Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Transformando los datos en ganancias: el poder del análisis de datos en las finanzas

Proyecto de la asignatura

Proyecto I, Comprensión de datos

Grado en Ciencia de Datos

Autores: Aguilar Mínguez, Adrià; Alvarruiz López, Andrés; Font Piña, Santiago; Gandía Miñana, Pablo; Gómez Cidad, Marc

Tutor: Francisco Pedroche Sánchez

Curso 2023-24

Resumen

En el mundo de las finanzas, los datos toman un papel importantísimo para comprender el funcionamiento de las empresas y el valor de sus acciones, además de que son factores clave para ayudarnos a tomar decisiones.

En este proyecto intentaremos, a partir de datos fundamentales de empresas, lograr entrenar modelos predictivos que combinados nos permitan conocer cuál será el crecimiento trimestral de las acciones de empresas en base a sus datos más recientes, con el objetivo de ofrecer información muy valiosa que ayude a cualquier usuario interesado en tener su dinero activo (invertir), para tomar decisiones correctas que le generen rentabilidad.

Además de proporcionar toda esta información en un formato amigable, visual e intuitivo, de tal forma que no sea necesario ser un experto en programación o en el mundo de las inversiones para poder utilizar estas ayudas.

Palabras clave: Finanzas, datos fundamentales, rentabilidad.

Resum

Al món de les finances, les dades prenen un paper importantíssim per comprendre el funcionament de les empreses i el valor de les seves accions, a més de ser factors clau per ajudar-nos a prendre decisions.

En aquest projecte intentarem, a partir de dades fonamentals d'empreses, aconseguir entrenar models predictius que combinats ens puguen peremtre conèixer quin serà el creixement trimestral de les accions d'empreses en base a les dades més recents, amb l'objectiu d'oferir informació molt valuosa que ajude a qualsevol usuari interessat en tindre els seus diners actius (invertir), per prendre decisions correctes que li generen rendibilitat.

A més de proporcionar tota aquesta informació en un format amigable, visual i intuïtiu, de manera que no siga necessari ser un expert en programació o en el món de les inversions per poder utilitzar aquestes ajudes.

Paraules clau: Finances, dades fonamentals, rendibilitat.

Abstract

In the world of finance, data plays a very important role in understanding the behavior of companies and the value of their actions, in addition to being key factors in helping us make decisions.

In this project we will try, based on fundamental company data, to train predictive models that combined allow us to know what the quarterly growth of company shares will be based on their most recent data, with the aim of offering very valuable information that helps any interested user to have their money active (invest), to make correct decisions that generate profitability.

In addition to providing all this information in a friendly, visual and intuitive format, so that it is not necessary to be an expert in programming or in the world of investments to be able to use these aids.

Keywords: Finances, fundamental company data, profitability

Indice

1. Introducción 5
    1.1 Motivación del trabajo 5
    1.2 Objetivos 5
    1.3 Metodología 5
2. Estado del arte 6
    2.1 Crítica al estado del arte 8
    2.2 Propuesta 7
3. Alcance del proyecto 7
    3.1 Requisitos, restricciones e hipótesis 7
    3.2 Productos entregables 8
    3.3 Límites y criterios de éxito 8
    3.4 Alineación con ODS 9
4. Preparación y comprensión de datos 9
    4.1 Obtención y descripción 9
    4.2 Limpieza 10
        4.2.1 Estudio de nulos 10
        4.2.2 Estudio de atípicos 12
5. Conocimiento extraído 13
    5.1 Modelos 14
6. Resultados 21
7. Despliegue 22
    7.1 Aplicación desarrollada 22
8. Conclusiones 22
9. Bibliografía 23
10. Anexos 23

1. Introducción

1.1 Motivación del trabajo

La sociedad actual, en la que se almacenan millones de datos en fracciones de segundo, ofrece innumerables oportunidades revolucionarias. La correcta gestión y análisis de estos datos son fundamentales para su aplicación en proyectos de diversos sectores, como la salud y la inteligencia artificial.

Nuestro proyecto se centra en los datos del ámbito empresarial y financiero. La principal motivación de nuestro trabajo es aplicar los conocimientos de ciencia de datos adquiridos en clase y de manera autodidacta, con el objetivo de obtener predicciones precisas sobre el crecimiento de las cotizaciones bursátiles de diversas empresas. De esta manera, buscamos generar beneficios económicos invirtiendo en las empresas que nuestro programa identifica como rentables.

1.2 Objetivos

El objetivo general es desarrollar una herramienta económica precisa, presentada a través de una interfaz gráfica sencilla e intuitiva, capaz de evaluar empresas según su potencial de crecimiento financiero. Esta herramienta, alimentada por los últimos datos trimestrales proporcionados por Stock Analysis, realiza predicciones precisas y crea rankings de empresas con mayor potencial de rentabilidad. Ayudando así a los inversores a tomar decisiones estratégicas a la hora de invertir, con la esperanza (siendo optimistas y basándonos en nuestros buenos resultados) de tener un beneficio económico significativo trimestral.

1.3 Metodología y distribución de tareas

Planificación Inicial:

Objetivos del proyecto: Definimos claramente que queríamos lograr con el proyecto de predicción de precios de acciones.

Investigación Preliminar: Investigamos cómo podríamos realizar el proyecto, identificando las metodologías adecuadas y las herramientas necesarias.

Recolección de Datos:

Búsqueda de Base de Datos: Buscamos una base de datos que contuviera una gran cantidad de datos históricos y precios de acciones. Primera fase —> Andrés

Web Scrapping: Generar la base de datos mediante web scrapping sobre la página stock analysis. Primera fase —> Pablo

Preprocesamiento de Datos:

Filtrado y Limpieza: Dejamos la base de datos inicial libre de nulos y atípicos, para proporcionar a los modelos datos de calidad. Además de realizar conversiones sobre las variables necesarias. Segunda fase —> Marc y Andrés

Desarrollo y evaluación de los modelos:

Entrenamiento de los modelos: Entrenamos diferentes modelos predictivos, adecuando las características propias de cada uno, para así tener modelos efectivos que realizarán predicciones. Tercer fase —> Pablo y Adrià

Evaluación: Evaluamos los modelos entrenandolos con los datos reservados para las pruebas, lo que condujo a plantear ciertas estrategias. Tercer fase —> Pablo y Adrià

Combinación de modelos: Con los modelos definitivos creados y las estrategias pertinentes establecidas, tuvimos que plantear cómo deberían funcionar los dos modelos en conjunto. Tercera fase —> Adrià

Implementación y puesta en práctica:

Desarrollo de página web: Creamos una interfaz gráfica para facilitar a cualquier usuario el uso sencillo de nuestros modelos, para que fueran funcionales en la vida real, implementando los diferentes códigos que teníamos. Cuarta fase —> Santiago

Prueba real: Una vez lo teníamos todo preparado pusimos nuestro proyecto a funcionar, enfrentándolo a los últimos datos reales, y viendo que el funcionamiento era más que satisfactorio. Quinta fase —> Santiago

2. Estado del arte

Nuestro proyecto se enfoca en la predicción del valor de las acciones de empresas en bolsa utilizando una combinación de modelos LSTM (Long Short-Term Memory) y Random Forest, aplicando técnicas avanzadas en el campo de la inteligencia artificial y el machine learning. Las redes LSTM permiten capturar las dependencias temporales a largo plazo en los datos históricos de precios, aprovechando su capacidad para manejar series temporales complejas y variables.

Este enfoque ha demostrado ser altamente efectivo en diversos estudios. Por ejemplo, investigaciones han mostrado que los modelos LSTM superan a los métodos tradicionales como ARIMA en la predicción de precios de acciones y tasas de cambio, debido a su capacidad para aprender patrones complejos y no lineales en los datos temporales.

Complementamos este enfoque con el modelo de Random Forest, conocido por su robustez y capacidad para manejar grandes conjuntos de datos con múltiples variables, lo que nos ayuda a capturar relaciones no lineales y mejorar la precisión de nuestras predicciones. Nuestro enfoque integrado asegura predicciones más precisas y confiables, y se complementa con técnicas de optimización de hiperparámetros para garantizar la transparencia e interpretabilidad de los resultados. Este proyecto representa un enfoque de vanguardia en la aplicación de inteligencia artificial para la toma de decisiones en el mercado bursátil.

Estudios y artículos respaldando estas ideas están mencionados en la bibliografía: [1],[2],[3]

2.1 Crítica al estado del arte

Limitaciones:

Los modelos LSTM y Random Forest dependen en gran medida de datos históricos, lo que puede ser problemático ya que las condiciones del mercado cambian constantemente y los patrones no siempre se repiten.

Son computacionalmente intensivos y requieren significativos recursos para entrenar, lo que puede ser una barrera para su implementación práctica.

Riesgo de sobreajuste, donde el modelo aprende demasiado bien los detalles del conjunto de datos de entrenamiento, pero no generalizan ante datos nuevos.

La interpretabilidad sigue siendo un desafío, ya que debido a cómo funcionan las redes neuronales dificultan la justificación de las predicciones ante clientes.

La selección y relevancia de las variables utilizadas son críticas, ya que una mala selección puede llevar a predicciones inexactas.

Estos límites subrayan la necesidad de mejorar continuamente estos modelos para aumentar su precisión y aplicabilidad.

3. Alcance del proyecto

3.1 Requisitos, restricciones e hipótesis

Requisitos:

- Obtener una gran cantidad de datos fundamentales históricos de una enorme cantidad de empresas.

Restricciones:

- Necesidad de recolectar nosotros mismos los datos con web scrapping.
- Gran variabilidad en los datos, y comportamientos extraños en el precio de las acciones, lo que puede conducir a que lo que ocurra en un futuro sea diferente a la actualidad.

Hipótesis:

- Partimos de la hipótesis que los valores financieros de una empresa tienen relación con las fluctuaciones de sus precios en bolsa. Por tanto, podemos predecir el crecimiento en bolsa entrenando redes neuronales y aplicando Machine Learning sobre estos datos.

3.2 Productos entregables

- Programas utilizados para obtener los datos necesarios, métodos predictivos desarrollados, e implementación de los modelos y pruebas.
- Entorno virtual para visualizar el funcionamiento del programa.
- Informe documentado con los contenidos del proyecto y las herramientas utilizadas.
- Vídeo.
- Códigos empleados.

3.3 Límites y criterios de éxito

Límites:

- Con las herramientas con las que contamos es imposible obtener métodos predictivos con una precisión exacta.
- La realidad futura puede diferir en gran medida con los datos históricos, ya que es un mundo muy volátil.

