso del seaborn para ver una gráfica de datos para analizar los datos de otra manera distinta, viendo la cantidad de datos de temperatura que se repiten en un mismo elemento.

Para preparar el dataframe para el uso con el algoritmo prophet, primero se va a seleccionar dos columnas x y donde la primera columna debe de ser el tiempo y la segunda columna debe de ser el tipo de valor a predecir que en este caso será algún tipo de valor capturado por un sensor y cambiar los nombres de las columnas por ‘ds’ y ‘y’ ya que esto es necesario para que el algoritmo pueda manipular estos datos.

Finalmente, ya con el dataframe preparado para los datos, se aplicarán los métodos específicos para que el algoritmo haga sus predicciones.

Una vez realizadas las predicciones, hicimos el mismo análisis de datos que se realizó anteriormente con el primer dataframe y se procedió con el proceso de graficar para observar las predicciones y una visualización de las predicciones, apoyándonos del método tail( ) para observar los últimos datos que serán estos los predichos.

Una vez hecho el análisis de los datos y la predicción de datos, se continuó con la evaluación de los experimentos mediante la implementación de la validación cruzada haciendo uso de la librería prophet.diagnostics importando cross_validation para el empleo de métricas con ayuda de la validación cruzada, se le asignó en el parámetro un horizonte de 30 minutos ya que es la cantidad máxima de datos que le puedo añadir a la validación cruzada y una cantidad mínima podría comprometer las métricas.

Después bajo esa misma librería se importó perfomance_metrics para observar métricas donde se les hizo énfasis a las métricas mae (mean absolute error – error absoluto medio) y mape (mean absolute porcentual error – error absoluto medio porcentual) para el análisis de confiabilidad.

Finalmente, se importó plot_cross_validation_metric desde prophet plot para poder analizar de manera gráfica las métricas.

Metodología de evaluación

Para el proceso de la evaluación del algoritmo, se tomó como referencia los resultados obtenidos en la evaluación en cada uno de los 3 experimentos en el apartado de la evaluación métrica, tomando como referencia las métricas de error medio absoluto, el error medio absoluto porcentual, el error cuadrático medio, y las gráficas que se recuperaron de dichas métricas.

Se tuvo en cuenta las gráficas obtenidas sobre las métricas para el posterior análisis de cada uno de los experimentos y llegar a un análisis en conjunto del algoritmo en base a los 3 experimentos previos llegando a una conclusión de los puntos que se encontraron.

Posterior a eso, se realizó una evaluación comparativa con otro algoritmo de machine learning llamado random forest recuperando los datos de experimentos realizados por el compañero Adrián Fernando Agundis Martínez, algoritmo el cual se encargó principalmente de la clasificación de varios atributos.

Dicha comparación se realizó con el apoyo de cada una de las gráficas recabadas de ambos algoritmos así como también las tablas de métricas de cada experimento, en dicha comparación solo fueron expuestos los primeros dos experimentos ya que están enfocados en los mismos puntos no así el tercero que fue un experimentos que cada uno en sus investigaciones tuvo giros distintos aunque el propósito de experimentar con el algoritmo haya sido el mismo, compararlos no habría sido posible.

Se analizó y discutió acerca de ambos algoritmos situaciones como las métricas de errores, los casos en los que se estuvieron usando los algoritmos, los datos de entrada y salida dentro de cada algoritmo en que caso un algoritmo conviene más que el otro, así como comparar la confiabilidad de ambos en los experimentos de ambos llegando a una conclusión de esta evaluación.

CAPITULO IV. RESULTADOS

Resultados de experimento

La cantidad de experimentos consistió en un total de 3, con los cuales en todas se siguió la metodología antes mencionada de las fases de machine learning para unificar y sistematizar el procedimiento, el primer experimento consistió en 4 sensores, que midieron valores dentro de una cantidad determinada de tiempo, en donde se ilustró únicamente los valores de un sensor que en este caso fue el de la temperatura, el segundo experimento fue la medición de valores de temperatura únicamente de un horno en otra cantidad de tiempo definido y por último fue un tercer experimento recuperado de la página Kaggle que consistió en la medición de temperatura promedio de dos motores dentro de un lapso de tiempo.

Experimento 1

El Data set consistió en datos recuperados por segundo de unos sensores en un lapso de 2 horas y media aproximadamente entre 12:57:09 horas hasta las 15:13:12 horas.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4378 entries, 0 to 4377
Data columns (total 5 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Fecha/Hora   4378 non-null   object
 1   Temperatura  4378 non-null   float64
 2   CO2          4378 non-null   int64
 3   Presion      4378 non-null   int64
 4   Aire         4378 non-null   int64
dtypes: float64(1), int64(3), object(1)
memory usage: 171.1+ KB
```

(a)

