



# TRADING ALGORÍTMICO

*GJR TRADES*

Juan Jeronimo Jauregui Prince

Gabriel David Castillo Rodriguez

Ricardo Andrés Pérez Porras



# WHAT IS THE STOCK MARKET?



Es un lugar donde se intercambian acciones; se pueden comprar o vender acciones.

Pero no solo eso.

Involucra una compleja red de compra y venta de todo tipo de instrumentos, cuyo precio es volátil y puede presentar ciertos patrones detectables para los algoritmos.

Prácticamente todos los países tienen una bolsa de valores local, en la que se ubican las principales empresas.



# WHAT IS THE MOTIVATION

¿Cuánto dinero se mueve en la bolsa de valores?

"En lo que va del año 2023, la NYSE ha tenido un volumen de negociación diario promedio de más de 15 mil millones de acciones."

Fuente: Rankia colombia

"Cerca de tres cuartas partes de los intercambios en la Bolsa de Valores de Nueva York y Nasdaq [la segunda bolsa de valores electrónica y automatizada más grande de Estados Unidos] son hechos por algoritmos", dice Belton." (BBC NEWS,2018)

## ¿Es una estafa?

Invertir en la bolsa de valores es seguro siempre y cuando se haga de manera informada y a través de canales regulados. Los organismos supervisores, como la SEC en Estados Unidos o la CNBV en México, han establecido reglas estrictas para proteger a los inversionistas y garantizar la transparencia de los mercados.

Fuente: Theinvestor.u



## ¿Sirven los humanos para invertir?

Los humanos no podemos competir con la rapidez de las computadoras. En apenas unos segundos, una máquina es capaz de realizar muchas más operaciones.

"Las computadores pueden hacer varias transacciones en unas fracciones de segundo, aprovechando cambios diminutos en precios de mercado e índices para obtener un beneficio", explica Belton.

Fuente: BBC

# MOTIVACIÓN PRINCIPAL

“Comprender el mercado financiero , identificar los patrones y tendencias, y junto a los recursos de computación, tomar las mejores decisiones con base a los datos”

“El trading se basa principalmente en el análisis técnico, el análisis fundamental y la aplicación de una estrategia concreta para operar”.(WSJ JOURNAL,2018)





# DE QUE SE TRATA EL PROYECTO?



01

Medir movimientos en la bolsa

02

Encontrar patrones

03

Predecir movimientos

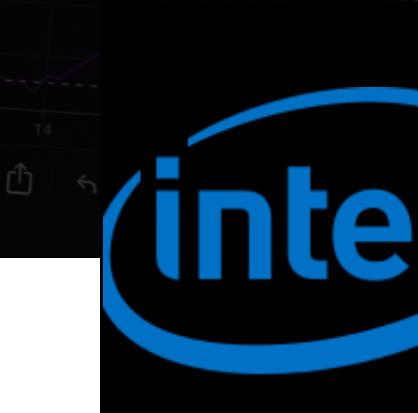
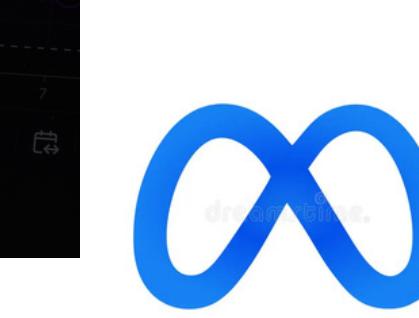
04

Generar señales



# DATASETS

1HR



\$AAPL

\$AMZN

\$NVDA

\$MSFT

\$META

\$INTC

\$AMD



# PROCESAMIENTO DATASET

01

## Moving average

Usamos la media móvil o SMA en un intervalo determinado de tiempo, así identificamos tendencias, retrocesos y cambios de tendencia

02

## Niveles de soporte y resistencia

Usamos la moda de los precios en un intervalo de tiempo, para encontrar niveles de donde el precio puede rebotar o romper.

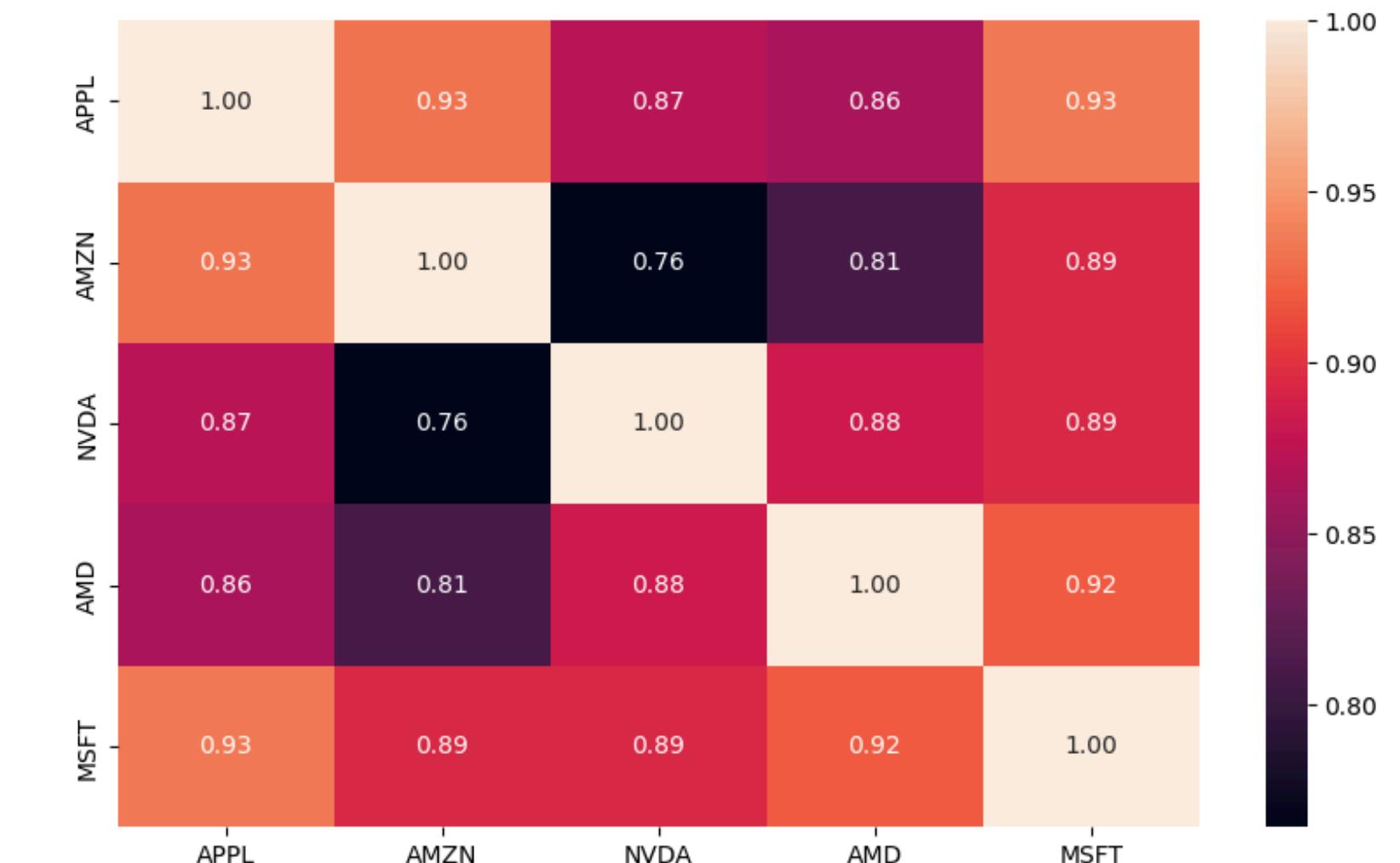
03

Usamos rangos de precios para encontrar, zonas de retrocesos y equilibrio para determinar la validez de una tendencia y puntos adecuados de entrada

04

## Tomas de liquidez

Usamos precios que sean muy lejanos de la media para determinar estrategias usadas por grandes firmas.