Criterios de éxito:

- Si la combinación de los modelos genera predicciones acertadas, que logren seleccionar empresas que verdaderamente van a crecer, nos daremos por satisfechos.

- Además, si al aplicar un método de inversión con dichas predicciones, obtenemos una rentabilidad superior a la del Nasdaq (el fondo indexado más importante del mundo con una rentabilidad media del 11%), consideraríamos

esto como un éxito, ya que estaríamos superando la rentabilidad de un índice que refleja el crecimiento de la economía.

3.4 Alineación con ODS

¿Qué son los ODS (Objetivos de Desarrollo Sostenible)? Fueron establecidos por las Naciones Unidas como una guía para poder abordar desafíos en todo el mundo y promover un desarrollo sostenible. Dentro de esta situación, es imprescindible hablar de la alineación de nuestro proyecto y nuestras metas con estos objetivos, así que observemos en qué medida hemos podido nosotros conseguir eso:

Primero, el ODS número 8 (trabajo decente y crecimiento económico), ya que desde los inicios del proyecto estamos en la constante búsqueda de fomentar el empleo pleno al ofrecer información que facilita la inversión en empresas con potencial de crecimiento. La herramienta que hemos desarrollado está basada en el análisis de datos financieros y la predicción, promoviendo la innovación en el campo de las finanzas y estando esto relacionado con el ODS número 9 (innovación). El ODS número 17 tiene como foco las alianzas para lograr los objetivos y, en este caso, al querer ofrecer una herramienta que ayude en el tema de las inversiones responsables y de manera estratégica, conseguimos colaborar con otros sectores en la promoción de objetivos comunes de desarrollo sostenible.

Por tanto, hemos podido ver que, aunque nuestra meta al principio estaba más guiada hacia la eficiencia económica y la maximización de ganancias, nuestro proyecto ha ido adquiriendo bastante impacto indirecto en diversos ODS. No solo buscamos maximizar ganancias, sino que también queremos contribuir al bienestar económico y social a medio y largo plazo.

4. Preparación y comprensión de datos

4.1 Obtención y descripción

Puesto que habíamos planteado entrenar modelos que hicieran predicciones trimestrales acertadas de los precios de bolsa de empresas, a partir de sus datos fundamentales (también trimestrales), es obvio que lo fundamental es disponer de una gran base de datos.

Primero hicimos una amplia búsqueda por internet para ver si podíamos encontrar alguna base de datos pública con las características que buscábamos, pero ninguna se adaptaba completamente a nuestros requisitos, por lo que nos vimos obligados a generar nosotros mismos los datos, empleando Web-Scrapping.

Después de comparar entre varias páginas, como Morningstar o Investing, que proporcionaban datos fundamentales de las principales empresas mundiales, acabamos eligiendo realizar el scrapping sobre la página: https://stockanalysis.com/, que no solo es mucho más sencilla de escrapear por su forma de disponer la información en tablas bien estructuradas, si no que también dispone, de forma gratuita, de mucha más información temporal que el resto de páginas.

Hay aproximadamente 5600 empresas, y por cada una encontramos información trimestral de cuatro tipos diferentes:
Ingresos, balances, fondos y ratios, cada tipo con sus diferentes variables.
Aquí observamos un ejemplo de una tabla normal de ingresos de una empresa:

(VER FIGURA 1 DE ANEXOS)

Vemos que cada columna es la información de esa empresa en una fecha determinada (trimestral), por lo que decidimos que nuestra base final se basaría en:

Cada fila contendrá la información de una empresa en una fecha determinada, y las columnas serían todas las variables (tanto de ingresos como balances, fondos y ratios) que se encuentran en sus cuatro tablas diferentes, más dos columnas añadidas que serían su precio en bolsa en esa fecha, Close Price 0M, y su precio 3 meses después, Close Price 3M, que sería la variable a predecir.

El scrapping lo realizamos con la ayuda de librerías como selenium, requests, BeautifulSoup, pandas, yahoo finance… entre otras. Y se puede encontrar en los notebooks con el nombre de “scraping stockanalysis.ipynb” .

4.2 Limpieza

4.2.1 Estudio de nulos

Después de haber realizado el scrapping general (tras muchas horas de ejecución) nos quedó una base de datos con más de 100 columnas (variables diferentes) y más de 160.000 filas, aunque evidentemente no toda esta información era útil, ya que muchas de las empresas no presentaban información en todos los parámetros sobre los que realizamos el scrapping, por lo que quedaron muchos valores nulos.

¿ Por qué se pueden dar valores nulos en nuestra base?
Hay variables que únicamente aparecen en los datos de muy pocas empresas, por lo que son columnas prácticamente vacías.

(VER FIGURA 2 DE ANEXOS)

Puesto que realizamos el scrapping sobre la versión gratuita de la página web, la información estaba limitada temporalmente, es decir, existía más información pero no era accesible en la versión gratuita:

Esto en la base de datos inicial genera una fila más por empresa donde casi todos los valores son nulos, ya que es imposible acceder a ellos, puesto que no aparecen en el código fuente de la página.

Por cada empresa de la base de datos había una fila de este tipo.

2014-07-31	2014-04-30	+63 Quarters
1,009	988	Upgrade
-38.92%	-42.96%	Upgrade
507	503	Upgrade
502	485	Upgrade
285	304	Upgrade
86	87	Upgrade
371	391	Upgrade
131	94	Upgrade
28	30	Upgrade
-66	-104	Upgrade

También hay muchas empresas sin información trimestral de una o varias de sus secciones, esto genera que todos los valores en las variables de estas secciones sean nulos, como se puede ver en ejemplo siguiente:

(VER FIGURA 3 DE ANEXOS)

Por último, y como es lógico, hay casos particulares de empresas con muy poca información u otras que de forma aleatoria, en una fecha determinada, tienen algún valor nulo o tienen '-' en alguna de sus variables.

¿ Cómo tratamos los valores nulos ?
Puesto que necesitamos que todas las filas puedan utilizarse como vectores para entrenar nuestros modelos, es necesario deshacerse de los valores nulos.
Se puede ver todo el código en el notebook "tratado_de_base.ipynb" .

Sin embargo, sería un gran error tomar la decisión de eliminar todas las filas con nulos o todas las columnas con nulos, el motivo es el siguiente: ¡Nos quedaríamos sin datos!

```
Hay  163634  de  163634  filas con nulos
Hay  122  de  123  columnas con nulos
```

Por ejemplo, si tomáramos la decisión de eliminar todas las filas con 1 o más valores nulos, como existen columnas con prácticamente todos sus valores nulos, estaríamos eliminando una fila totalmente útil por culpa de estas variables. Por lo que antes hay que asegurarse de dejar únicamente las columnas y filas útiles.

Primero, hay que deshacerse de las filas totalmente inservibles, es decir, las filas que tienen valores nulos en las columnas que contienen información esencial sobre las predicciones, es decir, Close Price 3M y Close Price 0M:

```
Hay 26219 valores nulos entre las variables de Close Price 0M y 3M
```

Después decidimos quitar por completo las columnas con un gran número de valores nulos, las que tuvieran más de un tercio de los valores faltantes, que consideramos variables inutilizables. Esto quitó 34 columnas de 123.

También era importante deshacernos de las filas completa, o parcialmente nulas, que eran inutilizables, decidimos quitar todas aquellas que tuvieran más de un 25% de datos faltantes. Este paso quitó 45868 filas,que aunque eran muchas, para conseguir una base limpia que nos ofrezca información útil, es necesario sacrificarlas.

Después de esta limpieza de columnas y filas, volvimos a ver cuantos faltantes seguían teniendo las columnas, observando que seguían habiendo variables com más de casí el 20% de valores ausentes:

(VER FIGURA 4 DE ANEXOS)

En este punto antes de comenzar a quitar columnas indiscriminadamente hicimos una comprobación de que columnas eran iguales, es decir, cuales tenían una correlación lineal cercana a 1. Un ejemplo sería el caso de una hipotética variable ‘Ganancias’ y otra hipotética ‘Ganancias antes de impuestos’, estas variables no tendrían valores idénticos, pero como los valores de una son una conversión de la otra, estarían aportando la misma información, por lo que podríamos quedarnos únicamente con una de ellas, la que tenga menos valores nulos.

Ordenamos las parejas no repetidas de variables numéricas según su correlación, y observamos que efectivamente, si existían variables de este tipo:

(VER FIGURA 5 DE ANEXOS)

Por ejemplo ‘Shares Outstanding Diluted’ (total de acciones en circulación, incluyendo opciones y bonos) tiene casi 15% de valores nulos y ‘Shares Outstanding Basic’ (total de acciones de una empresa) que ofrece la misma información no tiene nulos. Por lo que haciendo esto, no solo reducimos las variables, también reducimos los nulos.

Aún así siguieron quedando 5 columnas con más del 90 % de valores, nulos, que acabamos eliminando, ya que si las hubiéramos conservado, solo hubieran quedado 32000 filas útiles.

En este punto, quitamos todas las filas con nulos, que eran un total de 26166 (de 91547 que quedaban), ya que si seguimos quitando columnas nos arriesgaríamos a perder información valiosa con tal de ahorrar muy pocas filas. Quedando finalmente:

```
Hay  0  de  65381  filas con nulos

Hay  0  de  62  columnas con nulos
```

4.2.2 Estudio de atípicos

Objetivo:

El estudio de atípicos que se va a realizar a continuación tiene el objetivo de detectar datos anómalos. Una vez detectados dichos datos se realizará una valoración sobre si es necesario eliminarlos o si por el contrario se han de mantener porque aportan información necesaria.
Con la intención de tener una base de datos más sencilla, permitiendo que los modelos que aplicaremos tengan menos error en sus predicciones y que encuentren patrones con mayor facilidad.

Estudio de datos atípicos:

Para identificar datos anómalos en cada columna, hemos empleado como criterio marcar aquellos que en valor absoluto se alejan de la media más de 4 desviaciones típicas.
Al ejecutar el programa (NOTEBOOK CON TÍTULO ATÍPICOS) recibimos como output cada uno de los datos atípicos encontrados y un diccionario con el total de datos atípicos de cada columna como se muestra en la figura 6 (DISPONIBLE EN ANEXOS).

