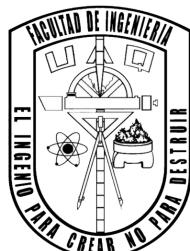


Universidad Autónoma
de Querétaro



Facultad de
Ingeniería

UNIVERSIDAD AUTÓNOMA DE QUERÉTARO

1 de diciembre de 2023

Autor: *Ing. Carlos Manuel Sánchez Martínez*

*Machine Learning (curso), Dr. Marco Antonio Aceves
Fernández*

Tarea 5: Series de Tiempo

Resumen

Como último de los escritos en el curso, describiremos y mostraremos las aplicaciones del modelo "Informers", el cual, puede ser una herramienta poderosa que nos proveerá información clave en la toma de decisiones de las distintas áreas aplicativas en las que este pueda tener, como: predicciones del clima, trayectorias de objetos, movimientos de robots o en nuestro caso, aplicado en un contexto de análisis de series de tiempo y predicciones de precios. La importancia de incorporar pronósticos que ayuden en la demanda de almacenamiento en productos perecederos dentro de la cadena de frío, tiene importancia de impacto económico y social, ya que es posible el estimar, volúmenes de almacenamiento que preparan a las empresas de adquirir posibles requerimientos adicionales de sus instalaciones y así mismo para el personal y materiales necesarios para la movilidad de los productos

Palabras clave: Deep Learning; Aprendizaje Supervisado; Redes Neuronales; Redes Recurrentes; LSTM; Transformers; Informers; Series de Tiempo; Minería de Datos; Soporte a la Decisión;

1. Introducción

El análisis de series de tiempo ha sido aplicado en distintos temas que han ido evolucionando a través de los años, algunos de ellos van desde encontrar patrones de datos para predecir la calidad de productos, la predicción de precios, costos para la medición de las altas y bajas del mercado, demanda de energía, entre otros [1][13]. Esta sección de la estadística siempre ha sido un área muy importante, ya que, reconocer la naturaleza compleja de los datos de mercado, pueda ayudar a incrementar nuestras ganancias y disminuir nuestros costos de producción. Estos datos, recopilados en un periodo de tiempo, presentan desafíos, errores humanos, tendencias no lineales y factores externos impredecibles. Por lo tanto, el preprocesamiento y la normalización de estos datos se convierten en pasos fundamentales antes de aplicar cualquier técnica analítica avanzada.

El uso de modelos avanzados de series de tiempo, como Transformers, modelos de redes neuronales y algoritmos de aprendizaje profundo, permite no solo realizar predicciones precisas, sino también identificar patrones y anomalías [9]. Estos modelos son capaces de manejar la complejidad dinámica de los datos de mercado, ajustándose a sus variaciones y ofreciendo predicciones confiables y valiosas para la toma de decisiones estratégicas en el ámbito comercial.

El análisis de series de tiempo para la predicción de precios de una tienda es un claro ejemplo de cómo las tecnologías de la información y las técnicas avanzadas de análisis de datos pueden ser aplicadas para obtener una comprensión más profunda y generar conocimientos prácticos en el mundo del comercio y los negocios. Con estas herramientas, los analistas no solo pueden predecir hacia dónde se dirigen los precios, sino también entender por qué se mueven en esa dirección, lo que permite a las empresas adaptarse de manera más eficiente y estratégica en un mercado en constante cambio.

2. Marco teórico

Con el paso de los años la generación de datos ha incrementado de manera exponencial, la generación de información de dispositivos conectados a internet y sistemas embebidos, ha generado un interés de las empresas en el procesamiento de toda esta información de manera masiva [19]. El primer acercamiento a la descripción de lo que es el big data, lo realizó Laney, quien presentó una definición para explicar lo que este era y lo hizo, utilizando 3 letras "v", que conformaban: el volumen, velocidad y variabilidad [14], pero años posteriores Borne incrementó la definición, añadiendo veracidad, validez, vocabulario y vaguedad [4]. En abril del 2014 el IDC (International Data Corporation) confirmó el gran crecimiento de los datos, fundamentado a partir de la cantidad de información generada a partir de dispositivos de IoT, como (servidores, sensores, PC's, cámaras, etc.), se predijo que para los años que rondan el 2020, el consumo de información sería de 44 zettabytes (1 Zettabyte=1000 millones de Gigabytes), el cual se quedó un poco corto con los datos generados a partir del covid-19 que fue 64.2 zettabytes para el 2020 [20] y actualmente con un aproximado de 163 zettabytes, según la firma IDC en 2022 [22]. Con esto ponemos en el lector, la cantidad de información generada día con día y cuál importante es la persona que sepa manejarla.

El análisis de datos es un área de la estadística que tiene como objetivo: crear conjuntos de información representativa, los cuales son capturados computacionalmente y representados con valores numéricos [29][23], esto lo hace mediante la construcción de matrices con codificaciones a una mínima pérdida de información. Los análisis de las matrices nos ayuda a encontrar relaciones entre grupos, los cuales nos hacen posible el encontrar nuevas observaciones que nos ayudarán en las futuras decisiones que pudieran ser de gran importancia en nuestra empresa. Sin importar en el área en que nosotros estemos, ya sea, biología, geografía, derecho, finanzas, etc. existe un principal problema y es el que las empresas tienen malos protocolos al manejar enormes cantidades de data. Desde el mal uso de discos de duros de almacenamiento, malas prácticas al llenar bases de datos, hasta no saber qué información es la que se tiene almacenada. Esto deriva una problemática mayor, ya que,

según estudios sobre datos únicos y replicados, menciona que en la esfera de datos global existe una proporción de 1:9 a 1:10 datos únicos/replicados [21]. Uno de los principales desafíos de las empresas es el extraer información y darle un valor a estos datos [30].

