

*Universidad Austral*  
*Facultad de Ingeniería*

Maestría en Ciencia de Datos

**Informe Final**  
Deep Learning - Equipo 7

**Tema Asignado: LSTM**

Julio 2024

**Autores:**

- Aureliano Chavarría
- Gastón Larregui
- Patricia Nuñez

**Asignatura:** Laboratorio 3

**Tema:** LSTM (Long Short-Term Memory)

**Profesor:** Gustavo Denicolay

## Contenido

Repositorio GitHub y GoogleDrive del trabajo	2
Hipótesis Experimental	4
Diseño experimental y materiales	6
Procedimientos y recopilación de resultados	8
Análisis exploratorio de los datos	8
Modelos exploratorios	14
Modelo final	23
Modelo: Función Principal	24
Análisis de los resultados	28
Conclusión	29
Resultados en Kaggle	29
Conclusión Final	30
Referencias	31

## Repositorio GitHub y GoogleDrive del trabajo

El Grupo utilizó el repositorio GitHub del alumno Aureliano Chavarría para cumplir con el requerimiento planteado en la consigna.

### GitHub

#### Repositorio:

[https://github.com/AurelianoChavarria/MCD\\_LAB3\\_Grupo7](https://github.com/AurelianoChavarria/MCD_LAB3_Grupo7)

#### Actualización de archivos

```
# git add . (git add -A)
# git commit -m "Actualizar cambios locales"
# git push origin main
```

### Google Drive

[020 Informe Final Rev02.pdf](#)

## Hipótesis Experimental

### Definición del Problema

La tarea consiste en predecir la cantidad de ventas de toneladas (tn) para el mes de febrero de 2020, utilizando técnicas de machine learning. Esta previsión se basará en el análisis de datos históricos proporcionados por la multinacional, los cuales abarcan el período comprendido entre enero de 2017 y diciembre de 2019.

Es fundamental señalar que la multinacional ha establecido una fórmula específica de error para evaluar los resultados de esta predicción. Todos los resultados serán medidos y comparados utilizando esta fórmula, lo que permitirá al departamento de planificación realizar una evaluación precisa y consistente de la eficacia de las previsiones generadas por nuestro equipo de trabajo.

### Análisis Exploratorio de Datos (EDA)

El análisis exploratorio de datos (EDA) es un paso fundamental en el proceso de preparación de datos para modelos predictivos. En el proyecto, se llevó a cabo un EDA exhaustivo para comprender mejor los datos de ventas y productos proporcionados por la multinacional.

### Fórmula de Error de Pronóstico

A continuación se puede observar la fórmula de error a utilizar:

$$Total\ ForecastError = \frac{\sum_{sku} |(ActualSales - ForecastSales)|}{\sum_{sku} ActualSales}$$

Para soportar la hipótesis experimental, se han estudiado diferentes publicaciones relacionadas con técnicas de modelado de redes neuronales profundas, en particular redes LSTM. La investigación se ha fundamentado principalmente en una variedad de fuentes académicas y técnicas que proporcionan un contexto sólido para el trabajo realizado.

Las bibliografías consultadas incluyen el libro "Long Short-Term Memory Networks with Python", que ofrece una comprensión detallada y práctica de las redes LSTM y su aplicación en diversos problemas de predicción de series temporales. Este recurso fue esencial para el desarrollo y la implementación de los modelos utilizados en los experimentos.

Adicionalmente, se han revisado artículos presentados en la materia "Análisis de Series Temporales 2023", los cuales proporcionaron una visión actualizada y académica sobre las técnicas más avanzadas y las mejores prácticas en el campo del análisis de series temporales. Estos artículos fueron fundamentales para entender las tendencias recientes y los enfoques metodológicos utilizados en la industria y la academia.

Toda la documentación relevante y las referencias bibliográficas utilizadas en esta investigación están adjuntadas en la sección de referencias.

## Diseño experimental y materiales

Durante el periodo de trabajo del presente proyecto se realizaron diferentes experimentos con el fin de obtener el menor error de predicción de los 780 productos seleccionados por la multinacional.

Inicialmente se realizó un análisis exploratorio de los datos con el fin de entender los mismos y obtener algún tipo de información relevante que añadiera mayor valor a la información brindada por la multinacional.

Los experimentos mencionados anteriormente se desarrollaron desde métodos clásicos como promedios de ventas de productos hasta métodos complejos utilizados en minería de datos.

El proyecto comenzó con un análisis exploratorio exhaustivo de los datos disponibles. Esta etapa inicial fue crucial para entender la naturaleza y las características de los datos proporcionados. El análisis exploratorio de datos (EDA) incluyó la identificación de patrones, tendencias y anomalías dentro del conjunto de datos. Se examinaron diferentes aspectos como la distribución de las ventas, la estacionalidad, las tendencias de crecimiento o decrecimiento, y la identificación de posibles outliers. Además, se llevó a cabo una evaluación de la calidad de los datos, abordando cualquier problema relacionado con valores faltantes, datos duplicados o inconsistencias.

A través del EDA, se buscó extraer información relevante que pudiera añadir valor a la base de datos inicial brindada por la multinacional. Se generaron visualizaciones como gráficos de series temporales, diagramas de dispersión e histogramas que facilitaron la comprensión de la dinámica de las ventas de los productos. Esta información preliminar sirvió como fundamento para la construcción de modelos más sofisticados y permitió establecer hipótesis iniciales sobre los factores que podrían influir en las ventas.

### Identificación y Tratamiento de Datos Faltantes

Para asegurar que el modelo LSTM pudiera aprender de manera efectiva, se completaron los periodos faltantes en los datos de ventas con valores cero. Este paso garantizó que todas las series temporales estuvieran completas, incluso en meses sin ventas, proporcionando un conjunto de datos consistente para el entrenamiento del modelo. Luego durante la generación del modelo estos valores a cero fueron transformados en promedios de tn (toneladas) por producto.

### Procesamiento de Datos y Generación del Archivo `sell-z-780-all-LTSM.csv`

Se realizó un procesamiento exhaustivo de los datos de ventas para los 780 productos seleccionados. Este procesamiento incluyó la agregación de datos de ventas mensuales y la imputación de valores faltantes. Los datos se estructuraron de manera uniforme para

garantizar su consistencia y se escalaron adecuadamente para ser utilizados en modelos de redes neuronales LSTM. Finalmente, se generó el archivo `sell-z-780-all-LTSM.csv`, que contiene los datos preprocesados y listos para el entrenamiento del modelo. Este archivo incluye todas las series temporales completas y ajustadas, proporcionando una base sólida para el análisis y la predicción de ventas futuras.

### **Generación de Archivos CSV para PostgreSQL y Visualización en Grafana**

Para facilitar el análisis y la visualización de los datos, se generaron archivos CSV específicos para su carga en una base de datos PostgreSQL. Estos archivos, `sell-in-all-to-postgres.csv` y `tb_productos_descripcion_ac.csv`, contienen las ventas registradas y la descripción de productos, respectivamente. Además, se añadió una columna `en780` para identificar los productos a predecir y otra columna `cant_periodos` para indicar la cantidad de meses que cada producto tuvo ventas. Una vez preparados, estos archivos se cargaron en PostgreSQL, permitiendo su posterior explotación mediante herramientas de visualización como Grafana. Esta integración facilita el monitoreo y análisis interactivo de los datos, proporcionando insights valiosos para la toma de decisiones.

Una vez completada la fase de análisis exploratorio, se procedió a la fase de experimentación con modelos predictivos. Esta etapa involucró tanto métodos clásicos como enfoques avanzados utilizados en el ámbito de la minería de datos y el machine learning. Los métodos clásicos incluyeron modelos estadísticos simples como promedios, que, a pesar de su simplicidad, pueden ofrecer una línea base importante para la comparación con métodos más complejos. Estos modelos básicos ayudaron a establecer expectativas iniciales sobre el comportamiento de las ventas y proporcionaron una referencia contra la cual se pudieron evaluar los modelos más avanzados.

Por otro lado, se exploraron técnicas más sofisticadas como redes neuronales profundas, en particular redes LSTM. Durante el proceso de modelado se realizaron ajustes a los hiperparámetros buscando minimizar el error de predicción en comparación con el método clásico.

Cada experimento fue documentado meticulosamente, con un registro detallado de las configuraciones del modelo, los resultados obtenidos y las lecciones aprendidas. Esta documentación permitió una evaluación continua y una iteración sobre los enfoques utilizados, asegurando que cada paso adicional en el proceso de modelado estuviera basado en un conocimiento sólido y una comprensión clara de los datos y los métodos.

## Procedimientos y recopilación de resultados

Esta sección se estructura de la siguiente manera:

- **Análisis exploratorio de los datos:**
  - Se procesaron los datos con PostgreSQL y Grafana con el objetivo de comprender mejor los datos de negocio. También se realizó un análisis exploratorio de datos (EDA) utilizando la librería ydata-profiling para obtener estadísticas detalladas sobre los datos.
- **Modelos exploratorios:**
  - En esta etapa, se realizaron varias pruebas para determinar la red LSTM a utilizar, probando diferentes agrupaciones por periodo y evaluando atributos complementarios a las toneladas vendidas (tn).
- **Modelo final:**
  - Se propone un modelo final para realizar la predicción objetivo para febrero de 2020.

