

# Trabajo Fin de Grado

## Aplicación de redes neuronales a la predicción de quiebra financiera

Autor/es

Frank M. Disla del Orbe

Director/es

Emilio Martín Vallespín

Facultad / Escuela

Facultad de Economía y Empresa de la Universidad de Zaragoza

Año

Agosto 2025

**Título del trabajo:** Aplicación de redes neuronales a la predicción de quiebra financiera  
/ Application of neural networks to financial bankruptcy prediction

**Titulación:** Grado de Economía

**Autor del trabajo:** Frank Martin Disla del Orbe

**Director del trabajo:** Emilio Martín Vallespín

**Resumen:**

Este trabajo analiza la aplicación de modelos de redes neuronales profundas, a la predicción de quiebra empresarial a partir de ratios financieros. Se construyó una base de datos con información de 684 empresas que cotizan en la Bolsa de Nueva York, de las cuales 75 atravesaron procesos de bancarrota entre 2000 y 2024. A partir de los estados financieros disponibles en la base de datos Refinitiv Eikon, se calcularon 34 ratios, aunque tras un riguroso proceso de preprocesamiento, limpieza y transformación estadística se utilizaron finalmente 22.

El modelo principal es una red neuronal con una capa oculta de 22 neuronas y función de activación ReLU. Los resultados muestran una exactitud aproximada del 90% en entrenamiento, validación y prueba, destacando una elevada capacidad de detección de quiebras (recuperación del 81%), aunque con una precisión más limitada (alrededor del 40–68% según la partición). Esto implica que el modelo identifica la mayoría de los casos de bancarrota, pero genera un número considerable de falsas alarmas.

Se concluye que este tipo de modelos no sustituyen al analista humano, pero constituyen una herramienta complementaria valiosa para filtrar empresas y priorizar análisis financieros en contextos de riesgo.

**Abstract:**

This paper analyzes the application of deep neural network models to predict corporate bankruptcy from financial ratios. A dataset was constructed with information from 684 companies listed on the New York Stock Exchange, 75 of which underwent bankruptcy proceedings between 2000 and 2024. From the financial statements available in the Refinitiv Eikon database, 34 ratios were calculated, although after a rigorous preprocessing, cleaning, and statistical transformation process, 22 were finally used.

The main model is a neural network with a 22-neuron hidden layer and a ReLU activation function. The results show an accuracy of approximately 90% in training, validation, and testing, highlighting a high bankruptcy detection capacity (81% recall), although with more limited precision (around 40–68% depending on the partition). This implies that the model identifies most bankruptcy cases, but generates a considerable number of false alarms. It is concluded that these types of models do not replace human analysts, but constitute a valuable complementary tool for screening companies and prioritizing financial analysis in risk contexts.

## Índice

1. Introducción y objetivos:	4
2. Precedentes:	5
3. Desarrollo de la metodología empleada:	6
3.1. Conjunto de datos:	6
3.1.1. Ratios financieros utilizados:	8
3.2. Modelo de inteligencia artificial:	13
3.2.1. Selección de modelo	14
3.3. Redes neuronales:	15
3.3.1. Propagación hacia atrás	16
3.4. Pre procesamiento de datos:	20
3.4.1. Procesamiento en todo el conjunto de datos:	20
3.4.2. División en conjunto de entrenamiento, validación y prueba:	22
3.4.3. Eliminación de ratios demasiado sesgados, extremos o constantes:	24
3.4.4. Winsorización para reducir la asimetría y las colas:	25
3.4.5. Auto Yeo-Johnson para columnas explosivas:	25
3.4.6. RobustScaler:	26
3.4.7. Pesos de clase:	28
4. Resultados:	28
4.1. Planteamiento del modelo a utilizar:	28
4.2. Entrenamiento:	30
5. Conclusiones:	35
6. Bibliografía:	37
7. Anexos	38
7.1. Apendice.1: Enumeración de los ratios antes del escalado:	38
7.2. Apendice.2: Enumeración de los ratios utilizados en el entrenamiento:	39

## 1. Introducción y objetivos:

El objetivo de este trabajo es analizar el desempeño del aprendizaje automático, así como analizar el proceso de diseño y fundamentos de dichos modelos. Con ello se pretende profundizar en las relaciones entre la información contable de una empresa, evaluada a lo largo del tiempo mediante ratios financieros, y su salud financiera, en particular para determinar si la empresa podría entrar o no en bancarrota. En la disciplina de *machine learning*, este tipo de problemas se aborda mediante modelos de predicción de clasificación binaria que pueden responder a la pregunta de si la empresa entrará en bancarrota dentro de dos años, con una respuesta de ‘sí’ o ‘no’; de ahí el término binaria, esto puede proveer de una herramienta útil a la hora del análisis contable.

La contribución de este trabajo es aportar más ejemplos prácticos del uso de modelos de inteligencia artificial en el ámbito económico-académico. Otra razón que motivó esta investigación es la estrecha conexión entre el análisis financiero y los requisitos de la mayoría de los modelos de inteligencia artificial. Esto se debe principalmente a que dichos modelos requieren grandes volúmenes de datos. Si no estuviéramos en la era de la información, los algoritmos de inteligencia artificial difícilmente habrían alcanzado el auge que tienen hoy en día, ya que su desempeño depende en gran medida de la cantidad y calidad de los datos disponibles (Djerdjouri, 2020). En este sentido, existe una relación natural con el análisis financiero, pues gracias a las regulaciones de transparencia, las empresas que cotizan en bolsa están obligadas a publicar periódicamente sus estados financieros. Estas regulaciones, establecidas en la mayoría de países desarrollados, exigen que las compañías presenten reportes constantes de su situación financiera para ofrecer a los inversionistas una visión clara de su salud económica, siendo esto una piedra angular en la transparencia y la confianza en la economía global (Joshi, P. & India, 2023). Como resultado, al contar con una gran cantidad de empresas que cotizan en bolsa y con la obligación de publicar información financiera cada año, se genera un enorme volumen de datos que puede aprovecharse para alimentar modelos de inteligencia artificial.

Los principales estados financieros que proporciona la información a utilizar son, el balance general, el estado de resultados y el estado de flujos de efectivo (el resto de estados financieros son el estado de cambios en el patrimonio neto y la memoria). A partir de estos documentos es posible calcular ratios financieros, que se utilizan para evaluar la

situación financiera de una empresa. Por ello, en este trabajo se emplearán como variables de entrada en los modelos finales de inteligencia artificial, y también estudiaremos lo que ocurre dentro de dichos modelos.

## 2. Precedentes:

Como se mencionó anteriormente, en los últimos años se ha dado un auge en los modelos de inteligencia artificial, y por supuesto esto también ha llegado al sector de los reportes financieros, dejándonos algunos precedentes que sirven como cimientos para el presente trabajo.

Uno de ellos es el estudio *Natural Language Processing and Text Mining Algorithms for Financial Accounting Information Disclosure*, de Huanhuan (2024). En dicho trabajo se exploran modelos de inteligencia artificial aplicados al procesamiento de lenguaje natural. Como base de datos se utilizan informes anuales, presentaciones regulatorias, transcripciones de resultados, artículos periodísticos y publicaciones en redes sociales, los cuales sirven para entrenar al modelo con el fin de identificar los sentimientos generales que existen en el mercado sobre la empresa en cuestión, constituyendo una poderosa herramienta para el análisis cuantitativo, más allá del análisis de los estados financieros que realizaremos en este trabajo.

Los modelos de inteligencia artificial también han encontrado aplicación en el ámbito de la seguridad, particularmente en la detección de fraudes. En los últimos años han surgido numerosos modelos de predicción de fraude financiero, y uno de los más completos es *Detection of Financial Statement Fraud Using Deep Learning for Sustainable Development of Capital Markets under Information Asymmetry*, de Chia-Hsuan Jan (2021). Este trabajo propone el uso de modelos de IA para hacer frente a los problemas de asimetría de información en los estados financieros publicados por las empresas.

El trabajo de Chia-Hsuan Jan no ha sido el único que se ha enfocado en tareas de predicción, lo cual era de esperarse, ya que la capacidad predictiva de los modelos de IA es uno de sus grandes fortalezas. Esa capacidad puede aplicarse ampliamente en contabilidad, como se demuestra en *Neural Networks in Accounting: Bridging Financial Forecasting and Decision Support Systems*, de Alin Emanuel Artene y Aura Emanuela Domil (2025). En este artículo se propone un marco teórico y empírico para aplicar redes neuronales a tareas de predicción, específicamente orientadas a estimar la rentabilidad corporativa de una empresa a partir de variables contables. El objetivo es mostrar que

estas técnicas predictivas pueden mejorar la toma de decisiones estratégicas en el entorno financiero actual.

El uso de la IA en el análisis de información financiera ofrece oportunidades significativas, automatiza la extracción de conocimiento de grandes volúmenes de texto, mejora la detección de anomalías y fraudes, y potencia modelos predictivos para apoyar decisiones estratégicas. No obstante, existen limitaciones relevantes, como la calidad y sesgo de los datos, la opacidad de algunos modelos, el riesgo de sobreajuste y la dependencia tecnológica, sin contar los retos legales y de gobernanza que exigen supervisión experta.

### 3. Desarrollo de la metodología empleada:

#### 3.1. Base de datos:

El conjunto final de datos está compuesto por 684 empresas estadounidenses, de las cuales solo 75 han entrado en proceso de bancarrota. Para cada una se calcularon 34 de los ratios más utilizados en el análisis financiero, aunque no todos estos ratios serán utilizados en la fase de entrenamiento (ver apartado 3.4). Inicialmente, el conjunto de datos incluía alrededor de 1498 empresas que forman parte del índice compuesto de la Bolsa de Nueva York (NYSE). Sin embargo, tras aplicar un filtro, se eliminaron aquellas compañías para las que no se disponía de información suficiente en la mayoría de los ratios o en la cantidad de tiempo analizada.

Es importante aclarar el “horizonte temporal” analizado. Los ratios de las empresas que estudiamos no están todos en el mismo periodo de tiempo; es decir, no corresponden necesariamente a los años 2015-2024, sino que cada empresa aporta 5 años de ratios, pero estos pueden ubicarse en distintos momentos dentro del intervalo 2000-2024. Por ejemplo, una empresa puede tener disponibles los ratios de 2020-2024, mientras que otra puede aportar los de 2000-2004 o 2009-2013.

La selección de estos periodos de 5 años no es aleatoria, se priorizan aquellos en los que la empresa haya atravesado un proceso de bancarrota. Así, si una compañía mantuvo buena salud financiera entre 2000 y 2020, pero experimentó bancarrota entre 2020 y 2024, se tomarían precisamente esos últimos 5 años para entrenar nuestro modelo de inteligencia artificial.

La razón principal de trabajar de esta forma es la disponibilidad de datos. Lo ideal sería contar con 20 años completos de ratios para un gran número de empresas, pero eso es extremadamente costoso y difícil de conseguir, además, aun si se lograra, no necesariamente se obtendría un número suficiente de empresas que hayan pasado por bancarota. Por eso, no utilizamos un rango fijo de 5 años (por ejemplo, 2019-2024), ya que de esa manera el grupo de empresas con quiebra quedaría muy limitado: se excluirían aquellas que, aunque se mantuvieron estables entre 2020 y 2024, sí experimentaron crisis entre 2000 y 2020.