Con ayuda de la evolución del software, los problemas a resolver han ido cambiando conforme a las necesidades actuales de la humanidad y con los años surgió uno de los paradigmas de la ingeniería en software, el cual es el paradigma con soporte a la decisión, el cual aborda temas para la generación, evaluación y priorización de soluciones [24]. Los sistemas con soporte a la decisión se utilizan en una variedad de campos y su objetivo principal es mejorar la eficiencia, consistencia, y calidad de las decisiones tomadas a partir de las características principales de nuestra data empresarial, la arquitectura es capaz de procesar, administrar y almacenar información que es representativa en el mercado y con el análisis correcto es posible mostrar gráficas, o elementos visuales que nos ayuden a tomar decisiones de mercado, como posibles predicciones en los incrementos de costos, o alteraciones que sufrirá el mercado que podamos prevenir. Existen artículos como el de Tofan [27], que proporcionan una visión sistemática del estado realizado sobre artículos publicados entre 2002 y 2012 que hablan sobre arquitecturas de software que se han aplicado, así como la escalabilidad, confiabilidad y decisiones arquitectónicas en grupo, Este trabajo es relevante tanto para la comunidad académica como para los practicantes del área, ya que ofrece direcciones prometedoras para investigaciones futuras.

2.0.1. Aprendizaje de Máquinas

El Aprendizaje de Máquinas (AM) es un subcampo de la inteligencia artificial (IA) dedicado al desarrollo de algoritmos y modelos que permiten a los sistemas mejorar su rendimiento en una tarea específica mediante la experiencia [18]. El Aprendizaje de Máquina puede categorizarse principalmente según el tipo de aprendizaje o entrenamiento que implementan los algoritmos. Uno de los tipos entrenamiento que existen es el aprendizaje supervisado. El cual es un enfoque donde los modelos se entrena utilizando un conjunto de datos etiquetado, es decir, cada muestra de entrenamiento está asociada con una etiqueta o resultado [33]. Este método es ampliamente aplicado en tareas como clasificación y regresión. Lo podemos definir como un proceso en el que un modelo es entrenado mediante un conjunto de datos de entrada donde las respuestas correctas son conocidas. El modelo hace predicciones o clasificaciones basadas en la entrada y es corregido cuando sus predicciones son incorrectas [26].

En 1943, McCulloch y Pitts presentaron un modelo simplificado de cómo funcionan las neuronas en el cerebro. En su modelo, cada neurona la representa como una unidad simple, realizando cálculos con los operadores lógicos [17], aunque esto como tal no representa toda la complejidad pura que tiene el cerebro, pero ha sido la que encamino a todos a realizar increíbles cosas en la actualidad. Uno de los trabajos derivados de esto son los de aprendizaje profundo y como representan las activaciones entre capas neuronales [15], el cual un método de aprendizaje que representa de forma más compleja el aprendizaje realizado por una computadora, en este método se tienen diferentes niveles de representación compuestos por módulos que pueden ser o no lineales, estos empleados para la transformación de entradas numéricas destinadas a obtener una cantidad de salidas deseadas.

2.0.2. Redes Recurrentes

Las redes neuronales recurrentes, o por sus siglas RNNs, son otro tipo de redes neuronales que surgieron como solución a uno de los muchos problemas que contenían las redes neuronales tradicionales, como lo fue la perdida de contexto en una secuencia de información. Uno de los primeros acercamientos que se realizaron para tratar de imitar el pensamiento humano en una computadora, fue el de Rumelhart [25], el cual estableció algunos fundamentos del aprendizaje mediante retropropagación, aunque no aporto como tal en temas de redes

recurrentes, hizo la base para la optimización de pesos, la cual es fundamental en entrenamientos de redes neuronales [15].

Una de la característica clave que posee, es que le permite recordar información previa en la red, por medio de ciclos internos, donde la salida de una capa de neuronas, en un tiempo determinado, retroalimenta como entrada para fungir como ayuda para nuestro siguiente tiempo de entrenamiento. Esto permite a este tipo de red el procesar no solo entradas individuales, sino también el aceptar secuencias de datos.

Definición:

$$h_t = f(W \cdot h_{t-1} + U \cdot x_t + b) \quad (1)$$

donde:

- h_t es el estado oculto en el tiempo t .
- h_{t-1} es el estado oculto en el tiempo $t - 1$ (estado anterior).
- x_t es la entrada en el tiempo t .
- W y U son matrices de pesos.
- b es un vector de sesgo.
- f es una función de activación no lineal, como la función tanh o ReLU.

2.0.3. Long Short-Term Memory (LSTM)

Por otro lado, las redes LSTM (Long Short-Term Memory) son una variante avanzada de las Redes Neuronales Recurrentes (RNNs), diseñadas para superar los desafíos de las RNNs tradicionales, como el problema del gradiente desvaneciente, fue desarrollada a partir de los avances en temas como el de HOPFIELD [11], en el cual, se tratan temas interesantes de como construir organismos biológicos con propiedades colectivas entre neuronas. Abordo temas como el significado de la memoria direccional, esto mediante flujos espaciales para nuestro estado en el sistema. Esto encamino a Sepp Hochreiter y Jürgen Schmidhuber en 1997 a desarrollar las LSTMs [10].

