

# Análisis de sentimientos en la formación de una cartera de inversión

Alejandro Sellés Pérez

Máster universitario en Big Data y Ciencia de Datos

26 de septiembre de 2025



**Universidad**  
Internacional  
de Valencia

## 1 Introducción

## 2 Marco teórico

- Análisis de series temporales
- Análisis de sentimientos

## 3 Experimentación

- Datos y variables
- Metodología
- Resultados

## 4 Conclusiones

## 5 Siguientes pasos

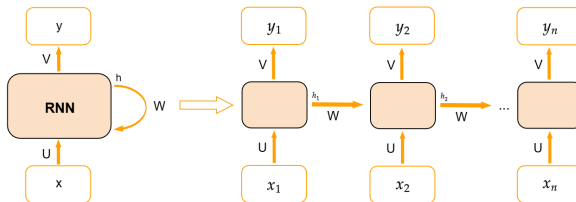




## Motivación de este trabajo

- La mayoría de estudios combinando **redes neuronales** y **sentimiento de noticias** se centran en **horizontes medios o largos**.
- Nuestro objetivo: evaluar si estas técnicas pueden ser útiles en **muy corto plazo** (de un día para otro).
- Idea: desarrollar un **bot de trading** a partir de los modelos obtenidos capaz de tomar decisiones diarias.
- Somos conscientes de la dificultad y de la alta probabilidad de obtener modelos con fiabilidad limitada.

- Diseñadas para procesar datos secuenciales: la salida depende de la entrada actual y del estado previo.
- Incorporan **memoria temporal** mediante un estado oculto  $h_t$ .
- Adecuadas para tareas donde el orden importa: lenguaje, voz, predicción de series temporales.



**Figura:** Esquema de una RNN

- Problema del **desvanecimiento y explosión de gradientes**.
- Dificultad para aprender dependencias a largo plazo.
- Cuando las redes son muy largas, tienden a “olvidar” información de pasos temporales lejanos.

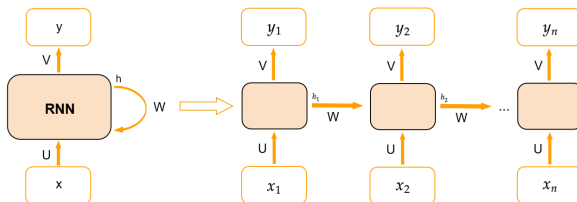


Figura: Esquema de una RNN

- 

**Figura:** Esquema de una celda LSTM

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \quad h_t = o_t \odot \tanh(c_t)$$



## Arquitectura GRU

- Propuestas por Cho et al. (2014) como alternativa más simple a las LSTM.
- Mantienen un único estado oculto  $h_t$  que combina memoria corta y larga.
- Compuertas:

- Actualización:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)$$

- Reinicio:

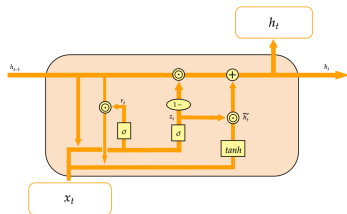
$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)$$

- Estado candidato:

$$\tilde{h}_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h)$$

- Actualización del estado:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$$



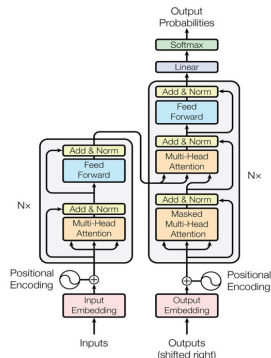
**Figura:** Esquema de una celda GRU

# Limitaciones de RNN, LSTM y GRU en PLN

- Procesamiento secuencial → lento y difícil de paralelizar.
- Persisten problemas de gradiente en secuencias largas.
- La memoria se condensa en un único vector → pérdida de contexto.

# Mecanismo de atención en Transformers

- Cada palabra de la secuencia se representa con tres vectores:
  - **Query (Q)**: lo que busca en las demás palabras.
  - **Key (K)**: la información que ofrece cada palabra.
  - **Value (V)**: el contenido que se combina en la salida.
- La **query** de una palabra se compara con las **keys** de todas las demás para asignar un peso de relevancia.
- Para cada palabra el resultado es una ponderación de las relaciones que tiene con el resto.
- Con **atención multi-cabeza**, distintas cabezas aprenden distintas relaciones.



**Figura:** Arquitectura Transformer  
(Vaswani et al., 2017)

# BERT y FinBERT para Análisis de Sentimientos

- **BERT** (Devlin et al., 2019): basado en **Transformers**, aprende representaciones **bidireccionales** del texto.
- Se añade el token especial [CLS] al inicio; su vector  $h_{[CLS]}$  resume el significado global de la secuencia.
- Para clasificación:

$$y = \text{softmax}(Wh_{[CLS]} + b)$$

donde  $y$  es un vector de probabilidades.