05

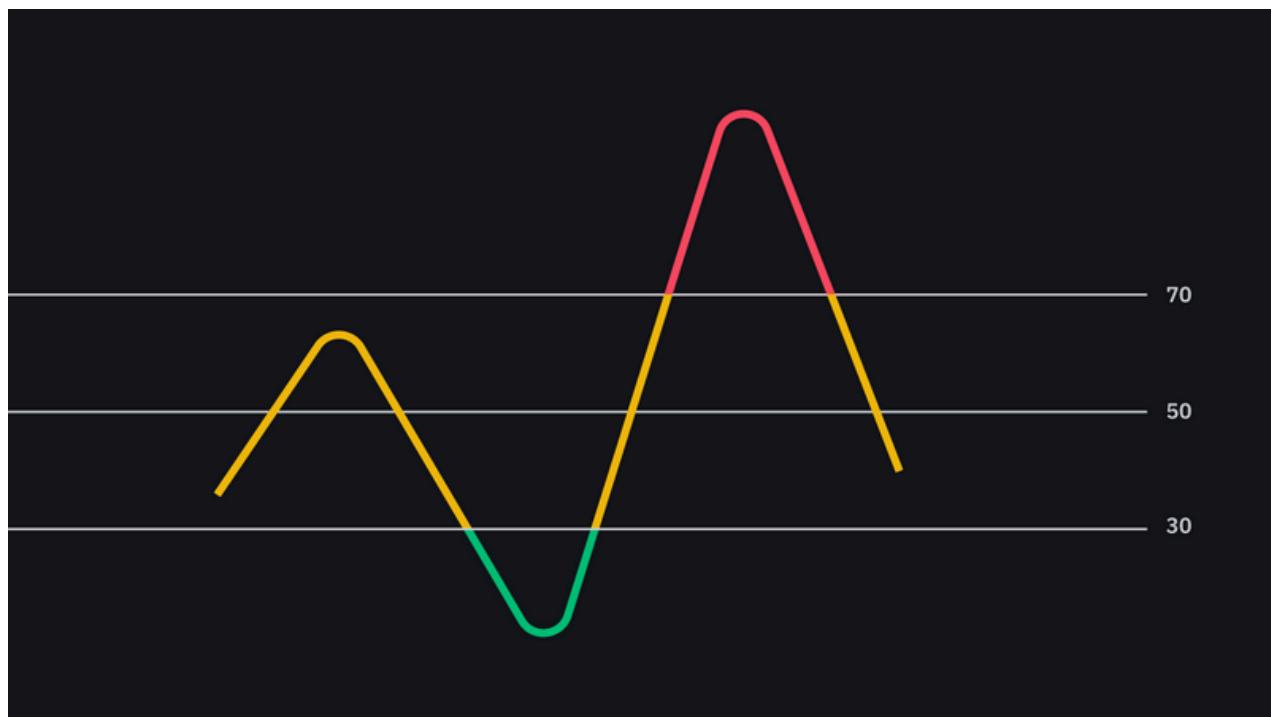
## Correlación de empresas

Medimos cómo se mueven dos activos en relación uno con el otro, lo que permite diversificar, observar tendencias fuertes y activos líderes.

# SELL - SEÑALES DE VENTA EN ACCIONES

## ¿Qué se realizó?

Un modelo que identifique señales de venta en acciones, basado en datos históricos de precios y métricas técnicas.



## ¿Qué información se utilizó?

- Dataset de 6640 registros con columnas: open, high, low, close, volume, RSI, MA\_5, MA\_20.
- Target: Señal de venta (1 si el precio cae  $\geq 2\%$  desde el máximo reciente).

# SELL

# DL - PROCEDIMIENTO

# BUY

```
▶ import pandas as pd
import numpy as np

#Cargar datos
df = pd.read_csv(AAPL_D1R)
df["datetime"] = pd.to_datetime(df["datetime"])

#Calcular el máximo de los últimos 5 días
df['max_5d'] = df['high'].rolling(5).max()

#Etiqueta: 1 si el precio cae >=2% desde el máximo reciente
df['signal'] = (df['close'] < 0.98 * df['max_5d']).astype(int)

#Eliminar filas con NaN (primeras 5 filas sin max_5d)
df = df.dropna()
```

- Sobrecomprado (RSI > 70)
- Sobrevendido (RSI < 30)

```
▶ #Rango diario (high - low)
df['range'] = df['high'] - df['low']

#Cambio diario (close - open)
df['change'] = df['close'] - df['open']

#Media móvil de 5 y 20 días
df['MA_5'] = df['close'].rolling(5).mean()
df['MA_20'] = df['close'].rolling(20).mean()

#RSI (Relative Strength Index)
delta = df['close'].diff()
gain = delta.clip(lower=0)
loss = -delta.clip(upper=0)
avg_gain = gain.rolling(14).mean()
avg_loss = loss.rolling(14).mean()
df['RSI'] = 100 - (100 / (1 + avg_gain / avg_loss))

#Eliminar filas con NaN (por indicadores)
df = df.dropna()

#features finales
features = ['open', 'high', 'low', 'close', 'volume', 'range', 'change', 'MA_5', 'MA_20', 'RSI']
```

$$RS = \frac{\text{Average Gain}}{\text{Average Loss}}$$
$$RSI = 100 - \frac{100}{(1 + RS)}$$

# DL - PROCEDIMIENTO

```
[89] from sklearn.preprocessing import MinMaxScaler
     from sklearn.model_selection import train_test_split

     #Normalización (escalar a [0, 1])
     scaler = MinMaxScaler()
     X_scaled = scaler.fit_transform(X)

     #entrenamiento 80% y prueba 20%
     X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2)

[90] import tensorflow as tf

     model = tf.keras.Sequential([
         tf.keras.layers.Input(shape=(X_train.shape[1],)),
         tf.keras.layers.Dense(128, activation='relu'),
         tf.keras.layers.Dense(64, activation='relu'),
         tf.keras.layers.Dense(32, activation='relu'),
         tf.keras.layers.Dense(1, activation='sigmoid')
     ])

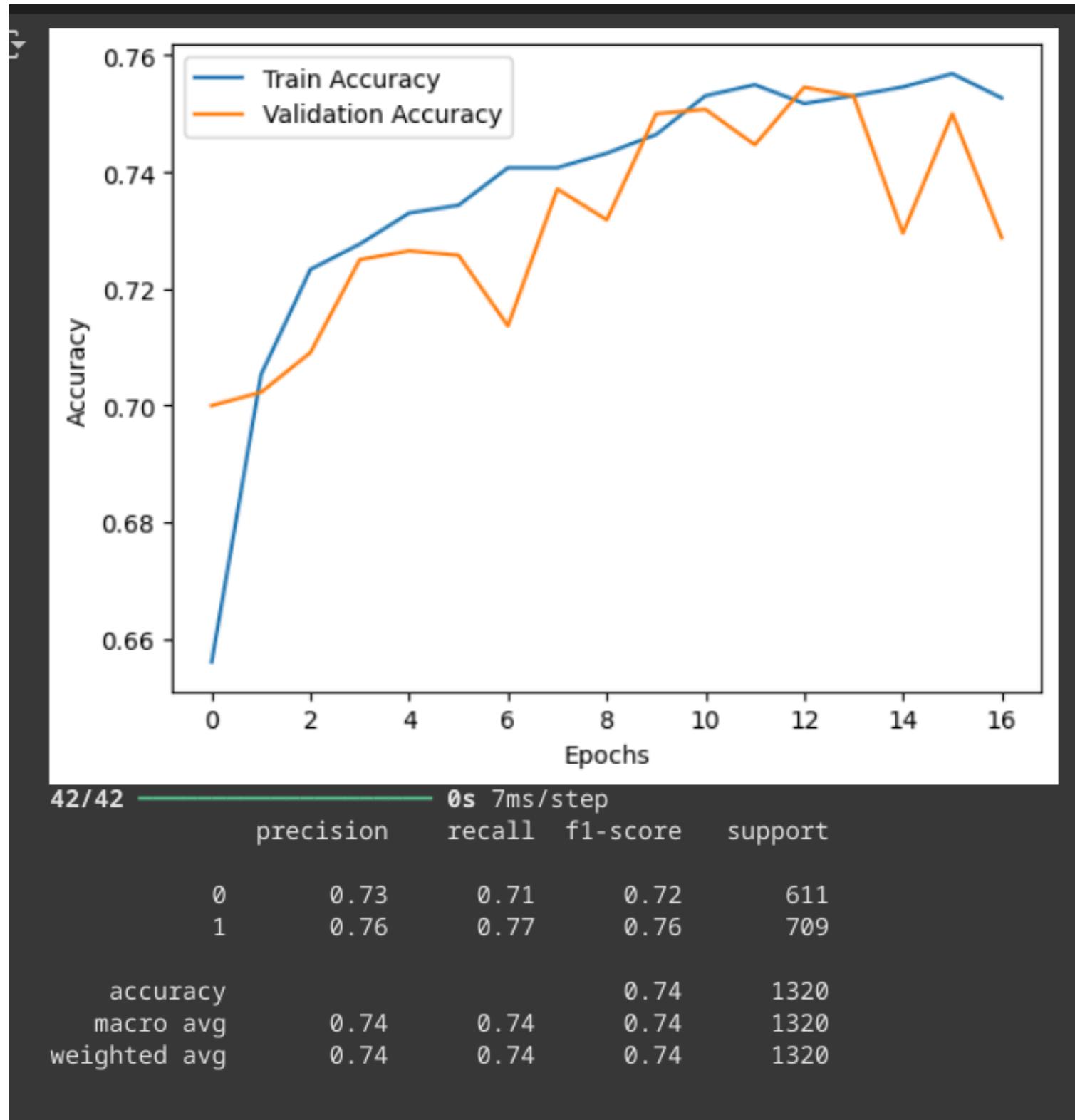
     model.compile(
         optimizer=tf.keras.optimizers.Adam(),
         loss='binary_crossentropy',
         metrics=['accuracy']
     )
```