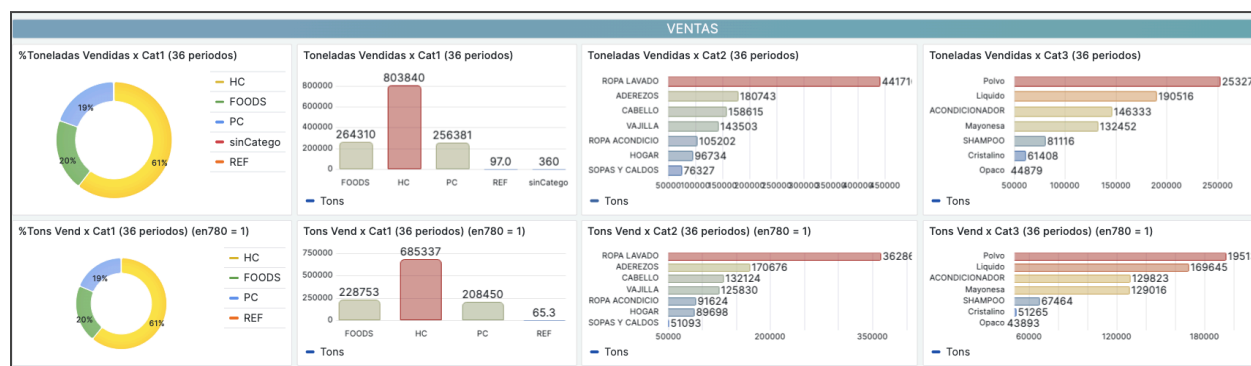
### Análisis exploratorio de los datos

#### Análisis con PostgreSQL y Grafana

Se procesaron los datos utilizando PostgreSQL para gestionar y almacenar la gran cantidad de información de ventas proporcionada por la multinacional. En primer lugar, se generaron archivos CSV específicos, **sell-in-all-to-postgres.csv** y que contenían las ventas registradas y la descripción de los productos, respectivamente. Estos archivos se cargaron en la base de datos PostgreSQL, lo que permitió una estructuración eficiente y una fácil manipulación de los datos. Se llevaron a cabo consultas SQL para extraer datos relevantes y realizar un análisis preliminar, identificando patrones, tendencias y anomalías en las ventas a lo largo del tiempo.

Para la visualización y un análisis más interactivo, se integró PostgreSQL con Grafana. Grafana permitió crear dashboards dinámicos y personalizados, donde se visualizó la evolución de las ventas mensuales, la estacionalidad de los productos y las comparaciones entre diferentes categorías de productos. Las visualizaciones incluyeron gráficos de series temporales, diagramas de dispersión e histogramas, lo que facilitó la comprensión de la dinámica de las ventas. Este enfoque no solo mejoró la capacidad para detectar tendencias y outliers, sino que también proporcionó una plataforma para monitorear en tiempo real las métricas clave de rendimiento, ayudando a tomar decisiones informadas durante el desarrollo del modelo predictivo.

En la siguiente imagen se muestra un dashboard de Grafana con diversas visualizaciones que representan las ventas de toneladas (tn) de diferentes categorías de productos a lo largo de 36 períodos. Se incluyen gráficos de anillos que desglosan el porcentaje de toneladas vendidas por categorías principales (Cat1) y subcategorías (Cat2 y Cat3). Además, se presentan gráficos de barras que muestran las toneladas vendidas por cada subcategoría, permitiendo una comparación clara entre los diferentes tipos de productos. Este dashboard facilita el análisis visual de las ventas, destacando patrones y tendencias importantes para la toma de decisiones.



La siguiente imagen muestra un dashboard de Grafana que presenta varias visualizaciones relacionadas con las toneladas vendidas en diferentes categorías de productos (Cat1) a lo largo de 36 períodos.

Un gráfico de anillo desglosa el porcentaje de toneladas vendidas por cada categoría principal (Cat1), destacando que HC representa el 61% de las ventas, seguido por FOODS (20%), PC (19%), REF y sinCatego.

Un gráfico de barras muestra las toneladas vendidas por cada categoría principal (Cat1) en términos absolutos, indicando que HC tuvo la mayor cantidad de toneladas vendidas (803,840 tn), seguido por FOODS (264,310 tn) y PC (256,381 tn).

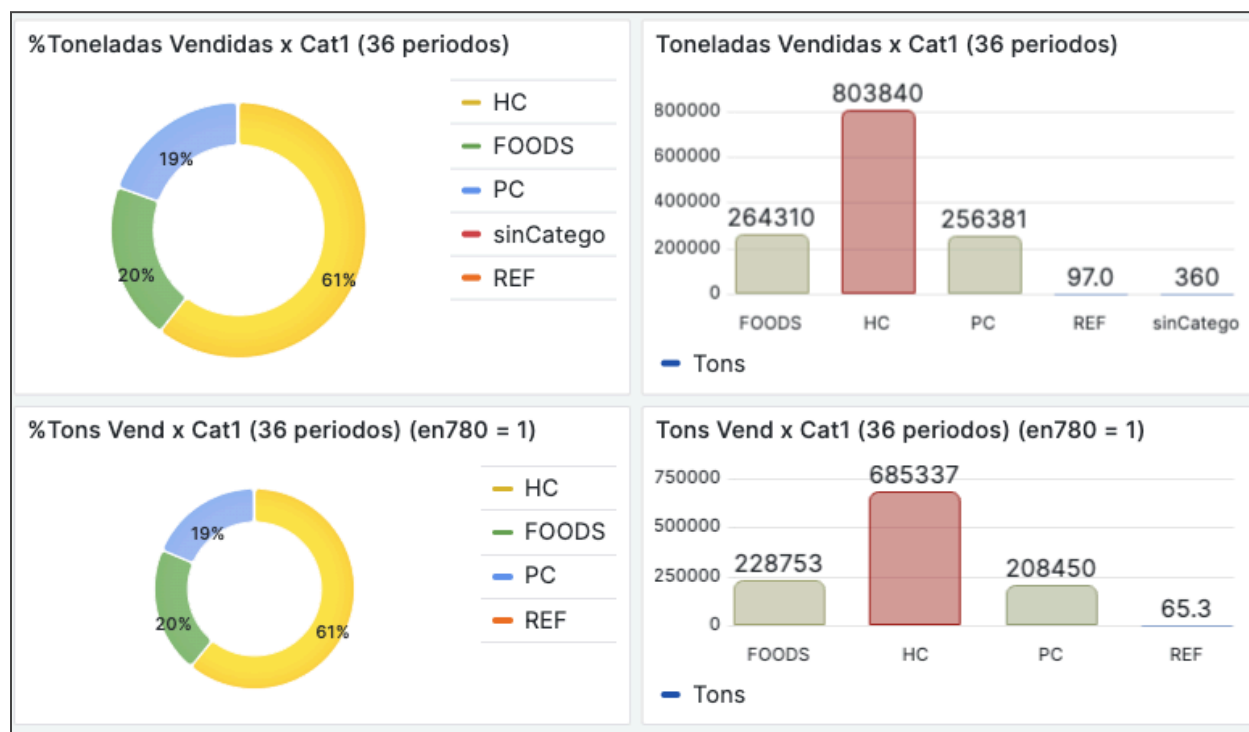
Otro gráfico de anillo muestra el mismo desglose de porcentajes de toneladas vendidas, pero específicamente para los productos marcados con `en780 = 1`, que son aquellos sobre los cuales se debe hacer una predicción para febrero de 2020.

Un gráfico de barras similar al anterior presenta las toneladas vendidas para los productos con `en780 = 1`, destacando nuevamente a HC como la categoría con mayores ventas (685,337 tn), seguido por FOODS (228,753 tn) y PC (208,450 tn).

Estas visualizaciones permiten analizar y comparar las ventas de toneladas entre diferentes categorías de productos y subgrupos de productos, tanto los incluidos en la



predicción ( $\text{en780} = 1$ ) como los que no se incluyen en la predicción ( $\text{en780} = 0$ ), facilitando la identificación de tendencias y patrones de ventas.



La siguiente imagen muestra un dashboard de Grafana que presenta varias visualizaciones relacionadas con las toneladas vendidas en diferentes subcategorías de productos (Cat2 y Cat3) a lo largo de 36 períodos.

Un gráfico de barras muestra las toneladas vendidas por cada subcategoría (Cat2) en términos absolutos. Ropa Lavado tiene la mayor cantidad de toneladas vendidas (441,711 tn), seguida por Aderezos (180,743 tn), Cabello (158,615 tn), y así sucesivamente.

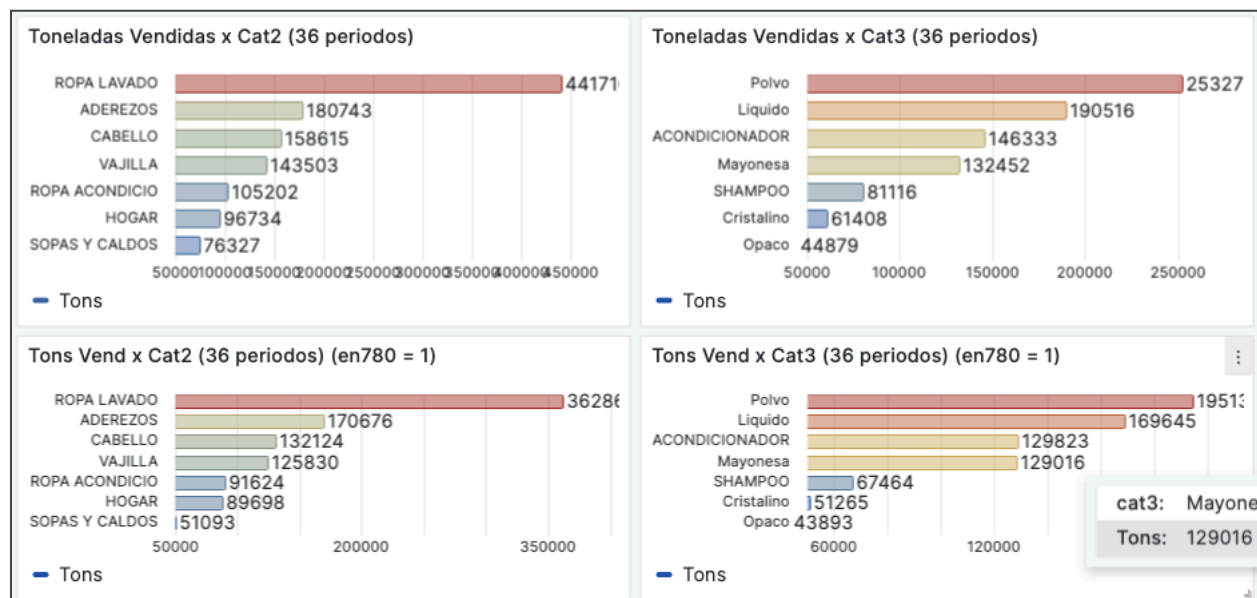
Otro gráfico de barras muestra las toneladas vendidas por subcategoría (Cat3). Polvo tiene la mayor cantidad de toneladas vendidas (253,271 tn), seguido por Líquido (190,516 tn), Acondicionador (146,333 tn), y otros.