Finalmente, construir el conjunto de datos con periodos variados dentro de 2000-2024 también favorece la generalización del modelo. Si solo se tomaran ratios de 2020-2024, todas las empresas habrían pasado por la crisis de la pandemia de COVID-19, lo que introduciría un patrón muy marcado en los datos. El modelo podría aprender a identificar ese efecto específico, pero tendría dificultades para generalizar en contextos distintos, como al evaluar la salud financiera de empresas en periodos futuros (por ejemplo, 2025-2029), donde los ratios no estarían condicionados por la pandemia.

La razón por la que en este trabajo se utilizan exclusivamente empresas listadas en la NYSE es, en primer lugar, que el formato de índice compuesto resulta muy útil para la construcción del conjunto de datos, ya que en la mayoría de las bases de datos es posible descargar todas las empresas que conforman un índice, en lugar de buscar manualmente las compañías que puedan servir como muestra. En segundo lugar, a diferencia de otros índices como el NASDAQ-100 o el S&P 500 (Standard & Poor's 500), cuyo número de empresas es considerablemente menor, la NYSE incluye una cantidad mucho más amplia. Además, mientras que índices como el S&P 500 tienden a estar compuestos por empresas grandes y relativamente estables, la NYSE ofrece una representación más diversa de la economía estadounidense, al incluir tanto compañías con una gran solidez financiera como un número significativo de empresas que han atravesado procesos de bancarota. Esto permite obtener una muestra más equilibrada y representativa para los fines de este estudio (New York Stock Exchange [NYSE], 2025).

En la Tabla 1 se muestra un ejemplo de la estructura del conjunto de datos utilizado en este trabajo. Para cada empresa se cuenta con 25 años de ratios financieros de los cuales se tomarán 5 siguiendo el proceso antes explicado, y en cada uno de esos años se asigna un valor a la variable dependiente u objetivo, denominada *bankruptcy\_dummy*. Esta variable ficticia toma el valor de 1 si la empresa entra en bancarota en cualquiera de los

dos años siguientes, y 0 en caso contrario. Cabe precisar que, en este trabajo, consideramos como bancarrota la situación en la que el patrimonio de una empresa es negativo ( $A - P < 0$ ).

Tabla.1

Ticker	Year	...	Ratio de capacidad de devolucion de los prestamos	Ratio de gasto financiero / ventas	Ratio de coste deuda	...	bankruptcy_dummy
AMN.N	2000	...	-0.012531495	0.032488363	0.038559682	...	0
AMN.N	2001	...	0.005387014	0.04335994	0.26106922	...	0
AMN.N	2002	...	2.836883749	0.042870331	0.356108125	...	0
AMN.N	2003	...	17.91265306	-0.000442191	-0.004881068	...	1
AMN.N	2004	...	0.138574514	0.003224546	0.044271009	...	1
AMN.N	2005	...	0.12089558	0.013417783	0.049084036	...	0
AMN.N	2006	...	0.076995122	0.013551172	0.066759961	...	0
AMN.N	2007	...	0.131870704	0.01543677	0.094883626	...	0
AMN.N	2008	...	0.160932992	0.145306532	1.118118356	...	0
AMN.N	2009	...	0.184738787	0.008782452	0.084643433	...	0
AMN.N	2010	...	-0.615199629	0.261824978	1.177251746	...	0
AMN.N	2011	...	-0.122660838	0.10537802	0.453311672	...	0
AMN.N	2012	...	0.104120716	0.026735672	0.125763595	...	0
AMN.N	2013	...	0.192593155	0.027274986	0.146692112	...	0
AMN.N	2014	...	0.312131896	0.009552132	0.064140851	...	0
AMN.N	2015	...	0.343669251	0.008915791	0.054734696	...	0
AMN.N	2016	...	0.545261752	0.005324439	0.030782484	...	0
AMN.N	2017	...	0.373222168	0.008129953	0.060831103	...	0
AMN.N	2018	...	0.471628268	0.009895627	0.048975234	...	0
AMN.N	2019	...	0.415290724	0.007557322	0.034456703	...	0
AMN.N	2020	...	0.27951954	0.012792813	0.043624583	...	0
AMN.N	2021	...	0.19109974	0.047182746	0.154770086	...	0
AMN.N	2022	...	0.511781718	0.01620813	0.075698911	...	0
AMN.N	2023	...	0.688983468	0.007704775	0.037631869	...	0
AMN.N	2024	...	0.284845879	0.014287773	0.064184563	...	0
...	...	...	...	...	...	...	...

### 3.1.1. Ratios financieros utilizados:

Antes de la introducción de los ratios financieros utilizados como variables de entrada para los modelos, es necesario aclarar que éstos han sido escogidos, en su mayoría, tomando como referencia las indicaciones del trabajo de Amat (2013). Para la construcción de dichos ratios se emplearon los campos financieros disponibles en la base de datos Eikon Refinitiv, cuya información fue descargada entre mayo y julio de 2025.

Los ratios de endeudamiento reflejan la cantidad y la calidad de la deuda que posee una empresa. Estos indicadores son fundamentales para el análisis financiero y la predicción de bancarrota, ya que un exceso de endeudamiento puede implicar una insuficiencia de fondos propios y, en consecuencia, una menor autonomía frente a terceros, aunque este efecto puede variar según el sector. Los ratios de endeudamiento que utilizaremos en este trabajo son:

1. Ratio de endeudamiento:
  - Que se calcula de la siguiente manera: *Deudas totales / Total activo*
2. Ratio de calidad de la deuda:
  - *Deudas a corto plazo / Deudas totales*



2. Ratio de capacidad de devolución de los préstamos:
  - $(\text{Beneficio neto} + \text{Amortizaciones}) / \text{prestamos recibidos}$
3. Ratio de gasto financiero / ventas:
  - $(\text{BAII} - \text{BAI}) / \text{ventas}$
4. Ratio de coste deuda:
  - $(\text{BAII} - \text{BAI}) / \text{saldo medio deuda con coste}$

Los ratios de liquidez o de tesorería reflejan la relación entre los flujos de efectivo, procedentes de actividades operativas, de inversión y de financiación, y la posición financiera de la empresa, lo cual permite identificar variaciones en los balances a lo largo del tiempo, un modelo de inteligencia artificial podría detectar patrones en la evolución de dichos indicadores y su vínculo con los fondos propios, los cuales se ven afectados tanto por la tesorería generada en la actividad ordinaria como por las decisiones de inversión y financiación (Antwi et al., 2024), aunque se es consciente de su posible limitada aportación como ratio calculado, puesto que en el estado de flujos de efectivo puede encontrarse información más precisa sin necesidad de hacer ningún cálculo. Los ratios de liquidez que utilizaremos serán:

1. Flujo de caja económico:
  - $\text{Beneficio neto} + \text{Amortizaciones} + \text{Provisiones}$
2. Flujo de caja libre:
  - $\text{Beneficio neto} + \text{Amortizaciones} + \text{Provisiones} - \text{Inversiones en activos circulantes} - \text{Inversiones en activos inmovilizados}$
3. Ratio de liquidez:
  - $\text{Activo circulante} / \text{Exigible a CP}$
4. Ratio de tesorería:
  - $(\text{Realizable} + \text{Disponible}) / \text{Exigible a CP}$
5. Ratio de disponibilidad:
  - $(\text{Disponible} + \text{Inversiones financieras CP}) / \text{Exigible a CP}$

Los ratios de rotación miden el rendimiento y la eficiencia con que una empresa utiliza sus activos para desarrollar su actividad. Estos ratios puede considerarse complementarios a los ratios de endeudamiento, ya que un modelo de inteligencia artificial podría detectar correlaciones entre las variaciones en las inversiones en el activo y, en consecuencia, las variaciones en el pasivo y las deudas, lo que se relaciona con la

salud financiera de la empresa a lo largo del tiempo. En términos generales, un aumento en la inversión en activos suele acompañarse de un incremento en el pasivo; sin embargo, si una empresa aumenta sus activos durante varios años sin una mejora proporcional en sus ingresos, ello podría indicar una pérdida de eficiencia en el uso de sus recursos. Además, el valor de los ratios de rotación se ve reforzado en nuestro conjunto de datos, ya que contamos con su evolución a lo largo de varios años, lo cual permite que la inteligencia artificial detecte patrones y tendencias en su desempeño (Berry et al, 1987). Los ratios de rotación que utilizaremos en esta categoría son:

1. Rotacion del activo fijo:
  - $Ventas / Activo\ fijo$
2. Rotacion del activo circulante:
  - $Ventas / Activo\ circulante$
3. Rotacion de las existencias:
  - $Coste\ de\ ventas / Existencias$

Tanto los plazos de cobro a clientes como los plazos de pago a proveedores, junto con el tiempo de permanencia de las existencias en almacén, aportan información valiosa para complementar otros ratios clave en la predicción de una posible quiebra, como los ratios de liquidez. En general, un menor plazo de cobro a clientes y mayores plazos de pago a proveedores aumentan la liquidez disponible para la empresa. No obstante, estos indicadores dependen del sector y no siempre constituyen un signo de buena salud financiera (Amat, 2013). Por otro lado, en lo que respecta al plazo de existencias, plazos más cortos suelen reflejar una mejor gestión de inventarios, ya que implican menores costos de almacenamiento o mayores ventas, lo que a su vez se relaciona con los ratios de ventas que analizaremos más adelante. Este tipo de relaciones puede ser detectado por la inteligencia artificial, contribuyendo a mejorar la exactitud de las predicciones. Los ratios que utilizaremos en esta categoría son:

1. Ratio de plazo de cobro:
  - $(Realizable / Ventas) * 365$
2. Ratio de plazo de pago:
  - $(Saldo\ de\ proveedores / Coste\ de\ ventas) * 365$
3. Plazo de almacenamiento de las existencias:
  - $(Existencias / Coste\ de\ ventas) * 365$

La variación en las ventas a lo largo de los años puede reflejar patrones que resultan difíciles de identificar a simple vista. Aunque las ventas tienen una gran importancia individual, no constituyen la única variable relevante (Amat, 2013); por ello, es fundamental que la inteligencia artificial las considere en conjunto con otros ratios, como los de rotación, o plazo, para construir una visión más completa de los factores que pueden llevar a una empresa a la bancarrota o, por el contrario, a mantenerse estable. Los ratios de ventas que tendremos en cuenta son:

1. Ratio de expansión de ventas:
  - $Ventas\ año\ n / Ventas\ año\ n-1$
2. Ratio de ventas por empleado:
  - $Ventas / Numero\ medio\ empleados$

Complementarios a los ratios de ventas están los ratios de gasto, que ponen en perspectiva los gastos en relación con el volumen de ventas. En definitiva, estos indicadores pueden ayudar a identificar patrones perjudiciales para la empresa, como un aumento sostenido de los ratios que relacionan gastos y ventas a lo largo de los años, lo cual podría poner en riesgo la salud financiera de la empresa:

1. Ratio de gasto fijo / ventas:
  - $Gastos\ fijos / Ventas$
2. Ratio de gasto variable / ventas:
  - $Coste\ de\ ventas / ventas$
3. Umbral de rentabilidad:
  - $Gastos\ fijos / (1 - (Coste\ de\ ventas / Ventas))$
4. Ratio de gastos de publicidad:
  - $Gasto\ publicidad / Ventas$

Estos ratios ponen en relación los beneficios con los capitales propios. Dado que los capitales propios afectan directamente a nuestra variable objetivo, el patrimonio, los siguientes ratios son de gran importantes para que nuestro modelo aprenda a predecir quiebras empresariales:

1. Ratio de rentabilidad:
  - $BAIL / Activo\ total$
2. Ratio de rentabilidad financiera:
  - $Beneficio\ neto / fondo\ de\ maniobra$

3. Apalancamiento financiero:

- $(BAII / \text{Activo total}) / (\text{Gasto financiero} / \text{Deuda total})$

4. Valor agregado:

- $BAII - (\text{Activo total} * \text{Coste medio ponderado pasivo})$

En general, la capacidad de autofinanciación mediante los recursos generados por el flujo de caja e invertidos en la propia empresa es un buen indicador de salud financiera. Esta capacidad está determinada, entre otras cosas, por la política de distribución de beneficios de la empresa. Por ello, incluir los ratios de autofinanciación en el entrenamiento de nuestra IA puede, de manera indirecta, ayudar al modelo a detectar patrones en la gestión administrativa de la empresa y a evaluar su grado de dependencia de entidades de crédito.