Observando la parte de la izquierda de la figura 6 podemos ver como las columnas de la base de datos presentan relativamente pocos datos atípicos para su gran número de filas.
En la imagen situada en la parte derecha de la figura 6 observamos un ejemplo de una de las filas eliminadas, en ella podemos observar cómo los datos atípicos en este caso se encuentran en las columnas “Close Price 0M”, “Close Price 3M” y “EPS” (Indicador de la cantidad de ingresos netos obtenidos por una empresa por acción de sus acciones comunes en circulación). Observamos que esta fila pertenece a la empresa “Vislink Technologies”, analizando su cotización en bolsa a lo largo de los años, como se muestra en la figura 7 (DISPONIBLE EN ANEXOS), observamos cómo se produce en el año 2013 un crecimiento aberrante de la empresa que causa los datos atípicos de la base de datos.

Dichas fluctuaciones se deben en su mayoría a eventos corporativos, como splits de acciones, que alteran fuertemente el precio de las acciones en bolsa, estas fluctuaciones son impredecibles, por ello decidimos eliminar todos los datos atípicos que provienen de este tipo de naturaleza ya que de esta manera facilitaremos el entrenamiento de los modelos predictivos que más tarde desarrollaremos.

Estudiando el resto de los datos atípicos observamos como la mayoría de ellos son de la naturaleza del ejemplo de la figura 7 (DISPONIBLE EN ANEXOS) explicado anteriormente, por lo que decidimos eliminarlos.
Como observamos en la figura 8 (DISPONIBLE EN ANEXOS) después de la limpieza de atípicos obtenemos una base de datos con 64101 filas y 62 columnas, por lo que se han eliminado aproximadamente 1000 filas.

5. Conocimiento extraído

5.1 Modelos

A continuación vamos a realizar el entrenamiento de dos modelos predictivos, redes neuronales LSTM y Random Forest, aunque antes de explicar a fondo cada uno, hemos visto importante explicar puntos comunes entre ambos:

Funcionamiento: Se busca que ambos modelos sean capaces de recibir la última información trimestral de empresas y hagan predicciones sobre el futuro del precio de las acciones en el trimestre siguiente (LSTM predice directamente el porcentaje de crecimiento y Random Forest el precio), para así mostrar el conjunto de empresas con las que se espera obtener más rentabilidad (tendrán mayor crecimiento).

Formas de medir el éxito: Para comprobar que los modelos están funcionando correctamente existen diferentes mediciones para realizar, en nuestro contexto utilizaremos tres:

Error de las predicciones**:** Es la diferencia entre las predicciones y el valor real, un menor error puede indicar que el modelo hace predicciones más acertadas.

Rentabilidad del modelo: Es la media del crecimiento trimestral real que tienen las empresas que el modelo decide escoger. Esto es un porcentaje, tal que 0.2 sería un 20% he indicaría que si se invierten 100 euros, se tendrán 120 el trimestre después del crecimiento.

Datos: Para el entrenamiento utilizamos la base de datos general que previamente hemos limpiado, retirando datos nulos y atípicos que pudieran interferir en el éxito.
Además como las librerías no aceptan variables categóricas, como lo es la variable sector, que tiene gran importancia, hicimos una codificación one-hot, para poder utilizarla.
Esto se basa en añadir columnas con el nombre de los sectores,que contengan falso (0) si no es ese sector, o verdadero (1) si se trata de el.

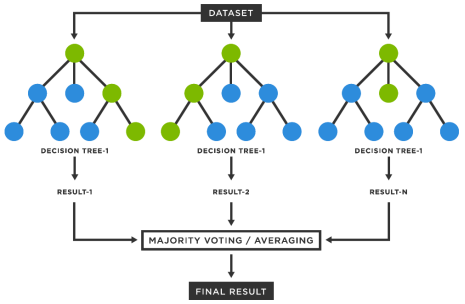
Estos datos los dividimos en tres, los datos de entrenamiento (Train) con los que se entrenará el modelo (más o menos el 80%), otros datos para hacer las comprobaciones (Test), y otro tipo de datos que hemos visto muy necesarios y que hemos llamado datos de validación. Estos últimos corresponden al tipo de datos a los que se enfrentará el modelo en la realidad, es decir, un solo vector por empresa, a diferencia de la data de test. Gracias a esto, nuestras comprobaciones y pruebas podrán ser más exactas y cercanas a la realidad.

PRIMER MODELO PREDICTIVO: Random Forest

El primer modelo predictivo que utilizaremos por su gran simplicidad será la técnica de Machine Learning llamada Random Forest, más concretamente el del tipo de regresión.

¿Qué es un Random Forest?

Este tipo de técnica de ensemble emplea una combinación de Árboles de Decisión (otra técnica de Machine Learning) que entrena con muestras de datos ligeramente diferenciadas, aumentando la precisión y la robustez en las predicciones.
Estas combinaciones de Árboles de Decisión se basan en generar árboles independientes, por medio de un proceso llamado bagging, generando sus propias predicciones, permitiendo que se pueda obtener una media de todas ellas para llegar a la predicción final.



Funcionamiento: El éxito de todos los modelos de aprendizaje automático dependen de dos factores clave:

Bias o sesgo: Cuánto dista la predicción del modelo con el valor real. Encontrar relaciones reales entre variables.

Varianza: El nivel de dependencia del modelo con los datos con los que ha sido entrenado, y lo susceptible que sería a que cambien datos (vida real).

Es la perfecta combinación de estos dos factores lo que llevará a un correcto funcionamiento de nuestro modelo predictivo. Y esto se hace mediante dos técnicas estadísticas:

Bagging: Se basa en entrenar varios sub-modelos (árboles de decisión) con datos ligeramente diferentes, es decir, que diferentes métodos se enfrenten a problemas distintos para que se presente más varianza y que el modelo final pueda funcionar mejor en el mundo real.
Boosting: Consiste en ir haciendo entrenamientos con submuestras que no presentan todas las variables, sino que se van haciendo combinaciones diferentes de variables para así poder encontrar relaciones verdaderas entre estas, y que no sean solo coincidencias circunstanciales. De esta forma se asigna el peso final (la importancia que tiene su presencia) a las diferentes variables.

En la práctica:
En nuestro caso utilizaremos un tipo de random forest llamado de regresión, ya que la variable que queremos predecir es numérica (precio en bolsa el siguiente trimestre).

El modelo será entrenado con datos históricos para poder visualizar su éxito, para que, cuando lo pongamos a funcionar en la realidad, pueda generar predicciones acertadas en base a los datos más actuales que tomará como entrada.

¿Por qué utilizamos el Random Forest?

En nuestro contexto de predecir el precio de las acciones de una empresa en base a datos fundamentales, este tipo de modelos presentan varias ventajas.

Por su diseño, y como hemos mencionado antes, es capaz de encontrar relaciones no lineales entre variables, dotándolo de robustez y haciendo apto para funcionar con una gran cantidad de datos sin llegar a sobre ajustarse (overfitting).
Además es muy sencillo de utilizar, y con la librería sklearn y optuna se nos ofrecen muchas posibilidades para usarlo en python, como el manejo de parámetros esenciales como:
N_esimators (número de árboles), Max_depth (para reducir el sobreajuste), Max_features (número máximo de variables que se selecciona en las submuestras)...
Y métodos de ajuste como la optimización bayesiana.

Primer modelo ejemplo:

(Se puede ver todo el código en el notebook "random_forest_3.ipynb")

Para comenzar entrenamos un random forest con los datos prácticamente por defecto para ver el comportamiento:

```
# PRIMER MODELO CON HIPER-PARÁMETROS (CASI) DEFAULT

model1 = RandomForestRegressor(n_estimators= 20, random_state= 42, n_jobs = -1, oob_score = True)

model1.fit(X_train, Y_train)
```

Obtuvimos un error medio absoluto en la data de Test de 24.55 y una rentabilidad máxima de 82% si se hubiera establecido como mínimo a invertir todas aquellas con más de un 80% de crecimiento (a esto lo llamaremos filtro).

Mientras que en los datos de validación vemos un error medio de 61.92 y una rentabilidad máxima del 10% con un filtro del 10%.

Lo que nos indica que, aunque el modelo ya es capaz de hacerse una idea aproximada de los precios de empresas en base a sus datos, hay un gran margen de mejora.

También podemos observar que importancia se le asigna a cada variable:

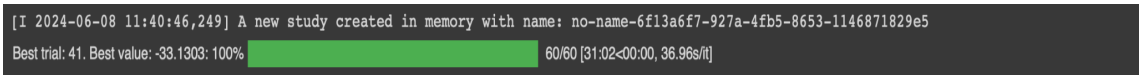
(VER FIGURA 9 DE ANEXOS)

Ajuste de hiper-parámetros:

Para mejorar el éxito de nuestro modelo necesitamos ajustar sus hiper-parámetros (número de árboles, max_depth…), para ver cuál se adapta mejor a nuestro contexto.
Esto se puede hacer de varias formas, o probando de uno en uno valores de parámetros y haciendo combinaciones aleatorias, cosa que resulta poca efectiva y tiene gran coste, o mediante la optimización bayesiana, que utiliza el 'out of the bag error' mencionado antes para ir haciendo combinaciones acertadas de hiper-parámetros durante un largo y minucioso entrenamiento, esta última técnica es la que utilizamos. Aquí podemos ver un ejemplo de lo que aparece en terminal:

En este proceso dejamos que se probaran las combinaciones de hiper-parámetros que presentarán menor error y mayor rentabilidad en el entrenamiento, y trás un largo tiempo de ejecución (60 pruebas) obtuvimos que los mejores parámetros eran:

{'n_estimators': 180, 'max_depth': 13, 'min_samples_split': 6, 'min_samples_leaf': 3, 'max_features': 0.9032028742983398, 'ccp_alpha': 0.3318358395791241}



Con el que bajamos el error del test a 14.25 y el de validación a 33.112.
Y aunque la rentabilidad máxima en la data de test subió mucho (700%), la rentabilidad en los datos de validación se mantenía baja (11%), y el error bastante alto, por lo que vimos necesario probar otro tipo de modelos o una estrategia diferente.

Modelo de cuantiles:

(Se puede ver todo el código en el notebook "cuantil_regression-2.ipynb")

Fijándonos en las predicciones que realiza el modelo general, y el tipo de datos con el contábamos, vimos que era la gran diferencia entre los precios de acciones lo que estaba conduciendo a un éxito menor al esperado, y es que, al haber empresas con precios de acciones de más de 10 mil dólares, mientras que el de otras era menor a 1 dólar, el comportamiento (en cuanto a crecimiento) era muy diferente, y las predicciones se veían afectadas por las grandes diferencias.