Un gráfico similar se presenta para los productos con  $\text{en780} = 1$ , que son aquellos sobre los cuales se debe hacer una predicción para febrero de 2020. En este caso, Ropa Lavado sigue siendo la categoría con mayores ventas (362,869 tn), seguida por Aderezos (170,676 tn) y Cabello (132,124 tn).

Finalmente, otro gráfico de barras muestra las toneladas vendidas para los productos con  $\text{en780} = 1$  en la subcategoría (Cat3). Polvo nuevamente tiene la mayor cantidad de

toneladas vendidas (195,137 tn), seguido por Líquido (169,645 tn), y Acondicionador (129,823 tn).

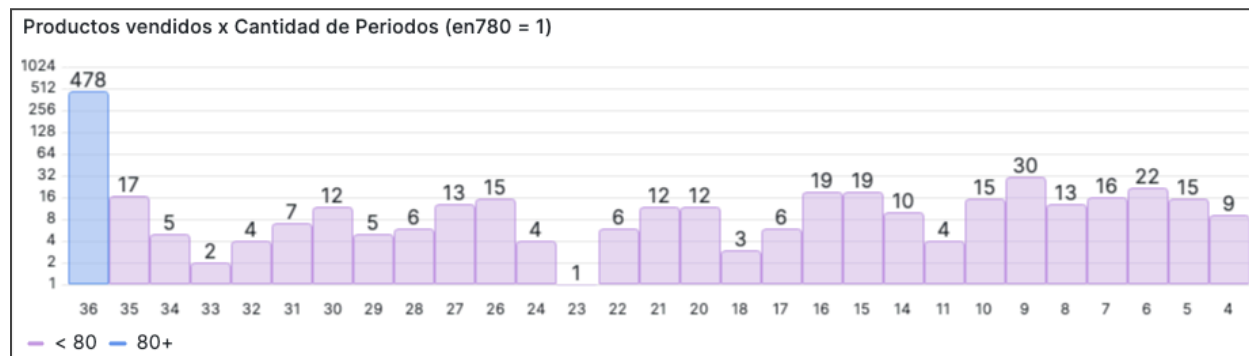
Estas visualizaciones permiten analizar y comparar las ventas de toneladas entre diferentes subcategorías de productos y subgrupos de productos, tanto los incluidos en la predicción (`en780 = 1`) como los que no se incluyen en la predicción (`en780 = 0`), facilitando la identificación de tendencias y patrones de ventas.



La siguiente imagen muestra la distribución de los 780 productos sobre los cuales se debe generar una predicción, en función de la cantidad de períodos (meses) en los que se registraron ventas. Se cuenta con datos de ventas a lo largo de 36 meses (periodos).

El gráfico indica que hay 478 productos que tuvieron ventas en todos los 36 meses, lo cual se refleja en la barra más alta. Otros 17 productos tuvieron ventas en 35 meses, 9 productos tuvieron ventas en solo 4 meses, y así sucesivamente para cada barra. Cada barra representa la cantidad de productos que registraron ventas en un número específico de períodos, con una marcada disminución en la frecuencia de productos a medida que se reduce el número de meses con ventas.

Esta distribución permite visualizar la consistencia y periodicidad de las ventas de cada producto, proporcionando una base sólida para el análisis y la predicción de ventas futuras, específicamente para febrero de 2020.

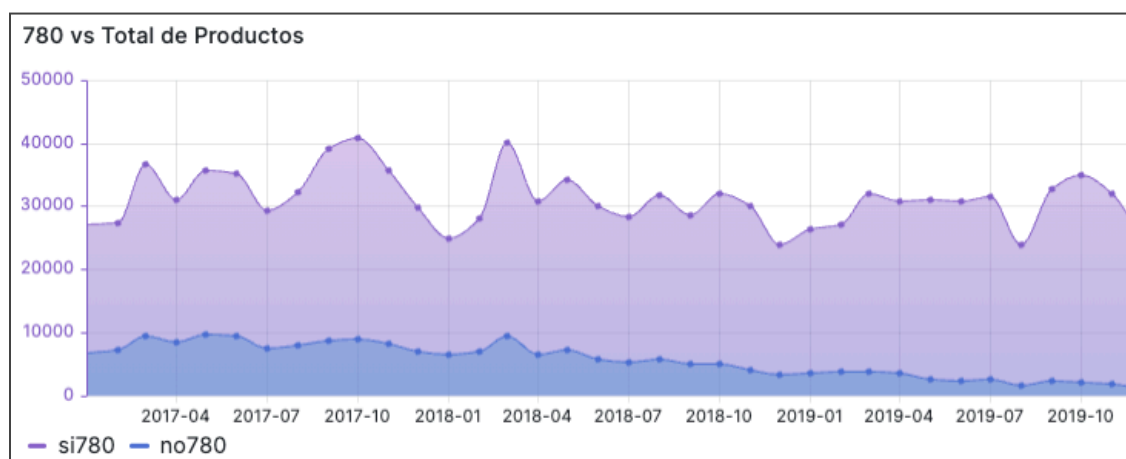


La siguiente imagen muestra la relación de productos y la distribución de toneladas (tn) para los productos que deben ser predichos (`si780`) y los que no deben ser predichos (`no780`).

El gráfico de área apilada ilustra cómo se distribuyen las ventas en toneladas a lo largo del tiempo, desde principios de 2017 hasta finales de 2019. La sección en azul representa los productos `no780`, aquellos que no se incluirán en la predicción. La sección en morado representa los productos `si780`, que son los 780 productos sobre los cuales se debe hacer una predicción para febrero de 2020.

Se observa que los productos `si780` constituyen la mayor parte de las ventas en toneladas a lo largo del período analizado, mostrando fluctuaciones estacionales y variaciones en la cantidad de ventas. Adicionalmente, se puede ver una tendencia decreciente en las ventas de los productos `no780`, que se achican en el tiempo, indicando una disminución en su contribución total a las ventas.

Incorporar esta tendencia puede proporcionar un contexto adicional al análisis, sugiriendo un posible cambio en el enfoque de ventas o en la popularidad de ciertos productos a lo largo del tiempo. Esta información es crucial para entender la evolución del mercado y ajustar las estrategias de predicción y ventas en consecuencia.



El análisis exploratorio de datos (EDA) se realizó utilizando la librería ydata-profiling, que permite generar perfiles exhaustivos de los datos para obtener estadísticas detalladas y visualizar la distribución y las características de los atributos.

En la siguiente imagen se muestra un ejemplo del potencial de la herramienta utilizando en el ejemplo el atributo "descripción" del dataset:

[illegible]

## Modelos exploratorios

### Promedios de toneladas

Se tomaron de la información brindada por la multinacional el promedio de las toneladas vendidas de los 780 productos durante los últimos 6 y 12 meses.

Posteriormente se obtuvieron resultados de predicciones en el mes de Febrero 2020 que se detallan a continuación:


	<b>promedio_6meses.csv</b> Complete · Gaston Larregui_Aus · 2m ago · promedio de los 6 últimos meses de 780 productos 00 - promedio ultimos 6 y 12 meses.ipynb	<b>0.313</b>
	<b>promedio_12meses.csv</b> Complete · Gaston Larregui_Aus · 20s ago · promedio de los 12 últimos meses de 780 productos 00 - promedio ultimos 6 y 12 meses.ipynb	<b>0.273</b>

En función de estos resultados, se obtuvo un valor de 0.273 correspondiente al promedio de los últimos 12 meses de las toneladas vendidas y dicho valor se toma como referencia a mejorar con técnicas más sofisticadas.


- notebook 00 - promedio ultimos 6 y 12 meses.ipynb

### LSTM - 780 Productos 36 meses

Como primer punto de partida, se seleccionaron los 780 productos para los cuales la multinacional desea obtener predicciones. Para cada uno de estos productos, se aplicó un modelo de redes neuronales LSTM durante el período de 36 meses, abarcando desde enero de 2017 hasta diciembre de 2019.

	periodo	36
	customer_id	597
	product_id	780
	plan_precios_cuidados	2
	cust_request_qty	84
	cust_request_tn	92001
	tn	91942
	dtype:	int64

Antes de proceder con la predicción cada uno de los productos fueron transformados como se puede observar en la siguiente imagen, luego de estos fueron escalados y estructurados de manera uniforme para garantizar la consistencia y precisión del modelo LSTM. Este proceso incluyó la transformación de los datos en secuencias de entrada y salida adecuadas, la imputación de datos faltantes por valor cero.



	periodo	20001	20002	20003	20004	20005	20006	20007
0	201701	934.772	550.157	1063.458	555.916	494.270	528.410	464.671
1	201702	798.016	505.886	752.115	508.200	551.431	599.186	638.630
2	201703	1303.358	834.735	917.165	489.913	563.900	868.342	840.833
3	201704	1069.961	522.354	525.826	512.054	662.590	565.319	741.172
4	201705	1502.201	843.438	620.482	543.367	515.587	813.176	858.045

Para este experimentos se utilizó la siguiente notebook adjunta en la documentación.