1. Autofinanciación sobre ventas:

- $(\text{Flujo de caja} - \text{dividendo}) / \text{Ventas}$

2. Autofinanciación sobre activo:

- $(\text{flujo de caja} - \text{dividendo}) / \text{Activo total}$

3. Política de dividendo:

- $\text{Dividendo} / \text{Beneficio neto}$

4. Autofinanciación de inversiones:

- $(\text{Flujo de caja} - \text{Dividendo}) / \text{Inversiones en activos inmovilizados}$

Todas las empresas presentes en la base de datos cotizan en bolsa. Por ello, analizar el desempeño de sus acciones a través de los ratios bursátiles puede contribuir a formar una imagen más detallada de cada empresa, dado que es esperable que las empresas pierden valor de mercado cuando su salud financiera se ve comprometida. Sin embargo, debido a la naturaleza volátil de los mercados financieros, la cotización no siempre refleja el valor real de una compañía. En este contexto, estos ratios bursátiles pueden ser útiles para mejorar la generalización del modelo, agregando robustez al aprendizaje y ayudando a evitar el sobreajuste al conjunto de entrenamiento.

1. Ratio de precio beneficio:

- $\text{Cotización} / \text{Beneficio por acción}$

2. Relación entre valor de mercado y contable:

- $\text{Cotización} / \text{Valor contable de la acción}$

3. Rentabilidad por dividendo:

- $\text{Dividendo por acción} / \text{Cotización}$

### 3.2. Modelo de inteligencia artificial:

Un algoritmo puede definirse como una lista de instrucciones diseñada para resolver una tarea determinada (Aho et al, 1974). Y junto con los datos son utilizados en los modelos de inteligencia artificial para mejorar su desempeño en una tarea específica sin necesidad de una intervención humana; en pocas palabras, aprenden. Dichos modelos son estudiados en el campo del *Machine Learning* (ML), disciplina dedicada al diseño y comprensión de sistemas capaces de aprender, y mejorar a través de la experiencia, imitando en cierta medida la forma en que los seres humanos adquirimos conocimiento: de manera gradual, repitiendo la tarea múltiples veces y perfeccionando el resultado en cada intento (Li et al., 2021).

Dentro de la disciplina de Machine Learning (ML) existen diferentes paradigmas de aprendizaje, entre ellos el aprendizaje supervisado, no supervisado y por refuerzo, entre otros. Dado que en este trabajo nos centraremos exclusivamente en el ML supervisado, no profundizaremos en la definición del resto de paradigmas para no desviarnos del enfoque del estudio. El aprendizaje supervisado se caracteriza porque sus modelos aprenden a partir de un conjunto de datos compuesto por variables de entrada y una variable objetivo (Deisenroth et al., 2024). En términos sencillos, el modelo aprende mediante ejemplos que indican qué debe predecir.

En el conjunto de datos utilizado en este trabajo, los ratios financieros calculados representan las variables de entrada (también llamadas variables independientes), mientras que la variable ficticia `bankruptcy_dummy` constituye la variable objetivo o dependiente. Cada muestra, es decir, los ratios de una empresa en un año, tiene asociado un valor de esta variable, que toma el valor 1 si la empresa entra en bancarrota en los dos años siguientes, y 0 en caso contrario. De este modo, el modelo aprende a identificar cómo los distintos ratios de las empresas a lo largo del tiempo se relacionan con la ocurrencia o no de bancarrota. En esencia, los modelos supervisados de ML encontrarían patrones en los ratios que permitan predecir si una empresa sufrirá una bancarrota en los próximos dos años.

En ML el concepto de aprender difiere de la definición habitual. Se considera que un modelo está aprendiendo a resolver una tarea cuando existe una función matemática subyacente que relaciona las variables de entrada con la salida, y el modelo logra aproximarse a dicha función. Dado que esta función real no es observable, se asume su

existencia y se evalúa la calidad del modelo en función de qué tan bien logra aproximarse a ella. En la práctica, diseñar un modelo de ML implica elegir las características y parámetros que determinan cómo el modelo aprende; estos parámetros configurables reciben el nombre de hiperparámetros.

### 3.2.1. Selección de modelo

A la hora de seleccionar el modelo final, existen diferentes métodos para determinar cuál presenta el mejor desempeño. En este trabajo utilizaremos la técnica de segmentar el conjunto de datos principal en tres subconjuntos y evaluar el rendimiento de los modelos en cada uno de ellos. Para medir dicho desempeño emplearemos la métrica denominada *exactitud* (*accuracy*), la cual calcula el porcentaje de aciertos del modelo sobre el conjunto evaluado. En nuestro caso particular, la exactitud corresponde al número de clasificaciones correctas realizadas por el modelo en el subconjunto que se está tratando.

El primer subconjunto es el de entrenamiento, que debe contener la mayor proporción de datos del conjunto total, ya que es el que el modelo utilizará para aprender. Luego está el conjunto de validación, que, junto con el de entrenamiento, se emplea para seleccionar el modelo con mejor desempeño. Es decir, buscamos un modelo que tenga una alta exactitud en el entrenamiento (aprende bien los datos) y también en la validación (generaliza bien a datos no vistos durante el entrenamiento). Por último, se encuentra el conjunto de prueba o *test*, que sirve como una evaluación final para comprobar cómo se comporta el modelo con información totalmente nueva. Es crucial respetar los roles de cada subconjunto de datos, pues un error común es elegir el modelo basándose en su desempeño en el conjunto de prueba, lo que implica indirectamente haberlo utilizado para la evaluación, y, en consecuencia, no sabríamos con certeza cómo funcionará con datos verdaderamente desconocidos. Asimismo, si se entrena el modelo únicamente sobre el conjunto completo de datos sin reservar validación ni prueba, se corre el riesgo de sobreajuste (*overfitting*), ya que no se estaría evaluando su capacidad de generalizar en información no vista (Arp et al., 2022).

Si diseñamos un modelo con supuestos muy fuertes sobre la distribución de los datos, este podría presentar una elevada varianza, haciéndolo muy sensible a nuevos datos. En tal caso, estaríamos frente a un problema de *sobreajuste*, que ocurre cuando el modelo memoriza el conjunto de entrenamiento en lugar de aprender patrones generales, logrando un error casi nulo en dicho conjunto, pero un error muy alto en validación o prueba. Por el contrario, si diseñamos un modelo con supuestos demasiado débiles, este podría perder

información relevante, generando un error de sesgo elevado y dando lugar a un problema de *infraajuste* (*underfitting*). Estos dos escenarios evidencian la necesidad de encontrar un equilibrio entre el ajuste a los datos de entrenamiento y la capacidad de generalización del modelo (Mitchell, 1997). Dicho equilibrio se logra, en parte, mediante la correcta selección de los hiperparámetros, proceso que suele alcanzarse a través de experimentación y ajuste sistemático.

De la necesidad de encontrar un equilibrio entre ajuste y sesgo surge el *No Free Lunch Theorem* (teorema de no hay almuerzo gratis), el cual establece que ningún algoritmo de aprendizaje es universalmente superior a otro en todos los problemas posibles. Cada algoritmo incorpora un sesgo inductivo particular, lo que explica la existencia de tantos modelos diferentes, ya que su desempeño depende del tipo específico de problema y de los datos utilizados (Wolpert y Macready, 1997).

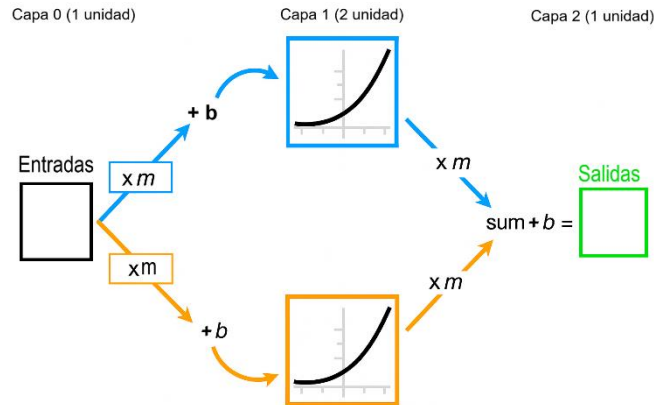
### 3.3. Redes neuronales:

En *Machine Learning* existen muchos modelos diferentes que podemos entrenar y usar para nuestra meta de predicción, pero sin lugar a dudas uno de los más utilizados son las redes neuronales. Estas se inspiran, como su nombre indica, en el funcionamiento de las redes neuronales del cerebro, aunque no son exactamente iguales a su equivalente biológico (Mitchell, 1997). Como se muestra en la figura.1, este tipo de modelos se compone de una primera capa que recibe las entradas provenientes del conjunto de datos. Luego, a cada entrada se le multiplican y suman ciertos parámetros que al inicio son desconocidos, y que el modelo ajusta durante el proceso de entrenamiento. Esta operación sigue la misma idea de la ecuación de una recta en su forma pendiente-intersección:

$$Y = m \times x + b$$

Donde  $Y$  es la salida del modelos, mientras que  $x$  es la variable de entrada. El parámetro  $m$  corresponde a la pendiente, también llamada *peso* en ML, y  $b$  es la intersección, conocida en ML como *sesgo*. Cabe mencionar que, en el contexto de ML, y todas estas variables suelen representarse como vectores (o matrices). En álgebra básica, esta fórmula se piensa con una sola variable  $x$  y un único par  $(m, b)$ . Sin embargo, en red neuronal existen muchos pesos (uno por cada conexión entre neuronas) y un sesgo asociado a cada neurona.

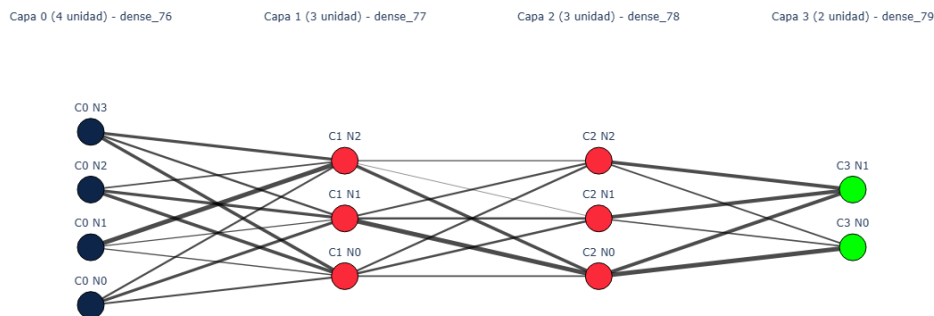
Figura.1



Eventualmente, las entradas llegan a las primeras unidades de procesamiento en la capa 1. Cada una de estas unidades o nodos aplica una función denominada *función de activación*, la cual toma como entrada los valores recibidos y produce una salida. Ese resultado se convierte en la entrada de la siguiente neurona, que puede ser la final (como en la figura 1) o bien alimentar a otras unidades de procesamiento (como en la figura 2).