Este tipo de redes son particularmente efectivas para aprender dependencias a largo plazo en secuencias de datos. En solución a problemas complejos de manejo de memoria en las RNNs, excediendo los 1000 datos en el tiempo, si producir un desborde de memoria, como sucedía en las tradicionales, esto debido a la cantidad de iteraciones para la retroalimentación de las últimas capas de nuestra red. LSTM contiene una entrada y una salida adicional. Este elemento adicional se conoce como celda de estado, el cual funge como un canal de transporte, en donde se pueden añadir o remover datos que no queremos que sean almacenados en la memoria de la red. Esto lo realiza por medio de compuertas que funcionan como mediadores, las compuestas abiertas permiten el paso de información que se almacenará en mi estado, y si está cerrada, no es de vital importancia el recordarlo, por lo que no lo tomamos en cuenta. Todo esto mencionado se realiza por medio de funciones matemáticas, las ecuaciones usadas para una LSTM son:

- Puerta de Olvido:

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

- Puerta de Entrada:

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

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

- Estado de la Celda:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

- Puerta de Salida:

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

$$h_t = o_t * \tanh(C_t)$$

2.0.4. Transformers

En la actualidad, la palabra ChatGPT, entre los usuarios de tecnología de la información, ha sonado cada vez más [2][3][5]. Esta herramienta desarrollada por OpenAI, explotó un interés que se tenía, pero no sobresalía en comparación de modelos de detección. Con la llegada de este modelo, capaz de entender textos, mantener el hilo de la conversación, con capacidad de procesar todo un artículo, generar respuestas de la mayoría de temas (Finanzas, Economía, Medicina, Programación, etc.) y todo esto en una velocidad nunca antes vista, la comunidad de programadores en la red, puso un gran interés sobre los nuevos paradigmas que este pueda tener.

Los Transformers son una arquitectura de procesamiento de lenguaje natural o secuencias de tiempo [31], la idea general de los Transformers se basa en tener dos capas principales, una codificadora y un decodificador. La codificación posee mecanismos especializados, los cuales llaman “atención”, que se basa en conceptos de “keys”(llaves), “values”(valores) y “queries”(consultas). Este mecanismo es fundamental para entender cómo funcionan estos modelos, donde cada elemento de entrada (ej. una palabra en una oración, o una secuencia en el tiempo) se transforma en un vector mediante una capa de embedding, a la cual se le añade una codificación posicional a estos vectores para incorporar información sobre la posición de cada elemento en la secuencia. Para después pasarlos por estos mecanismos. Donde realiza la parte más interesante de este tema, ya que calcula un puntaje de atención para cada par de elementos en la secuencia, comparando el vector del “query” de un elemento con el vector de “key” de otro. Calculándolo posteriormente con el producto punto entre el “query” y el “key”, seguido de una normalización y una función softmax que nos ayude a ponderar los vectores. Esto lo realizan n cantidad de veces por “head”.

- Codificación: Se compone de un conjunto de 6 capas, cada uno similares en figura y estructura. Dentro de cada capa, existen sub capas que desempeñan funciones específicas. La primera sub capa es un mecanismo de auto-atención de múltiples cabezales “heads”, que permite al modelo atender diferentes posiciones de la entrada en paralelo. La segunda sub capa es una red de alimentación directa simple, que procesa la información de manera secuencial y está completamente conectada. Una característica importante de esta arquitectura es que usa conexiones residuales alrededor de cada sub capa. Esto significa que la salida de cada sub capa se suma a su entrada original, antes de pasar por una normalización de capa. Matemáticamente, esto se representa como $\text{LayerNorm}(x + \text{Sublayer}(x))$, donde “x” es la entrada a la sub capa y “Sublayer(x)” es la función implementada por la sub capa misma. Este enfoque ayuda a mitigar el problema del desvanecimiento del gradiente en redes profundas, permitiendo que el modelo aprenda de manera más efectiva.

1. Positional Encoding:

$$PE(pos, 2i) = \sin\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right) \quad (2)$$

$$PE(pos, 2i + 1) = \cos\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right) \quad (3)$$

2. Multihead Attention:

$$(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \quad (4)$$

donde cada cabeza se define:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \quad (5)$$

y la función de activación:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (6)$$

3. Self-Attention:

$$\text{Self-Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (7)$$

■ Decodificación: Al igual que el codificador, consta de un conjunto de 6 capas, cada una idéntica en su estructura. No obstante, el decodificador incorpora una sub capa extra en cada una de sus capas, resultando en un total de tres sub capas por capa. La primera sub capa funge como mecanismo de autoatención, como el del codificador. La segunda es también una red de alimentación directa. La tercera sub capa, que es única del decodificador, realiza operaciones de atención sobre la salida de la pila del codificador. Esto permite que el decodificador procese la información recibida del codificador y la integre para generar la salida final.

Al igual que en el codificador, se emplean conexiones residuales alrededor de cada sub capa, seguidas por una normalización de capas. Esta arquitectura ayuda a evitar problemas de aprendizaje en redes profundas y asegura la estabilidad en el flujo de información. Una característica distintiva del decodificador es la modificación en la sub capa de autoatención para prevenir que las posiciones atiendan a posiciones futuras. Esto se logra mediante un mecanismo de enmascaramiento. Esta técnica, en combinación con el hecho de que las incrustaciones de salida están desplazadas por una posición, asegura que las predicciones para una posición dada solo puedan depender de las salidas conocidas en posiciones anteriores a i. Este enfoque es crucial para tareas como la generación de texto, donde el contexto futuro no debe influir en la predicción actual.

1. Self-Attention del Decodificador:

$$(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}} + M\right)V \quad (8)$$

M es una máscara que previene la atención a posiciones posteriores, asegurando que las predicciones para una posición solo dependan de posiciones conocidas.