- 01 -autoscript\_Primer LSTM (1).ipynb

Durante el presente experimento se realizaron varios cambios en los hiperparametros del modelo arrojando los siguiente resultados:

	<b>05_pred_202002.csv</b> Complete · Gaston Larregui_Aus · 1mo ago · Primer LSTM - Batch_size=1 *** epochs=100	<b>0.348</b>
	<b>04_pred_202002.csv</b> Complete · Gaston Larregui_Aus · 1mo ago · Primer LSTM - Batch_size=1 *** epochs=50	<b>0.321</b>
	<b>03_pred_202002.csv</b> Complete · Gaston Larregui_Aus · 1mo ago · Primer LSTM - Batch_size=1 *** epochs=10	<b>0.349</b>
	<b>02_pred_202002.csv</b> Complete · Gaston Larregui_Aus · 1mo ago · Primer LSTM - Batch_size=1 *** epochs=1	<b>0.409</b>
	<b>01_pred_202002.csv</b> Complete · Gaston Larregui_Aus · 1mo ago · Primer LSTM (rusticooo) - con Predicción para todos los productos	<b>0.441</b>

Como se puede observar en ningunos de los casos superó valor de 0.273 correspondiente al promedio de los últimos 12 meses de las toneladas vendidas

### LSTM - 780 Productos - Segregación por Meses

En este experimento, se realizó la segregación de cada producto de acuerdo con la cantidad de meses que tuvo ventas en el pasado y por último se agruparon en los siguientes grupos:

```
[ ] # Paquete 1: product_id con meses entre 36 y 24
paquete_1 = cant_meses.query('24 < meses <= 36').reset_index()
paquete_1['product_id'] = paquete_1['product_id'].astype(int)

# Paquete 2: product_id con meses entre 24 y 12
paquete_2 = cant_meses.query('12 < meses <= 24').reset_index()
paquete_2['product_id'] = paquete_2['product_id'].astype(int)

# Paquete 3: product_id con meses entre 12 y 5
paquete_3 = cant_meses.query('5 <= meses <= 12').reset_index()
paquete_3['product_id'] = paquete_3['product_id'].astype(int)
```

De esta forma quedaron 3 grupos

Grupo 1: Contiene 568 productos que tuvieron ventas durante 24 a 36 meses desde Diciembre de 2019.

```
▶ unique_counts = df_final_1.nunique()
print(unique_counts)
```

periodo	36
customer_id	596
product_id	568
plan_precios_cuidados	2
cust_request_qty	82
cust_request_tn	87462
tn	87402
dtype:	int64

Grupo 2: Contiene 88 productos que tuvieron ventas durante 23 a 12 meses desde Diciembre 2019.

```
▶ unique_counts = df_final_2.nunique()
print(unique_counts)
```

periodo	23
customer_id	542
product_id	88
plan_precios_cuidados	1
cust_request_qty	59
cust_request_tn	10622
tn	10631
dtype:	int64

Grupo 3: Contiene 124 productos que tuvieron ventas durante los últimos 11 meses desde Diciembre 2019.

```

unique_counts = df_final_3.nunique()
print(unique_counts)

periodo          11
customer_id      504
product_id       124
plan_precios_cuidados  2
cust_request_qty  50
cust_request_tn   7176
tn               7176
dtype: int64

```

Para este experimentos se utilizaron 3 notebook adjuntas en la documentación

- notebook 02 - autoscript\_Kaggle\_Paquete\_01\_LSTM\_01 (1).ipynb
- notebook 02 - autoscript\_Kaggle\_Paquete\_02\_LSTM\_01 (1).ipynb
- notebook 02 - autoscript\_Kaggle\_Paquete\_03\_LSTM\_01 (1).ipynb

Cada notebook corresponde a un grupo particular comentado anteriormente, luego fueron ejecutadas en 3 máquinas en googlecloud con el fin de optimizar los tiempos de trabajo.

Folder browser		Buckets > guigui666 > autoscript		
<div> <div>guigui666</div> <div> <div>autoscript/</div> <div> <div>ipynb_checkpoints/</div> <div>lightning_logs/</div> <div>datasets/</div> <div>exp/</div> <div>log/</div> </div> </div> </div>		<div> <div>UPLOAD FILES</div> <div>UPLOAD FOLDER</div> <div>CREATE FOLDER</div> <div>TRANSFER DATA</div> <div>MANAGE HOLDS</div> </div>		
		<div> <div>Filter by name prefix only</div> <div>Filter</div> <div>Filter objects and folders</div> </div>		
		Name		
		Size		Created
		<div> <div>ipynb_checkpoints/</div> <div>2018_REV1autoscript_Kaggle_Paquete_01_LSTM_01.ipynb</div> <div>2018autoscript_Kaggle_Paquete_01_LSTM_01.ipynb</div> <div>2024-06-05-Prophet.ipynb</div> <div>Kaggle_Paquete_01_LSTM_01.ipynb</div> <div>Kaggle_Paquete_02_LSTM_01.ipynb</div> <div>Kaggle_Paquete_03_LSTM_01.ipynb</div> <div>LSTM_02_Kaggle-Copy1.ipynb</div> <div>LSTM_02_Kaggle.ipynb</div> <div>Preparación data-Cambiomodelo-Copy1.ipynb</div> <div>Preparación data-Cambiomodelo-Copy2.ipynb</div> <div>Preparación data-Cambiomodelo-Copy3.ipynb</div> <div>Preparación data-Copy1.ipynb</div> <div>Preparación data.ipynb</div> <div>Primer LSTM.ipynb</div> </div>		
		-		-
		532.5 KB		Jun 18, 2024
		850 KB		Jun 18, 2024
		5.8 KB		Jun 5, 2024
		421 KB		Jun 18, 2024
		266.3 KB		Jun 17, 2024
		279.4 KB		Jun 17, 2024
		155 KB		Jun 16, 2024
		155 KB		Jun 16, 2024
		802.1 KB		Jun 29, 2024
		801.9 KB		Jun 29, 2024
		802 KB		Jun 29, 2024
		803.5 KB		Jun 21, 2024
		804.8 KB		Jun 21, 2024
		1.8 MB		Jun 11, 2024



A continuación se pueden observar los resultados finales del experimento, en donde nuevamente ningunos de los casos superó valor de 0.273 correspondiente al promedio de los últimos 12 meses de las toneladas vendidas

✓	<b>predicciones_07_04.csv</b> Complete · Gaston Larregui_Aus · 18d ago · paquete01_ts9_NU128_B6_E100.txt paquete02_ts3_NU128_B6_E100.txt pack03_ts3_NU128_B3_E100.txt	<b>0.382</b>
✓	<b>predicciones_07_03.csv</b> Complete · Gaston Larregui_Aus · 18d ago · paquete01_ts9_NU128_B6_E100.txt paquete02_ts3_NU128_B6_E100.txt paquete03_ts6_NU128_B6_E100...	<b>0.329</b>
✓	<b>predicciones_06.csv</b> Complete · Gaston Larregui_Aus · 18d ago · 3 paquetes	<b>0.329</b>
✓	<b>predicciones_05.csv</b> Complete · Gaston Larregui_Aus · 19d ago · Negativos a zero (0) time_step = 6 N_UNITS = 128 BATCH = 6 EPOCHS = 100	<b>0.489</b>
✓	<b>predicciones_04.csv</b> Complete · Gaston Larregui_Aus · 19d ago · time_step = 6 N_UNITS = 128 BATCH = 6 EPOCHS = 100	<b>0.375</b>
✓	<b>predicciones_02.csv</b> Complete · Gaston Larregui_Aus · 19d ago · time_step = 3 N_UNITS = 128 BATCH = 6 EPOCHS = 300	<b>0.392</b>

### LSTM - 780 Productos - Segregación por categorías

En este experimento, se realizó la segregación de cada producto de acuerdo con la categoría “CAT1” y “CAT2”, en donde luego se agruparon de la siguientes manera:

#### Segregación por categoría CAT1

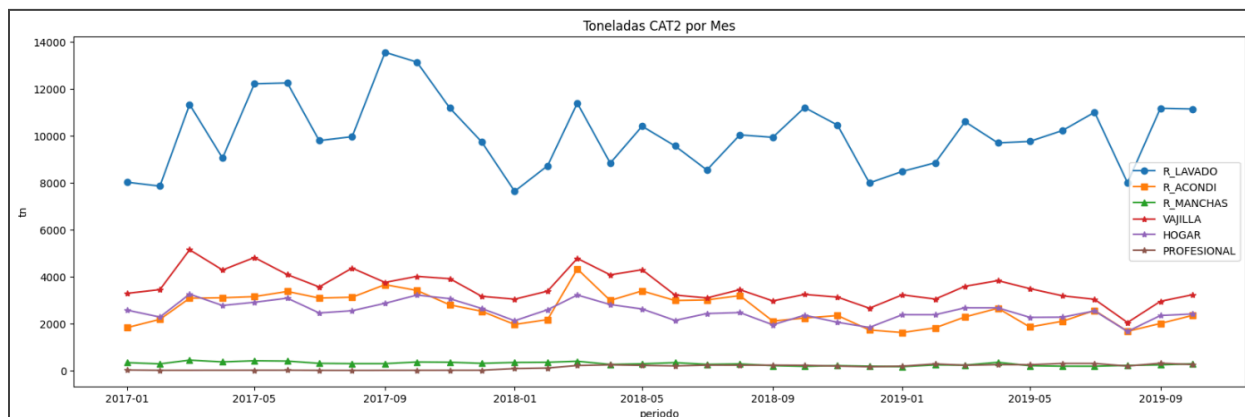
- HC

Se realizó una evaluación de las tendencia a lo largo de los 36 meses de los productos contenidos por la categoría “HC” y en función de estos se agruparon los productos por similitud de toneladas vendidas a lo largo de los 36 meses formando 3 subgrupos de acuerdo a la categoría CAT2