Las unidades de la última capa, en el caso de una tarea de clasificación, aplican una función especial que convierte los resultados en probabilidades. Estas probabilidades se interpretan como la predicción sobre a qué clase pertenece la entrada recibida.

Figura.2



### 3.3.1. Propagación hacia atrás

Todo lo descrito hasta ahora se denomina *propagación hacia delante* (*forward propagation*). Este es el proceso mediante el cual los datos recorren el modelo desde la entrada hasta convertirse en probabilidades de clase. Dichas probabilidades se obtienen introduciendo las salidas de la penúltima capa (resultados de las neuronas anteriores) en una función llamada *Softmax*. Esta función transforma la salidas de la combinación lineal



$Y = m \times x + b$  en un rango entre  $[0,1]$ , lo que nos permite interpretarlas como probabilidades (Cybenko, 1989).

$$\text{Softmax}(Y)_i = \frac{e^{Y_i}}{\sum_j^n e^{Y_j}}$$

Ahora, esa clasificación se compara con el valor de la variable dependiente y se calcula el error entre el resultado de la última unidad y dicho valor de la variable objetivo. Esto se hace a través de la función de coste denominada *función de error de entropía cruzada binaria*. Para su cálculo, primero asumimos la verosimilitud, es decir, la probabilidad de observar el valor del objetivo (en nuestro caso, la variable ficticia de bancarrota) dado el resultado de nuestro modelo. Suponemos que sigue una distribución binomial, ya que queremos un clasificador con solo dos clases: bancarrota o no bancarrota.

Estadísticamente, esto se traduce en que la probabilidad de que la variable objetivo (que empezaremos a denotar como  $t$ , de *target*) sea 1 es igual al resultado de nuestro modelo (que ya denotábamos como  $Y$ ):

$$P(t = 1) = Y$$

Mientras que la probabilidad de que  $t$  sea 0 corresponde a la clase opuesta, es decir:

$$P(t = 0) = 1 - Y$$

Esto es posible porque las salidas del modelo son probabilidades. La forma generalizada de esta función sería:

$$P(t = t) = Y^t(1 - Y)^{1-t}$$

Y para obtener nuestra función de coste, la *entropía cruzada binaria*, debemos asumir que tanto las variables de entrada como la variable objetivo están independiente e idénticamente distribuidas (i.i.d.). Bajo esta suposición, podemos introducir en la función de máxima verosimilitud la expresión para la probabilidad:  $P(t = t) = Y^t(1 - Y)^{1-t}$ , la función de verosimilitud se calcula entonces como el producto (o, equivalentemente, la suma de logaritmos) de estas probabilidades para todos los datos (MacKay, 2003), y todo esto para encontrar el valor de los pesos ( $m$ ) que maximiza dicha verosimilitud:

$$m_{MV} = \arg_m \max \sum_{i=1}^n \log P(t^i | Y^i, m)$$

Así después de aplicar un poco las reglas de los logaritmos obtendríamos la *Función de error de entropía cruzada binaria* tenemos:

$$m_{MV} = \arg_m \max \sum_{i=1}^n t^i \times \log Y^i + (1 - t^i) \times \log(1 - Y^i)$$

La interpretación matemática de esta función (que denominaremos  $J(m)$ ) es que cuando el  $t$  resulta ser igual a 1 se maximiza el valor de salida del modelo ( $Y$ ) para hacer que se acerque lo más posible a 1, por otro lado, cuando  $t$  es igual a 0 se maximiza  $1 - Y$  (que no es más que minimizar  $Y$ ) para que se acerque lo más posible a 0 el cual es el valor objetivo.

Entonces, la propagación hacia atrás consiste en calcular el error ( $E$ ) de las unidades de la última capa (en la figura 1, la capa 3). Ese error se utiliza luego para calcular el error de las unidades de la capa anterior, el cual está ponderado por los pesos obtenidos en la función de error previamente calculada. Este procedimiento se repite sucesivamente hasta llegar a la primera capa de procesamiento.

Dado que el cálculo del error comienza desde la salida y avanza hacia la entrada, este proceso recibe el nombre de propagación hacia atrás. La idea central es que cada unidad contribuye parcialmente al error total, y esa contribución se mide mediante el peso calculado a través del descenso de gradiente aplicado a la función de error (Rumelhart et al., 1986). Dicho gradiente no es más que la derivada parcial de la función de error respecto al peso de cada neurona:

$$\Delta J(m) = \left[ \frac{\partial J}{\partial m_0}, \frac{\partial J}{\partial m_1}, \dots, \frac{\partial J}{\partial m_n} \right]$$

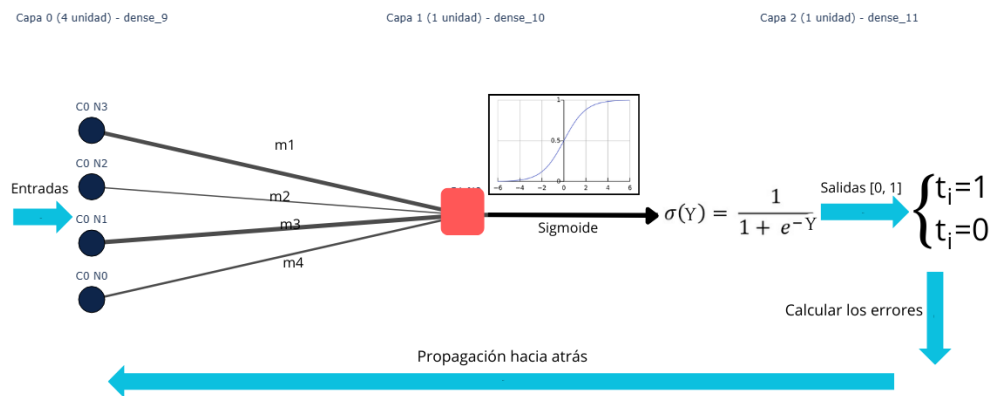
Dichos pesos están inicializados con valores aleatorios, y serán actualizados a lo largo de la dirección opuesta a la del gradiente, de nuevo *propagación hacia atrás*.

$$\Delta m = m - \alpha \Delta J(m)$$

Entonces, la actualización de los pesos se obtiene restando a su valor anterior el gradiente de la función de error multiplicado por una constante  $\alpha$ . Esta constante, denominada *tasa de aprendizaje*, regula el tamaño del paso en cada iteración. Si  $\alpha$  es demasiado pequeña, el descenso de gradiente será muy lento, en cambio si es demasiado grande, puede sobrepasar el mínimo. No obstante, incluso con una tasa de aprendizaje fija, el descenso de gradiente puede llegar a un mínimo local.

Como se muestra en la figura 3, el modelo de red neuronal es un proceso cíclico que comienza con las entradas, en nuestro caso, los *ratios* financieros a lo largo del tiempo para una determinada empresa. Estas entradas son procesadas por las unidades mediante la función de activación, cuyo resultado se transforma en probabilidades a través de la función sigmoide (que es la versión binaria de la softmax). Luego, se calculan los errores comenzando por la última capa y propagándolos hacia las capas anteriores. Finalmente, con esos errores se actualizan los valores de los pesos, y el proceso se repite.

Figura.3



Todo esto tiene como resultado que el modelo se ajuste a la distribución de los datos de entrada y de la variable objetivo. De forma más simplificada, el resultado final del modelo es una función cuyas variables de entrada son los *ratios* financieros, y cuya salida es la probabilidad de que dichos *ratios* correspondan a una empresa que quebrará en los próximos dos años. Gracias a la actualización de los pesos durante este proceso, la función resultante se vuelve sensible a los patrones más relevantes del conjunto de datos, sin que esto haya sido especificado directamente por nosotros. Además, en la mayoría de los casos, no se estudia de manera explícita qué considera importante el modelo, sino únicamente se estudia que tal se desempeña en la tarea asignada.

Como podemos notar, los modelos de redes neuronales se organizan en capas compuestas por unidades o neuronas. Al número de unidades en una capa se le denomina *ancho*, mientras que al número de capas se le llama *profundidad*. Esta organización se conoce como la *arquitectura* de la red neuronal, y constituye uno de los principales hiperparámetros a definir en el diseño del modelo.

Existen casos en los que solo está presente una capa, lo que hace que estos modelos sean más simples. En cambio, también existen modelos con múltiples capas (como se muestra en la figura 2). En este tipo de red neuronal, cada capa aplica una función sobre la salida

de la capa anterior. Cuando un modelo de red neuronal tiene varias capas, se dice que implementa *aprendizaje profundo*. Este enfoque parte de la idea de que los problemas complejos pueden descomponerse en problemas más simples, ya que el aprendizaje profundo combina muchas funciones sencillas para construir funciones más complejas (Goodfellow et al, 2016).

### 3.4. Pre-procesamiento de datos:

#### 3.4.1. Procesamiento en todo el conjunto de datos:

El pre-procesamiento de datos consiste en transformar los datos crudos a una forma que pueda ser interpretada por los modelos de ML. A continuación, se describe el procedimiento que se siguió para preparar el conjunto de datos y hacerlo interpretable para los modelos de redes neuronales que utilizaremos.

Lo primero es ver la composición de los datos, en este caso el conjunto de datos con el que empezamos integra alrededor de 1500 empresas del índice compuesto de NYSE, siendo un numero increíble, sin embargo, no todas eran útiles lo primero que se noto es que habían muchos datos faltantes, empresas que tenían años completos en los que no tenían información con la que calcular los ratios y otras empresas tenían datos para un año y para el siguiente no tenían nada de información, estos ejemplos de muestras deben ser mitigados, ya que esa falta de información puede causar distorsión a la hora de que nuestro modelo aprenda, para ello lo primeros que hicimos fue contabilizar el porcentaje de información faltante que había por cada columna, que serían nuestros ratios, después eliminamos los que tengan más de un 50% de porcentajes de valores faltantes, que en este caso fueron el *Valor agregado*, *Ratio de gastos de publicidad*, *Rentabilidad por dividendo*, *Rotacion de las existencias*, *Plazo de las existencias*, *Ratio de ventas por empleado*, *Ratio de precio beneficio*, *Flujo de caja libre*, *Autofinanciación de inversiones*, *Ratio de tesoreria* y *Flujo de caja economico*, aunque puede parecer un poco brusco eliminar datos sin más dilación, pero hay que considerar que esto aumentaría la calidad de nuestro conjunto de datos. Posteriormente podemos hacer lo mismo para los años que las empresas tienen datos faltantes que corresponde a las filas de nuestro conjunto de datos, y después al igual que antes eliminamos todas las filas que tengan más de un 50 % de datos faltantes, después de esta primera pulga de los campos que tenían una extrema falta de datos, en la cual se eliminaría más del 70 % de las filas y solo un 30% de todas las columnas del conjunto de datos con el que empezamos.

Claro que después de haber eliminado el 70% de las filas algunas empresas se quedaron con información temporal incompleta, haciendo que si por ejemplo tenían los ratios para el año 2024 y 2023, luego tenían el año 2022 totalmente eliminado por lo que deberíamos imponer la ventana de tiempo que queremos usar, que recordemos que son 5 años en cualquier momento del intervalo de 2000-2024 priorizando aquellos donde se ha sufrido de bancarrota, por lo que, agrupamos todas las empresas que cuyos 5 años no fueron eliminados por la pulga de campos con extrema falta de datos y eliminando el resto que son todas las empresas que no cumplan los requisitos de tiempo ya mencionados, como resultado de estas dos pulgas el conjunto de datos paso de alrededor de 1500 empresas a solo 684, pero todas ellas con información más completa.