- Train 80%
- Test 20 %
- 3 Capas ocultas
- Optimizador Adam

```
▶ from tensorflow.keras.callbacks import EarlyStopping
  #Parar el entrenamiento si no mejora en 5 épocas
  early_stop = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)

  history = model.fit(
      X_train, y_train,
      epochs=20,
      batch_size=100,
      validation_data=(X_test, y_test),
      callbacks=[early_stop],
      verbose=1
  )
```

# DL - RESULTADOS

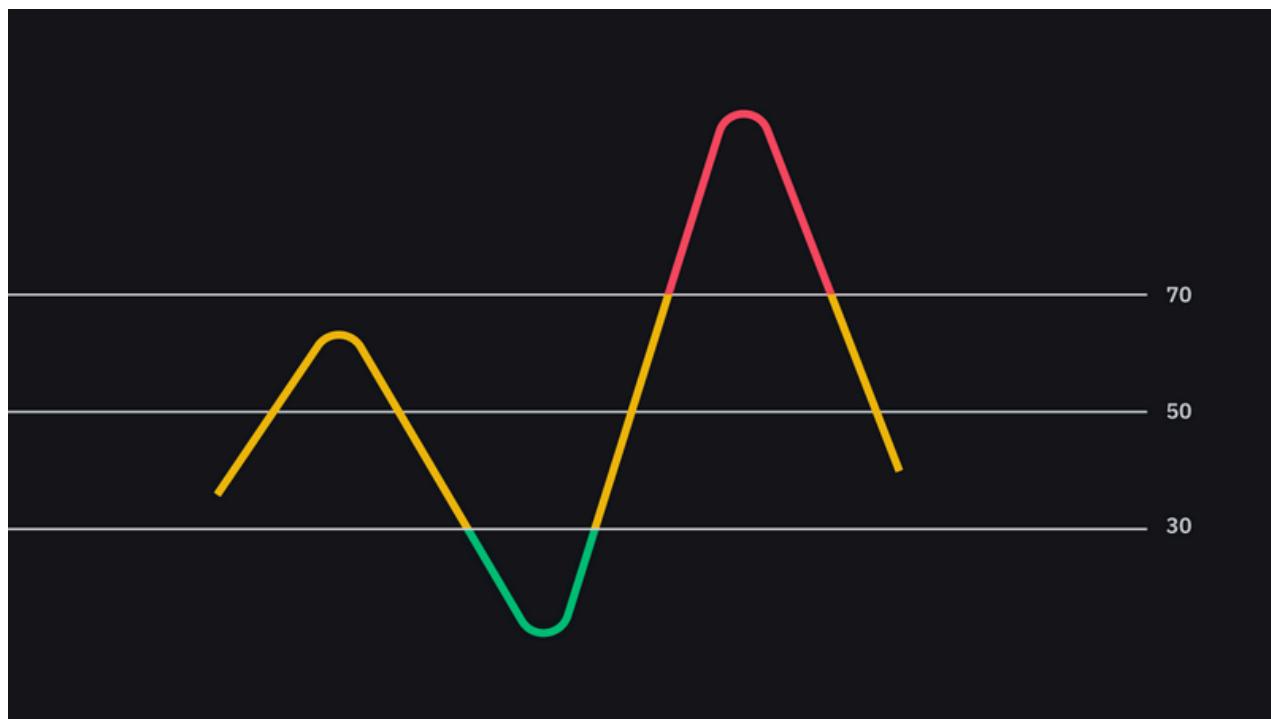


- 74% de accuracy global
- Estabilidad entre train/validation
- Buen desempeño en la clase 1 (vender)
- Buen equilibrio (F1-score)

# SELL ML - DETECCION DE TOMA DE LIQUIDEZ

## ¿Qué se realizó?

Un modelo usando random forest, para lograr identificar los eventos de toma de liquidez.



## ¿Qué información se utilizó?

- Dataset de 6640 registros con columnas: open, high, low, close, volume
- Target: Sweep (1 si hubo un sweep, 0 si no hubo).

# ML- PROCEDIMIENTO

```
lookback = 5
df["prev_low"] = df["low"].shift(1).rolling(window=lookback).min()
df["prev_high"] = df["high"].shift(1).rolling(window=lookback).max()

# 3. Detectar rupturas
df["broke_prev_low"] = df["low"] < df["prev_low"]
df["broke_prev_high"] = df["high"] > df["prev_high"]

# 4. Detectar reversión
reversal_strength = 0.005 # 0.5%
df["reversed_up"] = df["close"] > df["open"] * (1 + reversal_strength)
df["reversed_down"] = df["close"] < df["open"] * (1 - reversal_strength)

# 5. Detectar volumen alto
df["vol_threshold"] = df["volume"].rolling(window=20).quantile(0.8)
df["high_volume"] = df["volume"] > df["vol_threshold"]

# 6. Etiquetas
df["liquidity_grab_low"] = (df["broke_prev_low"] & df["reversed_up"] & df["high_volume"]).astype(int)
df["liquidity_grab_high"] = (df["broke_prev_high"] & df["reversed_down"] & df["high_volume"]).astype(int)
df["liquidity_grab"] = (df["liquidity_grab_low"] | df["liquidity_grab_high"]).astype(int)

# 7. Features
```

```
split_index = int(len(df) * 0.8)
x_train, x_test = x.iloc[:split_index], x.iloc[split_index:]
y_train, y_test = y.iloc[:split_index], y.iloc[split_index:]

# 11. Entrenamiento
clf = RandomForestClassifier(random_state=42, class_weight='balanced')
clf.fit(x_train, y_train)
```

# ML - RESULTADOS

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.98      | 0.71   | 0.82     | 260     |
| 1            | 0.07      | 0.60   | 0.13     | 10      |
| accuracy     |           |        | 0.71     | 270     |
| macro avg    | 0.53      | 0.66   | 0.48     | 270     |
| weighted avg | 0.95      | 0.71   | 0.80     | 270     |

- 71% de accuracy global
- Clases desequilibradas
- Muchas clase 0 (no hay toma)
- aumentar la sensibilidad a clase 1

SELL

¿LSTM?

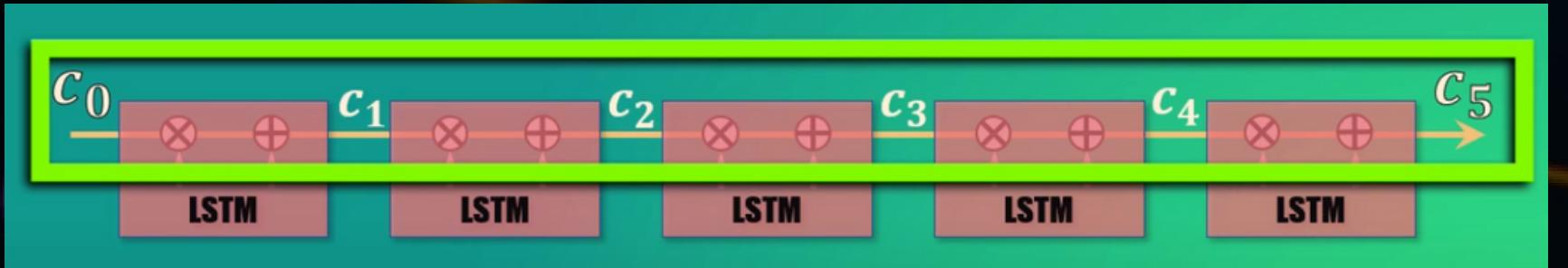
BUY

# LONG SHORT-TERM MEMORY

Tipo de red neuronal **recurrente**, diseñada para aprender patrones en secuencias de datos