- Devuelve **tres scores de probabilidad**:
  - $P(\text{positivo})$
  - $P(\text{neutral})$
  - $P(\text{negativo})$

La etiqueta final se elige según la clase con mayor probabilidad.

- Utilizaremos **FinBERT** (Araci, 2019): BERT adaptado al lenguaje financiero.

## Datos y variables utilizadas

- **Fuentes:** Github dataset FNSPID (Zhong, 2020) + precios históricos de apertura/cierre de `yfinance`.
- **Periodo:** 2012–2023
  - 2012–2021: entrenamiento
  - 2022: validación
  - 2023: test
- **Variables técnicas:** 92 indicadores (ta) → selección de las 20 más relevantes vía Random Forest.
- **Variables de sentimiento:** noticias → resumen + título → clasificación con FinBERT.
  - Probabilidades para cada noticia: positivo, neutral, negativo.
  - Agrupación diaria: medias, máximos/mínimos, desviaciones, recuento de noticias...
- **Compañías:** Gilead, Coca-Cola, Merck, QQQ, Wells Fargo, NVIDIA.

# Entrenamiento

- Se entrenaron un total de 12 modelos para cada compañía.
- Tres tipos de modelos:
  - LSTM
  - GRU
  - RandomForest
- Dos configuraciones de entrada:
  - Solo **variables de sentimiento**.
  - Todas las variables. **Variables de sentimiento + Variables técnicas**.
- Dos tareas predictivas:
  1. **Clasificación binaria:** predecir si la acción subirá o bajará al día siguiente.

$$y_t = \begin{cases} 1 & \text{si } P_{t+1} > P_t, \\ 0 & \text{si } P_{t+1} \leq P_t \end{cases}$$

2. **Regresión logarítmica:** modelar el retorno diario.

$$r_t = \ln\left(\frac{P_{t+1}}{P_t}\right), \quad \hat{y}_t = \begin{cases} 1 & \text{si } \hat{r}_t > 0, \\ 0 & \text{si } \hat{r}_t \leq 0 \end{cases}$$

## Evaluación y ensamblado

### ■ Métricas de clasificación:

- Accuracy:  $\frac{TP+TN}{TP+TN+FP+FN}$
- Precisión:  $\frac{TP}{TP+FP}$  (fiabilidad de las señales positivas).
- Recall:  $\frac{TP}{TP+FN}$  (capacidad de detectar todas las subidas).

### ■ Estrategia de ensamblado:

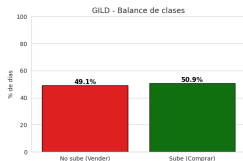
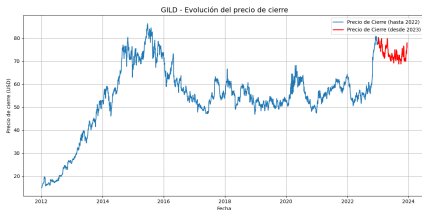
- Se combinan predicciones binarias  $\{0, 1\}$  de 5 modelos distintos a través de una votación ponderada, donde cada modelo recibe un peso  $w_i$  según sus métricas.
- Puntuación del modelo  $i$ :

$$s_i = \beta_{\text{acc}} \cdot \text{Accuracy}_i + \beta_{\text{prec}} \cdot \text{Precisión}_i$$

- Realizamos 2 ensamblados distintos:
  - $\beta_{\text{acc}} > \beta_{\text{prec}}$ : prioriza **accuracy global**.
  - $\beta_{\text{prec}} > \beta_{\text{acc}}$ : prioriza **precisión de la clase 1 (Comprar)**.

## Contexto - Gilead Sciences (GILD)

- Crecimiento fuerte 2012–2015 por tratamientos de hepatitis C.
- Repunte en 2020 en crisis Covid gracias a *Veklury (remdesivir)*.
- En 2023 alternancia de subidas y bajadas → incertidumbre.
- Balance de clases equilibrado entre jornadas de subida y bajada.





## GILD – Resultados Clasificación Binaria

Modelo	Accuracy	Precisión	Recall	F1	Peso Acc.	Peso Prec.
Combined LSTM	0.609	<b>0.690</b>	0.556	0.615	0.120	0.293
Combined GRU	0.606	0.667	0.458	0.543	0.116	0.252
Sentiment RF	0.540	0.556	0.615	0.584	–	0.099
Sentiment GRU	0.532	0.529	<b>1.000</b>	<b>0.692</b>	–	0.082
Sentiment LSTM	0.516	0.521	0.939	0.670	–	0.073
Combined RF	0.460	0.477	0.323	0.385	–	–

- Los modelos de redes neuronales dan mejores resultados que Random Forest.
- Destaca el LSTM con todas las variables, que alcanza un Accuracy y una precisión bastante buenas.
- Modelos con variables de sentimiento tienen un **Recall muy alto**, pero baja Precisión. Tendencia clara a predecir Clase 1.