El siguiente paso es calcular la proporción de empresas que sufrieron bancarrota (variable objetivo = 1). En nuestro conjunto de datos, el 89% corresponde a la clase “no bancarrota” (objetivo = 0) y el 11% a la clase “bancarrota”. Esta desproporción indica un desequilibrio de clases: si no intervenimos, el modelo tenderá a aprender solo los patrones de la clase mayoritaria. Es un problema común y puede abordarse tras dividir los datos en entrenamiento/validación/prueba, por ejemplo, mediante remuestreo (submuestreo/sobremuestreo) en entrenamiento, ajuste de pesos de clase, entre otras.

Nótese que, hasta ahora, hemos analizado y tratado el conjunto de datos en su totalidad, sin segmentarlo en los subconjuntos de entrenamiento, validación y prueba. Esto se debe a que estábamos resolviendo problemas estructurales del conjunto completo, pero sin usar información estadística global de los datos. Esto último sería un error conocido como contaminación de datos (data leakage), que ocurre cuando se calculan y utilizan estadísticas de todo el conjunto de datos antes de dividirlo en subconjuntos (Arp et al., 2022).

El problema de la contaminación de datos es que el subconjunto de entrenamiento recibiría información indirecta sobre la distribución de los subconjuntos de validación y prueba, lo que haría que la métrica de exactitud fuera demasiado optimista. En realidad, no sabríamos qué tan bien se está desempeñando el modelo y probablemente ese optimismo reflejaría sobreajuste más que un rendimiento real del modelo.

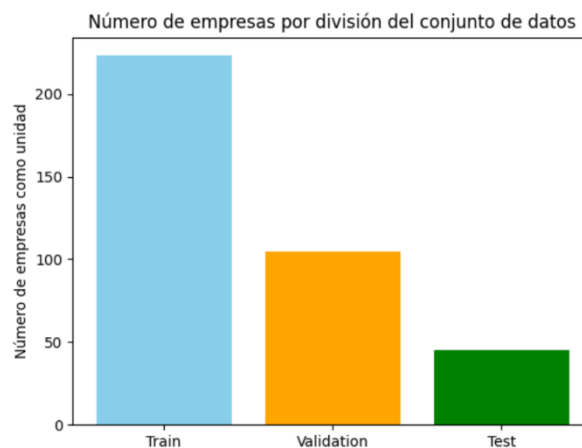
Cabe aclarar que, a pesar de lo dicho, no se eliminaron todos los casos con datos faltantes, ya que tampoco es deseable reducir en exceso el tamaño del conjunto disponible. En su lugar, después de dividir los datos en subconjuntos, estimaremos los valores faltantes rellenando los huecos con una estadística fija, como lo es la mediana. Esta medida es

rápida de calcular y robusta frente a valores faltantes. Además, la imputación se aplicará a cada subconjunto de forma independiente, evitando así la contaminación entre ellos.

#### 3.4.2. División en conjunto de entrenamiento, validación y prueba:

Para dividir el conjunto de datos, es recomendable mantener proporciones adecuadas entre los subconjuntos, asignando siempre la mayor parte al conjunto de entrenamiento. En este caso, las proporciones utilizadas fueron: 60% para entrenamiento, 28% para validación y 12% para prueba, (ver gráfico.1).

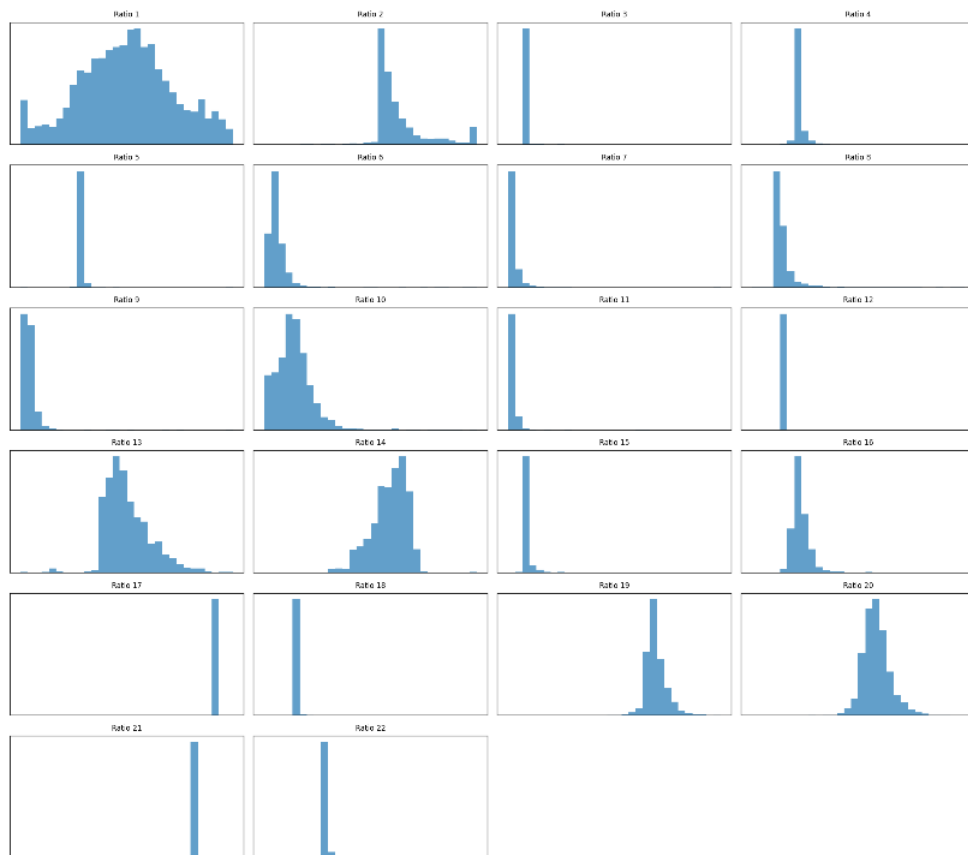
Grafico.1



Después de esto, como ya se ha mencionado, el conjunto de datos está estructurado a nivel de fila. Esto significa que, si entrenamos un modelo con este conjunto de datos, el modelo estaría tratando los ratios de cada año como si fueran independientes entre sí, y esto es un gran inconveniente, ya que no se estaría cumpliendo el supuesto de independencia e idéntica distribución de los datos (i.i.d.) (ver apartado 3.3.), ya que sabemos que la situación de una empresa en un año depende, en buena medida, de su situación en años anteriores y afecta también a la de años posteriores. Además, esta dependencia puede influir en el cálculo de la función de error para el modelo de red neuronal (ver apartado 3.3.), Este problema se soluciona agrupando los datos por *Ticker* (ver Tabla.1). De esta forma, obtenemos que en el conjunto de entrenamiento hay 410 empresas, 191 en el de validación y, por último, 83 en el de prueba.

Posteriormente a la imputación previamente mencionada, que recordemos, consistió en rellenar los valores faltantes con la mediana dentro de cada subconjunto correspondiente, podemos visualizar los histogramas de los ratios en el conjunto de entrenamiento para observar cómo se distribuyen:

Grafico.2



En el gráfico.2 se aprecia que algunos ratios, como el Ratio 1, el Ratio 13, el Ratio 14 y el Ratio 19 (ver Apéndice 1), presentan una distribución más equilibrada, con forma aproximada a una campana, concentrándose alrededor de un valor medio. En cambio, muchos otros ratios (como el Ratio 2, el Ratio 3, el Ratio 4, entre otros) muestran una fuerte asimetría: la mayoría de los valores se agrupan cerca de cero y se extienden en una larga cola hacia valores más grandes. Asimismo, ciertos ratios (por ejemplo, el Ratio 8, el Ratio 16, el Ratio 21 y el Ratio 22) exhiben picos muy marcados en un único valor (probablemente en cero) con escasa variabilidad en el resto del rango.

Estas características tienen implicaciones importantes para el modelado. Las variables con colas largas a la derecha y con valores extremos implican que la mayoría de las empresas comparten ratios similares, pero un pequeño subconjunto presenta valores atípicos muy altos. Si se entrena el modelo directamente con este conjunto de datos, existe el riesgo de que dichos atípicos dominen el proceso de aprendizaje, ya que los algoritmos de redes neuronales tienden a ajustarse con mayor intensidad a las variables que abarcan rangos numéricos más amplios, esto también aplica para la escala de los ratios, que de no tratarse puede inducir al modelo a centrarse en los patrones de los ratios con rangos altos,

ignorando en gran medida los de menor rango, lo que a su vez podría derivar en un entrenamiento inestable y dificultades para generalizar..

Finalmente, los ratios con valores casi constantes o con picos en un único punto aportan poca información predictiva de manera individual, aunque podrían resultar útiles en combinación con otras variables. En contraste, las distribuciones más balanceadas (como las del Ratio 1 o el Ratio 13) probablemente contengan señales más matizadas y útiles para el modelo.

#### 3.4.3. Eliminación de ratios demasiado sesgados, extremos o constantes:

Lo primero que debemos hacer es detectar y eliminar aquellos ratios que no resultan útiles y que no podemos “rescatar”. Para ello, calculamos un conjunto de estadísticas de diagnóstico únicamente sobre el conjunto de entrenamiento (para no contaminar los datos de validación ni de prueba). Estas estadísticas incluyen, para cada ratio: el mínimo, el percentil 1%, la mediana (50%), el percentil 99%, el máximo y el número de valores únicos.

Un ratio será eliminado del conjunto de entrenamiento, validación y prueba si cumple alguna de las siguientes condiciones:

1. Máximo absoluto mayor a 100 millones: se considera un valor extremo y el ratio se descarta. Esto permite controlar distribuciones con colas largas, donde unos pocos valores atípicos podrían dominar el entrenamiento.
2. Relación (percentil 99 / mediana) mayor que 1000: en este caso el ratio presenta una asimetría extrema, donde la mayoría de las observaciones se concentran en valores muy pequeños pero unos pocos atípicos distorsionan la distribución (como ocurre en el Ratio 2, con un pico cerca de cero y una larga cola).
3. Menos de 2 valores únicos: el ratio es prácticamente constante y no aporta información discriminativa. Este patrón coincide con casos como los Ratios 17, 21 y 22, donde la imputación concentró casi todos los valores en un único punto. Dichos ratios actúan más como indicadores binarios y su eliminación ayuda a que el modelo se centre en aquellos que realmente diferencian entre empresas que quebrarán y las que no.

En este caso el único ratio eliminado fue el numero 15 (el *Ratio de rentabilidad*), debido a que tenía un valor extremo ya que cumplió la primera condición y su máximo era superior a 100 millones.



#### 3.4.4. Winsorización para reducir la asimetría y las colas:

Tras la limpieza inicial, en la que eliminamos el único ratio con valores extremos, podemos aplicar winsorización. Esta técnica consiste en recortar los valores atípicos en los extremos de la distribución. Concretamente, para cada ratio se calculan el percentil 0,5 y el percentil 99,5 únicamente en el conjunto de entrenamiento, y luego se ajustan todos los valores: los que estén por debajo del 0,5 se reemplazan por ese límite y los que estén por encima del 99,5 se sustituyen por dicho percentil. Finalmente, los mismos límites se aplican de manera coherente a los conjuntos de entrenamiento, validación y prueba.