```
[ ] df.describe()
```

	Temperatura	CO2	Presion	Aire
count	4378.000000	4378.000000	4378.000000	4378.000000
mean	42.066743	2687.652810	1862.664002	7296.235724
std	3.012033	204.134289	113.466682	427.943217
min	0.000000	0.000000	0.000000	0.000000
25%	41.200000	2697.000000	1851.000000	7318.000000
50%	42.300000	2706.000000	1877.000000	7321.000000
75%	43.500000	2710.000000	1887.000000	7326.000000
max	51.400000	2737.000000	2021.000000	7684.000000

(b)

Ilustraciones 3. Análisis del dataframe. (a) Información del dataframe del experimento 1 y (b) descripciones matemáticas

Como podemos ver, al momento de hacer el primer análisis del dataframe, podemos ver las columnas, las cantidades de datos no nulos y el tipo de datos donde nos interesó en especial para la limpieza de los datos fue el de la cantidad mínima ya que, como tal, el primer análisis no nos detectó valores nulos, pero si existen valores ceros, lo cual para analizar si estos valores fueron ‘fieles’ se hizo una gráfica en base a la columna ‘Fecha/Hora’ y ‘Temperatura’, y nos arrojó lo siguiente:

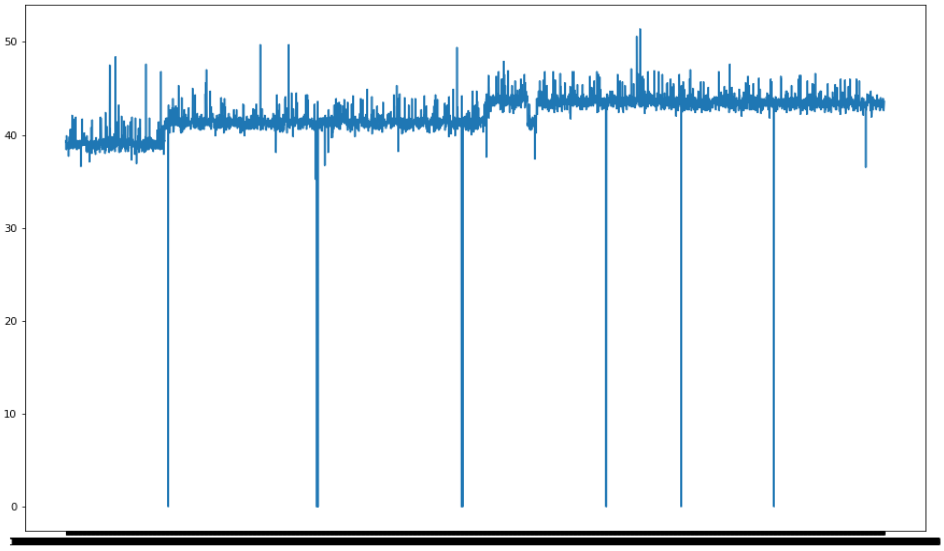


Ilustración 4. Primer análisis gráfico de dataframe experimento 1

Por este análisis podemos ver que hubo una anomalía en el sensor que detecto, en este caso la temperatura, por lo cual se omitieron estos datos para proceder a el uso de la herramienta de la separación de datos de entrenamiento y testeo.

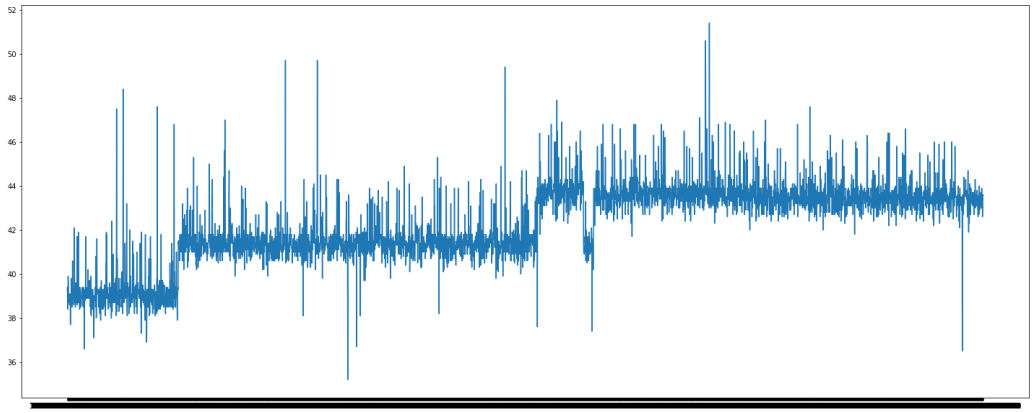


Ilustración 5. Análisis gráfico de dataframe experimento 1 una vez omitidos los valores anomalía

Una vez omitidos los datos anómalos, se usó la librería de sklearn para separar datos de entrenamiento y prueba. Ver Anexo 1.

Una vez separados los datos, se realizó una tercera gráfica arrojando el resultado siguiente:

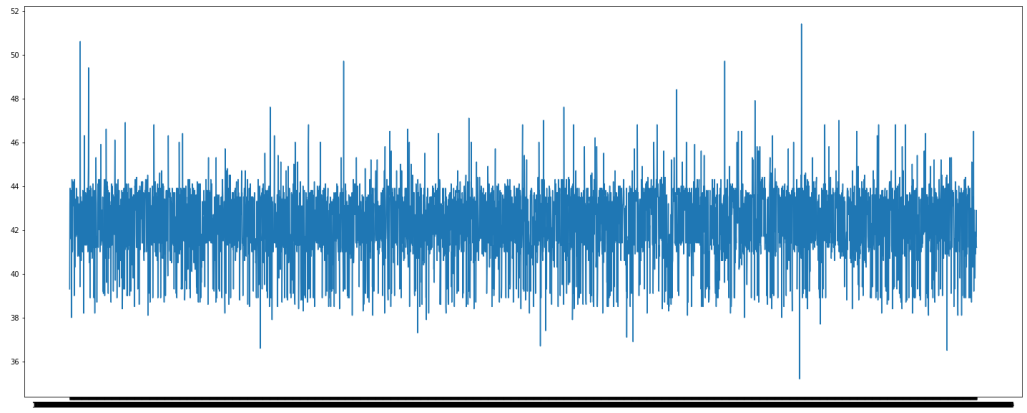


Ilustración 6. Gráfica de los datos train del dataframe del experimento 1

Otro análisis en base a los datos de entrenamiento que serán los que usaremos para la predicción fue el usado con la librería Seaborn para analizar la distribución de los datos:

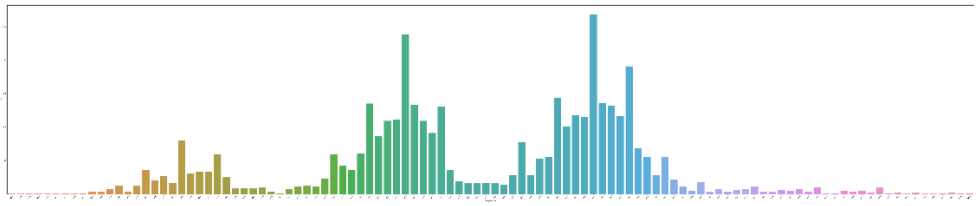


Ilustración 7. Análisis de la distribución de los datos

Una vez preparados los datos para el modelo se seleccionaron las columnas de ‘Fecha/Hora’ y ‘Temperatura’ para guardarles en otra variable para su uso y se les cambio el nombre de las columnas por ‘ds’ y ‘y’ respectivamente ya que es necesario para la función del algoritmo con el siguiente fragmento de código que se mostrará a continuación en donde se está implementando el dataframe anteriormente limpiado y procesado para el uso de este con el algoritmo Prophet.

Se agregaron parámetros dentro de make_future_dataframe como periodo el valor de 3600 que fueron la cantidad de datos a predecir, despues el de freq con valor ‘S’ lo cual nos indica que fueron de tipo segundo (osease 3600 segundos a predecir) y se le incluyó historia para poder observar datos previos que fueron los del dataframe. Ver Anexo 2

Despues de eso se procedió a analizar la gráfica del pronóstico.

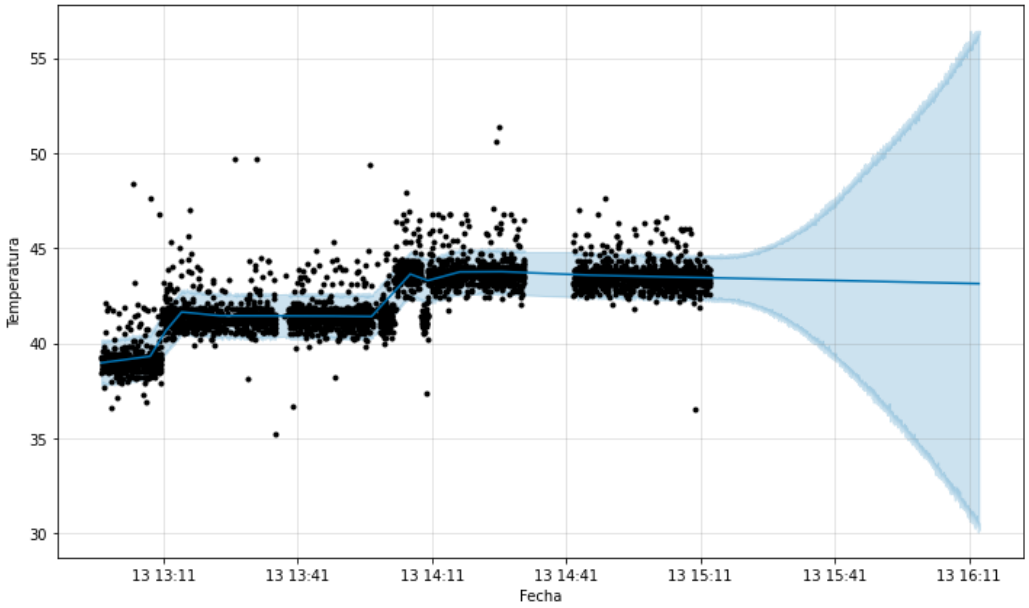


Ilustración 8. Predicción de datos usando el algoritmo Prophet

Se pudo observar que la tendencia de los datos continua de manera recta y que no se aleja mucho de lo observado a partir de las 15:11.

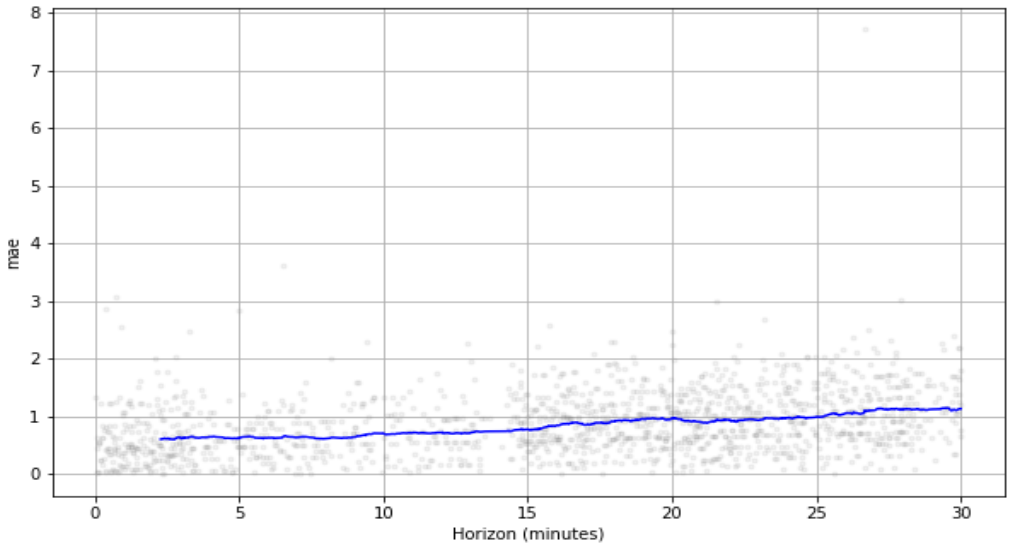
Se procedió finalmente al análisis de errores, para esto se hizo apoyo de la validación cruzada y las métricas de rendimiento importadas de la librería prophet.diagnostics. Ver Anexo 3

horizon	mse	rmse	mae	mape	mdape	smape	coverage
0 days 00:02:17	0.627146	0.791925	0.599317	0.013678	0.010836	0.013654	0.887417
0 days 00:02:18	0.626335	0.791413	0.598495	0.013659	0.010836	0.013635	0.887417
0 days 00:02:19	0.630937	0.794315	0.602559	0.013754	0.010836	0.013729	0.887417
0 days 00:02:20	0.637261	0.798286	0.607162	0.013862	0.010836	0.013835	0.887417
0 days 00:02:21	0.635415	0.797129	0.603730	0.013783	0.010812	0.013757	0.887417

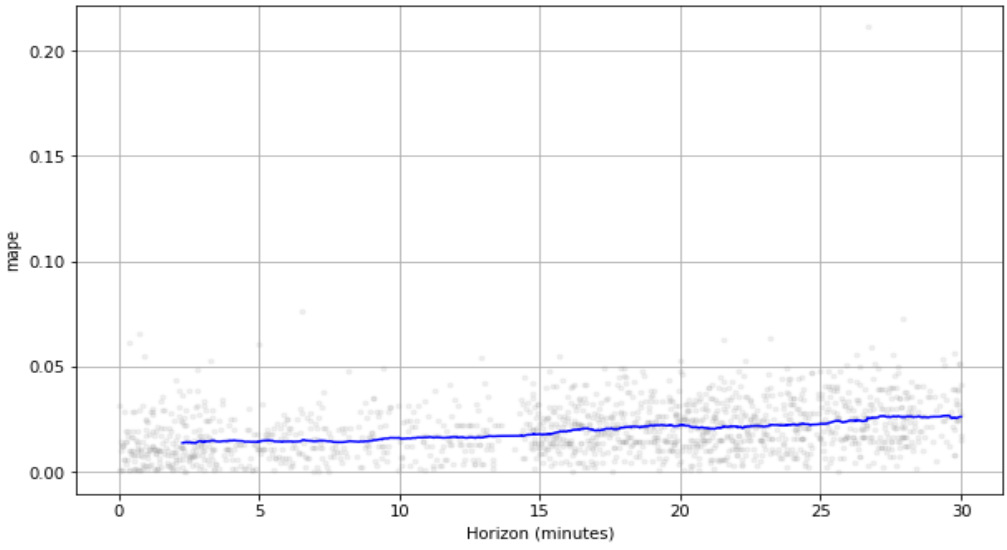
Tabla 1. Métrica de errores del algoritmo del experimento 1