## Resultados

## GILD – Resultados Regresión Logarítmica

Modelo	Accuracy	Precisión	Recall	F1	Peso Acc.	Peso Prec.
Reg Sentiment RF	<b>0.628</b>	0.258	0.271	0.265	0.144	–
Reg Combined LSTM	0.624	<b>0.378</b>	0.708	<b>0.493</b>	0.138	–
Reg Combined GRU	0.587	0.320	0.471	0.381	0.096	–
Reg Combined RF	0.586	0.250	0.339	0.288	–	–
Reg Sentiment GRU	0.571	0.308	0.471	0.372	–	–
Reg Sentiment LSTM	0.429	0.256	0.588	0.357	–	–

- Accuracy similar a los modelos de clasificación binaria. Pero esta vez es superior en los modelos de Random Forest. Sin embargo, el modelo más decente parece el LSTM con todas las variables.
- La **Precisión de compra es muy baja** en general (<30 %). Ninguno entra en el ensamblado por Precisión.
- Debido al alto Accuracy y baja precisión al predecir compras, podemos deducir que estos modelos tienen una buena precisión a la hora de **predecir cuándo No Comprar**.

## Resultados

## GILD – Ensamblados

## Ensamblado por Accuracy

Clase	Prec.	Recall	F1
No Comprar	0.774	0.761	0.768
Comprar	0.307	0.322	0.314
<b>Acc. global</b>	0.653		

Alto recall y precisión en la clase 0.  
Modelo fiable para descartar compras,  
pero falla en detectar oportunidades.

## Ensamblado por Precisión

Clase	Prec.	Recall	F1
No Comprar	0.605	0.441	0.510
Comprar	0.593	0.739	0.658
<b>Acc. global</b>	0.597		

Buena precisión y alto recall. Puede ser  
un buen apoyo para detectar  
oportunidades de compra.

## Conclusiones

Los **resultados del resto de compañías** son bastante similares a los de GILD, y nos permiten obtener unas conclusiones bastante similares:

- **Modelos de regresión logarítmica:** Precisión muy baja ( $<30\%$ ), útiles solo en ensembles por *Accuracy* para descartar compras.
- **Ensamblados:**
  - Por *Accuracy*: en general muy fiables en todas las compañías identificando “No comprar” ( $>75\%$  en varios casos).
  - Por *Precisión*: Buen apoyo para detectar oportunidades de compra pero con resultados más heterogéneos; útiles en *GILD* o *NVDA*.
- **Comparación de modelos:** No se observaron ventajas claras de LSTM/GRU sobre Random Forest. Tampoco entre usar todas las variables o solo sentimiento.
- **Utilidad práctica:** Los modelos no constituyen una estrategia de *trading* directa, pero sí un **apoyo en la toma de decisiones**, sobre todo como filtro adicional.

## Siguientes pasos

- **Modelos de lenguaje de gran escala (LLMs):** Aprovechar técnicas como *RAG* para integrar noticias, informes o resultados en tiempo real.
- **Aprendizaje por refuerzo (RL):** Entrenar agentes de trading en entornos de simulación (*FinRL*, *dueling Q-networks*), optimizando beneficio acumulado con costes, liquidez y riesgo.
- **Redes neuronales aplicadas a finanzas cuantitativas:** Uso de *PINNs* para aproximar soluciones de ecuaciones diferenciales estocásticas como Black–Scholes.

## Referencias I



Araci, D. (2019). FinBERT: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*. Recuperado de <https://arxiv.org/abs/1908.10063>



Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*. <https://arxiv.org/abs/1406.1078>



Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*. <https://arxiv.org/abs/1810.04805>



Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL* (pp. 4171–4186). <https://doi.org/10.18653/v1/N19-1423>

## Referencias II



Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.  
<https://doi.org/10.1162/neco.1997.9.8.1735>



Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS 2017)* (pp. 5998–6008). <https://arxiv.org/abs/1706.03762>



Zhong, D. (2020). FNSPID: A financial news dataset for stock price movement prediction. *GitHub repository*. Recuperado de [https://github.com/Zdong104/FNSPID\\_Financial\\_News\\_Dataset](https://github.com/Zdong104/FNSPID_Financial_News_Dataset)