Grupo	Productos
G1_HC	53
G2_HC	117
G3_HC	7

### # GRUPO PARA CATEGORIA 2 DE LOS PRODUCTOS HC

```
G1_HC = HC_R_LAVADO.copy()
G2_HC = pd.concat([HC_R_ACONDI, HC_VAJILLA, HC_HOGAR], ignore_index=True)
G3_HC = pd.concat([HC_PROFESIONAL, HC_R_MANCHAS], ignore_index=True)
```



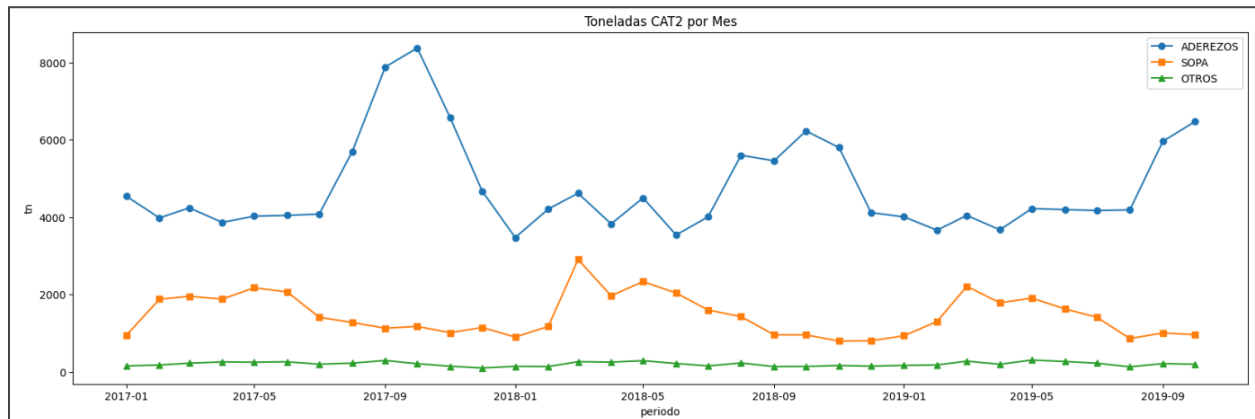
### ● FOODS

Se realizó una evaluación de las tendencias a lo largo de los 36 meses de los productos contenidos por la categoría “FOODS” y en función de estos se agruparon los productos por similitud de toneladas vendidas a lo largo de los 36 meses formando 3 subgrupos de acuerdo a la categoría CAT2

Grupo	Productos
G1_FOODS	49
G2_FOODS	85
G3_FOODS	9

### # GRUPO PARA CATEGORIA 2 DE LOS PRODUCTOS FOODS

```
G1_FOODS = FOODS_ADEREZOS.copy()
G2_FOODS = FOODS_SOPA.copy()
G3_FOODS = FOODS_OTROS.copy()
```



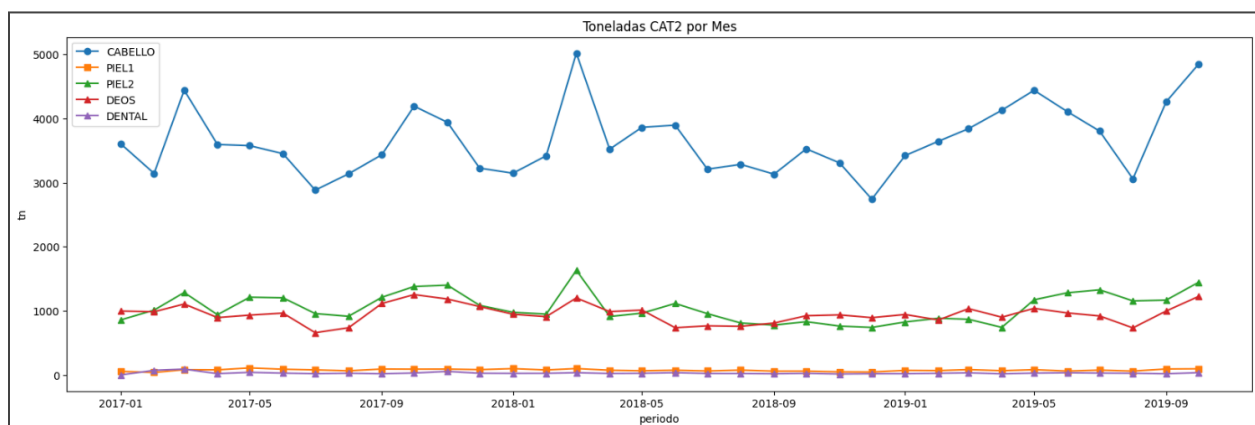
- PC

Se realizó una evaluación de las tendencias a lo largo de los 36 meses de los productos contenidos por la categoría “PC” y en función de estos se agruparon los productos por similitud de toneladas vendidas a lo largo de los 36 meses formando 3 subgrupos de acuerdo a la categoría CAT2.

Grupo	Productos
G1_PC	203
G2_PC	104
G3_PC	137

#### # GRUPO PARA CATEGORIA 2 DE LOS PRODUCTOS PC

```
G1_PC = PC_CABELLO.copy()
G2_PC = pd.concat([PC_PIEL1, PC_PIEL2], ignore_index=True)
G3_PC = pd.concat([PC_DEOS, PC_DENTAL], ignore_index=True)
```



- REF

En esta categoría se tomaron todos los productos de la misma en un único grupo.

Grupo	Productos
G1_REF	6

```
# GRUPO PARA CATEGORIA 2 DE LOS PRODUCTOS REF
G1_REF = prod_REF.copy()
```

Para este experimento se utilizó la notebook adjuntas en la documentación, el cual fue ejecutado en 3 máquinas en googlecloud con el fin de optimizar los tiempos de trabajo, donde en cada una se realizó variación de los hiperparámetros del modelo.

- notebook Preparación data-Cambiomodelo-Copy1.ipynb
- notebook Preparación data-Cambiomodelo-Copy2.ipynb
- notebook Preparación data-Cambiomodelo-Copy3.ipynb

A continuación se pueden observar los resultados finales del experimento, en donde nuevamente ninguno de los casos superó valor de 0.273 correspondiente al promedio de los últimos 12 meses de las toneladas vendidas

✓	nuevo_pred_sl3_actlineal_NU50_B12_E50.csv Complete · Gaston Larregui_Aus · 14d ago	0.438
✓	pred_nuevomodelo_sl3_actline_NU128_B6_E200.csv Complete · Gaston Larregui_Aus · 15d ago · Modificado, 0.4	0.399
✓	nuevo_pred_sl12_actIRELU_NU128_B6_E50.csv Complete · Gaston Larregui_Aus · 15d ago	0.400
✓	nuevo_pred_sl12_actIRELU_NU128_B6_E50.csv Complete · Gaston Larregui_Aus · 15d ago	0.553
✓	nuevo_pred_sl12_actlineal_NU128_B6_E50.csv Complete · Gaston Larregui_Aus · 15d ago	0.436
✓	pred_nuevomodelo_sl6_actline_NU128_B6_E50.csv Complete · Gaston Larregui_Aus · 15d ago	0.470

### **Análisis de los resultados**

De acuerdo a los experimentos realizados, utilizando diferentes metodologías y agrupaciones de datos, no se logró mejorar el valor de referencia de 0.273 en las predicciones de toneladas vendidas. Esto sugiere que las técnicas de LSTM aplicadas, a pesar de la variación en la preparación y segmentación de datos, no fueron suficientemente eficaces para superar el promedio de los últimos 12 meses. Este resultado refleja una limitación en la capacidad de los modelos utilizados para capturar los patrones necesarios para una predicción más precisa, lo que implica la necesidad de explorar enfoques alternativos.

Teniendo en cuenta que los experimentos mencionados anteriormente se realizaron con un escalamiento de datos únicamente entre 0 y 1, se ha decidido implementar otros tipos de escalamiento en los datos. Este cambio en la metodología busca mejorar la eficacia del modelo al permitirle aprender patrones de manera más efectiva. Por ello, se diseñaron y ejecutaron nuevos experimentos, denominados Experimentos 5 y 6, que incluyen estas variaciones en el preprocesamiento de datos.

Los resultados de estos últimos experimentos fueron alentadores. En particular, se obtuvo un valor de error en la predicción para el mes de febrero de 2020 de 0.244, el cual es inferior al error del promedio de los últimos 12 meses. Este logro indica una mejora significativa en la precisión del modelo, validando la hipótesis de que un escalamiento diferente puede tener un impacto positivo en el rendimiento de las predicciones.

Dado que el valor de error obtenido en estos experimentos es menor que el error previo de 0.273, se ha decidido tomar como valor final de referencia el 0.244. Este nuevo valor no solo representa una mejora cuantitativa, sino que también respalda la eficacia de las modificaciones implementadas en el proceso de escalamiento y preprocesamiento de los datos.