Podemos observar que en el mae nos arrojan datos que inician desde el 0.59 hasta el 0.60 en los primeros 5 segundos del experimento, lo cual para las métricas se considera que son resultados buenos. Asi como también en el mape se obtuvo un rango entre 0.0136 y 0.137 que vendría siendo porcentajes de 1.36% y 1.37% los

cuales según la documentación de Prophet, nos menciona que los porcentajes de las métricas tanto de mae como de mape deben de acercarse al 0 lo cual nos hace concluir que los resultados de las métricas obtenidas de este experimento 1 demuestran que el algoritmo es de confianza en este caso.



a)



b)

Ilustración 9. Métricas gráficas (Exp 1). a) Error medio absoluto a lo largo del tiempo b) Error medio absoluto porcentual a lo largo del tiempo

Experimento 2

El segundo data set fue un compendio de temperaturas capturadas por segundo de un horno esta serie de tiempo tuvo como rango desde las 19:20:32 horas hasta las 20:07:19 horas, lo que sería poco menos de una hora.

Como análisis podemos recalcar que fue similar que el ejemplo anterior, debido a que no presentaba datos nulos, pero en este caso solo media lo que era la temperatura. Sin embargo, a la hora de mostrar la primera grafica se observaron de manera más claras las anomalías.

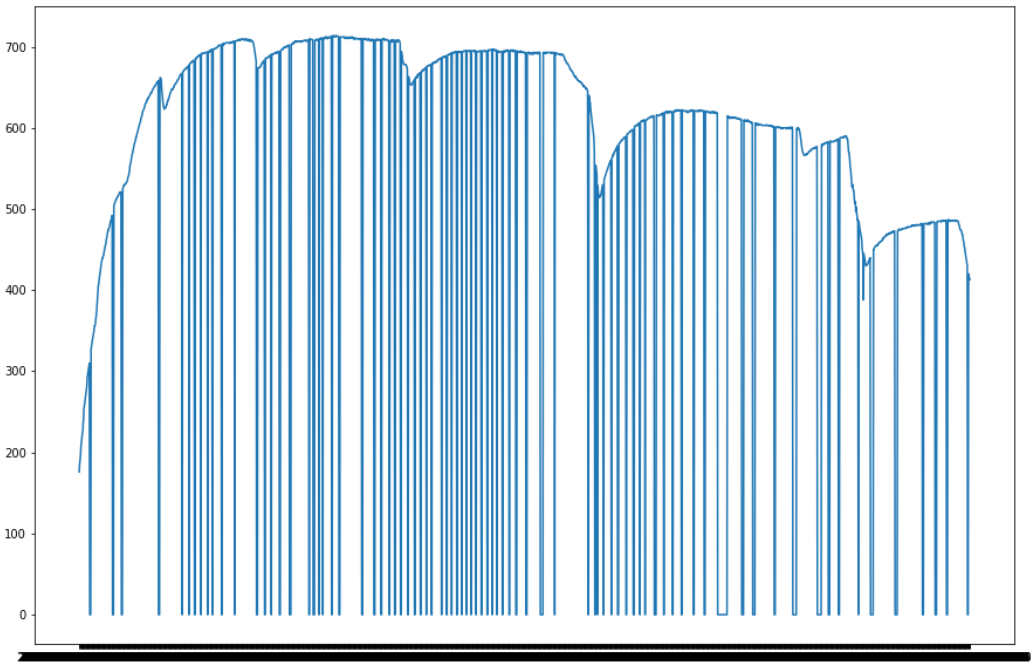


Ilustración 10. Primer análisis de la data set experimento 2

En este primer análisis podemos observar que en los primeros segundos hay un momento donde la temperatura comienza a subir, esos datos podrían considerarse como realistas, sin embargo después empiezan a haber picos muy bajos y repentinos donde seguramente sucedió alguna anomalía en los datos, ya sea que no capturaron los datos de temperatura o que simplemente hubo alguna especia de corto circuito, así que se planteó la posibilidad de eliminar estos datos de la estadística arrojándonos la siguiente tabla:

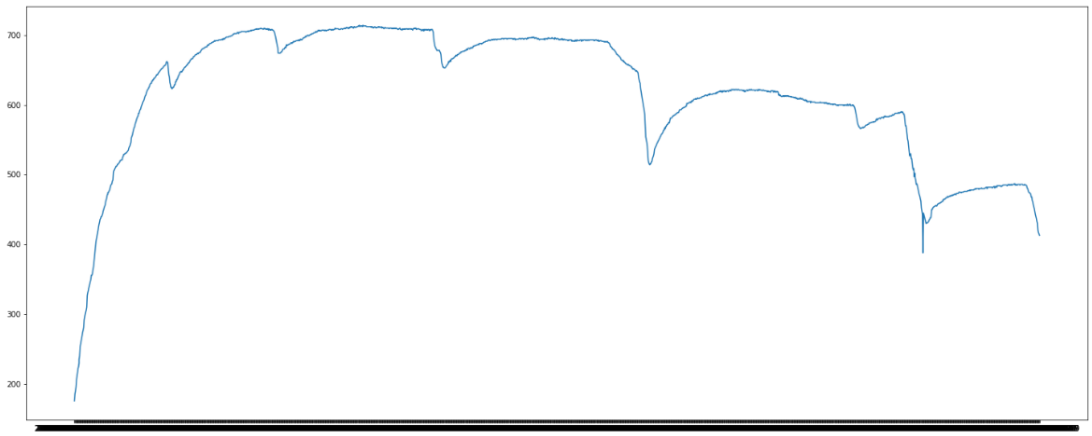


Ilustración 11. Dataframe sin anomalías.

Después se hizo la separación de datos entrenamiento-prueba sin embargo en esta ocasión el tamaño de prueba lo reduje a 0.05 debido a la poca cantidad de datos que se tiene para un algoritmo que requiere de big data, sumado a la limpieza previa de una considerable cantidad de datos.

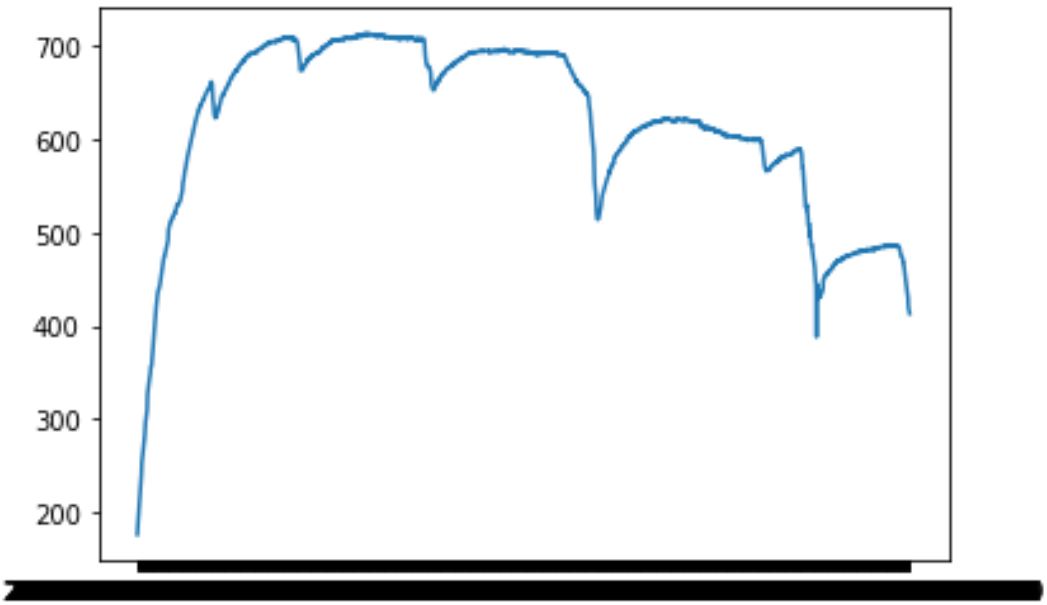


Ilustración 12. Dataframe de entrenamiento

Después se hizo el análisis con la librería seaborn donde podemos observar que gran cantidad de datos se ubican en las altas temperaturas.

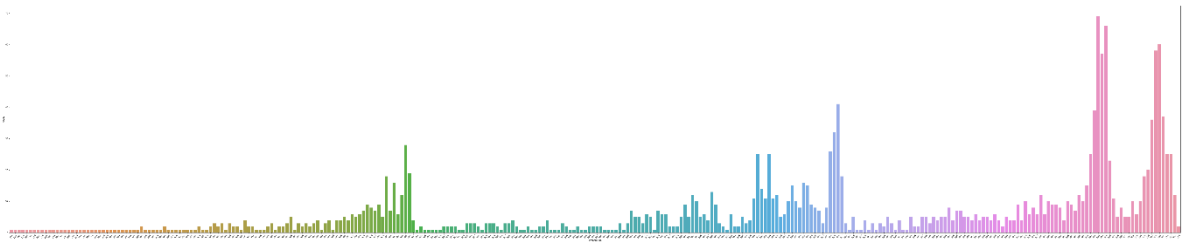


Ilustración 13. Análisis de la distribución de los datos

Después de la preparación de los datos se ejecutó dentro del modelo arrojándonos la siguiente tabla:

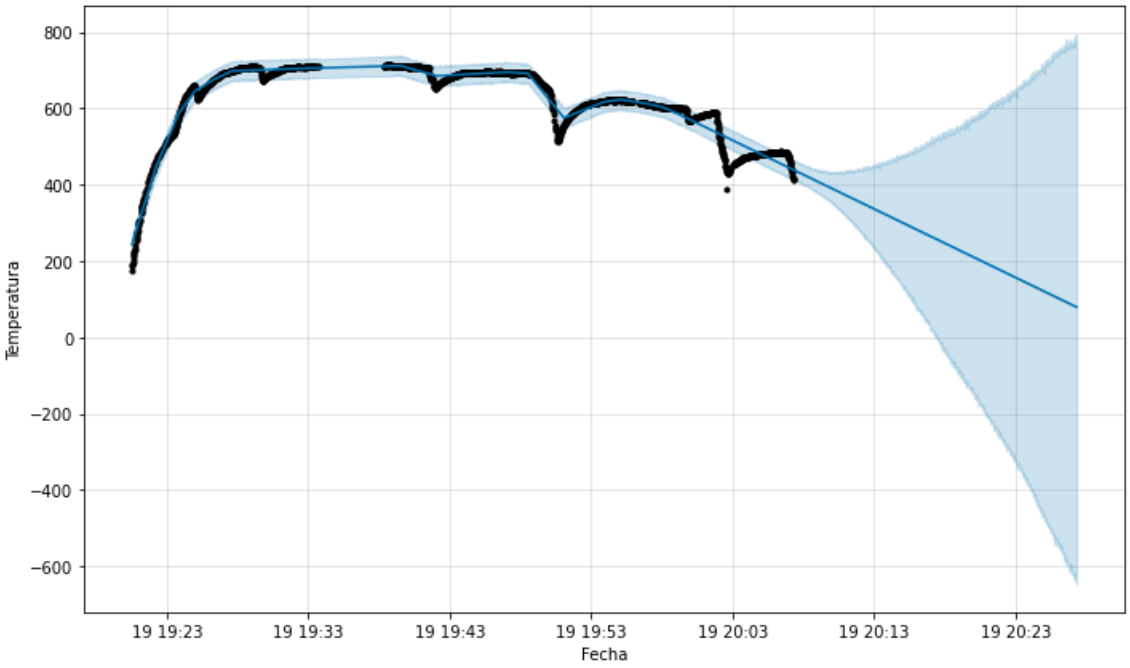


Ilustración 14. Pronóstico de temperatura con uso del algoritmo prophet

Sin duda claramente esa bajada de temperatura que pronostica el algoritmo se debe a la caída en picada que el data set de los hornos se pudo manifestar en que el horno ya se estaba apagando, el ejemplo no es alguno que podría aplicarse normalmente ya que son acciones estacionarias no relacionadas con algún mantenimiento predictivo, pero de igual forma nos puede servir como documentación ante futuros casos

Finalmente, se hizo la prueba de errores por métrica y validación cruzada.

	horizon	mse	rmse	mae	mape	mdape	smape
0	0 days 00:01:03	2884.115516	53.703962	50.557834	0.083055	0.067482	0.085329
1	0 days 00:01:04	2957.220486	54.380332	51.198147	0.084090	0.067729	0.086434
2	0 days 00:01:05	3026.787089	55.016244	51.772403	0.085011	0.069008	0.087464
3	0 days 00:01:06	3098.114518	55.660709	52.377663	0.085988	0.069008	0.088512
4	0 days 00:01:07	3175.569533	56.352192	53.042116	0.087059	0.069750	0.089717

Tabla 2. Métrica de errores del algoritmo del experimento 2

Podemos observar un buen porcentaje en el mape con un 8% aproximadamente en los primeros 5 segundos del pronóstico sin embargo el error es desalentador en el mae con un 53.70 lo cual se puede llegar a concluir con esto que el experimento no es confiable, esto pude concluir que fue causa a factores en el data set que no ayudan a que el algoritmo se use de manera correcta, los factores son los siguientes:

1. Poca cantidad de datos: 1 hora de datos para pronosticar es muy poca si queremos dar resultados acertados, al menos se debe tener 3-5 veces la cantidad de datos para sacar 1 vez de pronósticos. Ejemplo un data set de 3-5 años para sacar 1 año de pronóstico.
2. Fallas en el sensor: las fallas en el sensor como observamos en la gráfica 2.1 afectaron directamente a la hora que querer limpiar datos puestos que si no las quitábamos iban a sesgar las predicciones y el quitarlos nos provocó la perdida de información.
3. El enfoque de los datos: este podría ser considerado la razón principal del error, un horno que se prendió y se apagó en un periodo corto de tiempo no puede ser usado para predecir algo como su temperatura, puesto que no se nos indica nada más que eso

Experimento 3

Para el siguiente documento se extrajo y modifico un data set recuperado de la página de Kaggle la cual consta de la temperatura de motores en un lapso de 23 días, esta se consideró como una buena cantidad de datos para una predicción de corto tiempo.

Primero se hizo el análisis de los datos para averiguar el tipo de datos así como los cálculos matemáticos de las columnas, viendo datos mínimos, máximos, media, estándar donde nos dimos cuenta de que no hay ningún dato cero ni nulo, lo cual es un buen paso e indica que no hay anomalías, sin embargo esto se terminó de corroborar con la gráfica de los datos, sin embargo en este caso la primera gráfica salió con las fechas en aleatorio por lo que se aplicó el siguiente método para acomodar los valores por la fecha en el dataframe. Ver anexo 4.

Se tomó en cuenta la columna ‘tiempo’ y la columna ‘avgTemp’ para la predicción en este experimento.

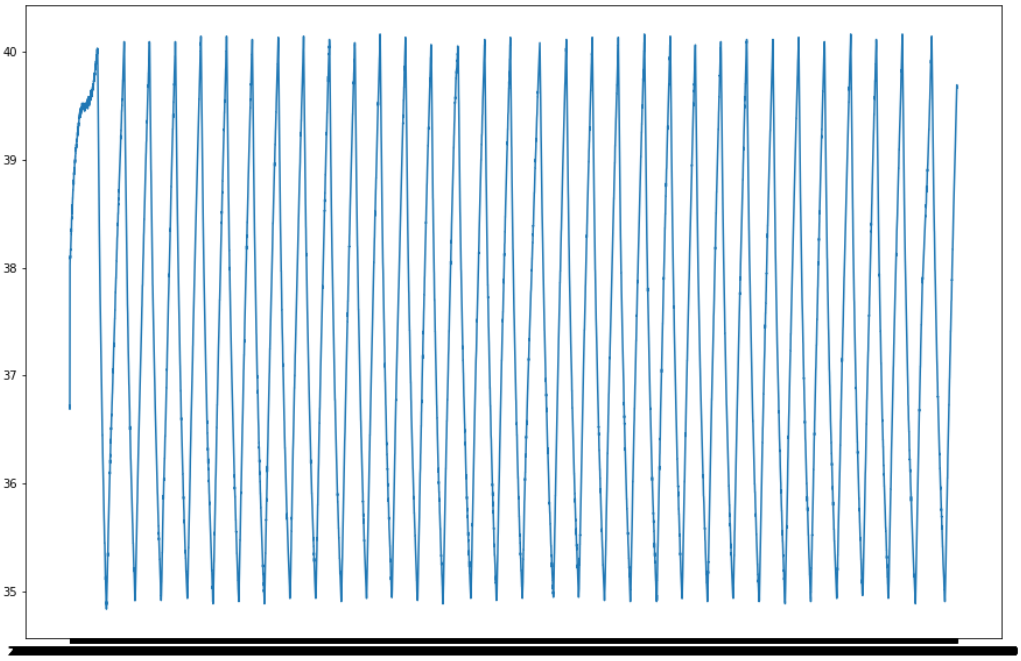


Ilustración 15. Primer análisis de la dataframe del experimento 3

Se pudo observar que los datos oscilaban entre 40 a 35 grados durante la mayoría del tiempo.

Después de procedió a realizar la separación de datos de prueba y entrenamiento, en este caso por ser más datos que los pasados experimentos el tamaño de la prueba cambió a 0.15 que equivale al 15% de los datos, la gráfica fue la siguiente:

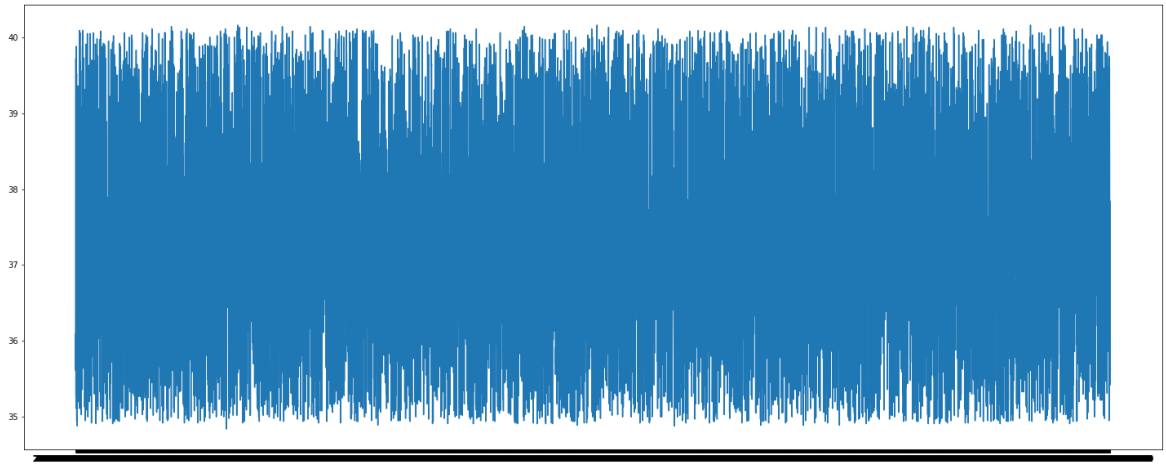


Ilustración 16. Dataframe de entrenamiento

Después se realizó el análisis de la distribución de los datos con ayuda de la librería seaborn.

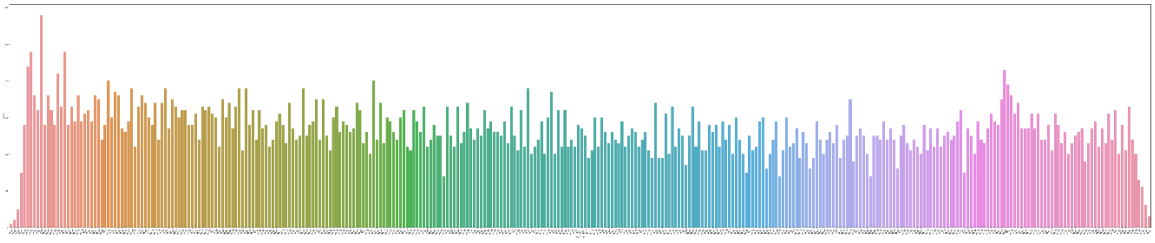


Ilustración 17. Gráfica de distribución de los datos

Se preparó el dataframe con las dos columnas para uso del modelo con los nombres ‘ds’ y ‘y’ para el ‘tiempo’ y ‘avgTemp’ respectivamente.

Después se procedió a realizar la predicción mediante prophet:

En este caso se pronosticó de 14,400 minutos a lo que equivale a 10 días, se tenía previsto hacer uso de su equivalencia en segundos, sin embargo, la RAM de sistema que proporciona Google Colab no soporta tales cantidades de datos. Ver anexo 5.

Una vez realizado el pronóstico se hizo la gráfica en base a la variable `forecast:`

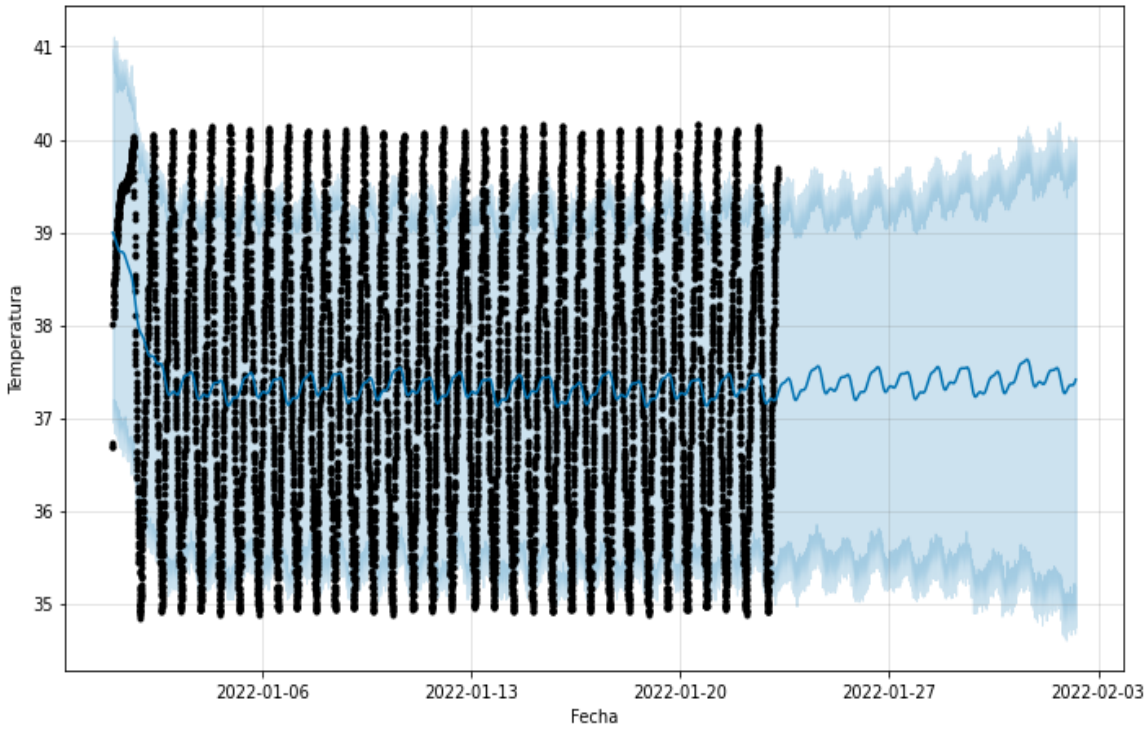


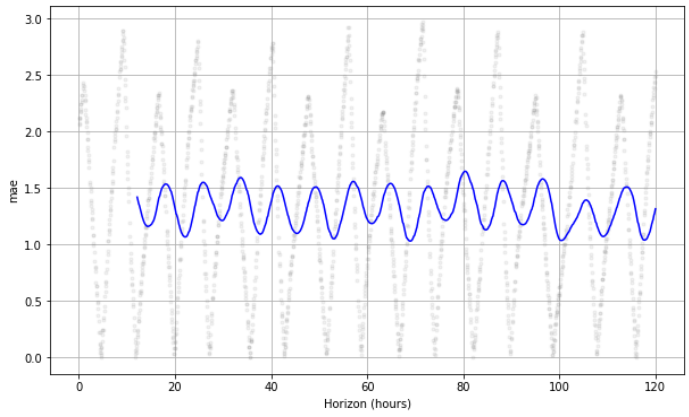
Ilustración 18. Pronostico del experimento 3 haciendo uso del algoritmo Prophet

Finalmente se realizaron las pruebas de errores haciendo uso de la validación cruzada.

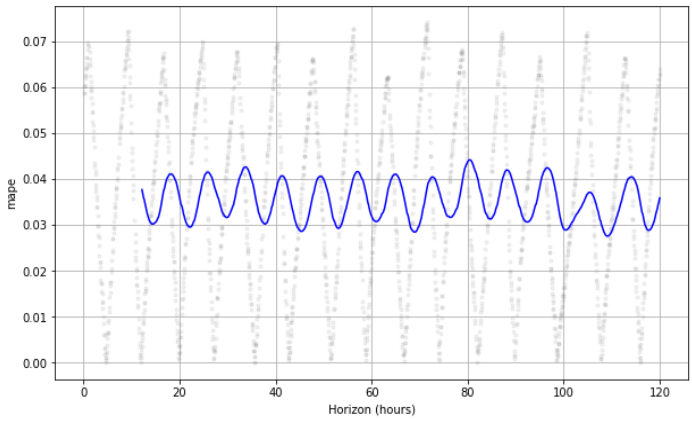
	horizon	mse	rmse	mae	mape	mdape	smape	coverage
0	0 days 12:09:00	2.739162	1.655041	1.419295	0.037626	0.038059	0.037809	0.633663
1	0 days 12:12:00	2.718274	1.648719	1.410095	0.037364	0.037957	0.037555	0.638614
2	0 days 12:15:00	2.697355	1.642363	1.401013	0.037104	0.037785	0.037303	0.643564
3	0 days 12:18:00	2.675532	1.635705	1.391752	0.036839	0.037578	0.037047	0.648515
4	0 days 12:21:00	2.653831	1.629058	1.382855	0.036584	0.037289	0.036800	0.653465

Tabla 3. Métrica de errores del algoritmo del experimento 3

Se pudo observar en especial enfoque que en el mae se obtuvo un 1.4 lo cual es considerado un resultado aceptable y en el caso del mape se obtuvo un 0.0376, ósea un 3.76% lo cual es un porcentaje muy bueno así que podemos concluir que el experimento es confiable.



a)



b)

Ilustración 19. Métricas gráficas (Exp 3). A) Error medio absoluto a lo largo del tiempo b) Error medio absoluto porcentual a lo largo del tiempo

Resultados de evaluación