(VER FIGURA 10 DE ANEXOS)

La mejor solución que se nos ocurrió para combatirlo fue separar los datos según sus precios, para así entrenar modelos diferentes, para que así cada modelo esté más adaptado para el tipo de acción que va a predecir y no tome ninguna influencia de acciones de diferente tipo. Para que así en la puesta en práctica, también se separen los datos según sus precios, y cada modelo trate de predecir únicamente las empresas más similares a las que usó en el entrenamiento.

Para ello miramos los diferentes percentiles de la variable 'Close Price 0M' (el último precio de la acción del que se dispone), para así separar los datos por intervalos, surgiendo los siguientes modelos:

Un modelo para los datos entre el percentil 0 y 25, otro para el intervalos de entre 50 y 75, otro para 75 y 90, entre 90 y 95, entre 95 y 95.5, y entre 95.5 y 100. Un total de 6 modelos.

Cada modelo lo entrenamos con los datos que le pertenecían y lo ajustamos con optimización bayesiana para que funcionarán lo mejor posible. Obteniendo finalmente una combinación de modelos que se adapta más a los datos y tiene mayor rentabilidad.

(VER FIGURA 11 DE ANEXOS)

Y esque, aunque a pesar de que ahora el error medio absoluto ha subido, tanto en validación (58), como en el test (28), es evidente que con esta estrategia se consigue captar mucho mejor las razones que provocan crecimiento en las acciones de empresas, ya que la rentabilidad máxima, que es lo que verdaderamente nos interesa, ha subido ahora al 27% en validación.

Combinación:

Una vez tenemos dos modelos totalmente funcionales, es necesario desarrollar una estrategia que seguir para su puesta en práctica, y es que, como tenemos dos modelos random forest, la predicción final de estos deberá ser la combinación de sus predicciones siguiendo alguna norma.
Lo que hemos planteado es que a partir de los datos nuevos se hagan las predicciones con los modelos (dividiendo los datos según cuantiles para el segundo modelo), y una vez tenemos los precios predichos, calculamos el crecimiento respecto al precio anterior.

Empezamos a almacenar en la lista de empresas finales de cada modelo todas las empresas que tengan un crecimiento mayor al 700%, y vamos bajando este filtro hasta que la lista tenga una cantidad mínima de empresas, y es que según hemos ido probando, esto es más útil que establecer un crecimiento mínimo, ya que es importante contar con un cierto número de empresas cuyo crecimiento pueda contrarrestar el crecimiento negativo de otras que el modelo ha predicho que sería positivo.

Una vez tenemos estas listas, que son diferentes en ambos modelos, para emitir la lista final, cogemos todas las del modelo de cuantiles, siempre y cuando el modelo general no haya predicho que van a tener un crecimiento negativo, y añadimos las del modelo general que tengan un crecimiento mayor al percentil 80 de los crecimientos del modelo de cuantiles (siempre y cuando en este no tengan crecimientos negativos).

Para establecer el mínimo de empresas a añadir en sus listas de cada modelo, hemos generado muestras aleatorias similares a la realidad con los datos de test, y hemos ido probando diferentes mínimos, hasta obtener que, tras 1000 pruebas con muestras aleatorias diferentes:

(Se puede ver todo el código en el notebook "Copy_of_buscar_filtros.ipynb")

	FILTROS	minimo	maximo	media	cantidad_negativos	media_negativos	cantidad_mas_02
0	40 | 20	-0.11440	5.8711	1.5436	11	-0.069512	481
1	60 | 50	0.06120	2.4673	0.8053	0	0.000000	531
2	70 | 40	0.08123	2.9816	0.8173	0	0.000000	711

Con un mínimo de 40 empresas como salida final del modelo general, y 20 en el modelo de cuantiles se tiene una rentabilidad media de 154% (media de crecimientos reales de las que predijo), aunque hubieron 11 casos donde la rentabilidad fue negativa, y solo 481 casos (de 1000) se tuvo una rentabilidad superior al 20%, mientras que con un mínimo de 70 para el general y 40 para el de cuantiles, se tiene una media más baja (80%) pero no tiene casos de rentabilidad negativa, y en el 71% de los casos (711), se tuvo rentabilidad superior al 20%. Por ello elegiremos estos mínimos de longitud de empresas finales elegidas para los modelos, ya que, aunque a la larga tendremos menos rentabilidad media, será menos probable tener pérdidas.

Segundo modelo predictivo: Red Neuronal LSTM/MLP:

Con el fin de predecir el crecimiento de las empresas en bolsa. En este trabajo se ha propuesto el uso de una red neuronal LSTM con un MLP (Multilayer perceptron) previo a su capa de salida. Las redes neuronales LSTM ofrecen una alta efectividad para tareas que implican datos secuenciales. Es por ello por lo que han sido elegidas en este trabajo para intentar predecir el crecimiento de empresas basándose en los datos financieros de los últimos X trimestres. Así mismo hemos tomado la decisión de añadir un MLP previo a la capa de salida ya que este tipo de redes neuronales permiten la transformación de características obtenidas con las capas LSTM así como un aumento de la capacidad de generalización, y la flexibilidad del modelo. Las capas Dense que conforman el MLP permiten refinar y ajustar las predicciones, asegurando que el modelo no solo captura dependencias temporales de los datos financieros, sino que también las procesa y la combina de manera óptima para crear predicciones precisas y robustas.

Estructura del input y output:

La red neuronal propuesta tomará como input los valores financieros de la empresa de los 10 trimestres anteriores al trimestre que se intenta predecir y tomará como output el porcentaje de crecimiento de la empresa en el próximo trimestre. Así mismo hemos realizado una estandarización de todos los valores de entrada de la red neuronal y

hemos aplicado one-hot encoding a la variable sector para facilitar el entrenamiento de la red neuronal.

Ajuste de hiperparámetros :

Dado que los datos utilizados son complejos y la red neuronal empleada es de gran tamaño. Decidimos emplear la optimización bayesiana para de esta manera no tener que probar todas las combinaciones posibles de hiperparámetros en busca de la combinación que nos diera la mejor red neuronal para nuestro problema.

La optimización bayesiana es un método que explora de manera más eficiente todas las combinaciones de hiperparámetros posibles . El proceso comienza al especificar los hiperparámetros que se deben maximizar. En nuestro caso elegimos como parámetros a optimizar la cantidad de capas, número de neuronas por capa, número de capas de normalización, el *dropout rate* y el *batch size.* En vez de iterar a través de todas las posibles combinaciones de hiperparámetros, la optimización bayesiana emplea un modelo probabilístico para estimar el rendimiento del modelo de red neuronal en diferentes conjuntos de hiperparámetros.

En primer lugar, se seleccionan al azar diferentes combinaciones del espacio de hiperparámetros. Entrenando la red neuronal con los hiperparámetros para analizar cada una de estas combinaciones se mide su desempeño mediante el cálculo del error medio absoluto. Utilizando esta información inicial, podemos generar una función de probabilidad que estimará el desempeño del modelo en relación a los diversos hiperparámetros.

Para encontrar la combinación que genere el menor error absoluto, se utiliza esta función de probabilidad para calcular una función de adquisición que evalúa los siguientes hiperparámetros.

En la figura 12 (DISPONIBLE EN ANEXOS) vemos el output de un script de Python (NOTEBOOK CON TÍTULO OPTIMIZACIÓN BAYESIANA ENTREGABLE) que ejecuta la optimización bayesiana descrita anteriormente. Observamos así que el mejor resultado se obtiene una red neuronal con 5 capas intermedias LSTM cada una de ellas con 280 neuronas seguidas de una capa Flatten y de 3 capas Dense con 168 neuronas la primera 84 la segunda y 42 la tercera. Así mismo observamos como el proceso no utiliza las capas de normalización y usa un *dropout* de 0.37 después de cada capa. Por último observamos como el *batch size* empleado es de 133 y como el error medio absoluto objetivo con esta combinación de hiperparámetros es 0.1934. Obteniendo así una estructura de red neuronal como la que se muestra en la figura 13 (DISPONIBLE EN ANEXOS).

Entrenamiento:

Como ya hemos mencionado en el apartado anterior el entrenamiento realizado con un script de python (NOTEBOOK CON TÍTULO LSTM ENTREGABLE), lo hemos hecho con un *batch size* de 133, y hemos entrenado el modelo durante 30 *epochs* dándonos una evolución del error de entrenamiento y de validación que se puede observar en la figura 14 (DISPONIBLE EN ANEXOS)

Como se puede observar en esta figura, durante los 30 primeros epochs el error se reduce tanto en el error de validación como en el error de entrenamiento mientras que a partir de este punto el error de entrenamiento sigue bajando y el error de validación comienza a subir. Esto se debe a que la red neuronal está entrando en un proceso denominado *overfitting* o sobreajuste que consiste en que en vez de aprenderse los patrones de la base de datos, empieza a memorizarlos, reduciendo así el error. Por tanto cuando se le presentan datos nuevos como los de validación su error aumenta.

Estrategia:

Una vez la LSTM realiza las predicciones, es importante establecer con qué criterio decidir qué empresas son óptimas para invertir en base a sus crecimientos predichos. Para ello seguiremos la estrategia de elegir todas aquellas empresas que tengan una predicción de crecimiento de más del 40%, ya que, realizando pruebas con muestras aleatorias, tal y como se hizo en el random forest, se ha observado que es el filtro que mayor rentabilidad media ofrece. (NOTEBOOK CON TÍTULO LSTM FUTURO CRECIMIENTO ENTREGABLE)

<u>Combinación de los modelos:</u>

Tal y como habíamos visto en el random forest, la selección de empresas en las que invertir se hacía combinando ambos modelos en base a ciertos criterios, ahora con el random forest (como combinación del general y el de cuantiles) y la LSTM haremos lo mismo. La selección final de empresas en las que invertir será la combinación de las empresas seleccionadas por ambas, ya que en la práctica hemos visto, que habían muchas empresas comunes en ambas, aunque otras no, por lo que, combinar ambas respuestas es una buena opción.

Con este criterio de selección volvimos a hacer una prueba con muestras aleatorias similares a la realidad, obteniendo los siguientes resultados:

	Min LSTM	Filtros R_F	Total Pruebas	minimo	maximo	media	cantidad_negativos	media_negativos	cantidad_mas_02
0	40%	70 | 40	1000	0.09412	1.6126	0.8823	0	0.0	792

Es decir, en un 79% de los casos tendría una rentabilidad superior al 20%, y sería muy poco probable tener una rentabilidad negativa, al menos en base a los datos de los que disponemos, ya que es obvio que hubieran cambios de comportamiento en el mercado, podría ocurrir cualquier cosa.