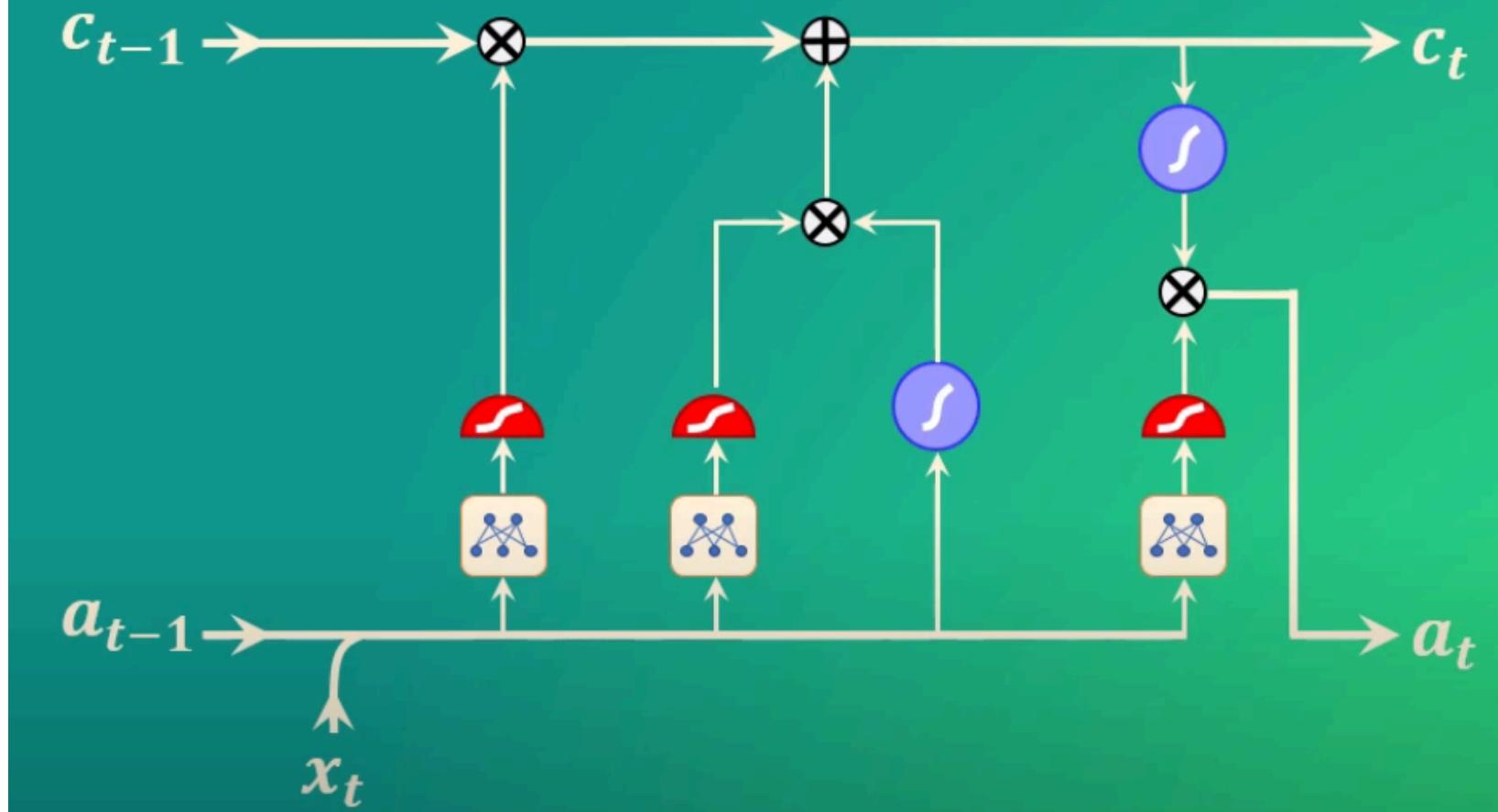
¿Qué se puede hacer?

- Recordar información
- Olvidar información
- Agregar información



# COMPONENTES

## UNA CELDA LSTM



- Forget gate
- Input gate
- Output gate

**SELL**

¿ROC?

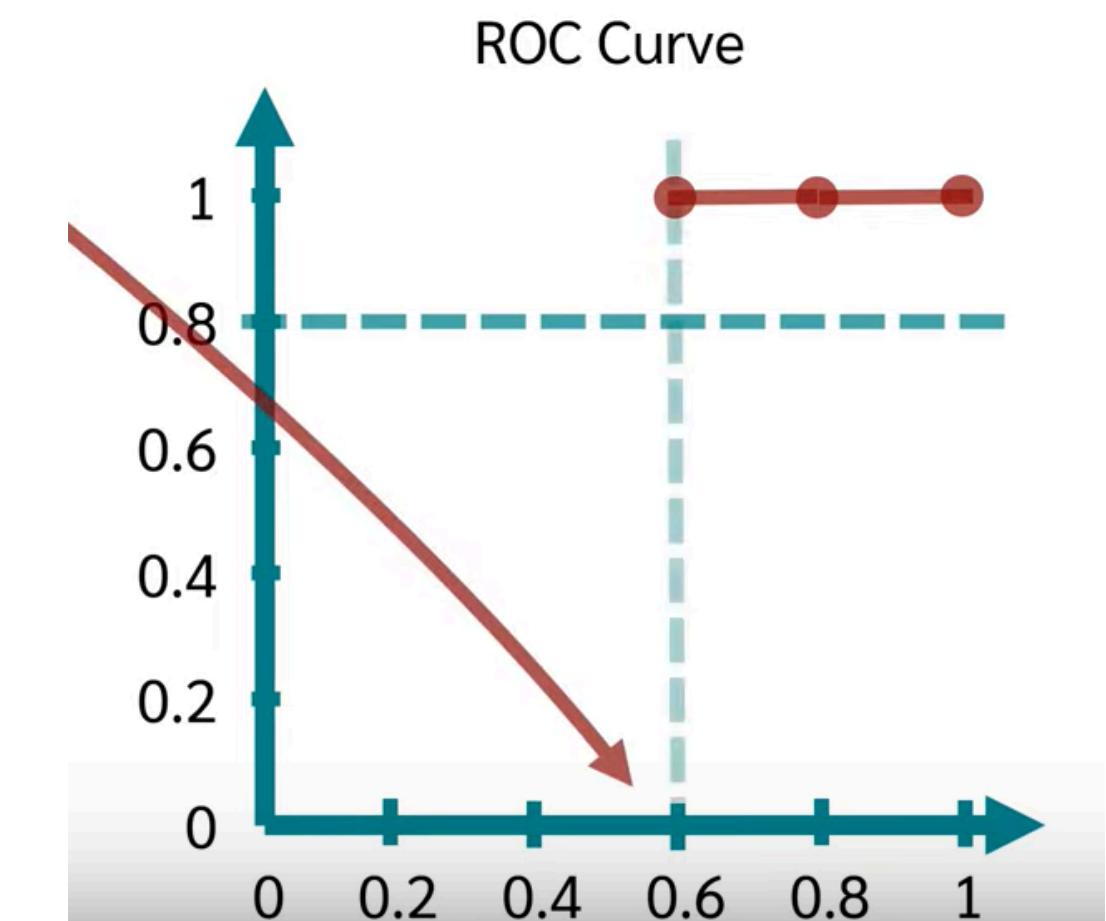
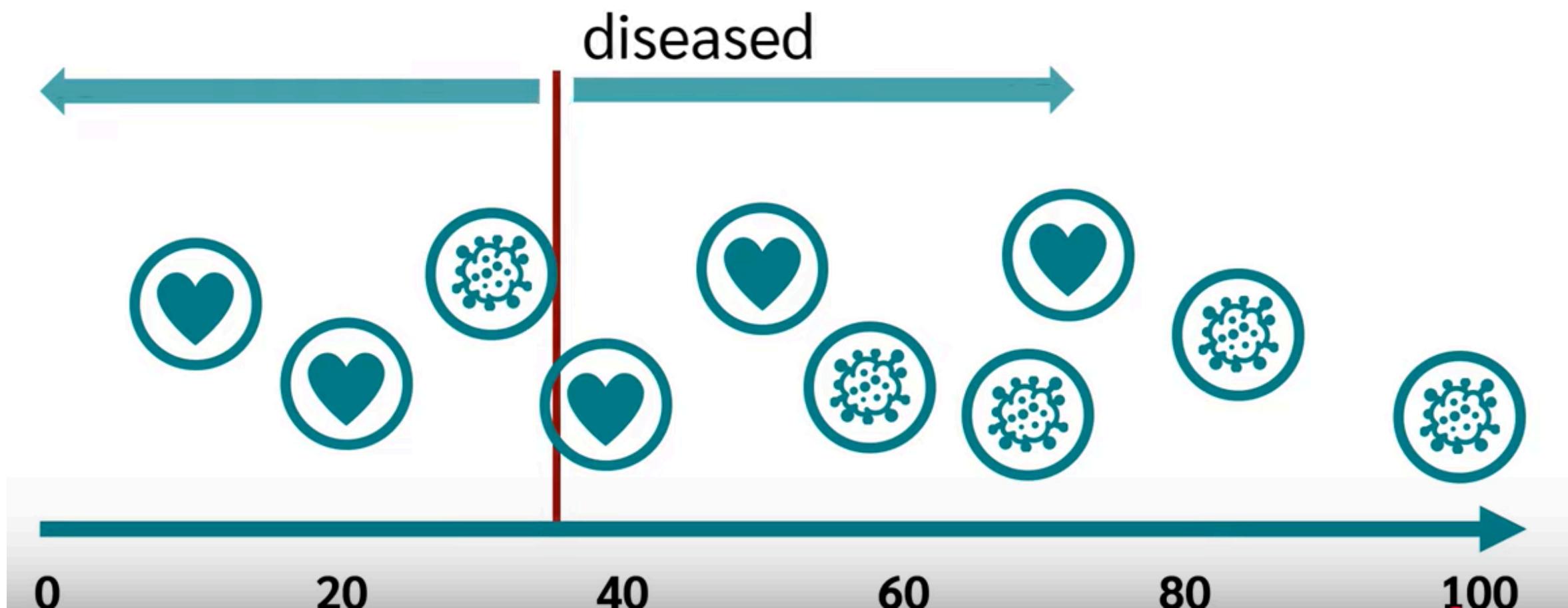
**BUY**