En el primer experimento del algoritmo prophet podemos observar en primera instancia la predicción que tuvo donde se muestra la predicción de la temperatura en un lapso finito de tiempo.

Los puntos en negro son los datos ya existentes, la línea azul es la tendencia que se esta generando, y la zona azul es todo el marco de la tendencia que se esta generando, interpretando las tendencias máximas y las tendencias mínimas.

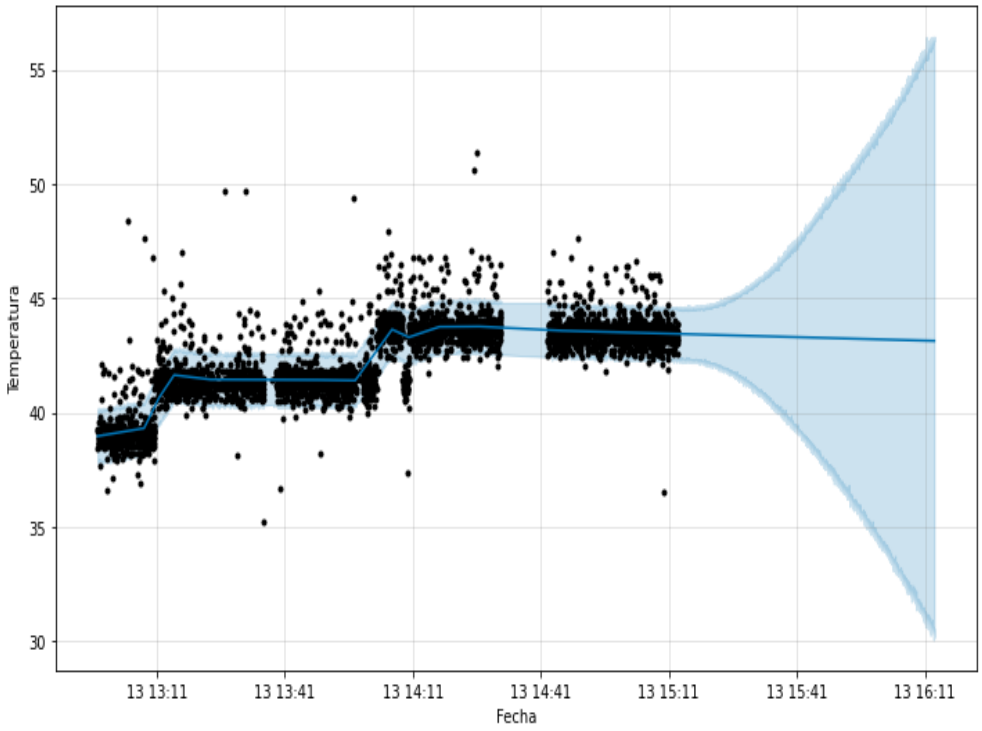


Ilustración 8. Predicción de datos usando el algoritmo Prophet

Por su parte, el algoritmo de random forest, por su naturaleza en la cual no maneja series de tiempo, se tomo uno de los otros sensores que se tenían a disposición en el primer experimento el cual es el Dióxido de carbono, el cual dicho experimento se encargó de identificar las relaciones entre el CO_2 y la temperatura para el análisis de relaciones entre ambas columnas.

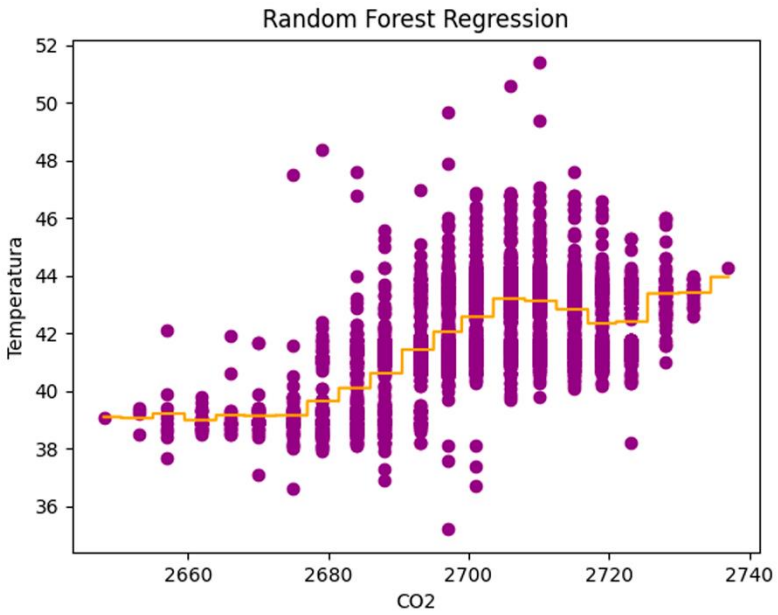


Ilustración 20. Uso del algoritmo de Random Forest en el experimento 1

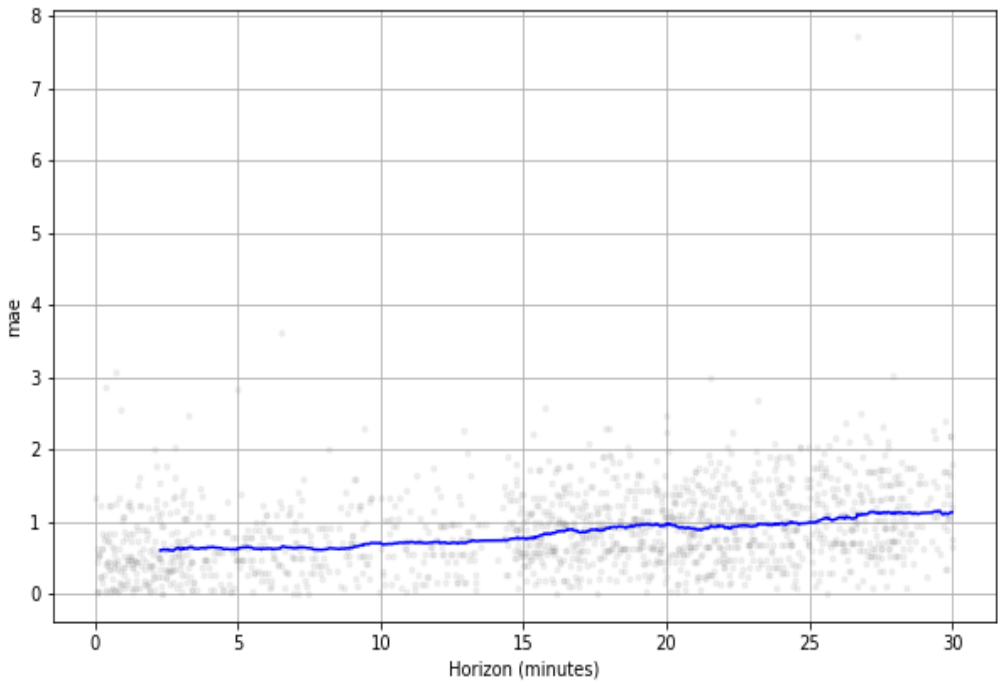
Como podemos observar, ambos experimentos a pesar de tener un mismo data set, los datos que se extraen, varían las columnas a extraer, así como también las necesidades que el algoritmo necesita para transformar, en el caso del prophet se necesita que los datos estén arreglados de manera cronológica, no el caso del Random Forest en este experimento

Como siguiente paso, una vez ejecutado el código para visualizar las métricas, podemos observar en el error absoluto medio que conforme el paso del tiempo, este va variando, algo que es distinto al algoritmo de Random Forest ya que es un algoritmo estático en el tiempo, no así dinámico como el prophet, por lo que conforme para el tiempo, el error absoluto medio va a aumentar progresivamente, esto debido a la incertidumbre que se genera con el paso del tiempo.

horizon	mse	rmse	mae	mape	mdape	smape	coverage
0 days 00:02:17	0.627146	0.791925	0.599317	0.013678	0.010836	0.013654	0.887417
0 days 00:02:18	0.626335	0.791413	0.598495	0.013659	0.010836	0.013635	0.887417
0 days 00:02:19	0.630937	0.794315	0.602559	0.013754	0.010836	0.013729	0.887417
0 days 00:02:20	0.637261	0.798286	0.607162	0.013862	0.010836	0.013835	0.887417
0 days 00:02:21	0.635415	0.797129	0.603730	0.013783	0.010812	0.013757	0.887417

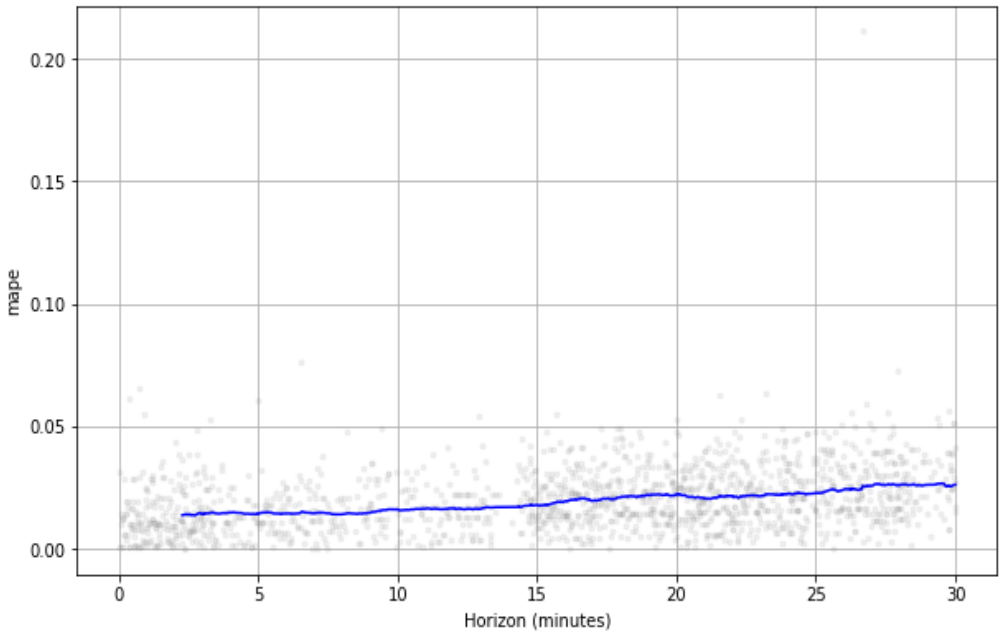
Tabla 1. Métrica de errores del algoritmo del experimento 1

Conforme pasa los minutos podemos observar dentro del algoritmo Prophet, el error absoluto medio y el error absoluto medio porcentual va aumentando progresivamente, en la gráfica podemos observar los datos por lapsos de cada 5 minutos.



a)

Ilustración 8. Métricas gráficas del Experimento 1 algoritmo Prophet. a) Error medio absoluto a lo largo del tiempo



b)

Ilustración 8. Métricas gráficas del experimento 1 algoritmo Prophet. b) Error medio absoluto porcentual a lo largo del tiempo

A continuación, podemos observar en cambio, las métricas recogidas con respecto al algoritmo de Random Forest en el primer experimento, en las tres métricas el resultado a esperar es que estas se acerquen lo mas posible al cero por lo que las métricas nos muestran que en este caso el algoritmo de random forest también es de confiar

Error Absoluto Medio	Error Absoluto Medio Porcentual	Error Cuadrático Medio
0.9983029222082913	0.023750424240294412	1.6335070046859357

Tabla 4. Tabla de métricas del experimento 1 usando el algoritmo Random Forest

Como conclusión en este experimento podemos observar que ambos, tanto el algoritmo Prophet como el algoritmo de Bosques aleatorios tienen unas buenas métricas puesto que ambas son valores cercanos a 0 lo que nos indica que son algoritmos confiables, sin embargo, ambos tienen propósitos distintos en este experimento, puesto que el algoritmo Prophet se encargó de predecir valores a futuro y el Random Forest a relacionar dos variables en relación a sus datos con lo cual podemos decir que en este caso ambos algoritmos pueden servir como

complemento si lo que se busca es información detallada de las variables en cuestión así como de posibles efectos secundarios que podrían surgir como el aumento del dióxido de carbono o de la presión por poner un ejemplo.