Gracias a este procedimiento, los ratios con distribuciones sesgadas y colas largas (como los Ratios 2, 10, 13,...) quedan acotados en el percentil 99,5. Esto evita que unas pocas empresas con ratios extremadamente elevados dominen las actualizaciones de gradiente en la red neuronal (ver apartado 3.3). El beneficio más claro se observa en casos con atípicos extremos, como ocurre en los Ratios 2 o 3, donde unas pocas empresas se sitúan muy alejadas del resto de la distribución. La winsorización hace que estos ratios resulten más compactos y comparables en escala, manteniendo intacta la gran mayoría de la información. Además, favorece la generalización del modelo, ya que los valores atípicos suelen corresponder a anomalías específicas de ciertas empresas; si no se corrigen, el modelo corre el riesgo de memorizarlos en lugar de aprender patrones generales de quiebra. De los ratios que más eliminamos valores atípicos fueron de los ratios: 11, 22, 1, 22, 20, 22, 19, 22, entre otros.

#### 3.4.5. Auto Yeo-Johnson para columnas explosivas:

De forma similar a lo que hicimos al eliminar los ratios demasiado sesgados, extremos o constantes, ahora aplicamos un enfoque más suave mediante la transformación de Yeo-Johnson, que ajusta las distribuciones para que se asemejen más a una distribución normal. Para ello, en cada ratio se calculan medidas de diagnóstico como la asimetría, la curtosis y los percentiles 1% y 99%, con estas métricas se definen los siguientes umbrales:

1. Si la asimetría de un ratio es mayor a 3, se considera muy asimétrico.
2. Si la curtosis supera 15, se interpreta como que la distribución tiene colas extremadamente densas.
3. Si la razón ( $p_{99} / \text{mediana}$ ) es mayor que 50, se considera que la cola derecha es excesivamente larga respecto al centro.

4. Si la razón ( $p1 / \text{mediana}$ ) es menor que -50, se interpreta que la cola izquierda es excesivamente baja en relación con el centro.

Estos umbrales están motivados por ratios como el 2, 3 y 4, que presentaban un fuerte sesgo y colas muy marcadas. Los ratios que cumplen estos criterios se transforman mediante Yeo-Johnson, reemplazando a los originales primero en el conjunto de entrenamiento y luego se aplica la misma transformación a validación y prueba. Esta técnica suaviza las colas pesadas que permanecieron después de la winsorización y hace que las distribuciones sean más simétricas y comparables en escala, reduciendo la influencia de valores extremos. De este modo, el modelo evita ser dominado por atípicos y puede generalizar mejor. En nuestro caso, los ratios transformados fueron: 3, 5, 8, 11, 12, 16, 17, 18, 21 y 22.

#### 3.4.6. RobustScaler:

Otro inconveniente que debemos enfrentar antes de entrenar nuestro modelo es la comparabilidad entre los ratios. No todos los ratios tienen la misma interpretación de sus valores: algunos, como los de rotación, mejoran cuanto mayores son sus valores, mientras que otros, como los ratios de endeudamiento, tienen un rango óptimo dentro del cual su valor es más favorable. Por supuesto, en este trabajo no es necesario indicarle al modelo cuáles son estos valores óptimos; el modelo, a través del aprendizaje automático, debe descubrir los patrones en los ratios que determinan la salud financiera de una empresa.

Sin embargo, para que esto sea posible, los datos deben ser comparables entre sí. Para lograrlo, utilizaremos la técnica de escalado robusto (RobustScaler), que consiste en transformar los valores de los datos restando la mediana y dividiendo por el rango intercuartílico (IQR), es decir, la diferencia entre el percentil 75 y el percentil 25:

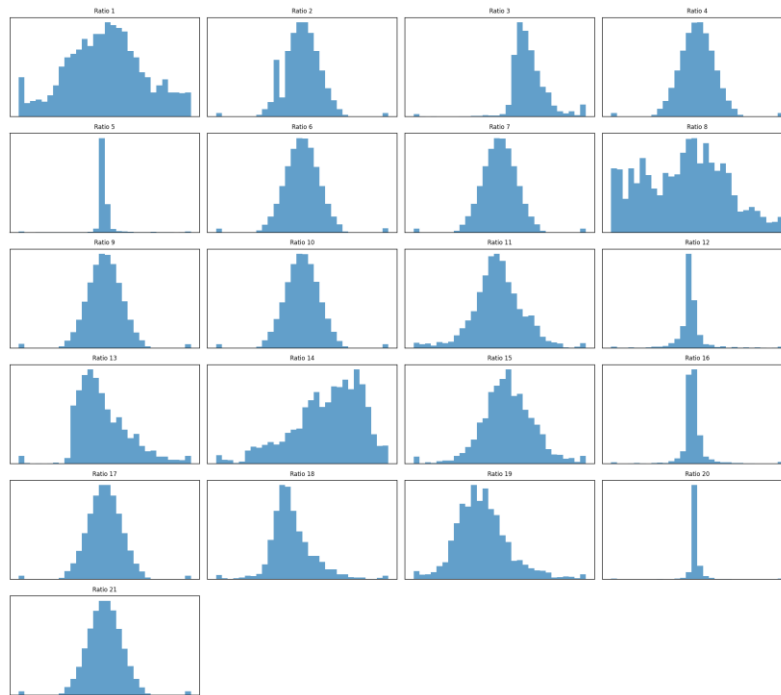
$$x' = \frac{x_i - \text{mediana}(x)}{IQR}$$

Donde  $x_i$  es un valor individual de un determinado ratio en un año específico para una empresa concreta. A este valor se le resta la mediana del ratio y el resultado se divide entre su rango intercuartílico. Este método es especialmente útil cuando los datos contienen valores atípicos o distribuciones muy sesgadas, ya que reduce el efecto de los extremos y proporciona variables más equilibradas para el entrenamiento. Esto resulta importante al usar modelos de redes neuronales, que son sensibles a las escalas de los

datos, pues aprenden patrones más fácilmente cuando las variables están normalizadas de manera robusta.

Ahora que están hechas las principales medidas de pre-procesamiento de datos, podemos proyectar nuevamente los datos para hacernos una idea de cómo se distribuyen ahora. Esto es precisamente lo que muestra el Gráfico.3:

Grafico.3

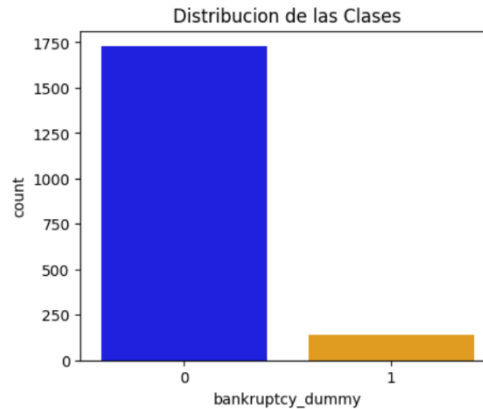


Como resultado, al revisar nuevamente los histogramas y estadísticas, se observa que muchos ratios presentan ahora una forma cercana a la campana de Gauss o, al menos, distribuciones más simétricas (por ejemplo, los Ratios 2, 4, 6, 7, 9, 10, 11, 15, 17, 18, 19 y 21). Algunos otros aún muestran picos marcados o colas largas (como los Ratios 5, 12, 16 y 20), lo cual es normal en ratios financieros, ya que ciertos denominadores pueden provocar explosiones en los valores.

Después de aplicar la winsorización, las transformaciones (Yeo-Johnson y de cuantiles) y el escalado, la red neuronal profunda (DNN) puede entrenarse casi sin problemas de gradientes explosivos ni de rangos de características desbalanceados.

### 3.4.7. Pesos de clase:

Ahora sí, justo antes de empezar a entrenar nuestro modelo, estaríamos en condiciones de abordar el problema de que los datos no están balanceados, como se observa en el Gráfico.4:



Una de las formas más sencillas, y no por eso menos efectiva, de abordar este problema es modificar la métrica de exactitud, asignando una mayor ponderación a la clase menos representada (las empresas que sufrirán bancarrota) y una menor ponderación a la clase más representada (las empresas con buena salud financiera). Esto se puede lograr mediante el ajuste de pesos de clase usando la siguiente fórmula:

$$w_i = \frac{n_{muestra}}{2 \times n_i}$$

Siendo  $n_{muestra}$  el número de observaciones en el subconjunto para el que estamos calculando la proporción, en el denominador aparece el número de clases individuales, que en nuestro caso son 2. Esto se multiplica por el número de muestras de la clase correspondiente (por ejemplo, el número de empresas que entraran en bancarrota). De esta forma, en el conjunto de entrenamiento, la proporción hace que un acierto en la clase infrarrepresentada tenga un peso de 4.64, mientras que un acierto en la clase sobrerrepresentada tenga un peso de 0.56. Así, se mitiga el problema de desbalance de datos durante el entrenamiento del modelo.

## 4. Resultados:

### 4.1. Planteamiento del modelo a utilizar:

El primer paso para entrenar nuestro modelo es elegir los hiperparámetros que definirán su arquitectura. Entre ellos se encuentran la profundidad (número de capas), el ancho (número de neuronas por capa) y la función de activación. Estos parámetros suelen

seleccionarse mediante prueba y error, evaluando el desempeño del modelo en los conjuntos de entrenamiento y validación hasta encontrar una arquitectura adecuada.

En este trabajo simplificamos el proceso de selección mediante la construcción de un bucle de búsqueda sistemática, que prueba y compara diferentes combinaciones de profundidad y ancho. Concretamente, se evaluaron arquitecturas de 1 a 4 capas y con tamaños de capa de 22, 44, 88 y 176 neuronas. La elección de estos tamaños no es arbitraria: se basan en el número de entradas del modelo, que es 22. Aunque existen 21 ratios, se incluye una entrada adicional correspondiente a una constante igual a 1, lo que se conoce como el truco del sesgo (bias trick). Este recurso permite representar el término de sesgo  $b_0$  como una entrada fija en el vector de características.

De manera más visual, el truco del sesgo puede expresarse así:

$$\begin{aligned} h_m(x) &= \sigma(b_0 + \mathbf{m}^T \mathbf{x}) \\ &= \sigma\left(b_0 + \begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_n \end{pmatrix}^T \times \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}\right) \end{aligned}$$

Donde  $\sigma$  es la función de activación que use el modelo, y esto gracias al bias trick se reescribe como:

$$\begin{aligned} h_m(x) &= \sigma(\mathbf{m}^T \mathbf{x}) \\ &= \sigma\left(\begin{pmatrix} b_0 \\ m_1 \\ m_2 \\ \vdots \\ m_n \end{pmatrix}^T \times \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}\right) \end{aligned}$$

De esta forma, el sesgo  $b_0$  se incorpora directamente como un peso adicional asociado a la entrada constante “1”.

Después de aplicar el bucle de búsqueda mencionado anteriormente, encontramos la arquitectura que mejor se adapta a nuestro objetivo de predicción de bancarrota, dadas las características del conjunto de datos, es una red neuronal con 1 capa de 22 neuronas y la función de activación ReLU.