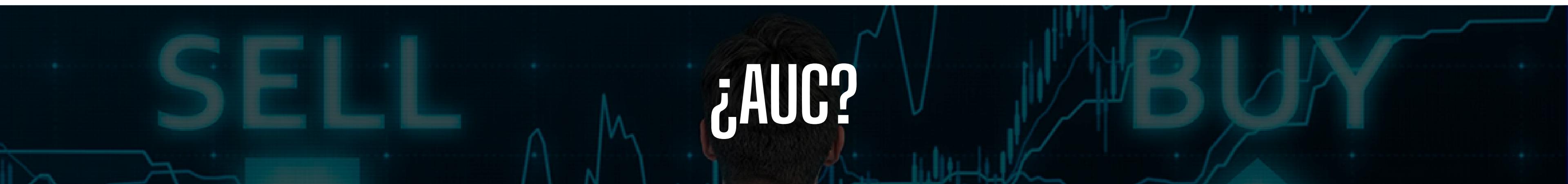


# R.O.C

## RECEIVING OPERATING CHARACTERISTIC

Ejemplo cancer



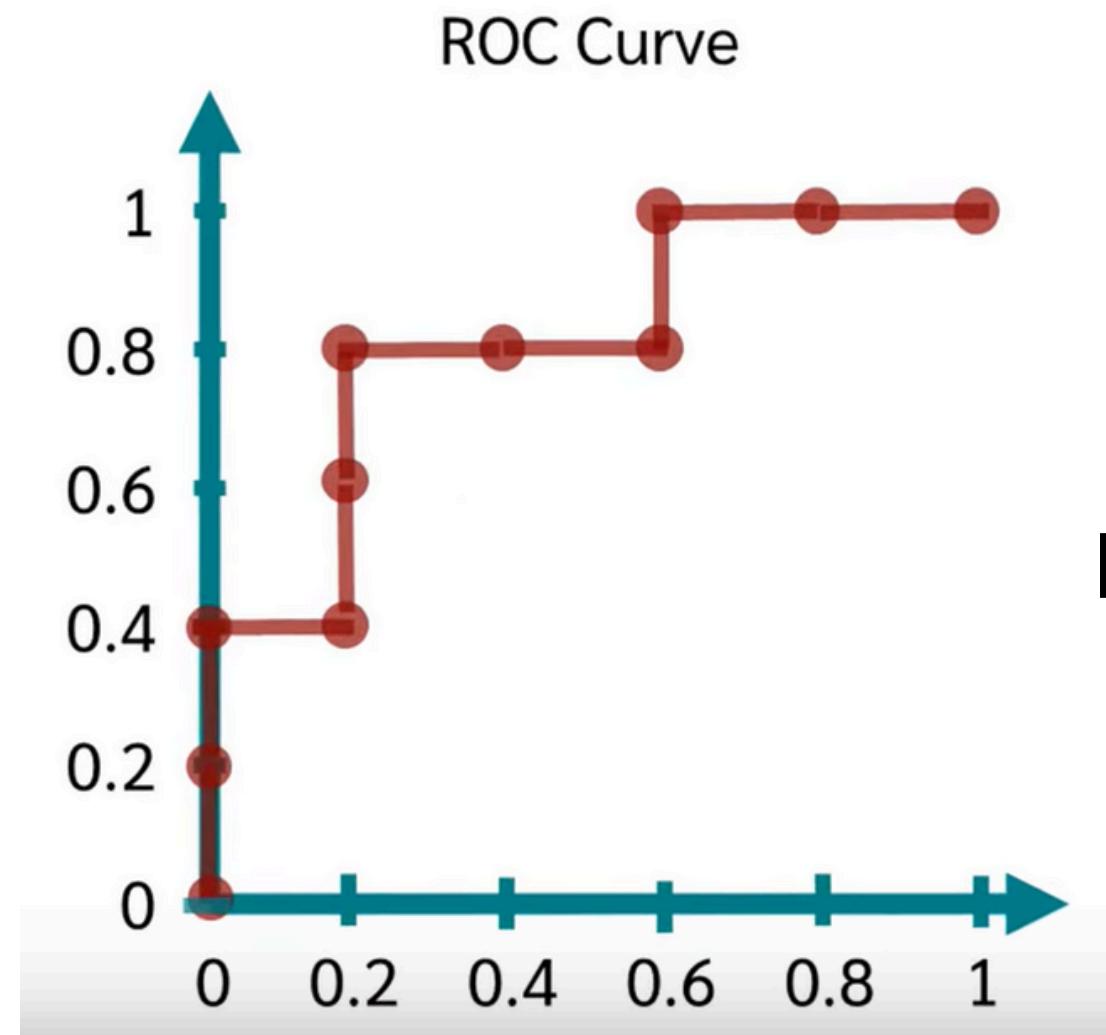


¿AUC?

# A.U.C

## AREA UNDER CURVE

Ejemplo cancer

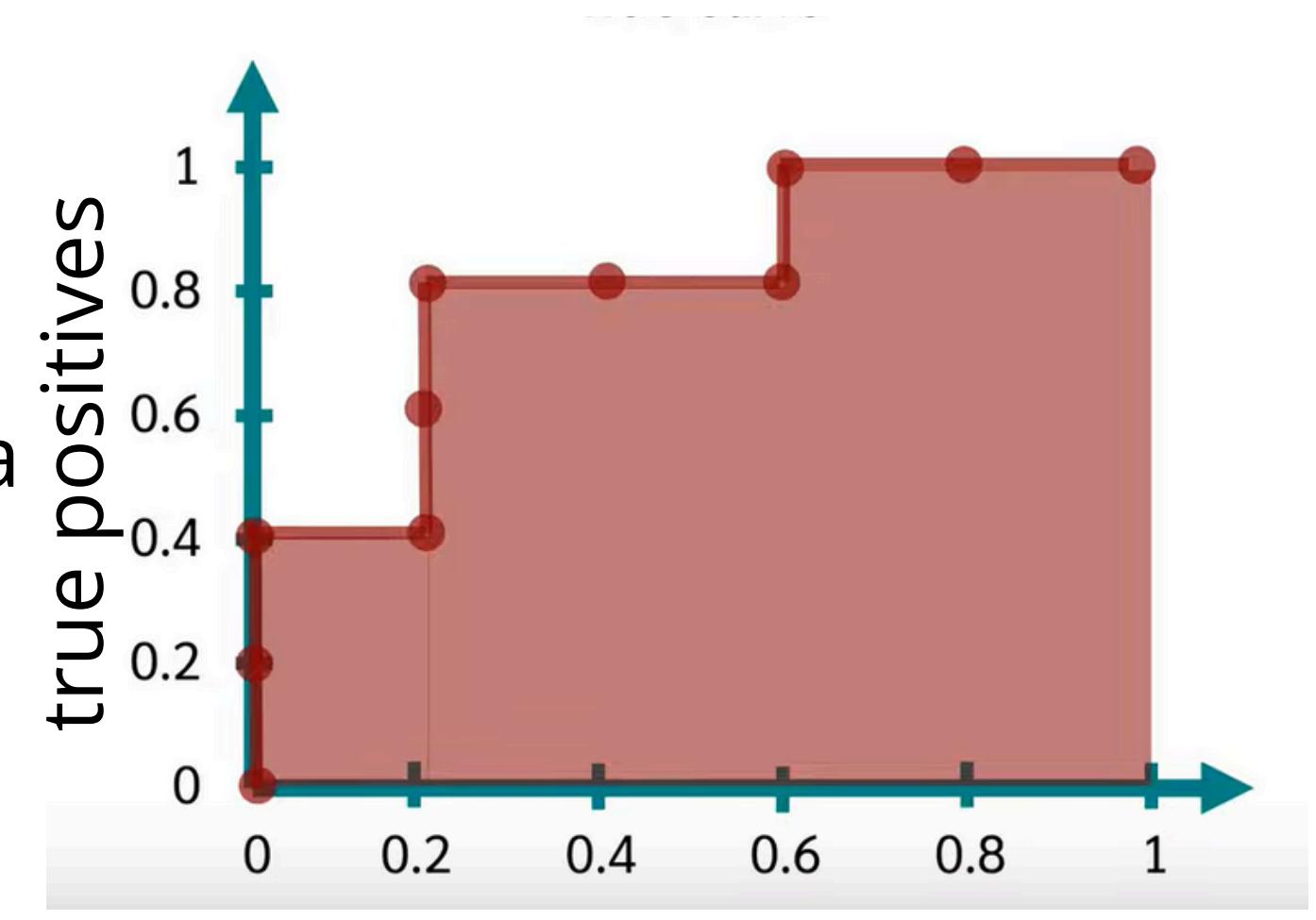


Entre más elevada sea la curva mejor el modelo

Mayor A.U.C →

precision

A.U.C



Mejor modelo

False Negatives

# R.O.C

## RECEIVING OPERATING CHARACTERISTIC

Ejemplo cancer

| AUC Score | Significado  |
|-----------|--|
| 1.0       | Clasificación perfecta (podría significar overfitting) |
| 0.9–1.0   | Excelente  |
| 0.8–0.9   | Muy bueno  |
| 0.7–0.8   | Bueno  |
| 0.6–0.7   | Regular  |
| 0.5       | Sin capacidad (equivalente a adivinar al azar)         |
| < 0.5     | Peor que adivinar al azar (modelo invertido)           |