Siguiendo con el segundo experimento, este consistió en un dataset que consistió en la captura de los datos de la temperatura de un horno en un lapso de menos de una hora, se puede observar en primera instancia que los datos empezaron desde una temperatura elevada, hay un segmento en el que se observan valores nulos poco despues de las 19:33 horas, sin embargo uno de los problemas que se mostró en este dataset es que hubo muchos lapsos que tenían una picada de datos a ceros lo cual se identificó como un error en el sensor lo cual redujo la cantidad de datos considerablemente.

El experimento se puede ver como en la gráfica de prophet despues de los datos los cuales son los puntos negros, el algoritmo pronostica que la temperatura va a caer en picada, lo cual nos puede decir que el horno ya se apagó para en ese momento.

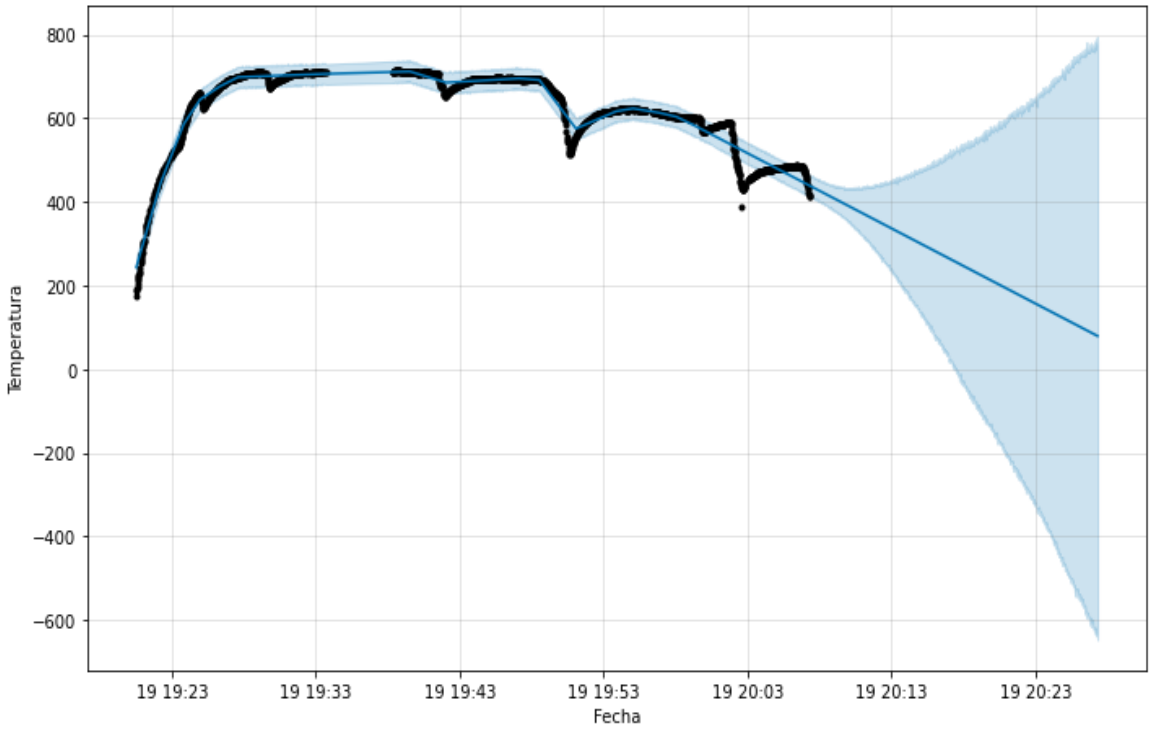


Ilustración 14. Pronóstico de temperatura de un horno con uso del algoritmo prophet en el experimento 2

	horizon	mse	rmse	mae	mape	mdape	smape
0	0 days 00:01:03	2884.115516	53.703962	50.557834	0.083055	0.067482	0.085329
1	0 days 00:01:04	2957.220486	54.380332	51.198147	0.084090	0.067729	0.086434
2	0 days 00:01:05	3026.787089	55.016244	51.772403	0.085011	0.069008	0.087464
3	0 days 00:01:06	3098.114518	55.660709	52.377663	0.085988	0.069008	0.088512
4	0 days 00:01:07	3175.569533	56.352192	53.042116	0.087059	0.069750	0.089717

Tabla 2. Métrica de errores del algoritmo Prophet en el experimento 2

En las métricas del algoritmo prophet podemos observar como las métricas mae y mse los valores se disparan a números muy grandes para una métrica que debería de acercarse al cero lo cual nos indica que el algoritmo no fue capaz de realizar un buen trabajo con el dataset.

El principal problema que se le identificó fue que la cantidad de datos fue muy pequeña puesto que solo eran poco mas de 2 mil datos recabados en menos de una hora, sumado a que también hubo fallas en el sensor por lo que se tuvieron que eliminar esos datos ceros y redujo aun mas la cantidad de los datos y el enfoque que tiene este dataset el cual no está preparado para emplearse con el algoritmo prophet, con lo que no se puede predecir con este algoritmo de manera correcta que se caracteriza por ser un predicador automático con tantas fallas en el dataset.

Asi mismo, al momento de aplicarse en el algoritmo de bosques aleatorios podemos observar en la gráfica que se encargó de comparar los valores dentro del data set con los valores que llegó a predecir el algoritmo Random Forest se pudo observar que siguen siendo valores en picada lo cual es algo que se dio por hecho por el apagado del horno.

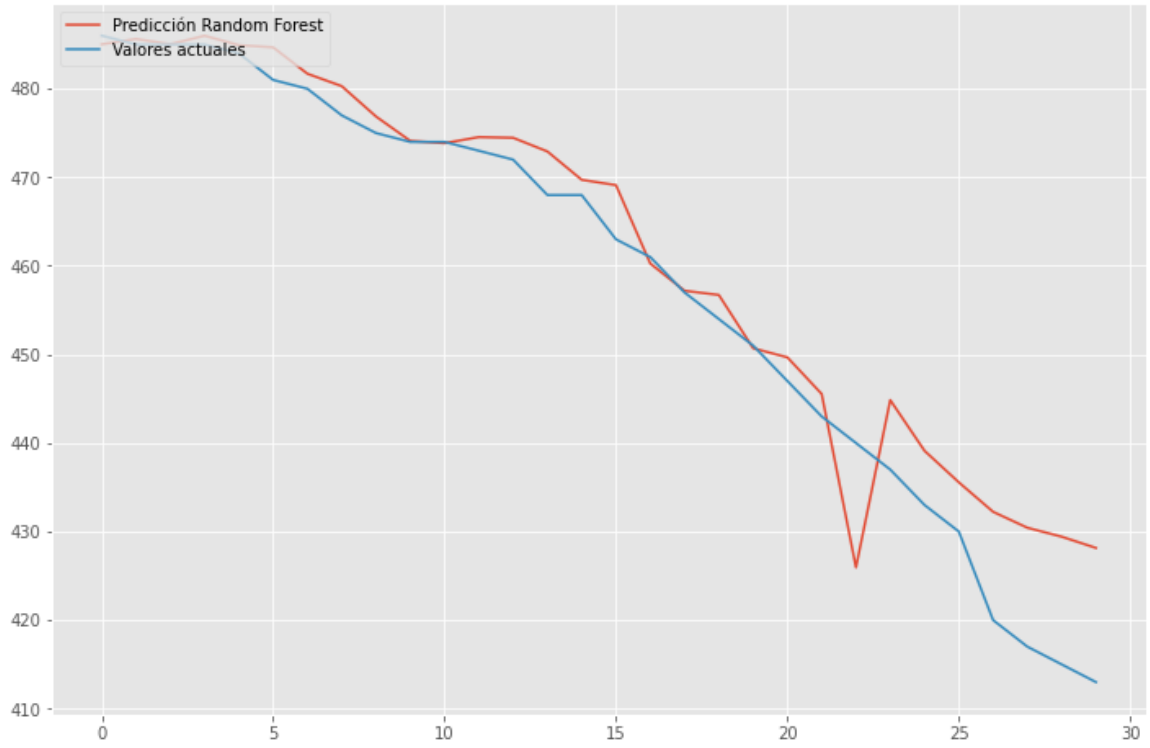


Ilustración 22. Pronóstico de temperatura de un horno con uso del algoritmo Random Forest en el experimento 2

Error Absoluto Medio	Error Absoluto Medio Porcentual	Error Cuadrático Medio
4.303114682539697	**0.009863143513462238**	**6.38629626721255**

Tabla 5. Métricas de errores del algoritmo Random Forest en el experimento 2

Viendo sus métricas, podemos observar que las métricas son más cercanas al valor de cero lo cual comparado con el algoritmo Prophet que tiene métricas muy elevadas en el caso del mae y del mse, este algoritmo es más adaptable a los obstáculos que se presentaron en el dataset, así como también se pudo llegar a la conclusión de que este algoritmo es más confiable ante dichas adversidades que se presentaron en el experimento 2 a comparación del algoritmo Prophet.

CAPITULO V. CONCLUSIONES

Con los resultados obtenidos se pudo lograr el objetivo de adaptar el algoritmo en distintos casos usando una variable en un proceso de manufactura como lo fue la temperatura dentro de una serie de tiempo definida.

En cuanto al algoritmo Prophet, gracias a lo que se observó en el segundo experimento, se concluyó que es un requisito tener un conjunto de datos con una cantidad considerable de información para su análisis, y que la cantidad a predecir no debe de superar a la cantidad original, siempre debe ser incluso 2 o 3 veces menor para que los datos a predecir sean los óptimos, también se detectó de la necesidad de usar la librería de SciKitLearn para dividir los datos de entrenamiento y de prueba ya que esta acción, en los resultados de los tres experimentos demostró que ayuda a mejorar las métricas, así como también un requisito indispensable es que los datos se encuentren en orden histórico ascendente puesto que el tiempo es la variable inamovible en el algoritmo.