La función ReLU (*Rectified Linear Unit*) es una de las más utilizadas en redes neuronales modernas. Matemáticamente se define como:

$$f(x) = \text{Max}(0, x)$$

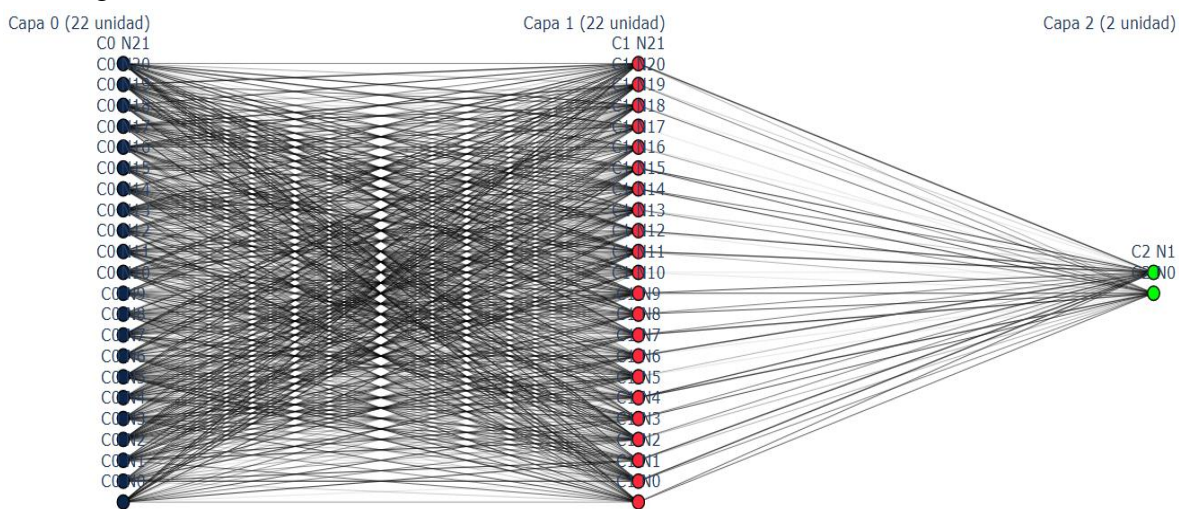
Esto significa que, si la entrada es positiva, la salida es igual a la entrada; en cambio, si la entrada es negativa o cero, la salida es 0. En otras palabras:

- Para valores positivos, la neurona transmite la señal sin cambios.
- Para valores negativos, la salida se anula, “apagando” esa neurona.

ReLU es computacionalmente muy eficiente, lo que facilita un aprendizaje más rápido y estable. Además, al transmitir únicamente la parte positiva de la señal, contribuye a que la red capture patrones útiles sin saturarse como ocurre con funciones sigmoides o tanh.

De esta forma, la arquitectura final de nuestro modelo puede representarse como:

Figura.4



Dejando de lado las cuestiones estéticas, pues los modelos muy grandes son difíciles de representar gráficamente, podemos observar que, para los datos con los que trabajamos, no se necesitan redes demasiado profundas para lograr un buen ajuste. Esto se debe, muy probablemente, al extenso tratamiento aplicado previamente a los datos: la imputación, limpieza, normalización y transformaciones realizadas redujeron la complejidad necesaria para que el modelo representara adecuadamente la información.

## 4.2. Entrenamiento:

Salida.1.

```
Eval(train): {'loss': 0.47385838627815247, 'compile_metrics': 0.8956097364425659}
Eval(val):   {'loss': 0.5300600528717041, 'compile_metrics': 0.8617801070213318}
65/65 ————— 1s 6ms/step
30/30 ————— 0s 10ms/step

Confusion Matrix (Train):
[[1662 167]
 [ 47 174]]
Confusion Matrix (Validation):
[[744 114]
 [ 18 79]]
```

En la Salida 1 observamos que los resultados del entrenamiento muestran una función de coste del 47% en el conjunto de entrenamiento y del 53% en el conjunto de validación. Esto indica que la pérdida en entrenamiento es menor, pero la diferencia es reducida, por lo que no hay señales de sobreajuste significativo. Si la brecha fuese grande, sí sería un síntoma de memorización en lugar de generalización.

En cuanto a la métrica de exactitud, ponderada para dar mayor peso a la clase minoritaria, es decir, las empresas que entraran en bancarrota, el modelo alcanza un 90% en entrenamiento y un 86% en validación. Esto significa que el modelo logra discriminar bien entre empresas sanas y quebradas, incluso en un conjunto de datos muy desbalanceado y ruidoso.

Pasando al análisis de la matriz de confusión en entrenamiento:

$$\begin{bmatrix} 1662 & 167 \\ 47 & 174 \end{bmatrix}$$

Donde 1662 es el número de observaciones de no bancarrota y fueron clasificadas correctamente, mientras que hay 167 observaciones que también de no quiebra, pero fueron marcadas erróneamente como que quebrarían. Después tenemos que hay 47 observaciones de quiebra, pero se clasificaron como observaciones de no quiebra, es decir no se detectó la quiebra lo cual son errores peligrosos en esta tarea, además tenemos 174 que se detectó correctamente la futura quiebra. Dejándonos unas métricas de entrenamiento de:

- Precisión =  $(1662+174) / 2050 = 89,4\%$
- Recuperación =  $174 / (174+47) = 78,7 \%$
- Precisión =  $174 / (174+167) = 51,0 \%$

Esto nos indica que, aunque el modelo identifica la mayoría de las quiebras, la precisión es baja (51%), lo que significa que cerca de la mitad de las quiebras previstas resultan ser falsas alarmas.

Ahora, en el conjunto de validación, la matriz de confusión es:

$$\begin{bmatrix} 744 & 114 \\ 18 & 79 \end{bmatrix}$$

De aquí se obtienen:

- Precisión =  $(744+79)/955 = 86,0 \%$
- Recuperación =  $79 / (79+18) = 81,4 \%$

- Precisión =  $79 / (79+114) = 40,9 \%$

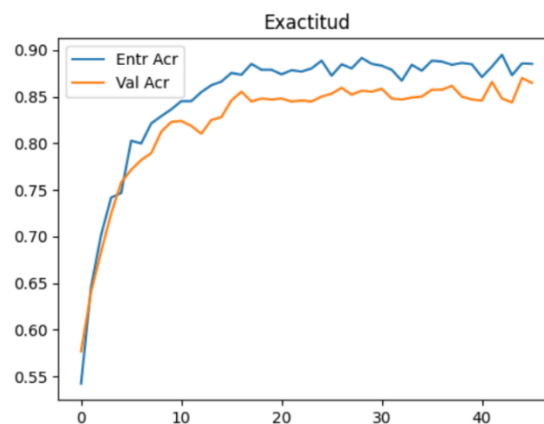
Aquí vemos una ligera caída de 10 puntos porcentuales en la precisión, lo que confirma que el modelo tiene la tendencia a producir muchos falsos positivos. Sin embargo, mantiene una recuperación sólida de 81,4%, lo que significa que detecta 8 de cada 10 quiebras. Esto refleja que el modelo, al estar ponderado para dar más importancia a las quiebras, está más inclinado a clasificarlas en exceso, lo que explica la baja precisión.

En conclusión, el modelo logra una alta recuperación (81%), lo que implica que rara vez omite quiebras. No obstante, el principal reto sigue siendo la baja precisión: aproximadamente 6 de cada 10 predicciones de “quiebra” son falsas alarmas.

El modelo está ajustado para detectar quiebras a expensas de algunas falsas alarmas, lo cual podría tener sentido en un caso de uso real (omitir quiebras es peor que investigar algunas empresas seguras).

Gracias a los siguientes gráficos podemos ver el desempeño de nuestro modelo de una manera más visual, al ver la comparación entre el conjunto de entrenamiento y el conjunto de validación:

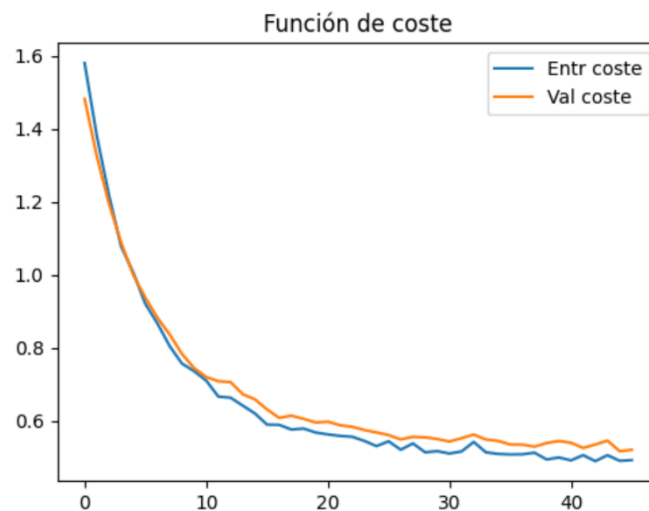
Gráfico.5



El Gráfico 5 muestra la comparación entre la exactitud obtenida en los conjuntos de entrenamiento y validación. Como se observa, ambas métricas aumentan de manera constante durante las primeras 10 épocas. Posteriormente, entre las épocas 10 y 20, la exactitud del entrenamiento se estabiliza alrededor de 0,88–0,89, mientras que la de validación lo hace en torno a 0,85–0,87. La pequeña brecha entre ambas curvas, con el entrenamiento ligeramente por encima de la validación, es un comportamiento esperado y saludable. Esto sugiere que el modelo generaliza bien en lugar de memorizar, lo cual es precisamente lo deseado, ya que indica que el modelo identificó patrones relevantes con rapidez y luego convergió a un rendimiento estable.



Gráfico.6



El Gráfico 6 muestra la evolución de la función de coste (o pérdida) en los conjuntos de entrenamiento y validación. Se observa que ambas disminuyen gradualmente durante las primeras 10 épocas. Posteriormente, la pérdida de entrenamiento continúa descendiendo lentamente hasta situarse en torno a 0,45, mientras que la de validación se estabiliza alrededor de 0,55–0,60. Es importante destacar que la pérdida de validación no llega a aumentar, lo cual sería un indicio de sobreajuste, esto sugiere que el modelo mantiene un comportamiento estable; en todo caso, podría explorarse un entrenamiento ligeramente más corto para optimizar la eficiencia, aunque no parece un problema grave. Si analizamos este gráfico junto con los anteriores, las curvas en conjunto confirman la estabilidad descrita en la salida.1.

## Salida.2

```
Keras Test Eval: {'loss': 0.47402969002723694, 'compile_metrics': 0.908433735370636}
Sklearn ROC AUC: 0.9265867847529737
Sklearn log_loss: 0.373196749510534
[[324 25]
 [ 13 53]]
      precision    recall  f1-score   support

     0       0.9614       0.9284       0.9446        349
     1       0.6795       0.8030       0.7361         66

 accuracy          0.9084          415
 macro avg       0.8205       0.8657       0.8404          415
 weighted avg    0.9166       0.9084       0.9114          415

Test Results: {'loss': 0.47402969002723694, 'compile_metrics': 0.908433735370636}
```

En la Salida 2 se muestra el desempeño de nuestro modelo en el conjunto de prueba. Antes de analizar estos resultados, es importante remarcar que no deben usarse como criterio para elegir el modelo, ya que el conjunto de prueba suele ser relativamente pequeño (12% de todo el conjunto de datos) y, por lo tanto, no es representativo de todas

las empresas que se quisieran analizar en el futuro. Su función es únicamente ofrecer una referencia de cómo se comporta el modelo con información nunca vista.

Dicho esto, los resultados confirman lo observado previamente la exactitud del modelo alcanza aproximadamente un 90%, y su capacidad de discriminación se mantiene elevada, clasificando correctamente a una empresa que quebrará frente a una empresa sana en torno al 93% de los casos.

En cuanto a la función de pérdida, el valor obtenido, que es de 0,373, indica que las probabilidades están razonablemente bien calibradas. En otras palabras, estos resultados sugieren una buena capacidad de generalización, dado que el rendimiento en el conjunto de prueba es muy cercano al observado en validación.