Respecto a lo que hicimos con el random forest, vemos que ahora la rentabilidad media ha subido ligeramente gracias a la combinación con la LSTM, y aunque la rentabilidad máxima ha bajado bastante, ahora el mínimo es más alto.
Aunque hay que tener en cuenta que son muestras aleatorias generadas a partir de los datos de test, y aunque estas muestras están hechas de tal forma que sean similares a la realidad, no dejan de ser datos históricos, por lo que también hay que ser crítico y escéptico, y tener en cuenta que en la realidad se puede comportar diferente.

6. Resultados

Resultados reales de los modelos en conjuntos:

Puesto que salió información del trimestral de marzo en stock analysis, trimestre que no aparecía en nuestra base principal, pudimos hacer una prueba completamente real de los modelos, y pudimos hacer comparaciones ya que también contábamos con los precios del siguiente trimestre (junio). Esta base aparece como 'Last Stock.csv'.

Tras realizar las predicciones con ambos modelos y realizando la combinación explicada en el punto anterior, obtuvimos que se hubiera recomendado invertir en 44 empresas, que el siguiente trimestre tuvieron un crecimiento medio real del 17.6 %, es decir, si hubiéramos invertido en dichas empresas dividiendo nuestro portfolio en partes iguales, la rentabilidad hubiera sido bastante positiva para ser trimestral, ya que el crecimiento medio de todas las empresas del trimestre fue negativo (-0.87%).

Por lo que podemos concluir con que hemos creado una forma de seleccionar empresas en las que invertir, que genera beneficios con bastante seguridad.

7.Despliegue

7.1 Aplicación desarrollada

Para desarrollar la aplicación web, hemos empleado la librería de Python llamada Streamlit, que facilita la creación de interfaces de usuario interactivas para aplicaciones de datos. Además, utilizamos un poco de CSS y HTML para personalizar el diseño.

La aplicación proporciona acceso a este informe. Además, se explica todo lo relevante sobre el proyecto. En el sitio web, los usuarios pueden encontrar un resumen explicativo de cada modelo utilizado en el proyecto. Además, tienen acceso a gráficos de velas y volumen, así como a bases de datos que abarcan todas las empresas analizadas. Y lo más importante, se muestra un ranking elaborado por cada modelo,

destacando las empresas con mayor crecimiento esperado, facilitando el trabajo al usuario.

La página es accessible en el link:

https://tradingstocks.nyxglow.com/

8.Conclusiones

Tras realizar este trabajo hemos llegado a ciertas conclusiones que convendría recalcar. Por ejemplo, hemos descubierto como los datos tienen una gran importancia en el sector financiero. Así mismo hemos podido comprobar la eficacia de ciertos modelos de vanguardia como lo son el Random Forest y las redes neuronales, como se ha demostrado en los apartados anteriores.

Otra conclusión a la que hemos llegado gracias a este trabajo es la capacidad de los modelos de aprendizaje profundo y machine learning de abarcar grandes cantidades de datos que un humano sería incapaz de procesarlas, ya que en este informe se ha demostrado cómo los modelos descritos anteriormente llegan a “comprender” los datos y pueden realizar predicciones sobre el futuro precio de las empresas de manera certera en base a dichos datos.

Como tercera conclusión tenemos el éxito de nuestro modelos, pues hemos logrado superar la rentabilidad media del Nasdaq 100 (el fondo indexado más importante del mundo) mediante el uso de inteligencia artificial y machine learning, obteniendo una rentabilidad trimestral del 17% superando así teóricamente al Nasdaq 100 el último año en un 36% anual.

Como última conclusión nos gustaría recalcar la tarea que hemos realizado de integración de dichos modelos en una página web muy intuitiva que permite que cualquier usuario sin experiencia programando pueda usar dichos modelos y así se pueda beneficiar de la rentabilidad teórica de los mismos.

9.Bibliografía

[1] *IEEE Xplore search results*. (s. f.). https://ieeexplore.ieee.org/search/searchresult.jsp?newsearch=true&queryText=Stock%20Price%20Prediction%20Using%20LSTM%20Neural%20Network

[2] Kebede, G. A., Lo, S., Wang, F., & Chou, J. (2024). Transfer learning-based deep learning models for proton exchange membrane fuel remaining useful life prediction. *Fuel*, *367*, 131461. https://doi.org/10.1016/j.fuel.2024.131461

[3] Raut, S. (2024). Stock Market Price Prediction and Forecasting Using Stacked LSTM. *Indian Scientific Journal Of Research In Engineering And Management*, *08*(03), 1-5. https://doi.org/10.55041/ijsrem29832

[4] Stock Analysis. (s. f.). *Stock Analysis - Free Online Stock Information for Investors*. https://stockanalysis.com/

[5] Gamez, M. J. (2022, 24 mayo). *Objetivos y metas de desarrollo sostenible - Desarrollo Sostenible*. Desarrollo Sostenible. https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/

10.Anexos

-Figura 1:

Income | Balance Sheet | Cash Flow | Ratios — Annual | Quarterly | Trailing

Agilent Technologies Income Statement

Financials in millions USD. Fiscal year is November - October. — Millions | Export

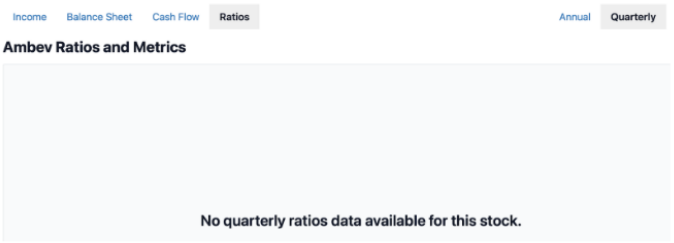
Quarter Ended	2024-01-31	2023-10-31	2023-07-31	2023-04-30	2023-01-31	2022-10-31	2022-07-31	202
Revenue	1,658	1,688	1,672	1,717	1,756	1,849	1,718	
Revenue Growth (YoY)	-5.58%	-8.71%	-2.68%	6.85%	4.90%	11.39%	8.32%	
Cost of Revenue	750	773	1,014	793	788	837	779	
Gross Profit	**908**	**915**	**658**	**924**	**968**	**1,012**	**939**	
Selling, General & Admin	396	393	407	415	419	422	412	
Research & Development	128	114	118	126	123	119	116	
Operating Expenses	524	507	525	541	542	541	528	
Operating Income	**384**	**408**	**133**	**383**	**426**	**471**	**411**	
Interest Expense / Income	22	22	24	24	25	23	19	
Other Expense / Income	-41	-34	-23	-18	-9	-7	-5	
Pretax Income	**403**	**420**	**132**	**377**	**410**	**455**	**397**	

(Captura de pantalla de https://stockanalysis.com/)

-Figura 2:

```
La variable:  Goodwill  tiene:  0.01  valores presentes
La variable:  Intangible Assets  tiene:  0.01  valores presentes
La variable:  Common Stock Issued  tiene:  0.01  valores presentes
La variable:  Share Repurchases  tiene:  0.01  valores presentes
La variable:  Interest Income  tiene:  0.02  valores presentes
La variable:  Interest Expense  tiene:  0.02  valores presentes
La variable:  Total Liabilities and Equity  tiene:  0.02  valores presentes
La variable:  Asset Turnover  tiene:  0.02  valores presentes
La variable:  Return on Equity (ROE)  tiene:  0.02  valores presentes
La variable:  Return on Assets (ROA)  tiene:  0.02  valores presentes
La variable:  Earnings Yield  tiene:  0.02  valores presentes
La variable:  FCF Yield  tiene:  0.02  valores presentes
La variable:  Preferred Dividends  tiene:  0.24  valores presentes
La variable:  Net Income Common  tiene:  0.24  valores presentes
```

-Figura 3:



(Captura de pantalla de https://stockanalysis.com/)

-Figura 4:

```
0  -  La variable:  Change in Investments  tiene:  0.78  valores presentes
1  -  La variable:  Interest Coverage  tiene:  0.82  valores presentes
2  -  La variable:  Debt Growth  tiene:  0.84  valores presentes
3  -  La variable:  Acquisitions  tiene:  0.84  valores presentes
4  -  La variable:  Shares Outstanding (Diluted)  tiene:  0.86  valores presentes
5  -  La variable:  Comprehensive Income  tiene:  0.86  valores presentes
6  -  La variable:  Net Cash Per Share  tiene:  0.86  valores presentes
7  -  La variable:  Goodwill and Intangibles  tiene:  0.87  valores presentes
8  -  La variable:  Other Operating Expenses  tiene:  0.87  valores presentes
9  -  La variable:  Other Expense / Income  tiene:  0.91  valores presentes
10  -  La variable:  Debt / Equity Ratio  tiene:  0.91  valores presentes
11  -  La variable:  Interest Expense / Income  tiene:  0.93  valores presentes
12  -  La variable:  Share-Based Compensation  tiene:  0.94  valores presentes
13  -  La variable:  Income Tax  tiene:  0.95  valores presentes
14  -  La variable:  Debt Issued / Paid  tiene:  0.95  valores presentes
15  -  La variable:  Market Cap Growth  tiene:  0.95  valores presentes
16  -  La variable:  Cost of Revenue  tiene:  0.96  valores presentes
17  -  La variable:  Share Issuance / Repurchase  tiene:  0.96  valores presentes
18  -  La variable:  Revenue Growth (YoY)  tiene:  0.97  valores presentes
19  -  La variable:  Current Debt  tiene:  0.97  valores presentes
```

-Figura 5:

	Variable 1	Variable 2	Correlación	% Nulos Variable 1	% Nulos Variable 2
2	Total Shareholder Return	Shares Change	1.000000	0.01	0.02
3	Shares Outstanding (Diluted)	Shares Outstanding (Basic)	1.000000	0.14	0.00
4	EPS (Basic)	EPS (Diluted)	0.999999	0.01	0.02
5	EBIT Margin	EBITDA Margin	0.999965	0.01	0.01
6	Other Current Assets	Other Long-Term Assets	0.999396	0.00	0.01
7	Market Capitalization	Enterprise Value	0.999394	0.00	0.00
8	Total Assets	Total Long-Term Assets	0.998157	0.00	0.00
9	Total Liabilities	Total Current Liabilities	0.995720	0.00	0.00
10	Total Long-Term Liabilities	Total Liabilities	0.993939	0.00	0.00
11	Total Assets	Total Current Assets	0.993426	0.00	0.00
12	Revenue	Cost of Revenue	0.990597	0.00	0.04
13	Net Income	Pretax Income	0.990450	0.00	0.00
14	Long-Term Debt	Total Debt	0.984789	0.01	0.00
15	Total Long-Term Assets	Total Current Assets	0.984685	0.00	0.00