2. Atención Entre Codificador y Decodificador:

$$(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (9)$$

En esta atención, Q proviene de las salidas de la capa anterior del decodificador, y K y V son las salidas del codificador. Esto permite que cada posición en el decodificador preste atención a todas las posiciones en la secuencia de entrada del codificador.

3. Lineal de Salida y Softmax:

$$h = \text{softmax}(W^O \cdot \text{Salida del Decodificador}) \quad (10)$$

Donde W^O es una matriz de pesos, y h es la salida de la última capa del decodificador. Esta salida se procesa a través de una capa lineal y luego por softmax para generar las predicciones finales.

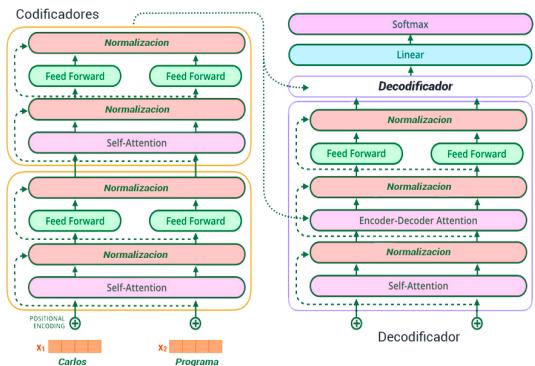


Imagen 1. Arquitectura de Transformers, separando en pasos generales, la codificación y decodificación de entradas de texto.

2.0.5. Informers, una versión mejorada

Los Informers [34], son una evolución de la arquitectura Transformer para el procesamiento de secuencias de tiempo largas, adaptándose a las necesidades de análisis en series temporales. Esta arquitectura se distingue por incorporar capas especializadas que optimizan el tratamiento de secuencias extensas. Los mecanismos clave de los Informers incluyen su método ProbSparse, que se basa en un enfoque selectivo hacia las keys (llaves) y values (valores) más relevantes, reduciendo la complejidad computacional y de memoria. Este enfoque es esencial para entender cómo los Informers procesan eficientemente grandes volúmenes de datos temporales.

Cada elemento de la secuencia (por ejemplo, un punto en el tiempo) se transforma en un vector a través de una capa de embedding, similar a los Transformers. Sin embargo, en los Informers, se añade una innovación en la codificación posicional y se utilizan generadores de máscaras de atención para focalizar en segmentos cruciales de la secuencia. Estos generadores de máscaras seleccionan partes de la secuencia que son más informativas, permitiendo que el modelo se concentre en los aspectos más relevantes de los datos.

1. ProbSparse Self-Attention:

$$PA(Q, K, V) = \text{softmax} \left(\frac{\hat{Q}K^T}{\sqrt{d}} \right) \hat{V} \quad (11)$$

Donde \hat{K} y \hat{V} son matrices reducidas de K y V respectivamente, seleccionadas mediante un proceso de muestreo que prioriza las entradas más informativas. Esto reduce la complejidad computacional y de memoria. \hat{Q} es una matriz dispersa del mismo tamaño de q y solo contiene las consultas Top-u bajo la medida de escasez

2. Generador de Máscaras de Atención:

$$M = \text{GeneradorMáscaras}(Q, K) \quad (12)$$

Este generador produce una máscara M que se aplica durante la auto-atención para enfocarse en las partes más relevantes de la secuencia.

3. Mecanismo de Atención Destilada:

$$\text{Atención Destilada}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V \quad (13)$$

Aquí, la atención se calcula de manera similar a los Transformers estándar, pero con un enfoque en filtrar la información más relevante de largas secuencias.

4. Positional Encoding con Convoluciones:

$$\text{Positional Encoding Conv}(X) = \text{Convolución}(X) \quad (14)$$

Los Informers pueden utilizar una capa de convolución para incorporar información posicional en los datos de entrada, lo que es especialmente útil para series temporales.

2.0.6. Otros modelos:

LSTM, Redes Recurrentes, GRU, Data2Vec, Redes Neuronales, Bayes, X-GBOOST.

2.0.7. Series de Tiempo

Las series de tiempo $\{X_t\}$, las podemos definir como secuencias de datos en el tiempo, tomados a partir de puntos de observación, los cuales, son capturados en algún instante en el tiempo y llevan un orden cronológico [32].

Definición:

$$\text{Serie de tiempo } X_t = f(t) + \epsilon_t \quad (15)$$

Las podemos clasificar según las características que estas poseen: Estacionarias y No estacionarias. Las estacionarias poseen una media y varianza constantes a lo largo del tiempo,

por lo que se les considera como estables. Los valores tienden a estar cercanos a la media constante. Por otro lado, las no estacionarias, son series en las que la tendencia cambia a lo largo del tiempo, por lo que la media tiende a crecer o decrecer a través del tiempo. Las series de tiempo pueden poseer componentes de tendencia, que son cambios a largo plazo en relación con la media, y componentes estacionales, los cuales son cambios periódicos (semestrales, mensuales, anuales, etc.), ejemplo ventas navideñas, por último tenemos los componentes aleatorios, los cuales nos corresponden a ningún patrón de comportamiento, esto es resultado de afectaciones aleatorias del universo [13][16].

A veces estos datos se comportan de diferentes formas, las cuales podemos representar como procesos estocásticos, son secuencias de datos que van evolucionando con el tiempo, una serie temporal es un caso particular de estos procesos, un claro ejemplo donde podemos encontrarlo es en el ruido blanco, donde los valores son independientes entre sí e idénticamente distribuidos. Los estocásticos estacionarios corresponden a secuencias de datos en las que su varianza y la media son constantes en todo el intervalo de tiempo. Existen formas de ver si una variable es dependiente de otra, ya sea entre observación y entre datos en la secuencia. Existen fórmulas como la función de autocorrelación, que nos ayudan a darnos un acercamiento a este tipo de peculiaridades.