Con ayuda de las otras librerías como Matplotlib y Seaborn fue posible hacer un análisis gráfico de los valores que se tenían dentro de los data set que iban siendo creados paso a paso en el Web IDE Google Colab, esto fue de gran ayuda para la limpieza de los datos, ya que el código se ejecutó por segmentos. Lo que permitía un análisis constante y por pasos de la ejecución de los métodos y la limpieza de los datos.

En cuanto a las métricas, se demostró que el algoritmo con un dataset bien adaptado y enfocado al algoritmo, puede arrojar un pronóstico de datos donde el algoritmo es muy confiable ya que tanto el error cuadrado medio, el error absoluto medio y el error absoluto medio porcentual tienen valores cercanos al cero al igual que el algoritmo Random Forest, no es el caso de un conjunto de datos donde presente muchos obstáculos como lo fue en el experimento 2 donde la cantidad de datos era muy poca y la historia era corta, donde se demostró que el algoritmo de Random Forest se adaptó de mejor manera que el algoritmo Prophet el cual arrojó datos muy elevados en las métricas.

Es muy importante saber el enfoque que se le quiere dar a los resultados y también lo que se espera conseguir en los datos de salida, ya que el propósito principal de ambos es distinto, esto se llegó a mencionar como el primer paso dentro de la metodología de resultados donde primero se tuvo que entender el problema (Ilustración 2), donde el algoritmo de prophet es el ideal si lo que se busca es hacer un pronostico de variables en el cual el conjunto de datos de entrada cumpla con las características que el algoritmo demanda, si no es así, el algoritmo Random Forest es de mayor ayuda, así como también en casos donde se requiera una clasificación de los datos el cual es una de sus funciones principales donde el algoritmo Prophet no se llega a involucran.

Para finalizar, se observó pero el algoritmo es muy confiable en base a sus métricas en casos donde los datos de entrada el que este trabaja posee un amplio volumen de datos para poder realizar una predicción con un rango de tiempo de no mas de la mitad del rango de tiempo del conjunto de datos, así como también otras consideraciones con respecto al preparado de los datos y entrenamiento para hacerlos trabajar en base al algoritmo, que la cantidad de datos de entrenamiento y de prueba tengan una relación de 10%, ordenar la tabla de manera cronológica del dato más antiguo al más reciente, así como también omitir los valores que puedan influir en deteriorar la predicción como valores en 0 o nulos como lo fue en el caso del experimento 1 y 2 donde el sensor en momentos hacia corto circuito y arrojaba esta clase de valores que lejos estaban de ser reales por la naturaleza de los eventos cuantificados.

TRABAJOS FUTUROS

Considero que este proyecto de tesis puede ser ampliado con otro tipo de ejemplos en donde se posean menores limitantes de tiempo y espacio, donde un sensor este monitoreando constantemente las variables criticas durante rangos de tiempo mas elevados, ya que una predicción de horas o minutos no es suficiente como lo fue en el experimento 1 y 2 para emplear el algoritmo en un proceso de mantenimiento predictivo.

FUENTES DE INFORMACIÓN

Amazon Web Services, Inc. (2021, marzo). *Amazon forecast: Guía para desarrolladores.* Amazon Forecast https://docs.aws.amazon.com/es_es/forecast/latest/dg/forecast.dg.pdf

Brea Guzmán, F. D. J. (2022, 3 junio). Introducción al Big Data Networking-T1. Course Hero. https://www.coursehero.com/file/134004250/Big-dato-networking-t1pdf/

Del Rosso, R. (2021, octubre). *Comparación de metodologías de Deep Learning para pronósticos en series temporales.* Universidad de Buenos Aires. https://www.consejo.org.ar/storage/attachments/PPT%20Rodrigo%20del%20Rosso.pdf-HAGgz5QxJm.pdf