## Modelo final

En esta etapa, se integraron los hallazgos del análisis exploratorio de datos y las pruebas realizadas con diferentes modelos exploratorios para construir un modelo predictivo robusto. Se seleccionó una red LSTM (Long Short-Term Memory) como el enfoque principal debido a su capacidad para capturar patrones complejos y dependencias temporales en los datos. El modelo final se presenta dentro de la documentación como `012_modelo_LSTM_v8.4.ipynb` y `012_modelo_LSTM_v8.4.py`. El archivo `.py` es una conversión a código Python realizada con ``jupyter nbconvert -to script``, lo que significa que contiene el mismo código para ser ejecutado fuera de Jupyter Notebook.

## Importaciones y Preparativos

Se importan bibliotecas esenciales como `numpy`, `pandas`, `matplotlib`, `sklearn` y `tensorflow`, indicando que el script maneja operaciones numéricas, manipulación de datos, visualizaciones, aprendizaje automático y modelos de deep learning.

## Preparación de Datos

Los datos se procesan para crear secuencias que alimentarán el modelo LSTM, incluyendo normalización y transformaciones logarítmicas, dependiendo de los parámetros seleccionados.

## Construcción del Modelo LSTM

El modelo LSTM se define con varias capas, incluyendo capas LSTM, capas Dense y una capa de salida. Las configuraciones de estas capas se determinaron en función de variables como el número de neuronas, las funciones de activación y otros hiperparámetros. Se utilizaron técnicas como dropout para regularizar el modelo y prevenir el sobreajuste.

## Compilación y Entrenamiento

El modelo se compila con la función de pérdida `mean_squared_error`, el optimizador Adam y métricas como MAE y RMSE. El entrenamiento incluye el uso de un número definido de épocas, un tamaño de lote y callbacks para la parada temprana y reducción de la tasa de aprendizaje en plataformas.

## Evaluación del Modelo

El script permite evaluar el modelo utilizando un conjunto de datos de validación, calculando métricas estándar como MAE, RMSE, MAPE y TFE, que es la métrica propuesta por La Multinacional. Las predicciones también se generan para un conjunto de datos de prueba, y el script maneja la inversión de cualquier normalización o transformación aplicada a los datos.

## Modelo: Función Principal

Una función central, llamada `master_of_the_universe()`, coordina la ejecución del script, manejando la preparación de los datos, la construcción del modelo, el entrenamiento, la evaluación y la generación de predicciones. La función devuelve las predicciones, los pesos del modelo y las métricas calculadas.

## Configuración y Ejecución de Combinaciones de Parámetros para el Modelo LSTM

Este bloque de código define una serie de parámetros y combina sus valores para entrenar y evaluar un modelo LSTM para la predicción de ventas, controlando si se incluyen predicciones futuras o no.

## Control de Predicciones Futuras

La variable `future_prediction` controla si se incluyen las predicciones para febrero de 2020 (``True``) o si el entrenamiento se limita hasta octubre de 2019 (``False``). Si `future_prediction` está activo (``True``), se realizan predicciones para febrero de 2020. Si no está activo (``False``), se utilizan los datos de 34 períodos (hasta octubre de 2019), y los datos de noviembre y diciembre de 2019 se usan para comparar las predicciones con los valores reales.

## Parámetros Definidos

**SEEDs:** Lista de semillas para reproducibilidad.  
**seq\_lengths:** Lista de longitudes de secuencia para el modelo LSTM.  
**epochs\_list:** Lista de cantidades de épocas para entrenar el modelo.  
**batch\_sizes:** Lista de tamaños de lote.  
**learning\_rates:** Lista de tasas de aprendizaje.  
**patience\_list:** Lista de valores de paciencia para el early stopping.  
**verbose\_list:** Lista de valores para el nivel de verbosidad del entrenamiento.  
**standard\_scalerS:** Lista de valores para indicar si se utiliza ``StandardScaler`` (1) o no (0).

## Cálculo del Número Total de Combinaciones

El código calcula el número total de combinaciones posibles de parámetros multiplicando las longitudes de las listas definidas para cada parámetro y se imprime por pantalla para control.

## Iteración sobre las Combinaciones de Parámetros

- Fijación de la Semilla: Para reproducibilidad, se fija la semilla para ``numpy``, ``tensorflow`` y ``random`` en cada iteración.
- Carga y Preprocesamiento de Datos: Se cargan los datos de ventas desde un archivo CSV y se convierte la columna ``periodo`` a tipo ``datetime``.
- Filtrado de Datos: Si ``future_prediction`` es ``False``, se filtran los datos para incluir solo hasta octubre de 2019.

- Definición de Parámetros: Se define un diccionario `parameters` con los valores actuales de los parámetros de la iteración.
- Registro de Tiempos: Se registra el tiempo de inicio y finalización de cada iteración para calcular el tiempo transcurrido.
- Impresión de Parámetros y Tiempo de Ejecución: Se imprimen los parámetros utilizados y el tiempo de ejecución de la iteración.

### Llamada a la Función Principal

Se llama a la función `master_of_the_universe()` con los parámetros definidos para entrenar y evaluar el modelo LSTM, y guardar los resultados y métricas.

Es importante que la llamada a la función principal permite experimentar con diferentes combinaciones de parámetros. Estos valores son sometidos a una estructura de repetición o loop (ciclo For) para realizar distintos entrenamientos y predicciones.

```
# Definir la variable para controlar si se incluyen las predicciones futuras o no
future_prediction = True # True para predecir febrero 2020, False para entrenar solo hasta octubre 2019

# Fijar la semilla para reproducibilidad
SEEDs = [52,12,11]
seq_lengths = [3,6,9,12]
epochs_list = [50,20,100,150]
batch_sizes = [32,16,64]
learning_rates = [0.0001,0.001]
patience_list = [10,20]
verbose_list = [0]
standard_scalerS = [0] # [0, 1]
```

```
for seed in SEEDs:
    # Fijar la semilla para reproducibilidad
    np.random.seed(seed)
    tf.random.set_seed(seed)
    random.seed(seed)
    for standard_scaler in standard_scalerS:
        for epochs in epochs_list:
            for batch_size in batch_sizes:
                for learning_rate in learning_rates:
                    for patience in patience_list:
                        for verbose in verbose_list:
                            for seq_length in seq_lengths:
                                # Cargar los datos
                                ventas_LSTM_path = './66_Datos/sell-z-780-all-LSTM.csv'
                                df_ventas = pd.read_csv(ventas_LSTM_path)

                                # Convertir la columna 'periodo' a tipo datetime
                                df_ventas['periodo'] = pd.to_datetime(df_ventas['periodo'], format='%Y-%m-%d')

                                # Filtrar los datos si future_prediction es False hasta octubre 2019
                                if not future_prediction:
                                    df_ventas = df_ventas[df_ventas['periodo'] <= '2019-10-01']

                                # Definir los parámetros utilizados
                                parameters = {
                                    "seq_length": seq_length,
                                    "epochs": epochs,
                                    "batch_size": batch_size,
                                    "learning_rate": learning_rate,
                                    "patience": patience,
                                    "verbose": verbose,
                                    "seed": seed,
                                    "standard_scaler": standard_scaler,
                                    "future_prediction": future_prediction
                                }
```



```
# Registrar el tiempo de finalización al final del script
fin, fin_str = registrar_tiempo()
# Calcular el tiempo transcurrido
tiempo_transcurrido = calcular_tiempo_transcurrido(inicio, fin)

print('\n')
print('# -----')
print('# Tiempo de Ejecucion:')
print(f"#    Tiempo de inicio: {inicio_str}")
print(f"#    Tiempo de actual: {fin_str}")
print('# ')
print(f"#    Tiempo transcurrido: {tiempo_transcurrido} segundos")
print('# ')
print('# LSTM Parameters:')
print(f'#    seed={seed}, seq_length={seq_length}, epochs={epochs}, batch_size={batch_size}, learn
print('# -----')
print('#\n')

# Llamar a la función principal con los parámetros
df_ventas2, weights_dict = master_of_the_universe()
```

## Recopilación de resultados

En el código se implementaron dos aspectos clave para la recopilación de resultados:

### 1. Asignación de Nombres a los Archivos

- Los nombres de los archivos de predicciones y de hiperparámetros con métricas se asignan de forma sistemática mediante una función específica, utilizando el formato `YYYYMMDD-HHMM` seguido de un sufijo descriptivo. Por ejemplo, nombres de archivos generados incluyen
  - 20240712-2017-LSTM\_v8v4\_feb\_tfe2\_0.3616.csv
  - 20240712-2017-LSTM\_v8v4\_feb\_tfe2\_0.3616.csv.metrics.json
  - 20240712-2213-LSTM\_v8v4\_feb\_tfe2\_0.2795.csv
  - 20240712-2213-LSTM\_v8v4\_feb\_tfe2\_0.2795.csv.metrics.json

Esta convención facilita la identificación y el análisis de los resultados asociados a cada conjunto específico de parámetros.

### 2. Generación de Archivos

- Por cada predicción, se generan dos archivos principales:
  - Un archivo con extensión `.csv`, que contiene las predicciones generadas por el modelo LSTM para cada combinación de hiperparámetros.
  - Un archivo con extensión `.metrics.json`, que registra los valores de los hiperparámetros utilizados y las métricas de rendimiento resultantes, como MAE, RMSE, MAPE y TFE.