Definición Autocorrelación:

$$\rho_k = \frac{\sum_{t=1}^{T-k} (X_t - \bar{X})(X_{t+k} - \bar{X})}{\sum_{t=1}^T (X_t - \bar{X})^2}$$

3. Materiales y métodos

1. Base de datos utilizada: En esta práctica, se utilizó conjunto de atributos proporcionados por el docente, la cual contiene las columnas: Lactose, Whey Central, Whey West, Whey East, DPSR Whey AVG, CME Whey AVG, NonFat Central/East, 34p WPc, entre otras. Información referente a los precios de lácteos y sueros proteína.
2. Frameworks usados: Con la ayuda de herramientas de desarrollo como Python, utilizamos una de sus muchas librerías, Jupyter Notebooks, el cual nos ayudó en el desarrollar, código interactivo, el cual es ejecutado en aplicaciones web y nos proporciona un kit de herramientas para una visualización intuitiva a los datos descriptivos de nuestra base de datos. Para la lectura de nuestra base de datos (.csv) utilizamos Pandas, la cual es una librería de programación en Python que proporciona herramientas para el análisis y la manipulación de datos, para el manejo de arreglos utilizamos NumPy, la cual nos proporciona una manera optimizada del uso de matrices de nxn. Y por último se utilizaron 2 librerías como medio para representar los datos de manera gráfica, seaborn y matplotlib, las cuales están enfocadas al 100 % en la visualización de datos estadísticos y nos han ayudado en crear gráficos atractivos.

3. Recursos de Cómputo:

- Procesador: 11th Gen Intel(R) Core(TM) i5-11600K @ 3.90GHz, 3912 Mhz, 6 procesadores principales, 12 procesadores lógicos.
- Tipo de sistema: x64-based PC.
- Sistema Operativo: Microsoft Windows 11 Home Single Language.
- Memoria Física (RAM): 48.0 GB.
- GPU: NVIDIA GeForce RTX 3060.
- Memoria GPU Dedicada: 12 GB.

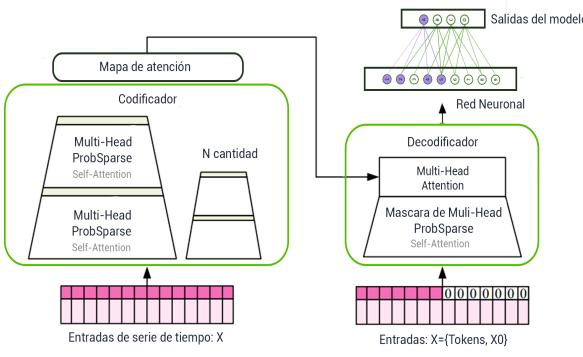


Imagen 2. Arquitectura de Informers, separando en pasos generales, la codificación y decodificación de entradas de series de tiempo.

3.0.1. Caso de estudio

La leche que más se produce en el país es la de bovino y ocupa el tercer lugar por su valor agregado en la rama de la industria de alimentos, con el 17.22 % del valor nacional, solo por detrás de la carne de bovino (30 %) y la carne de ave (23 %) [7]. En esta práctica, nos enfocaremos primordialmente sobre la lactosa, la cual, es un tipo de azúcar que se encuentra naturalmente en la leche y en los productos lácteos. Tiene varios usos y relevancias en nuestro día a día, como el consumo diario de productos derivados de la leche [8]. Otro de los puntos en los que se puede usar la lactosa es como una fuente importante de energía para el cuerpo humano. Al ser un carbohidrato, se descompone en glucosa y galactosa en el cuerpo, proporcionando energía para las actividades diarias, esto es usado en sueros para atletas o barras energéticas [6].

3.0.2. Métodos Empleados

a) Preprocesamiento: Para el correcto uso de la base de datos proporcionada, fue necesario aplicar un conjunto de protocolos que nos ayudaron a generar una base de datos nueva sin alterar los más que se pueda las distribuciones de los valores. Algunos de los procedimientos realizados fueron: eliminación de las filas que contenían el Avg, eliminación de filas duplicadas en el tiempo, aquellas que no contenían información relevante, se utilizó un formato de fecha ISO, MM/DD/YY, se añadieron 0 a los faltantes, se detectaron también inconsistencias en las columnas, donde existían en algunos años y en otros no, también inconsistencias con los nombres, poseían diferente nombre de columna en instantes del tiempo. Después del acomodo de columnas, se realizó un despliegue de la información, en gráficos, para ver las distribuciones de los datos, en los cuales se les aplicó una técnica de imputación, la cual consistía en una práctica antes realizada, en la que se aborda el algoritmo de KNN donde se buscan los valores cercanos a este. De la misma manera se observaron comportamientos con otro tipo de imputaciones, como regresiones lineales o con la media de la distribución.

b) Métricas

Cálculo de residuos

$$\text{Residuo}_i = Y_i - \hat{Y}_i \quad (16)$$

Error Cuadrático Medio (MSE):

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (17)$$

Error Medio Absoluto (MAE):

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (18)$$

Optimizador Adam [12]:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (19)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (20)$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (21)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (22)$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \hat{m}_t \quad (23)$$

4. Resultados

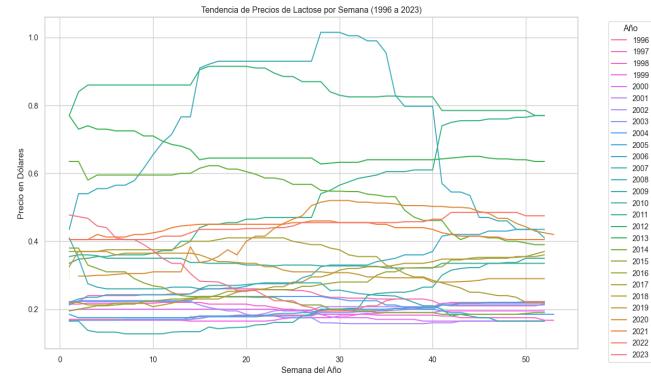


Imagen 3. Gráfica de Lactosa, se muestra en la serie de tiempo, como se comporta el precio de la lactosa, conforme pasan los años.