Galmés Mifsud, A. (2019, mayo). *Automatic forecasting y sus aplicaciones en Big Data: una comparativa entre algoritmos.* Universitat de Barcelona. http://diposit.ub.edu/dspace/bitstream/2445/142438/1/memoria_tfg_galmes mifsud.pdf

Gandomi, A., & Haider, M. (2014, 3 diciembre). *Beyond the hype: big data concepts, methods, and analytics.* Elsevier Enhanced Reader. Recuperado 15 de agosto de 2022, de https://reader.elsevier.com/reader/sd/pii/S0268401214001066?token=9E7FF79F5968826295D86036BF79CE19576F475DC9C4401046131369EE5AE99B99B9215810F965E32CFE13BA158F4DFC&originRegion=us-east-1&originCreation=20220815051055

Gracia, M. Á. (2020, 11 marzo). *Mantenimiento predictivo mediante Inteligencia Artificial y algoritmos de Deep Learning.* Instituto Tecnológico de Aragón. Recuperado 15 de agosto de 2022, de https://www.itainnova.es/blog/big-data-y-sistemas-cognitivos/mantenimiento-predictivo-mediante-inteligencia-artificial-y-algoritmos-de-deep-learning/

Heras, J. M. (2020, 10 octubre). *Análisis Descriptivo, Predictivo y Prescriptivo de datos.* IArtificial.net. https://www.iartificial.net/analisis-predictivo-y-prescriptivo-con-machine-learning/#Analisis_Descriptivo

IBM Cloud Education. (2020, 15 Julio). *Machine Learning.* IBM. Recuperado 2 de septiembre de 2022, de https://www.ibm.com/mx-es/cloud/learn/machine-learning#:%7E:text=El%20machine%20learning%20es%20una,amplia%20historia%20con%20machine%20learning.

Microsoft. (s. f.). *¿Qué es el aprendizaje automático? Microsoft Azure.* Recuperado 14 de agosto de 2022 de https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-machine-learning-platform/#benefits

Mitchell, T., Buchanan, B., DeJong, G., Dietterich, T., Rosenbloom, P., & Waibel, A. (1990). *Machine learning.* Annual review of computer science, 4(1), 417-433.

Moreno, A., Armengol, E., Béjar Alonso, J., Belanche Muñoz, L. A., Cortés García, C. U., Gavaldà Mestre, R., Gimeno, J. M., et al. (1994). *Aprendizaje automático.* Llibre, Ediciones UPC. http://hdl.handle.net/2099.3/36157

Mirete Blanco, H. (2019, septiembre). *Extracción y predicción de datos de series temporales de reservas de vuelo.* Universitat De València. https://www.uv.es/lapeva/Thesis/TFM_2019_Hector_Mirete.pdf

Nexus Integra. (2021, 28 diciembre). *Aplicaciones del Machine Learning en la industria.* Recuperado 15 de agosto de 2022, de https://nexusintegra.io/es/aplicaciones-del-machine-learing-en-la-industria/

Ortega, C. (2021, 13 abril). *Investigación mixta. Qué es y tipos que existen.* QuestionPro. Recuperado 16 de agosto de 2022, de https://www.questionpro.com/blog/es/investigacion-mixta/

Romero Gelvez, J. I., & Rincón Quintero, B. S. (2020). Aplicación de machine learning en el mantenimiento predictivo industrial con herramientas de código abierto. Universidad de Bogotá Jorge Tadeo Lozano.

https://expeditiorepositorio.utadeo.edu.co/bitstream/handle/20.500.12010/10108/Trabajo%20de%20grado.pdf?sequence=1&isAllowed=y

Romero Rojas, B. (2020, octubre). *Una introducción a los modelos de machine learning.* Benemérita Universidad Autónoma de Puebla. https://hdl.handle.net/20.500.12371/10527

Shin, T. (2021, 13 diciembre). *All Machine Learning Models Explained in 6 Minutes*. Towards Data Science. https://towardsdatascience.com/all-machine-learning-models-explained-in-6-minutes-9fe30ff6776a#aff7

Zhou, L., Pan, S., Wang, J., Vasilakos, A. (2017). *Machine learning on big data: Opportunities and challenges*. Neurocomputing, Volume 237, 350-361. https://doi.org/10.1016/j.neucom.2017.01.026.

ANEXOS

Anexo 1

```
from sklearn.model_selection import train_test_split

df1_train, df1_test = train_test_split(df1, test_size=0.10,

random_state=10)
```

Anexo 2

```
m = Prophet()

m.fit(dataframe)

future = m.make_future_dataframe(periods=3600, freq='S',

include_history=True)

forecast = m.predict(future)
```

Anexo 3

```
df_cv = cross_validation(m, horizon='30 minutes')

df_p = performance_metrics(df_cv)

df_p.head()

fig = plot_cross_validation_metric(df_cv, metric='mae')

fig = plot_cross_validation_metric(df_cv, metric='mape')
```

Anexo 4

```
df = df.sort_values('tiempo')
```

Anexo 5

```
m = Prophet()

m.fit(dataframe)

future = m.make_future_dataframe(periods=14400, freq='min',in
clude_history=True)

forecast = m.predict(future)
```