### 3. Recopilación de Datos

- Mediante el script `013_metrics_eval.ipynb`, se levantan todos los archivos `.metrics.json` y se genera un archivo `013_metrics_summary.xlsx` con los resultados de todas las predicciones. Este archivo consolidado permite una revisión y análisis exhaustivo de los resultados de todas las combinaciones de

parámetros, facilitando la evaluación del rendimiento del modelo en diferentes configuraciones.

En las siguientes tablas se muestra un ejemplo con los resultados de `013_metrics_summary.xlsx`

file_name	future_pre diccion	seed	seq_le ngth	epochs	batch size	learning_r ate	patience	verbose	standard_sca l	validation_av g_mae	validation_av g_rmse	validation_av g_mape	validation_av g_tfe	future_predic tions_mae_n ov_dec_2019	future_predic tions_rmse_n ov_dec_2019	future_predic tions_mape_n ov_dec_2019	future_predic tions_tfe_nov _dec_2019
20240713-1936-LSTM_v8v4_feb_tfe2_0.2622.csv.metrics.json	FALSO	52	6	100	16	0,0001	10	0	0	3,226966215	7,737307414	113,6092885	0,262218459	3,226966215	8,800021961	113,6092885	0,234806995
20240713-1820-LSTM_v8v4_feb_tfe2_0.2629.csv.metrics.json	FALSO	52	3	50	16	0,0001	10	0	0	3,295202325	7,827089361	101,4891503	0,262941326	3,295202325	8,825514617	101,4891503	0,23758271
20240712-2257-LSTM_v8v4_feb_tfe2_0.2659.csv.metrics.json	FALSO	52	3	50	32	0,0001	10	0	0	3,317300455	7,878135131	113,0663669	0,265857493	3,317300455	8,878550316	113,0663669	0,238597355
20240713-1825-LSTM_v8v4_feb_tfe2_0.2683.csv.metrics.json	FALSO	52	6	50	16	0,0001	10	0	0	3,309560004	8,253737901	116,3859735	0,268287635	3,309560004	9,411056817	116,3859735	0,240593444
20240713-1836-LSTM_v8v4_feb_tfe2_0.2698.csv.metrics.json	FALSO	52	3	20	32	0,0001	10	0	0	3,354298202	7,912903479	115,3743246	0,269835668	3,354298202	8,921426074	115,3743246	0,243240266
20240713-1930-LSTM_v8v4_feb_tfe2_0.2689.csv.metrics.json	FALSO	52	3	100	16	0,0001	10	0	0	3,375165263	7,856539303	113,7963127	0,268894409	3,375165263	8,919617036	113,7963127	0,243503018
20240712-2317-LSTM_v8v4_feb_tfe2_0.2691.csv.metrics.json	FALSO	52	3	20	32	0,0001	10	0	0	3,367390218	7,786031571	142,2717167	0,269082564	3,367390218	8,758124915	142,2717167	0,243562081
20240712-2240-LSTM_v8v4_feb_tfe2_0.2723.csv.metrics.json	FALSO	52	6	100	32	0,0001	10	0	0	3,351406221	8,133114658	111,7826048	0,272315849	3,351406221	9,231542871	111,7826048	0,244245382
20240713-1907-LSTM_v8v4_feb_tfe2_0.2683.csv.metrics.json	FALSO	52	3	100	32	0,0001	10	0	0	3,390810588	8,128915625	117,3711139	0,268348551	3,390810588	9,177955698	117,3711139	0,244357691
20240713-1352-LSTM_v8v4_feb_tfe2_0.268.csv.metrics.json	FALSO	14	3	50	32	0,0001	10	0	0	3,391134392	7,88950877	118,3332776	0,268043988	3,391134392	8,891878644	118,3332776	0,244493068
20240713-1759-LSTM_v8v4_feb_tfe2_0.2734.csv.metrics.json	FALSO	52	6	50	32	0,0001	10	0	0	3,359721686	8,065057302	120,6058051	0,273446497	3,359721686	9,103013336	120,6058051	0,245009103
20240712-2251-LSTM_v8v4_feb_tfe2_0.274.csv.metrics.json	FALSO	52	6	100	64	0,0001	10	0	0	3,383252107	8,048728921	116,9423632	0,273978782	3,383252107	9,080909549	116,9423632	0,246385168
20240713-1945-LSTM_v8v4_feb_tfe2_0.2736.csv.metrics.json	FALSO	52	12	100	16	0,0001	10	0	0	3,367730422	8,408286242	95,12147234	0,273602588	3,367730422	9,768876677	95,12147234	0,247482209
20240712-2307-LSTM_v8v4_feb_tfe2_0.2761.csv.metrics.json	FALSO	52	12	50	32	0,0001	10	0	0	3,376822883	8,52745606	101,4157324	0,276142301	3,376822883	9,841164196	101,4157324	0,248190589
20240713-1834-LSTM_v8v4_feb_tfe2_0.2735.csv.metrics.json	FALSO	52	12	50	16	0,0001	10	0	0	3,366110544	8,09807949	94,0173986	0,273472575	3,366110544	9,407949365	94,0173986	0,248229958
20240713-1857-LSTM_v8v4_feb_tfe2_0.2756.csv.metrics.json	FALSO	52	6	20	16	0,0001	10	0	0	3,423415704	8,116598329	120,8217	0,275603882	3,423415704	9,281866669	120,8217	0,248978598
20240712-2229-LSTM_v8v4_feb_tfe2_0.2766.csv.metrics.json	FALSO	52	6	150	64	0,0001	10	0	0	3,428465606	8,130451374	116,803407	0,276604861	3,428465606	9,168790852	116,803407	0,24899775
20240712-2235-LSTM_v8v4_feb_tfe2_0.2745.csv.metrics.json	FALSO	52	3	100	32	0,0001	10	0	0	3,473229685	8,404923641	111,051945	0,274498794	3,473229685	9,547825382	111,051945	0,250562323
20240713-1910-LSTM_v8v4_feb_tfe2_0.2781.csv.metrics.json	FALSO	52	6	100	32	0,0001	10	0	0	3,431186111	8,21283242	115,7039792	0,278123544	3,431186111	9,282261525	115,7039792	0,250768347
20240713-1403-LSTM_v8v4_feb_tfe2_0.2721.csv.metrics.json	FALSO	14	3	20	32	0,0001	10	0	0	3,46430595	8,138912045	112,5947949	0,272149426	3,46430595	9,212143174	112,5947949	0,250916616
20240712-2217-LSTM_v8v4_feb_tfe2_0.2745.csv.metrics.json	FALSO	52	6	150	32	0,0001	10	0	0	3,436329183	8,051677022	113,8478774	0,274465845	3,436329183	9,199100352	113,8478774	0,250997628
20240713-1918-LSTM_v8v4_feb_tfe2_0.2734.csv.metrics.json	FALSO	52	3	100	64	0,0001	10	0	0	3,491190665	8,094791819	117,0715818	0,273445578	3,491190665	9,138601911	117,0715818	0,2519991
20240713-1805-LSTM_v8v4_feb_tfe2_0.2746.csv.metrics.json	FALSO	52	12	50	32	0,0001	10	0	0	3,424978877	8,419117509	98,0947495	0,274576182	3,424978877	9,742223889	98,0947495	0,252482587
20240713-1903-LSTM_v8v4_feb_tfe2_0.2743.csv.metrics.json	FALSO	52	12	20	16	0,0001	10	0	0	3,42142175	8,239565418	102,2266898	0,274345491	3,42142175	9,551363071	102,2266898	0,252556868
20240713-1756-LSTM_v8v4_feb_tfe2_0.2749.csv.metrics.json	FALSO	52	3	50	32	0,0001	10	0	0	3,510485398	8,421436293	111,9442381	0,274870456	3,510485398	9,525934873	111,9442381	0,253073771
20240713-1843-LSTM_v8v4_feb_tfe2_0.2813.csv.metrics.json	FALSO	52	12	20	32	0,0001	10	0	0	3,469257016	8,757760383	97,95861737	0,281258481	3,469257016	10,13734979	97,95861737	0,254984336
20240713-1810-LSTM_v8v4_feb_tfe2_0.2845.csv.metrics.json	FALSO	52	6	50	64	0,0001	10	0	0	3,521152899	8,370970852	114,0259906	0,284531966	3,521152899	9,453942735	114,0259906	0,256859396
20240712-2223-LSTM_v8v4_feb_tfe2_0.2793.csv.metrics.json	FALSO	52	12	150	32	0,0001	10	0	0	3,49116545	9,121925256	101,2204032	0,279318472	3,49116545	10,67711068	101,2204032	0,256834954
20240713-1923-LSTM_v8v4_feb_tfe2_0.2795.csv.metrics.json	FALSO	52	3	150	32	0,0001	10	0	0	3,571090154	8,803146661	111,9623554	0,279528326	3,571090154	10,04818951	111,9623554	0,257089326
20240713-1923-LSTM_v8v4_feb_tfe2_0.2811.csv.metrics.json	FALSO	52	6	100	64	0,0001	10	0	0	3,516299211	8,185154442	114,7311617	0,281066209	3,516299211	9,25189623	114,7311617	0,25716651
20240713-1356-LSTM_v8v4_feb_tfe2_0.2846.csv.metrics.json	FALSO	14	6	50	32	0,0001	10	0	0	3,551957815	8,4611519	123,4249584	0,28462287	3,551957815	9,613258057	123,4249584	0,258159776