# NUESTRA IMPLEMENTACIÓN



# ¿Cómo ingiere la serie de precios?

Tomamos bloques de 60 horas de información previa (precios e indicadores) para predecir un evento en las próximas 72 h.

Cada bloque incluye:

Precio de apertura, máximo, mínimo y cierre.

Volumen de negociación.

Indicadores técnicos:

- Retorno logarítmico hora a hora.

- Medias móviles exponenciales (EMA12, EMA26) y su diferencia (MACD).

- Índice de fuerza relativa (RSI).

- Bandas de Bollinger (superior e inferior).

- Cambio porcentual de volumen.

## 1. Ventanas de Entrada (lookback=60)

- $X[t]$  = datos de precio e indicadores de las 60 horas previas
- $y[t]$  = etiqueta binaria: ¿sube  $\geq 0.25\%$  en próximas 72 h?

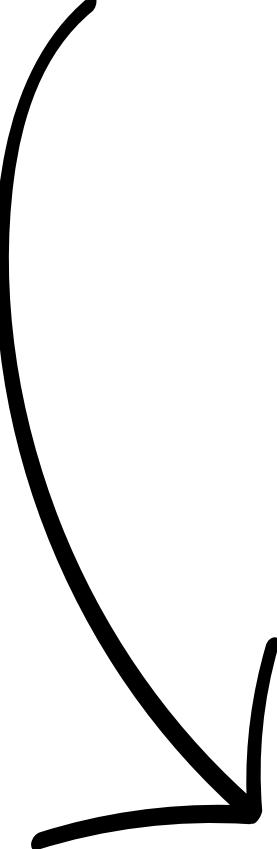
## 2. Los INDICADORES

- $\log_{\text{return}} = \ln(\text{close}_t / \text{close}_{t-1})$
- EMA12, EMA26, MACD = tendencias de corto vs. largo plazo
- RSI = fuerza relativa de compra/venta
- Bollinger Bands ( $\text{bb\_upper}$ ,  $\text{bb\_lower}$ ) = volatilidad
- Vol\_change = cambio porcentual de volumen

## 3. LAS WINDOWS

```
def create_windows(X, y, lookback=60):
    Xs, ys = [], []
    for i in range(lookback, len(X)):
        Xs.append(X[i-lookback:i])
        ys.append(y[i])
    return np.array(Xs), np.array(ys)
```

## RED LSTM

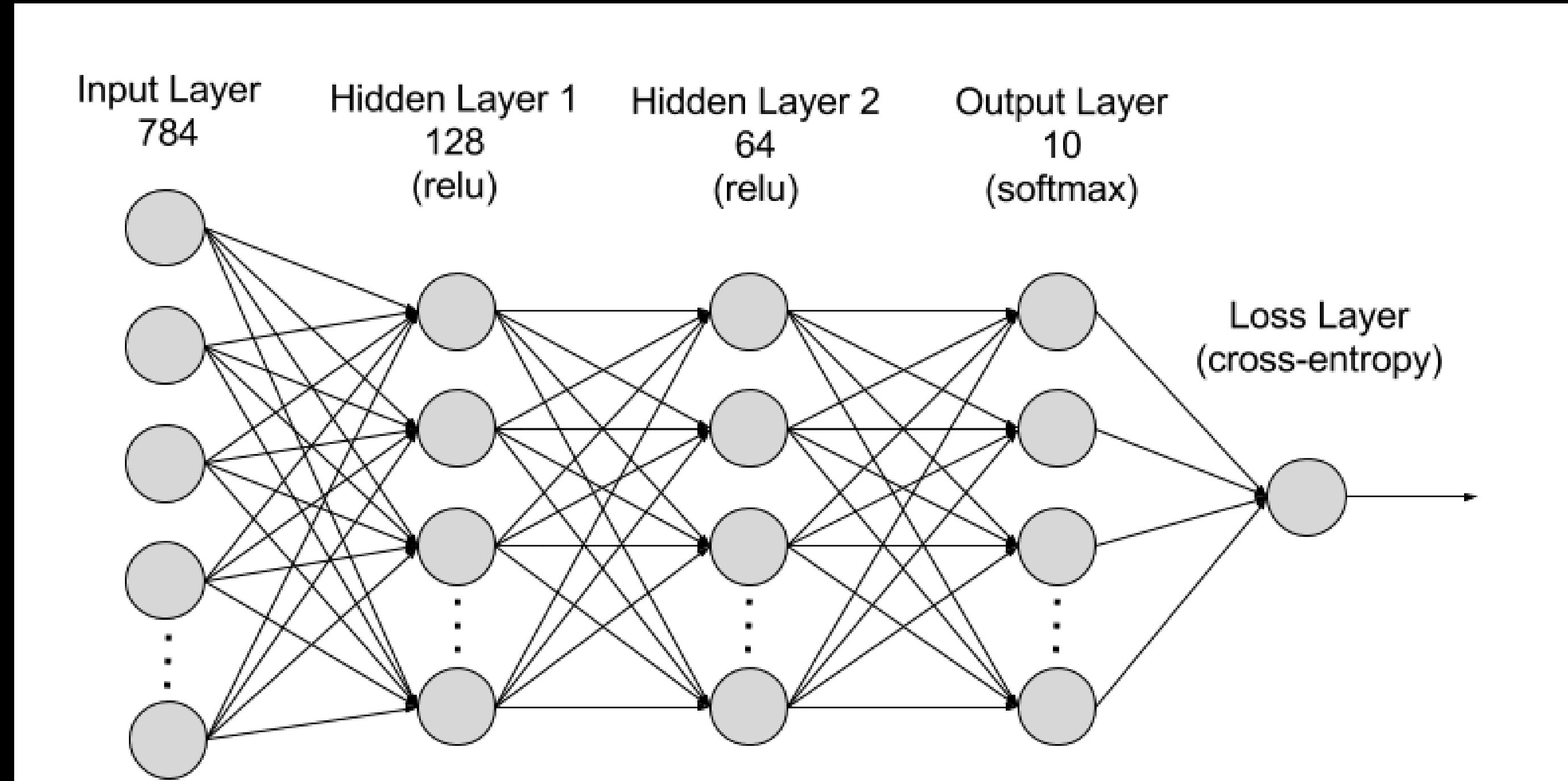


Red LSTM: memoria interna que captura patrones en la secuencia de precios:

1. Capa LSTM de 64 unidades (retiene la secuencia completa).
2. Capa LSTM de 32 unidades (resume en un vector de estado).
3. Capa densa de 16 neuronas (filtra patrones no lineales).
4. Salida sigmoide: probabilidad de “subida exitosa”.



```
def build_model(input_shape):  
    inp = Input(shape=input_shape)  
    x = LSTM(64, return_sequences=True)(inp)  
        x = LSTM(32)(x)  
        x = Dense(16, activation='relu')(x)  
        out = Dense(1, activation='sigmoid')(x)  
        model = Model(inp, out)  
        model.compile(  
            loss='binary_crossentropy',  
            optimizer='adam',  
            metrics=['accuracy',  
tf.keras.metrics.AUC(name='auc'),  
tf.keras.metrics.Precision(),  
tf.keras.metrics.Recall()  
        )  
    return model
```