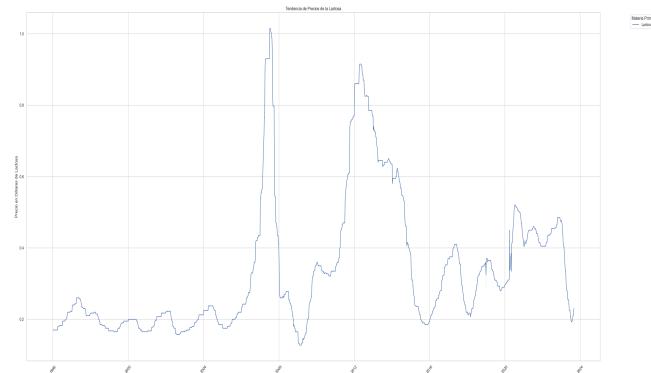


Imagen 4. Gráfica de Lactosa, se muestra en la serie de tiempo como se comporta la lactosa, conforme pasan el tiempo de manera general.

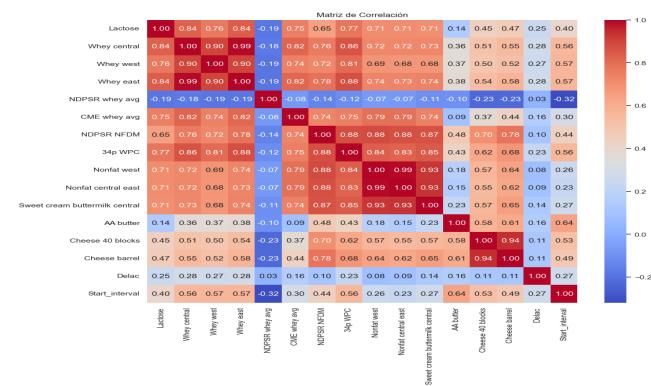


Imagen 4. Gráfica de Correlación, se muestra como es que se relacionan los datos entre sí.

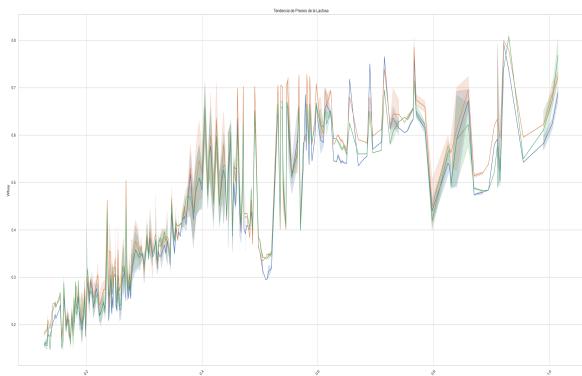


Imagen 6. Gráfica, Lactosa vs Suero en diferentes ubicaciones geográficas.

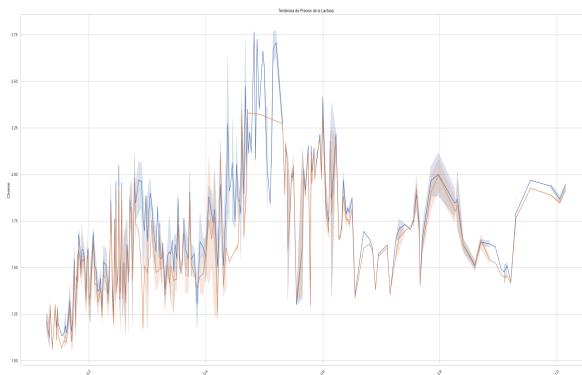


Imagen 7. Gráfica, Lactosa vs Cheese.

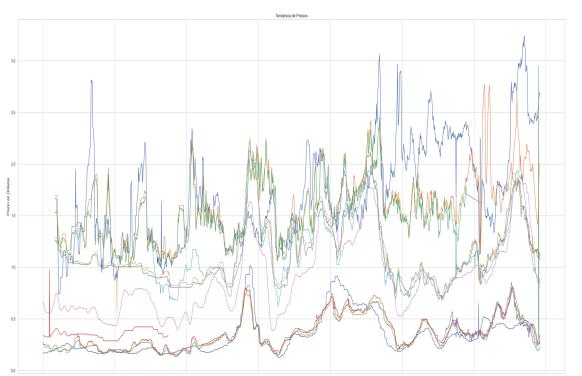


Imagen 8. Gráfica, de Columnas vs Tiempo.

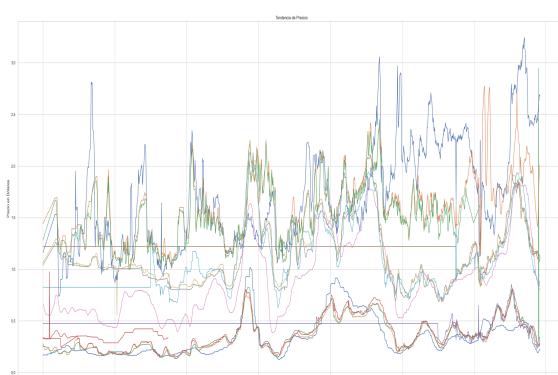


Imagen 9. Gráfica, Aplicando procesamiento de datos, pruebas.