	file_name	future_pre diccion	seed	seq_le ngth	epochs	batch size	learning_r ate	patience	verbose	standard_sca l
1										
2	20240713-1936-LSTM_v8v4_feb_tfe2_0.2622.csv.metrics.json	FALSO	52	6	100	16	0,0001	10	0	0
3	20240713-1820-LSTM_v8v4_feb_tfe2_0.2629.csv.metrics.json	FALSO	52	3	50	16	0,0001	10	0	0
4	20240712-2257-LSTM_v8v4_feb_tfe2_0.2659.csv.metrics.json	FALSO	52	3	50	32	0,0001	10	0	0
5	20240713-1825-LSTM_v8v4_feb_tfe2_0.2683.csv.metrics.json	FALSO	52	6	50	16	0,0001	10	0	0
6	20240713-1836-LSTM_v8v4_feb_tfe2_0.2698.csv.metrics.json	FALSO	52	3	20	32	0,0001	10	0	0
7	20240713-1930-LSTM_v8v4_feb_tfe2_0.2689.csv.metrics.json	FALSO	52	3	100	16	0,0001	10	0	0
8	20240712-2317-LSTM_v8v4_feb_tfe2_0.2691.csv.metrics.json	FALSO	52	3	20	32	0,0001	10	0	0
9	20240712-2240-LSTM_v8v4_feb_tfe2_0.2723.csv.metrics.json	FALSO	52	6	100	32	0,0001	10	0	0
10	20240713-1907-LSTM_v8v4_feb_tfe2_0.2683.csv.metrics.json	FALSO	52	3	100	32	0,0001	10	0	0

validation_av g_mae	validation_av g_rmse	validation_av g_mape	validation_av g_tfe	future_predic tions_mae_n ov_dec_2019	future_predic tions_rmse_n ov_dec_2019	future_predic tions_mape_n ov_dec_2019	future_predic tions_tfe_nov _dec_2019
3,226966215	7,737307414	113,6092885	0,262218459	3,226966215	8,800021961	113,6092885	0,234806995
3,295202325	7,827089361	101,4891503	0,262941326	3,295202325	8,825514617	101,4891503	0,23758271
3,317300455	7,878135131	113,0663669	0,265857493	3,317300455	8,878550316	113,0663669	0,238597355
3,309560004	8,253737901	116,3859735	0,268287635	3,309560004	9,411056817	116,3859735	0,240593444
3,354298202	7,912903479	115,3743246	0,269835668	3,354298202	8,921426074	115,3743246	0,243240266
3,375165263	7,856539303	113,7963127	0,268894409	3,375165263	8,919617036	113,7963127	0,243503018
3,367390218	7,786031571	142,2717167	0,269082564	3,367390218	8,758124915	142,2717167	0,243562081
3,351406221	8,133114658	111,7826048	0,272315849	3,351406221	9,231542871	111,7826048	0,244245382
3,390810588	8,128915625	117,3711139	0,268348551	3,390810588	9,177955698	117,3711139	0,244357691

## Análisis de los resultados

En esta sección se presentan los resultados obtenidos al ejecutar diferentes configuraciones del modelo LSTM para la predicción de ventas. El análisis se enfoca exclusivamente en los modelos configurados sin predicción futura (`future_prediction = False`) y se centra en la métrica propuesta por el curso, el TFE (**Total Forecast Error**). A continuación, se detallan las métricas de TFE de validación y predicción para los distintos modelos evaluados, destacando el modelo con los mejores resultados obtenidos.

### 1. Modelo con Seq\_length: 3, Epochs: 150, Batch\_size: 1

Este modelo presenta un TFE de validación de 0.3721, indicando un error total de predicción moderado. En cuanto a las predicciones futuras para noviembre y diciembre de 2019, el modelo muestra un TFE de 0.3225, reflejando un rendimiento consistente con las métricas de validación.

### 2. Modelo con Seq\_length: 6, Epochs: 150, Batch\_size: 1

En este caso, el modelo arroja un TFE de validación de 0.4484, sugiriendo una precisión ligeramente inferior en comparación con el modelo anterior. Las predicciones futuras presentan un TFE de 0.3286, manteniendo una consistencia razonable con los resultados de validación.

### 3. Modelo con Seq\_length: 3, Epochs: 150, Batch\_size: 16

Este modelo muestra un TFE de validación de 0.4360, indicando un error total de predicción algo mayor. Para las predicciones futuras, el modelo tiene un TFE de 0.3682, evidenciando una menor precisión en comparación con las métricas de validación.

### 4. Modelo con Seq\_length: 6, Epochs: 150, Batch\_size: 16

Con un TFE de validación de 0.4551, este modelo parece tener un ajuste menos preciso. Las predicciones futuras revelan un TFE de 0.3107, reflejando una mejora en comparación con el TFE de validación.

### 5. Modelo con Seq\_length: 12, Epochs: 150, Batch\_size: 16

Este modelo presenta un TFE de validación de 0.3900, sugiriendo un rendimiento razonablemente preciso. En cuanto a las predicciones futuras, el modelo muestra un TFE de 0.3373, manteniendo una buena consistencia con las métricas de validación.

### 6. Mejor Modelo con Seq\_length: 12, Epochs: 150, Batch\_size: 64

La mejor predicción se obtuvo con el modelo configurado con los siguientes parámetros: `seq_length: 12`, `epochs: 150`, `batch_size: 64`, `learning_rate: 0.0001`, `patience: 10`, `verbose: 0`, `seed: 52`. Este modelo presenta un TFE de validación de 0.2891, indicando un error total de predicción significativamente menor. Las predicciones futuras para

noviembre y diciembre de 2019 muestran un TFE de 0.2705, reflejando una excelente consistencia y precisión con respecto a las métricas de validación.

### Conclusión

En resumen, los modelos evaluados muestran diferentes niveles de precisión en términos de TFE. El modelo con `seq_length: 12`, `epochs: 150`, `batch_size: 64`, `learning_rate: 0.0001`, `patience: 10`, `verbose: 0`, `seed: 52` ha demostrado el mejor rendimiento, con un TFE de validación de 0.2891 y un TFE de predicción de 0.2705. Estos resultados sugieren que este modelo ofrece la mayor precisión y consistencia, haciendo que sea la configuración óptima para la predicción de ventas en este estudio.

Se recomienda continuar ajustando los parámetros y explorando diferentes configuraciones del modelo LSTM para optimizar la métrica TFE y mejorar la precisión de las predicciones.

### Resultados en Kaggle

Nota en el momento de realizar este documento el mejor resultado se obtuvo con los siguientes parámetros:

```
{  
  "seq_length": 12,  
  "epochs": 150,  
  "batch_size": 64,  
  "learning_rate": 0.0001,  
  "patience": 10,  
  "verbose": 0,  
  "seed": 52  
}
```

Los resultados en Kaggle se ven de la siguiente manera:

Labo III, edicion 2024v

Submit Prediction...

Overview

Data

Code

Models

Discussion

Leaderboard

Rules

Team

Submissions

Complete · AurelianoChavarria · 4d ago

✓















20240710-0438-LSTM\_v7\_feb\_tfe2\_0.3.csv


Complete · AurelianoChavarria · 4d ago

0.244

Public
Private

This leaderboard is calculated with approximately 50% of the test data. The final results will be based on the other 50%, so the final standings may be different.

#	Team	Members	Score	Entries	Last	Join
1	Deep Learning 8	   	0.101	873	5h	
2	Deep Learning - Equipo 5	   	0.216	97	1h	
3	Deep Learning 2	  	0.234	235	3d	
4	<b>Deep Learning - Equipo 7</b>	  	0.244	404	4h	



Your Best Entry!  
Your submission scored 0.254, which is not an improvement of your previous score. Keep trying!

## Conclusión Final

Como conclusión final y teniendo en cuenta los resultados obtenidos en los diferentes experimentos, así como la oportunidad de evaluar los experimentos realizados por otros competidores, se ha observado que el valor de error de la predicción para el método del promedio de los últimos 12 meses es levemente superado por métodos de machine learning complejos. Estos métodos, aunque ligeramente más precisos, requieren altas horas de procesamiento y generan costos significativos tanto en términos computacionales como de recursos humanos. Para el caso particular del trabajo en cuestión, no parece rentable embarcarse en un camino tan costoso, considerando que los resultados sólo mejoran ligeramente en comparación con un promedio matemático convencional.

No obstante, sería un error subestimar el valor agregado que los métodos avanzados de machine learning pueden ofrecer. Si bien en este contexto específico los beneficios no justifican los costos, en otros escenarios con diferentes condiciones y requisitos, la aplicación de técnicas más sofisticadas podría resultar altamente recomendable. La capacidad de estos métodos para adaptarse y aprender patrones complejos puede ser crucial en situaciones donde se necesite una mayor precisión y donde los beneficios económicos justifiquen la inversión en recursos.

Por otro lado, la experiencia adquirida al aplicar técnicas avanzadas de machine learning en el conocimiento del negocio ha mostrado una mejoría en el error de predicción. Este avance sugiere que, con más tiempo de análisis y una comprensión más profunda del negocio, se podrían lograr resultados aún más alentadores y competitivos. La clave reside en continuar perfeccionando los modelos y ajustándolos a las especificidades del negocio para maximizar su eficacia.

## Referencias

- [\*Autoregressive Models in Deep Learning — A Brief Survey | George Ho\*](#)
- [\*11.3 Neural network models | Forecasting: Principles and Practice \(2nd ed\) \(otexts.com\)\*](#)
- [\*An Autoregressive Neural Network Approach to Forecasting Bitcoin Price | by MAX Institutional Sales & Marketing | MAX—MaiCoin Asset Exchange | Medium\*](#)
- [\*Understanding LSTM Networks -- colah's blog\*](#)
- [\*https://colab.research.google.com/drive/1Si7z6htXpaKWw91GpxB0x3XoblKN73ip\*](https://colab.research.google.com/drive/1Si7z6htXpaKWw91GpxB0x3XoblKN73ip)