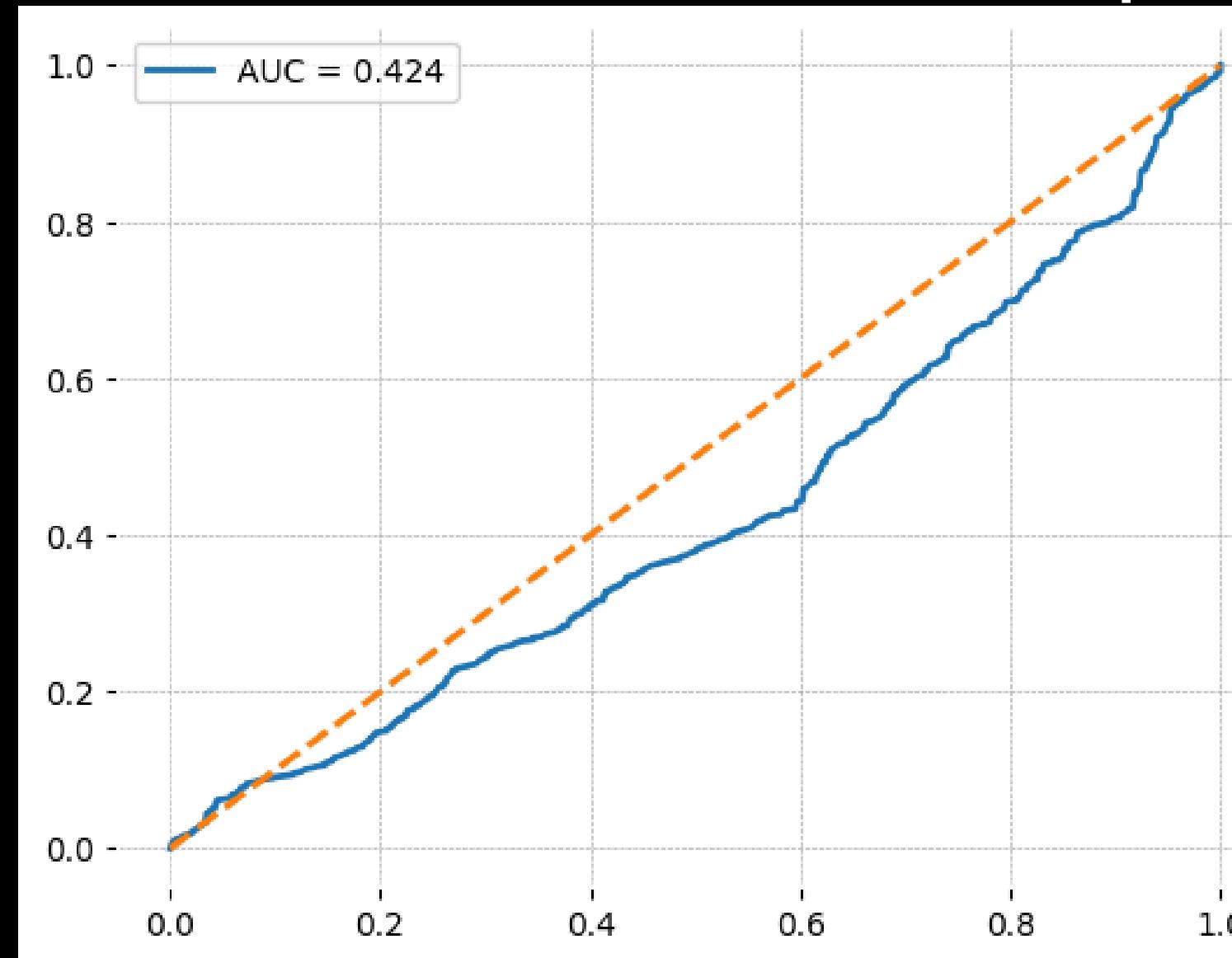


TPR(TRUE POSITIVE RATE)  
FPR

Una red de memoria a corto-largo plazo (LSTM) es RN recurrente.

# ¿Resultados?

<https://colab.research.google.com/drive/1fyeNVWpRJVeEirMZI80GRM3eCL4svVNq?usp=sharing>



## ALEATORIO?

“Los defensores del papel de las computadoras en las bolsas de valores argumentan que en realidad esos programas le imprimen eficiencia a los mercados al hacer más fáciles las transacciones, pese a los ocasionales colapsos.

Matt Wall, de Alpha Development, dice que el más reciente derrumbe de la bolsa “ocurrió muy, muy rápido”.

“Empezó faltando quince minutos para las 15:00, y en cosa de un par de minutos, el índice Dow Jones de la bolsa había perdido 600 puntos, la segunda caída más grande en un día en toda la historia”.

“A los 20 minutos ya se había recuperado”.

[https://www.bbc.com/mundo/noticias/2010/10/101005\\_transacciones\\_alta\\_velocidad\\_if](https://www.bbc.com/mundo/noticias/2010/10/101005_transacciones_alta_velocidad_if)  
2010

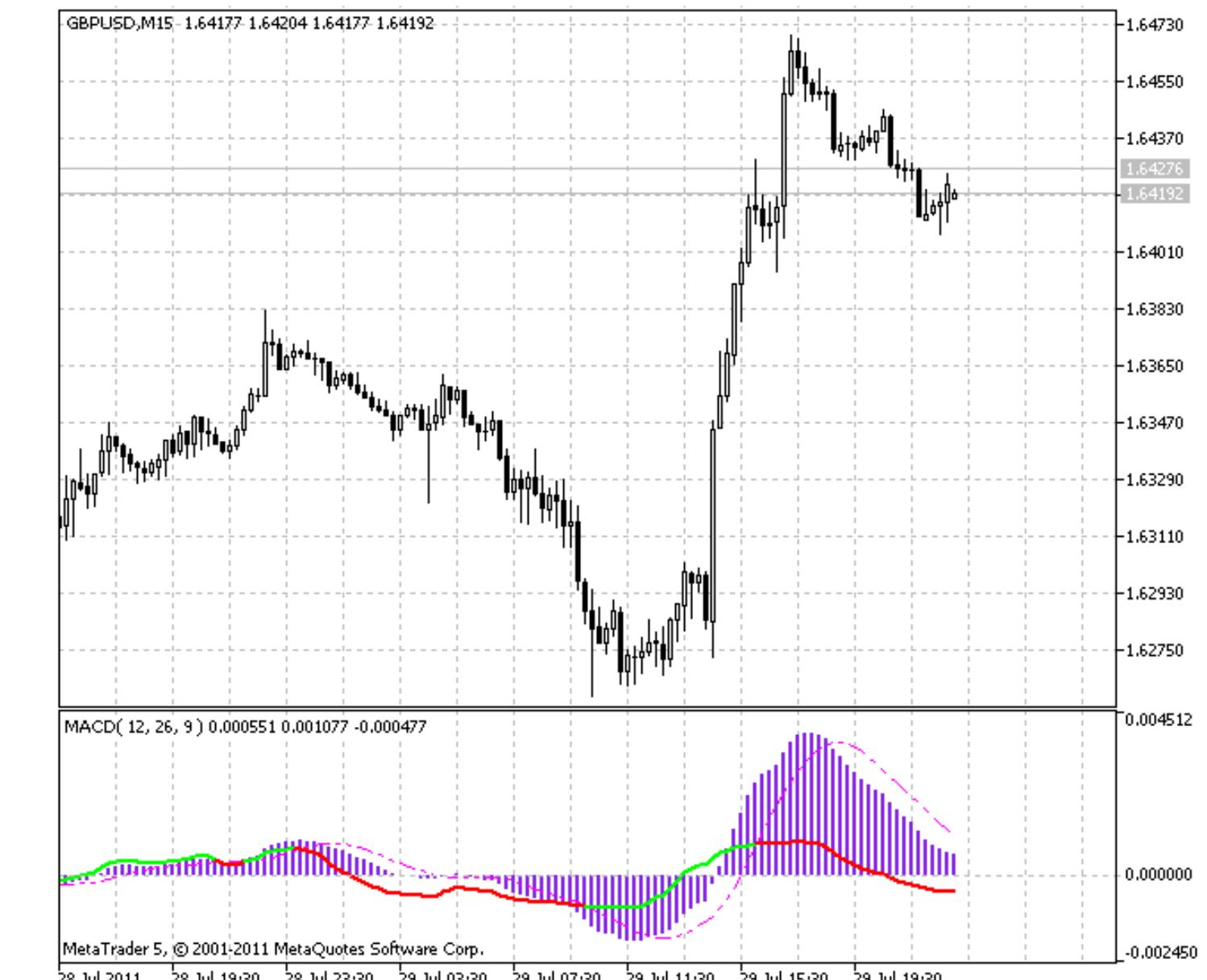




# MODELOS FINALES

# INDICADORES NUEVOS

- **RSI** : Fuerza de un movimiento
- **MACD**: convergencia de EMA
- **Señal alcista** : convergencia entre rsi y macd
- **Imbalance indicator**: Fvg + FIB
- **Price spike**: toma de liquidez