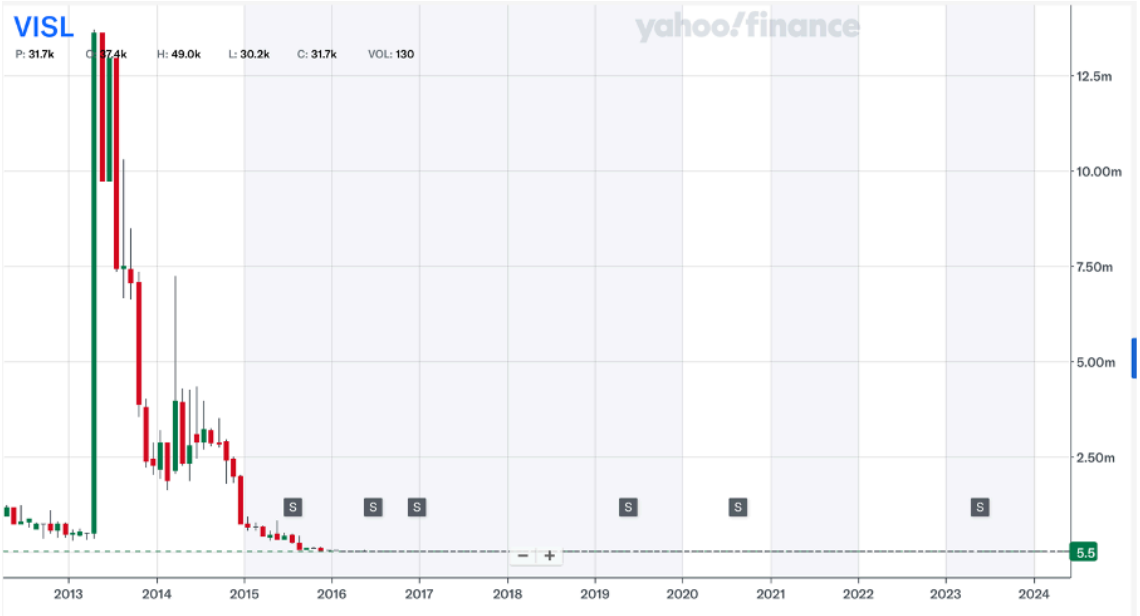
-Figuras 6:

	Columna	Número de atípicos
0	Shares Outstanding (Basic)	352
1	Market Capitalization	251
2	Revenue Growth (YoY)	25
3	Market Cap Growth	66
4	P/OCF Ratio	61
5	Quick Ratio	189
6	Current Ratio	270
7	EPS (Basic)	38
8	PE Ratio	42
9	Return on Capital (ROIC)	14
10	PB Ratio	58
11	Free Cash Flow Margin	42
12	Operating Margin	19
13	Profit Margin	22
14	EBIT Margin	19
15	Gross Margin	29
16	Book Value Per Share	27
17	Free Cash Flow Per Share	34
18	Close Price 0M	35
19	Close Price 3M	19
20	PS Ratio	30
21	Cash Growth	25
22	Revenue	130
23	Gross Profit	165
24	Selling	166
25	General & Admin	166
26	Operating Expenses	166
27	Operating Income	123
28	Income Tax	126
29	Net Income	128
30	EBITDA	126
31	Depreciation & Amortization	147
32	EBIT	109
33	Cash & Equivalents	155
34	Cash & Cash Equivalents	129
35	Other Current Assets	161
36	Property	117
37	Plant & Equipment	117
38	Total Assets	162
39	Accounts Payable	139
40	Current Debt	114
41	Other Current Liabilities	165
42	Other Long-Term Liabilities	137
43	Total Long-Term Liabilities	166
44	Total Debt	135
45	Retained Earnings	117
46	Shareholders' Equity	131
47	Net Cash / Debt	132
48	Working Capital	65
49	Other Operating Activities	104
50	Debt Issued / Paid	107
51	Financing Cash Flow	74
52	Net Cash Flow	104
53	Free Cash Flow	126
54	Operating Cash Flow	136
55	Investing Cash Flow	112
56	Capital Expenditures	100
57	Receivables	91
58	P/FCF Ratio	2
59	Total Shareholder Return	1

Columnas con valores atípicos: EPS (Basic), Close Price 0M, Close Price 3M

Revenue	0.15
Revenue Growth (YoY)	354.55
Gross Profit	0.09
Selling, General & Admin	1.50
Operating Expenses	4.33
Operating Income	-4.24
Income Tax	0.00
Net Income	-4.28
Shares Outstanding (Basic)	0.00
EPS (Basic)	-254879.95
Free Cash Flow	-4.00
Free Cash Flow Per Share	-235411.77
Gross Margin	60.00
Operating Margin	-2824.67
Profit Margin	-2854.00
Free Cash Flow Margin	-2668.00
EBITDA	-3.28
Depreciation & Amortization	0.96
EBIT	-4.24
EBIT Margin	-2824.67
Cash & Equivalents	1.89
Cash & Cash Equivalents	1.89
Cash Growth	-26.15
Receivables	0.95
Other Current Assets	0.48
Property, Plant & Equipment	0.75
Total Assets	24.96
Accounts Payable	1.67
Current Debt	0.13
Other Current Liabilities	1.00
Other Long-Term Liabilities	0.00
Total Long-Term Liabilities	2.03
Total Debt	2.15
Retained Earnings	-165.53
Shareholders' Equity	19.66
Net Cash / Debt	-0.27
Working Capital	4.33
Book Value Per Share	1156705.88
Other Operating Activities	-0.42
Operating Cash Flow	-3.46
Capital Expenditures	-0.54
Investing Cash Flow	-0.54
Debt Issued / Paid	-0.03
Financing Cash Flow	0.93
Net Cash Flow	-3.07
Market Capitalization	47.00
Market Cap Growth	-12.36
PE Ratio	-2.67
PS Ratio	50.68
PB Ratio	2.41
P/FCF Ratio	-2.40
P/OCF Ratio	-2.60
Quick Ratio	0.87
Current Ratio	2.32
Return on Capital (ROIC)	-68.32
Total Shareholder Return	-142.86
Close Price 0M	2822400.00
Close Price 3M	1368000.00

-Figura 7:



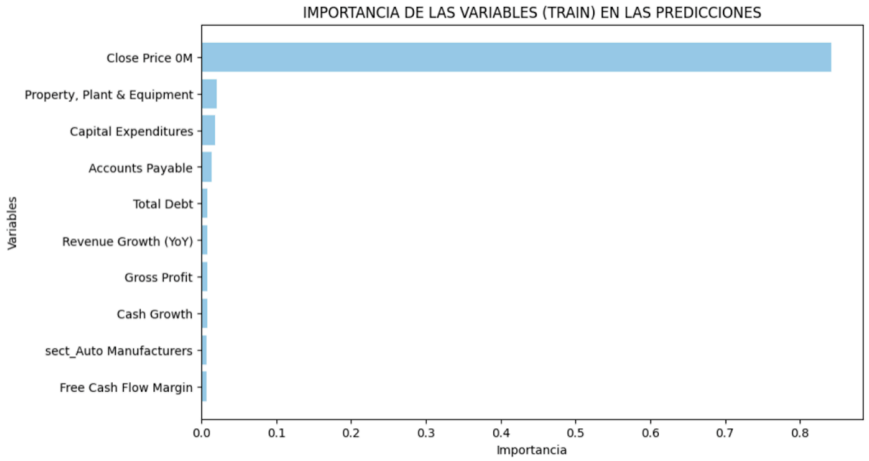
(Captura de pantalla de https://www.tradingview.com/)

-Figura 8:

	Ticker	Name	Sector	...	Total Shareholder Return	Close Price 0M	Close Price 3M
0	A	Agilent Technologies	Diagnostics & Research	...	1.24	103.370003	132.830002
1	A	Agilent Technologies	Diagnostics & Research	...	2.04	121.769997	105.639999
2	A	Agilent Technologies	Diagnostics & Research	...	1.93	136.080002	126.660004
3	A	Agilent Technologies	Diagnostics & Research	...	2.58	152.080002	133.250000
4	A	Agilent Technologies	Diagnostics & Research	...	3.86	138.350006	155.690002
...	...	...	...	...	...	...	...
64096	ZYXI	Zynex	Medical Distribution	...	-0.32	0.181818	0.163636
64097	ZYXI	Zynex	Medical Distribution	...	106.58	0.127273	0.163636
64098	ZYXI	Zynex	Medical Distribution	...	87.65	0.154545	0.136364
64099	ZYXI	Zynex	Medical Distribution	...	99.48	0.136364	0.136364
64100	ZYXI	Zynex	Medical Distribution	...	51.53	0.263636	0.145455

[64101 rows x 62 columns]

-Figura 9:



-Figura 10:

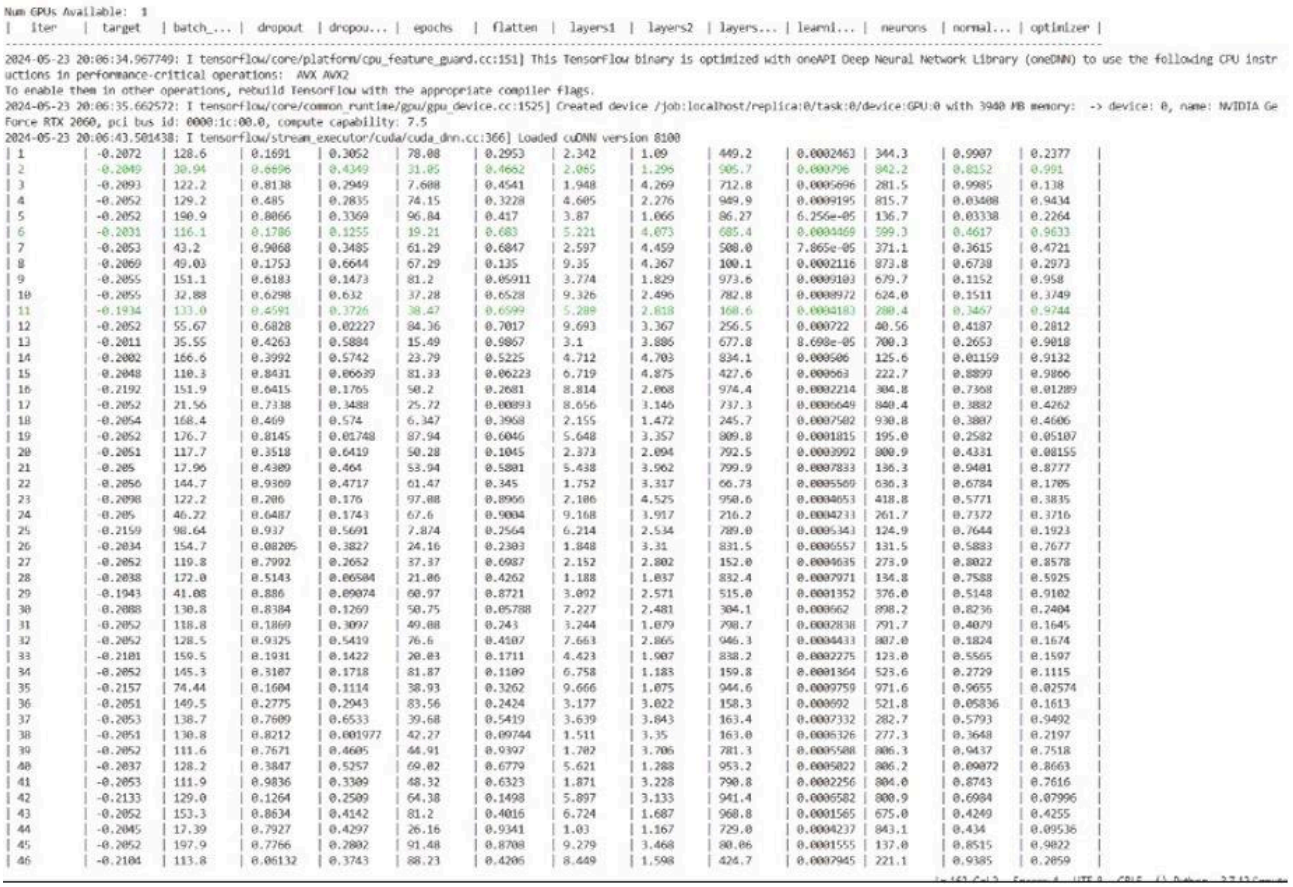
	Intervalo	Frecuencia Q0	Frecuencia Q25	Frecuencia Q75	Frecuencia Q90	Frecuencia Q99
0	(-1, -0.8)	27	35	4	5	5
1	(-0.8, -0.6)	201	235	37	37	32
2	(-0.6, -0.4)	884	901	204	155	81
3	(-0.4, -0.1)	4546	7570	1990	1281	200
4	(-0.1, 0.1)	4665	13015	4629	2656	194
5	(0.1, 0.4)	3385	8415	2444	1470	94
6	(0.4, 0.6)	941	1038	160	72	13
7	(0.6, 1)	753	451	50	35	9
8	(1, 2)	377	133	24	15	4
9	(2, 3)	69	13	2	1	3
10	(3, 4)	28	3	1	0	1
11	(4, 5)	10	2	1	0	0
12	(5, 7)	8	2	0	0	0
13	(7, 10)	8	1	0	0	0
14	(10, 80)	5	0	0	0	0

-Figura 11:

	Percentil	MAE
0	0.0	1.411129
1	25.0	3.591274
2	50.0	6.774599
3	75.0	12.161272
4	90.0	39.670653
5	99.5	2871.660248

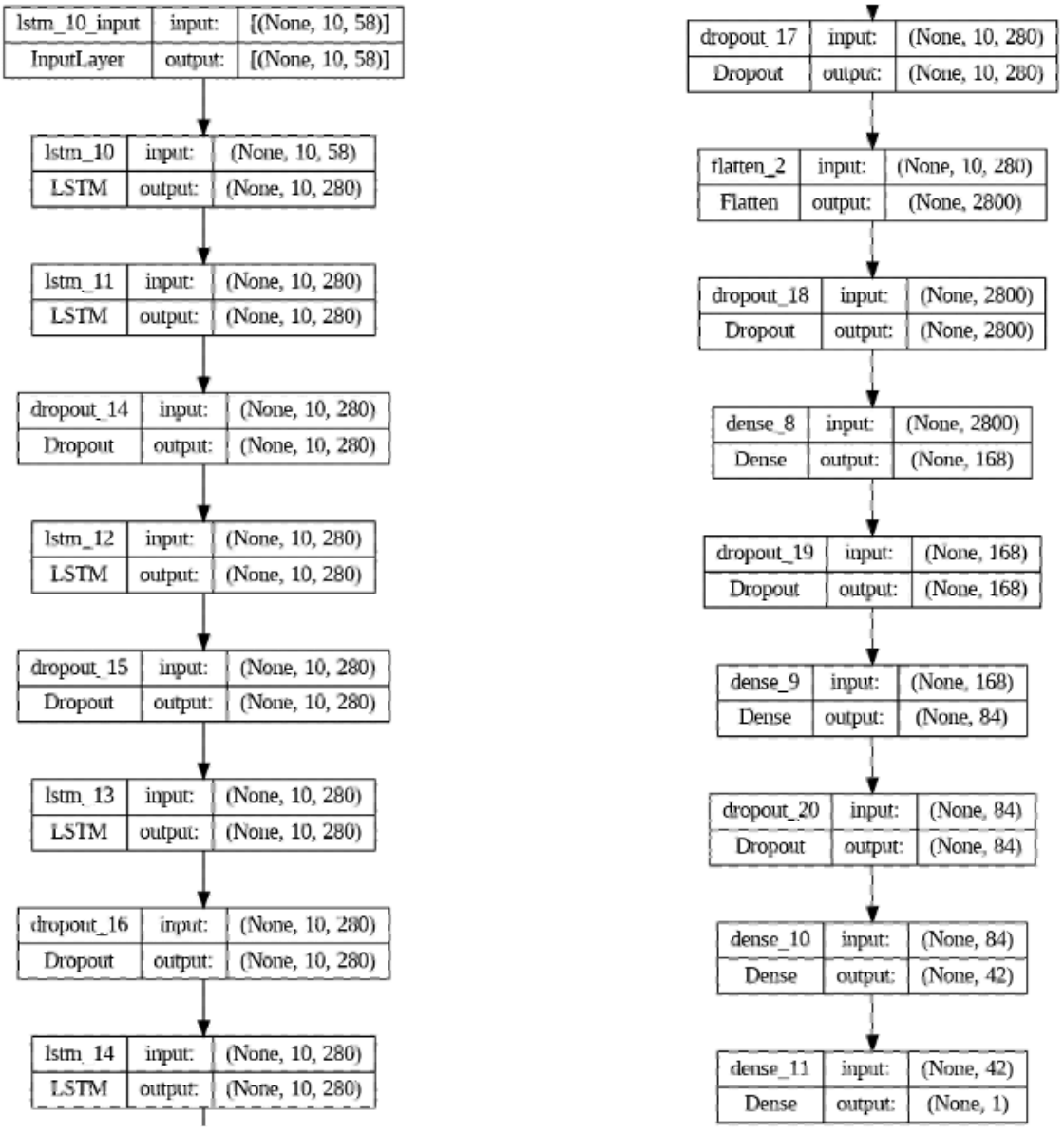
	Rentabilidad	Empresas	Filtros
0	0.277002	7	4.9
1	0.277002	7	5.0
2	0.274993	5	5.4
3	0.274993	5	5.5
4	0.274993	5	5.6

-Figura 12:

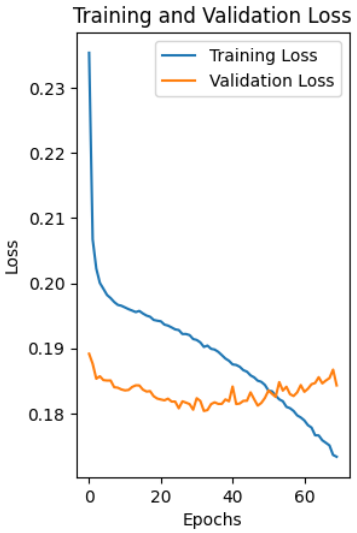


```
Num GPUs Available:  1
|   iter    |   target   | batch_... |  dropout  | dropou... |  epochs   |  flatten  |  layers1  |  layers2  | layers... | learni... |  neurons  | normal... | optimizer |
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------
2024-05-23 20:06:34.967749: I tensorflow/core/platform/cpu_feature_guard.cc:151] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instr
uctions in performance-critical operations:  AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2024-05-23 20:06:35.662572: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1525] Created device /job:localhost/replica:0/task:0/device:GPU:0 with 3940 MB memory:  -> device: 0, name: NVIDIA Ge
Force RTX 2060, pci bus id: 0000:1c:00.0, compute capability: 7.5
2024-05-23 20:06:43.501438: I tensorflow/stream_executor/cuda/cuda_dnn.cc:366] Loaded cuDNN version 8100
| 1         | -0.2072   | 128.6     | 0.1691    | 0.3052    | 78.08     | 0.2953    | 2.342     | 1.09      | 449.2     | 0.0002463 | 344.3     | 0.9907    | 0.2377    |
| 2         | -0.2049   | 30.94     | 0.6696    | 0.4349    | 31.05     | 0.4662    | 2.065     | 1.296     | 905.7     | 0.000796  | 842.2     | 0.8152    | 0.991     |
| 3         | -0.2093   | 122.2     | 0.8138    | 0.2949    | 7.608     | 0.4541    | 1.948     | 4.269     | 712.8     | 0.0005696 | 281.5     | 0.9985    | 0.138     |
| 4         | -0.2052   | 129.2     | 0.485     | 0.2835    | 74.15     | 0.3228    | 4.605     | 2.276     | 949.9     | 0.0009195 | 815.7     | 0.03408   | 0.9434    |
| 5         | -0.2052   | 190.9     | 0.8066    | 0.3369    | 96.84     | 0.417     | 3.87      | 1.066     | 86.27     | 6.256e-05 | 136.7     | 0.03338   | 0.2264    |
| 6         | -0.2031   | 116.1     | 0.1786    | 0.1255    | 19.21     | 0.683     | 5.221     | 4.073     | 685.4     | 0.0004469 | 599.3     | 0.4617    | 0.9633    |
| 7         | -0.2053   | 43.2      | 0.9068    | 0.3485    | 61.29     | 0.6847    | 2.597     | 4.459     | 508.0     | 7.865e-05 | 371.1     | 0.3615    | 0.4721    |
| 8         | -0.2069   | 49.03     | 0.1753    | 0.6644    | 67.29     | 0.135     | 9.35      | 4.367     | 100.1     | 0.0002116 | 873.8     | 0.6738    | 0.2973    |
| 9         | -0.2055   | 151.1     | 0.6183    | 0.1473    | 81.2      | 0.05911   | 3.774     | 1.829     | 973.6     | 0.0009103 | 679.7     | 0.1152    | 0.958     |
| 10        | -0.2055   | 32.88     | 0.6298    | 0.632     | 37.28     | 0.6528    | 9.326     | 2.496     | 782.8     | 0.0008972 | 624.0     | 0.1511    | 0.3749    |
| 11        | -0.1934   | 133.0     | 0.4591    | 0.3726    | 38.47     | 0.6599    | 5.289     | 2.818     | 168.6     | 0.0004183 | 280.4     | 0.3467    | 0.9744    |
| 12        | -0.2052   | 55.67     | 0.6828    | 0.02227   | 84.36     | 0.7017    | 9.693     | 3.367     | 256.5     | 0.000722  | 40.56     | 0.4187    | 0.2812    |
| 13        | -0.2011   | 35.55     | 0.4263    | 0.5884    | 15.49     | 0.9867    | 3.1       | 3.886     | 677.8     | 8.698e-05 | 700.3     | 0.2653    | 0.9018    |
| 14        | -0.2002   | 166.6     | 0.3992    | 0.5742    | 23.79     | 0.5225    | 4.712     | 4.703     | 834.1     | 0.000506  | 125.6     | 0.01159   | 0.9132    |
| 15        | -0.2048   | 110.3     | 0.8431    | 0.06639   | 81.33     | 0.06223   | 6.719     | 4.875     | 427.6     | 0.000963  | 222.7     | 0.8899    | 0.9866    |
| 16        | -0.2192   | 151.9     | 0.6415    | 0.1765    | 50.2      | 0.2681    | 8.814     | 2.068     | 974.4     | 0.0002214 | 304.8     | 0.7368    | 0.01289   |
| 17        | -0.2052   | 21.56     | 0.7338    | 0.3488    | 25.72     | 0.00893   | 8.656     | 3.146     | 737.3     | 0.0006649 | 840.4     | 0.3882    | 0.4262    |
| 18        | -0.2054   | 168.4     | 0.469     | 0.574     | 6.347     | 0.3968    | 2.155     | 1.472     | 245.7     | 0.0007502 | 930.8     | 0.3887    | 0.4606    |
| 19        | -0.2052   | 176.7     | 0.8145    | 0.01748   | 87.94     | 0.6046    | 5.648     | 3.357     | 809.8     | 0.0001815 | 195.0     | 0.2582    | 0.05107   |
| 20        | -0.2051   | 117.7     | 0.3518    | 0.6419    | 50.28     | 0.1045    | 2.373     | 2.094     | 792.5     | 0.0003992 | 800.9     | 0.4331    | 0.08155   |
| 21        | -0.205    | 17.96     | 0.4389    | 0.464     | 53.94     | 0.5881    | 5.438     | 3.962     | 799.9     | 0.0007833 | 136.3     | 0.9481    | 0.8777    |
| 22        | -0.2056   | 144.7     | 0.9369    | 0.4717    | 61.47     | 0.345     | 1.752     | 3.317     | 66.73     | 0.0005569 | 636.3     | 0.6784    | 0.1705    |
| 23        | -0.2098   | 122.2     | 0.206     | 0.176     | 97.88     | 0.8966    | 2.186     | 4.525     | 950.6     | 0.0004653 | 418.8     | 0.5771    | 0.3835    |
| 24        | -0.205    | 46.22     | 0.6487    | 0.1743    | 67.6      | 0.9004    | 9.168     | 3.917     | 216.2     | 0.0004731 | 261.7     | 0.7372    | 0.3716    |
| 25        | -0.2159   | 98.64     | 0.937     | 0.5691    | 7.874     | 0.2564    | 6.214     | 2.534     | 789.0     | 0.0005343 | 124.9     | 0.7644    | 0.1923    |
| 26        | -0.2034   | 154.7     | 0.08205   | 0.3827    | 24.16     | 0.2303    | 1.848     | 3.31      | 831.5     | 0.0006557 | 131.5     | 0.5883    | 0.7677    |
| 27        | -0.2052   | 119.8     | 0.7992    | 0.2652    | 37.37     | 0.6987    | 2.152     | 2.802     | 152.0     | 0.0004635 | 273.9     | 0.8022    | 0.8578    |
| 28        | -0.2038   | 172.0     | 0.5143    | 0.06584   | 21.86     | 0.4262    | 1.188     | 1.037     | 832.4     | 0.0007971 | 134.8     | 0.7588    | 0.5925    |
| 29        | -0.1943   | 41.08     | 0.886     | 0.09074   | 60.97     | 0.8721    | 3.092     | 2.571     | 515.0     | 0.0001352 | 376.0     | 0.5148    | 0.9102    |
| 30        | -0.2088   | 130.8     | 0.8384    | 0.1269    | 50.75     | 0.05788   | 7.227     | 2.481     | 304.1     | 0.000662  | 898.2     | 0.8236    | 0.2404    |
| 31        | -0.2052   | 118.8     | 0.1869    | 0.3097    | 49.88     | 0.243     | 3.244     | 1.079     | 798.7     | 0.0002838 | 791.7     | 0.4079    | 0.1645    |
| 32        | -0.2052   | 128.5     | 0.9325    | 0.5419    | 76.6      | 0.4107    | 7.663     | 2.865     | 946.3     | 0.0004433 | 807.0     | 0.1824    | 0.1674    |
| 33        | -0.2101   | 159.5     | 0.1931    | 0.1422    | 20.03     | 0.1711    | 4.423     | 1.907     | 838.2     | 0.0002275 | 123.0     | 0.5565    | 0.1597    |
| 34        | -0.2052   | 145.3     | 0.3107    | 0.1718    | 81.87     | 0.1109    | 6.758     | 1.183     | 159.8     | 0.0001364 | 523.6     | 0.2729    | 0.1115    |
| 35        | -0.2157   | 74.44     | 0.1604    | 0.1114    | 38.93     | 0.3262    | 9.666     | 1.075     | 944.6     | 0.0009759 | 971.6     | 0.9655    | 0.02574   |
| 36        | -0.2051   | 149.5     | 0.2775    | 0.2943    | 83.56     | 0.2424    | 3.177     | 3.022     | 158.3     | 0.000692  | 521.8     | 0.05836   | 0.1613    |
| 37        | -0.2053   | 138.7     | 0.7609    | 0.6533    | 39.68     | 0.5419    | 3.639     | 3.843     | 163.4     | 0.0007332 | 282.7     | 0.5793    | 0.9492    |
| 38        | -0.2051   | 130.8     | 0.8212    | 0.001977  | 42.27     | 0.09744   | 1.511     | 3.35      | 163.0     | 0.0006326 | 277.3     | 0.3648    | 0.2197    |
| 39        | -0.2052   | 111.6     | 0.7671    | 0.4605    | 44.91     | 0.9397    | 1.702     | 3.706     | 781.3     | 0.0005588 | 806.3     | 0.9437    | 0.7518    |
| 40        | -0.2037   | 128.2     | 0.3847    | 0.5257    | 69.02     | 0.6779    | 5.621     | 1.288     | 953.2     | 0.0005022 | 806.2     | 0.00872   | 0.8663    |
| 41        | -0.2053   | 111.9     | 0.9836    | 0.3309    | 48.32     | 0.6323    | 1.871     | 3.228     | 790.8     | 0.0002256 | 804.0     | 0.8743    | 0.7616    |
| 42        | -0.2133   | 129.0     | 0.1264    | 0.2509    | 64.38     | 0.1498    | 5.897     | 3.133     | 941.4     | 0.0006582 | 800.9     | 0.6984    | 0.07996   |
| 43        | -0.2052   | 153.3     | 0.8634    | 0.4142    | 81.2      | 0.4016    | 6.724     | 1.687     | 968.8     | 0.0001565 | 675.0     | 0.4249    | 0.4255    |
| 44        | -0.2045   | 17.39     | 0.7927    | 0.4297    | 26.16     | 0.9341    | 1.03      | 1.167     | 729.0     | 0.0004237 | 843.1     | 0.434     | 0.09536   |
| 45        | -0.2052   | 197.9     | 0.7766    | 0.2802    | 91.48     | 0.8708    | 9.279     | 3.468     | 80.86     | 0.0001555 | 137.0     | 0.8515    | 0.9822    |
| 46        | -0.2104   | 113.8     | 0.06132   | 0.3743    | 88.23     | 0.4206    | 8.449     | 1.598     | 424.7     | 0.0007945 | 221.1     | 0.9385    | 0.2859    |
```

-Figura 13:


lstm_10_input | input: | [(None, 10, 58)]
InputLayer | output: | [(None, 10, 58)]
lstm_10 | input: | (None, 10, 58)
LSTM | output: | (None, 10, 280)
lstm_11 | input: | (None, 10, 280)
LSTM | output: | (None, 10, 280)
dropout_14 | input: | (None, 10, 280)
Dropout | output: | (None, 10, 280)
lstm_12 | input: | (None, 10, 280)
LSTM | output: | (None, 10, 280)
dropout_15 | input: | (None, 10, 280)
Dropout | output: | (None, 10, 280)
lstm_13 | input: | (None, 10, 280)
LSTM | output: | (None, 10, 280)
dropout_16 | input: | (None, 10, 280)
Dropout | output: | (None, 10, 280)
lstm_14 | input: | (None, 10, 280)
LSTM | output: | (None, 10, 280)
dropout_17 | input: | (None, 10, 280)
Dropout | output: | (None, 10, 280)
flatten_2 | input: | (None, 10, 280)
Flatten | output: | (None, 2800)
dropout_18 | input: | (None, 2800)
Dropout | output: | (None, 2800)
dense_8 | input: | (None, 2800)
Dense | output: | (None, 168)
dropout_19 | input: | (None, 168)
Dropout | output: | (None, 168)
dense_9 | input: | (None, 168)
Dense | output: | (None, 84)
dropout_20 | input: | (None, 84)
Dropout | output: | (None, 84)
dense_10 | input: | (None, 84)
Dense | output: | (None, 42)
dense_11 | input: | (None, 42)
Dense | output: | (None, 1)

-Figura 14:


Training and Validation Loss
Training Loss
Validation Loss
0.23
0.22
0.21
0.20
0.19
0.18
Loss
0
20
40
60
Epochs