Luego la matriz de confusión para el conjunto de prueba es:

$$\begin{bmatrix} 324 & 25 \\ 13 & 53 \end{bmatrix}$$

Esto significa que 324 de las observaciones de no quiebra fueron clasificadas correctamente, mientras que 25 observaciones que también eran de no quiebra fueron clasificadas erróneamente (falsos positivos). Por otro lado, 13 observaciones de si quiebra no fueron detectadas (falsos negativos), mientras que 53 casos de quiebra se predijeron correctamente.

De aquí se refuerza lo ya señalado: el modelo rara vez omite quiebras (solo 13 casos), pero aún genera algunos falsos positivos (25 casos).

Clase 0 (la empresa no quiebra en los dos años siguientes)

- Precisión = 0,961 → casi todas las empresas "seguras" predichas lo son realmente.
- Recuperación = 0,928 → la mayoría de las empresas seguras se identifican correctamente.
- F1-score = 0,945 → rendimiento muy sólido.

Clase 1 (empresas que quebraran)

- Precisión = 0,680 → aproximadamente 2 de cada 3 quiebras alertadas son reales.
- Recuperación = 0,803 → el modelo detecta aproximadamente 8 de cada 10 quiebras.
- F1-score = 0,736 → inferior al de la clase 0, lo esperado dado que la clase de quiebra es menos común.

En definitiva, vemos nuevamente que el modelo presenta mejor desempeño en recuperación que en precisión para la clase de quiebra, lo cual sigue la lógica de que detectar quiebras puede llegar a ser más importante que evitar falsas alarmas.

Salida.3

Feature 18: importance = 0.005581	Feature 8: importance = 0.000875
Feature 3: importance = 0.006027	Feature 15: importance = 0.001043
Feature 12: importance = 0.006278	Feature 14: importance = 0.001069
Feature 6: importance = 0.008720	Feature 9: importance = 0.001069
Feature 7: importance = 0.019108	Feature 11: importance = 0.001544
Feature 5: importance = 0.020548	Feature 10: importance = 0.001649
Feature 21: importance = 0.035130	Feature 13: importance = 0.001771
Feature 16: importance = 0.045738	Feature 1: importance = 0.002996
Feature 17: importance = 0.053357	Feature 4: importance = 0.003927
Feature 0: importance = 0.076640	Feature 20: importance = 0.005266
Feature 2: importance = 0.081357	Feature 19: importance = 0.005466

El ratio 2 (capacidad de devolución de préstamos) y ratio 0 (endeudamiento) son los más importantes, lo cual tiene sentido económico ya que la probabilidad de quiebra está fuertemente determinada por la capacidad para cubrir deudas y por el apalancamiento. Unido a esto, los ratios 17, 16 y 18 muestran importancias relevantes, coherente con que la capacidad de generar caja e invertirla reduce riesgo de quiebra debido a que mientras más recursos puedes generar más difícil es que la cuenta de patrimonio sea negativa.

Por otro lado, tal vez el ratio 14 (umbral de rentabilidad) y 15 (rentabilidad financiera) aparecen muy bajos porque su información predictiva probablemente esté en parte capturada por los ratios 2, 16 y las medidas de autofinanciación, además las transformaciones y eliminación de variables por extremos, pudieron favorecer esta dinámica.

## 5. Conclusiones:

En general, este trabajo presenta un modelo con capacidad real para la predicción de quiebras, pero conviene no sobredimensionar sus resultados. Aunque el modelo muestra buena discriminación y una alta recuperación, detecta aproximadamente 8 de cada 10 quiebras, su precisión en la clase de quiebra es limitada (varía según partición: 51% en entrenamiento, 41% en validación y 68% en la prueba), por lo que muchas de las predicciones de quiebra son falsas alarmas. Por tanto, este sistema no reemplaza al analista contable, más bien funciona como herramienta complementaria, un filtro o primer diagnóstico que ayuda a priorizar empresas para un examen humano más profundo.

También hay que reconocer las limitaciones prácticas: la cantidad y calidad de los datos son determinantes para el rendimiento. No sería sorprendente que modelos más simples,

pero entrenados con datos de mayor calidad y cobertura temporal, sea igual o mejor que modelos más complejos con datos ruidosos o incompletos.

A su vez hay que tener en cuenta que este tipo de modelos aprende los patrones más frecuentes en los datos de entrenamiento: es decir, adquiere señales representativas de las empresas que lo alimentaron. Por ello resulta ilusorio considerarlo un clasificador universal aplicable sin más a cualquier empresa o contexto. En este trabajo, los datos provienen principalmente de empresas estadounidenses listadas en la NYSE, por lo que el modelo captura patrones y prácticas contables, regulatorias y de mercado propias de ese universo. Si se aplica a empresas de otro país (por ejemplo, España) o a firmas que no cotizan en la Bolsa de Nueva York, es probable que su rendimiento disminuya debido al desplazamiento de dominio (diferencias en distribución de variables, tamaño, sector, etc.).

Para mitigar este riesgo conviene, al desplegar el modelo fuera del dominio original, recopilar y usar datos locales para recalibrarlo o reentrenarlo, validar su rendimiento por subgrupos (país, sector, tamaño), ajustar umbrales de decisión y considerar técnicas de adaptación de dominio o transferencia de aprendizaje. En resumen: el modelo es una herramienta valiosa, pero su uso responsable exige validar y, si procede, adaptar el modelo al contexto concreto donde se quiera aplicar.

En cuanto a posibles mejoras, hay un amplio margen, desde incorporar ratios más especializados y relevantes para bancarrotas, hasta ampliar y limpiar el conjunto de datos, refinar la ingeniería de características, ajustar umbrales de decisión (equilibrio precisión/recuperación), emplear validaciones más robustas (por ejemplo, bootstrap o cross-validation estratificada), probar ensamblados y técnicas de calibración de probabilidades, o simplemente eliminar algunos de los ratios que no aportan tanto al modelo (como muestra la salida.<sup>3</sup>) y así sería un modelo más barato en cuanto información requerida se refiere. Con estas mejoras y en manos expertas, el modelo puede convertirse en una herramienta muy útil para analizar la salud financiera de empresas.

En definitiva, el modelo está orientado conscientemente a maximizar la detección de quiebras (aceptando falsas alarmas), una elección razonable cuando el coste de no detectar una quiebra supera el coste de investigar una alarma falsa. Sin embargo, su uso operacional exige supervisión humana, validación continua y mejoras en los datos y el diseño para aumentar la precisión y la confianza en sus predicciones.

## 6. Bibliografía:

Amat, O. (2013), *Claves del análisis de empresas*, Revista de Contabilidad y Dirección Vol. 2, 13-51

Aho, A. V., Hopcroft, J. E., Ullman J. D. (1974). *The Design and Analysis of Computer Algorithms*, Addison-Wesley

Arp, D., Quiring, E., Pendlebury, F., Warnecke, A., Pierazzi, F., Wressnegger, C., Cavallaro, L., Rieck, K., (2022), *Dos and Don'ts of Machine Learning in Computer Security*, 31st USENIX Security Symposium (USENIX Security '22), 3971–3988

Artene, A. E. y Domil, A. E. (2025), *NeuralNetworksinAccounting: Bridging Financial Forecasting and Decision SupportSystems*, Electronics, 14, 993. <https://doi.org/10.3390/electronics14050993>

BERRY, A., CITRON, D., JARVOS, R. (1987), *The information needs of bankers dealing with large and small companies*, CACA, London.

Cybenko, G. (1989), *Approximation by superpositions of a sigmoidal function*. Mathematics of Control, Signals, and Systems, volume 2, 303-314.

Deisenroth, M.P., Faisal, A. A., Ong, C. S.,(2024), *Mathematics for Machine Learning*, Published by Cambridge University Press

Djerdjouri, M. (2020), *Data and business intelligence systems for competitive advantage: prospects, challenges, and real-world applications*, Mercados y Negocios, (41), 5-18.

Goodfellow, I., Bengio Y., Courville A. (2016), *Deep Learning*, MIT Press, <http://www.deeplearningbook.org>

Huanhuan, S. (2024), *Natural Language Processing and Text Mining Algorithms for Financial Accounting Information Disclosure*, Journal of Electrical Systems 20-9s: 453-461.

Jan, C.-L. (2021), *Detection of Financial Statement Fraud Using Deep Learning for Sustainable Development of Capital Markets under Information Asymmetry*. Sustainability, 13, 9879. <https://doi.org/10.3390/su13179879>

Joshi, P., India, U. (2023). *Role of financial reporting standards in enhancing transparency and accountability: an empirical study*, Journal of cardiovascular Disease

Rumelhart, David E., Hinton, Geoffrey E., Williams, Ronald J. (1986), *Learning Representations by Back-Propagating Errors*. Nature, volumen 323(6088), 533–536.

Li, S., Deng, Y.Q., Zhu, Z.L., Hua, H.L., Tao, Z.Z. (2021), *A comprehensive review on radiomics and deep learning for nasopharyngeal carcinoma imaging*. Diagnostics, volumen 11(9), 1523.

MacKay, D. J. C. (2003), *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press.

Mitchell, T.M. (1997), *Machine Learning*, Book News, Inc

New York Stock Exchange. (2025). *Listings Directory*. NYSE. [https://www.nyse.com/listings\\_directory/stock](https://www.nyse.com/listings_directory/stock)

Antwi, B. O., Adelakun B. O., Eziefule A. O., (2024), *Transforming Financial Reporting with AI: Enhancing Accuracy and Timeliness*, International Journal of Advanced Economics, volumen 6, 205-223

Refinitiv. (2025), *Eikon database*. Retrieved May–July 2025, <https://www.refinitiv.com>

Wolpert, D. H., Macready, W. G. (1997), *No Free Lunch Theorems for Optimization*, IEEE Transactions on Evolutionary Computation Volumen 1: 67-82.

## 7. Anexos

### 7.1. Apendice.1: Enumeración de los ratios antes del escalado:

0. Ratio de endeudamiento
1. Ratio de calidad de la deuda
2. Ratio de capacidad de devolución de los prestamos
3. Ratio de gasto financiero / ventas
4. Ratio de coste deuda
5. Ratio de liquidez
6. Ratio de disponibilidad
7. Rotación del activo fijo
8. Rotación del activo circulante
9. Ratio de plazo de cobro
10. Ratio de plazo de pago
11. Ratio de expansión de ventas
12. Ratio de gasto fijo / ventas
13. Ratio de gasto variable / ventas
14. Umbral de rentabilidad

15. Ratio de rentabilidad
16. Ratio de rentabilidad financiera
17. Apalancamiento financiero
18. Autofinanciación sobre ventas
19. Autofinanciación sobre activo
20. Política de dividendo
21. Relación entre valor de mercado y contable

## 7.2. Apendice.2: Enumeración de los ratios utilizados en el entrenamiento:

0. Ratio de endeudamiento
1. Ratio de calidad de la deuda
2. Ratio de capacidad de devolución de los prestamos
3. Ratio de gasto financiero / ventas
4. Ratio de coste deuda
5. Ratio de liquidez
6. Ratio de disponibilidad
7. Rotación del activo fijo
8. Rotación del activo circulante
9. Ratio de plazo de cobro
10. Ratio de plazo de pago
11. Ratio de expansión de ventas
12. Ratio de gasto fijo / ventas
13. Ratio de gasto variable / ventas
14. Umbral de rentabilidad
15. Ratio de rentabilidad financiera
16. Apalancamiento financiero
17. Autofinanciación sobre ventas
18. Autofinanciación sobre activo
19. Política de dividendo
20. Relación entre valor de mercado y contable