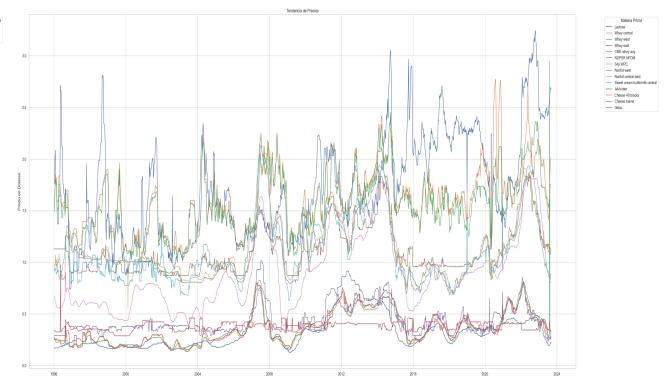


Imagen 10. Gráfica, Aplicando procesamiento de datos, imputación.

Predicciones De Lactosa	Dia
0.33859226	1
0.28946777	2
0.4732766	3
0.15462655	4
0.3286983	5
0.04484865	6
0.022695022	7
0.02790148	8
0.053448	9
0.2048891	10
0.05560016	11
0.4998224	12
0.37474492	13
0.430582	14
0.20114549	15
0.07740324	16
0.218656	17
0.10073473	18
0.022405097	19
0.1325227	20
0.007607089	21
0.35000512	22
0.36366132	23
0.10000001	24
0.008784344	25
0.083668703	26
0.07575781	27
0.050607346	28
0.045700908	29
0.0547999132	30

Imagen 11. Resultados, Predicciones obtenidas con el modelo.

Experimento	MSE por experimento	MAE por experimento	MSE
5	mse=0.72532820701991	mae=1.60323965549469	Entrenamiento Loss: 0.2193756 Validacion Loss: 1.1310687 Pruebas Loss: 5.4705334
4	mse=2.521484917074463,	mae=1.945710597445679	Entrenamiento Loss: 0.2152194 Validacion Loss: 3.229007
3	mse=1.7559610695239688,	mae=1.896603107452393	Entrenamiento Loss: 0.2773816 Validacion Loss: 1.7610815 Pruebas Loss: 3.4899728
2	mse=3.3509252071380615	mae=1.71558857268982	Entrenamiento Loss: 0.2687720 Validacion Loss: 1.7477277 Pruebas Loss: 2.3186407
1	mse=4.7895812988282125	mae=2.1139848232269287	Entrenamiento Loss: 0.2362071 Validacion Loss: 1.4931296 Pruebas Loss: 5.2300835

Imagen 12. Imagen con Métricas del modelo, MSE obtenidos con el modelo a través del entrenamiento, validación y pruebas.

5. Discusión

Uno de los principales retos obtenido en la práctica, fue la de unificar y procesar la información de tal manera de que fuera útil para los modelos, las técnicas usadas para la normalización constaron de pruebas con diferentes técnicas realizadas a lo largo del curso, fueron puestas en comparación con los resultados de la distribución de datos. Se verificó él antes y él después de imputar no afectar la distribución original, por lo que se descartaron técnicas como imputación a partir de la media, regresión lineal simple y se optó por usar la técnica empleada en la anterior práctica, que fue buscando los vecinos cercanos a partir de las características que tenemos. Se notó que hubo un incremento en el precio de la lactosa por los años 2000 al 2006, por lo que ayudo a la toma de decisiones al momento de configurar el modelo usado, ya que es necesario la parametrización de la cantidad que batch que el modelo puede recibir, a comparación de los modelos tradicionales de redes neuronales, este tiene una gran ventaja al tener un mayor contexto de los datos recurrentes en la serie de tiempo, esto nos ayudó a generar información nueva sin la perdida de información de los primeros datos de entrada en años anteriores. Se realizó una comparación entre velocidades de CPU y GPU para ver la cantidad de procesamiento que este era distribuido a través de los núcleos computacionales. Con este proyecto se busca buscar interés y agregar nuevas variables de factores locales a la base de datos que pueden ayudar a mejorar la precisión del pronóstico, como: costos de cereales, agua, hidrocarburos y factores ambientales.

6. Conclusiones

El principal factor de interés que tuvo el escritor de este documento respecto a los modelos evaluados, es la gran capacidad de respuesta que tiene el modelo al recibir grandes cantidades de información secuencial, el cual soluciona uno de los principales problemas encontrados en otros modelos basados en transformers, como obtener mayor cantidad de salidas a predecir o de no guardar el contexto de una masiva cantidad de información entrante. La principal innovación de este modelo es en cómo los Informers manejan la atención. A diferencia de los

Transformers, que calculan puntuajes de atención considerando todas las combinaciones de pares de elementos en una secuencia, los Informers emplean un método que distingue y prioriza ciertos elementos, haciendo el proceso mucho más eficiente para secuencias largas. Esta eficiencia los hace especialmente aptos para el análisis detallado y el pronóstico en series temporales extensas. El uso correcto de los datos puede ayudar a disminuir el impacto de los costos que nos pueden generar en la empresa, solo es cuestión de darles un buen uso y administrarlos de manera correcta desde la captura.

Referencias

- [1] ABONYI, J., Fell, B., Nemeth, S., and Arva, P. 2003. Fuzzy clustering based segmentation of time-series. In Proceedings of the 5th International Symposium on Intelligent Data Analysis (IDA 03). Springer, 275–285
- [2] Ahmad, A., Waseem, M., Liang, P., Fehmideh, M., Aktar, M.S., Mikkonen, T.: Towards human-bot collaborative software architecting with chatgpt. arXiv preprint arXiv:2302.14600 (2023)
- [3] Bang, Y., Cahyawijaya, S., Lee, N., Dai, W., Su, D., Wilie, B., Lovenia, H., Ji, Z., Yu, T., Chung, W., et al.: A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. arXiv preprint arXiv:2302.04023 (2023)
- [4] BORNE K. Top 10 big data challenges a serious look at 10 big data v's, Tech. Rep. 2014. [Online]. Available: <https://www.mapr.com/blog/10-big-data-challenges-look-10-big-data-v>.
- [5] Chen, Y., Eger, S.: Transformers go for the lols: Generating (humorous) titles from scientific abstracts end-to-end. arXiv preprint arXiv:2212.10522 (2022)
- [6] CRIBB, Paul J.; COUNCIL, US Dairy Export. Las proteínas del suero de leche de los Estados Unidos y la nutrición en los deportes. US Dairy Export Council, 2005, vol. 1, p. 1-12.
- [7] GALLEGOS-DANIEL, Cecilia; TADDEI-BRINGAS, Cristina; GONZÁLEZ-CÓRDOVA, Aarón F. Panorama de la industria láctea en México. Estudios sociales. Revista de alimentación contemporánea y desarrollo regional, 2023, vol. 33, no 61.
- [8] GUEVARA ALBARRACÍN, Ramón; MARISCAL VILLAO, Jackson. Elaboración de yogurt a partir de suero de leche. 2011. Tesis de Licenciatura. Universidad de Guayaquil. Facultad Ingeniería Química.
- [9] HELWE, Chadi; CLAVEL, Chloé; SUCHANEK, Fabian. Reasoning with transformer-based models: Deep learning, but shallow reasoning. En International Conference on Automated Knowledge Base Construction (AKBC). 2021.
- [10] HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long short-term memory. Neural computation, 1997, vol. 9, no 8, p. 1735-1780.
- [11] HOPFIELD, John J. Neural networks and physical systems with emergent collective computational abilities. Proceedings of the national academy of sciences, 1982, vol. 79, no 8, p. 2554-2558.
- [12] KINGMA, Diederik P.; BA, Jimmy. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.
- [13] KENDALL, Maurice George; HILL, A. Bradford. The analysis of economic time-series-part i: Prices. Journal of the Royal Statistical Society. Series A (General), 1953, vol. 116, no 1, p. 11-34.
- [14] LANEY D. 3D data management: controlling data volume, velocity, and variety. META Group, Tech. Rep. 2001. <https://studylib.net/doc/8000000/3d-data-management-controlling-data-volume-velocity-an>.
- [15] LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. Deep learning. nature, 2015, vol. 521, no 7553, p. 436-444.
- [16] NAVA, F. Alejandro. Procesamiento de series de tiempo. Fondo de cultura económica, 2015.
- [17] McCulloch, W. S., & Pitts, W. (1943). "A logical calculus of the ideas immanent in nervous activity". The bulletin of mathematical biophysics, 5(4), 115-133
- [18] MITCHELL, Tom M. Machine learning. 1997.
- [19] PEÑA, Daniel, et al. Análisis de datos multivariantes. Cambridge: McGraw-Hill España, 2013.
- [20] PETROC TAYLOR, Amount of data created, consumed, and stored 2010-2020, with forecasts to 2025, Nov 16 2023.
- [21] PETROC TAYLOR, Share of unique data and replicated data in the global datasphere in 2020 and 2024, May 23 2022.
- [22] REINSEL D., GANTZ J., RYDNING J. Data age 2025. the digitization of the world: From edge to core. an IDC white paper, zUS44413318. 2018.
- [23] PROVOST, Foster; FAWCETT, Tom. Data Science for Business: What you need to know about data mining and data-analytic thinking. O'Reilly Media, Inc.", 2013.
- [24] RUHE, Günther. Software engineering decision support—a new paradigm for learning software organizations. En International Workshop on Learning Software Organizations. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002. p. 104-113.
- [25] RUMELHART, D. E. David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams Learning representations by back-propagating errors Nature 323: 533-536. nature, 1986, vol. 323, p. 533-536.
- [26] SHALEV-SHWARTZ, Shai; BEN-DAVID, Shai. Understanding machine learning: From theory to algorithms. Cambridge University Press, 2014.
- [27] TOFAN, Dan, et al. Past and future of software architectural decisions—A systematic mapping study. Information and Software Technology, 2014, vol. 56, no 8, p. 850-872.
- [28] TSAI, Chun-Wei, et al. Big data analytics: a survey. Journal of Big data, 2015, vol. 2, no 1, p. 1-32.
- [29] TUKEY, John W., et al. Exploratory data analysis. 1977.
- [30] VAN DER AALST, Wil; VAN DER AALST, Wil. Data science in action. Springer Berlin Heidelberg, 2016.
- [31] VASWANI, Ashish, et al. Attention is all you need. Advances in neural information processing systems, 2017, vol. 30.
- [32] VILLAVICENCIO, Jhon. Introducción a series de tiempo. Puerto Rico, 2010.
- [33] ZHU, Xiaojin; GOLDBERG, Andrew B. Introduction to semi-supervised learning. Springer Nature, 2022.
- [34] ZHOU, Haoyi, et al. Informer: Beyond efficient transformer for long sequence time-series forecasting. En Proceedings of the AAAI conference on artificial intelligence. 2021. p. 11106-11115.