# SELL

# INDICADORES NUEVOS

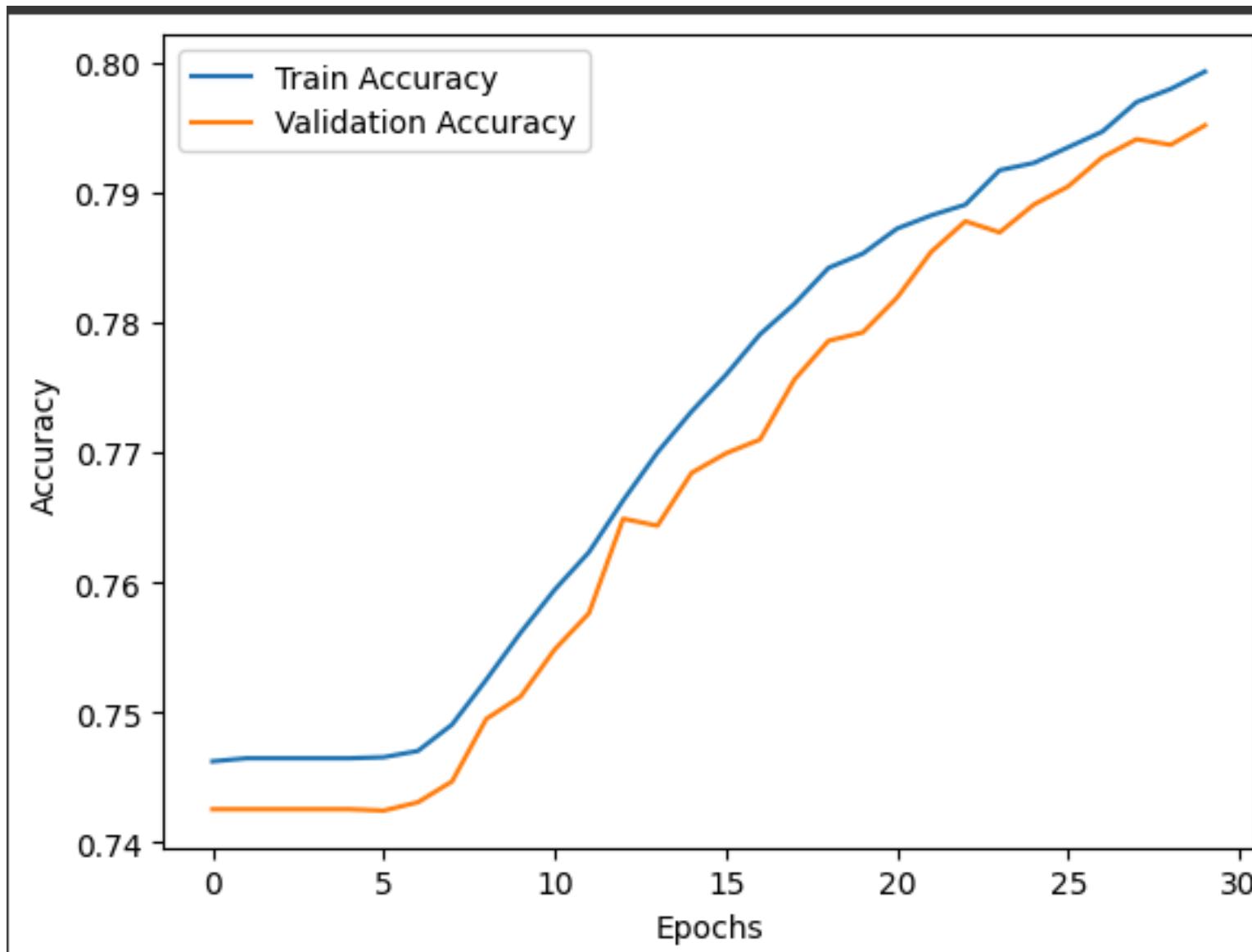
# BUY



# SELL DL - SEÑALES DE COMPRA

¿Qué se realizó?

Modelo de redes neuronales densas para detectar oportunidades de compra



|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.80      | 0.96   | 0.87     | 6939    |
| 1            | 0.73      | 0.32   | 0.45     | 2406    |
| accuracy     |           |        |          | 9345    |
| macro avg    | 0.77      | 0.64   | 0.66     | 9345    |
| weighted avg | 0.79      | 0.80   | 0.76     | 9345    |

¿Qué información se utilizó?

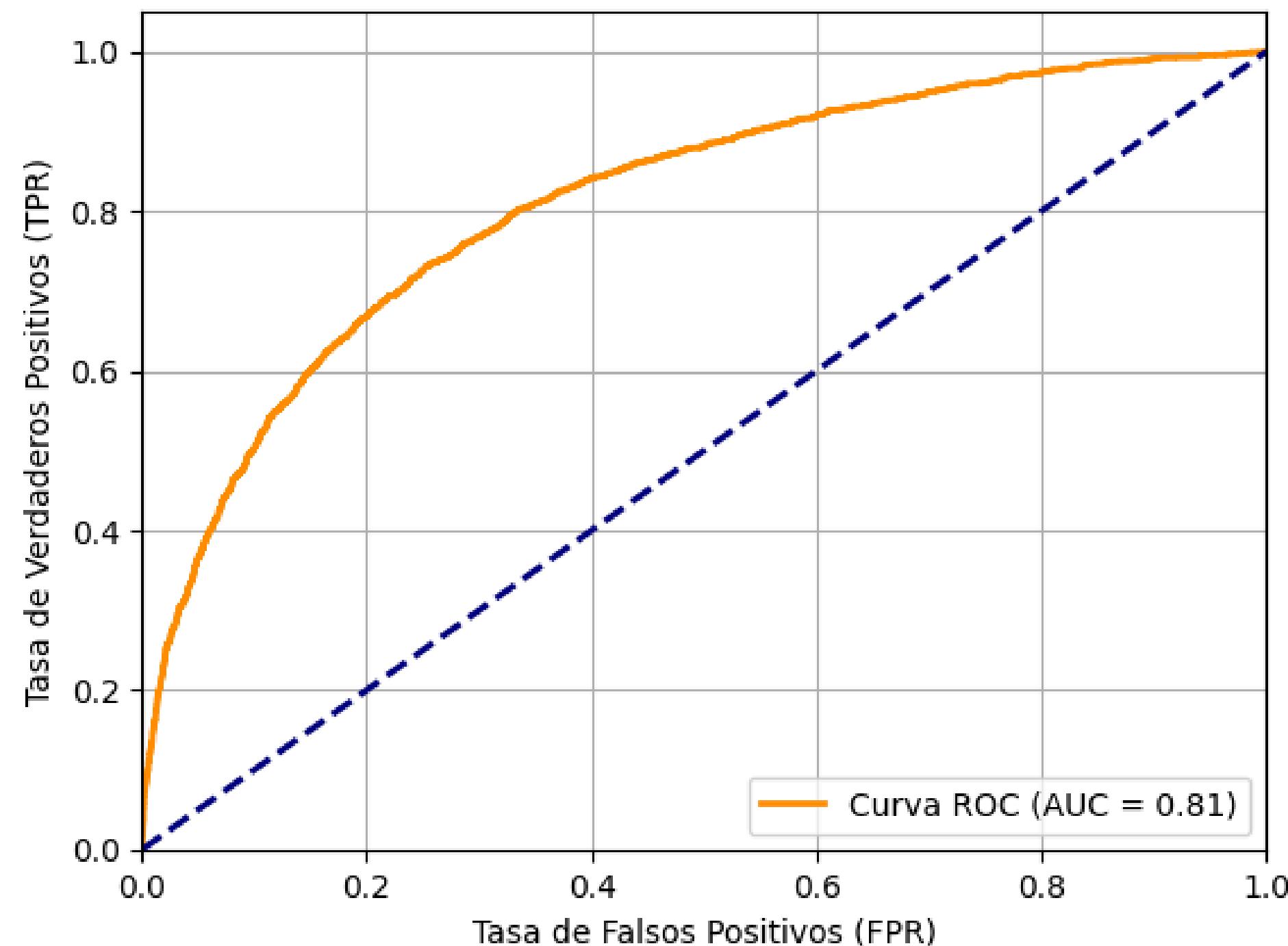
- Indicadores nuevos
- Target: Señal de venta (1 si el precio sube  $\geq 1\%$  desde el mínimo entre 3 velas.

# SELL

# AUC

# BUY

Curva ROC



# GRADIENT BOOSTING - SEÑALES DE COMPRA

¿Qué se realizó?

Modelo gradient boosting para detectar oportunidades de compra

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.87      | 0.94   | 0.90     | 6964    |
| 1            | 0.77      | 0.59   | 0.67     | 2381    |
| accuracy     |           |        | 0.85     | 9345    |
| macro avg    | 0.82      | 0.76   | 0.79     | 9345    |
| weighted avg | 0.85      | 0.85   | 0.84     | 9345    |

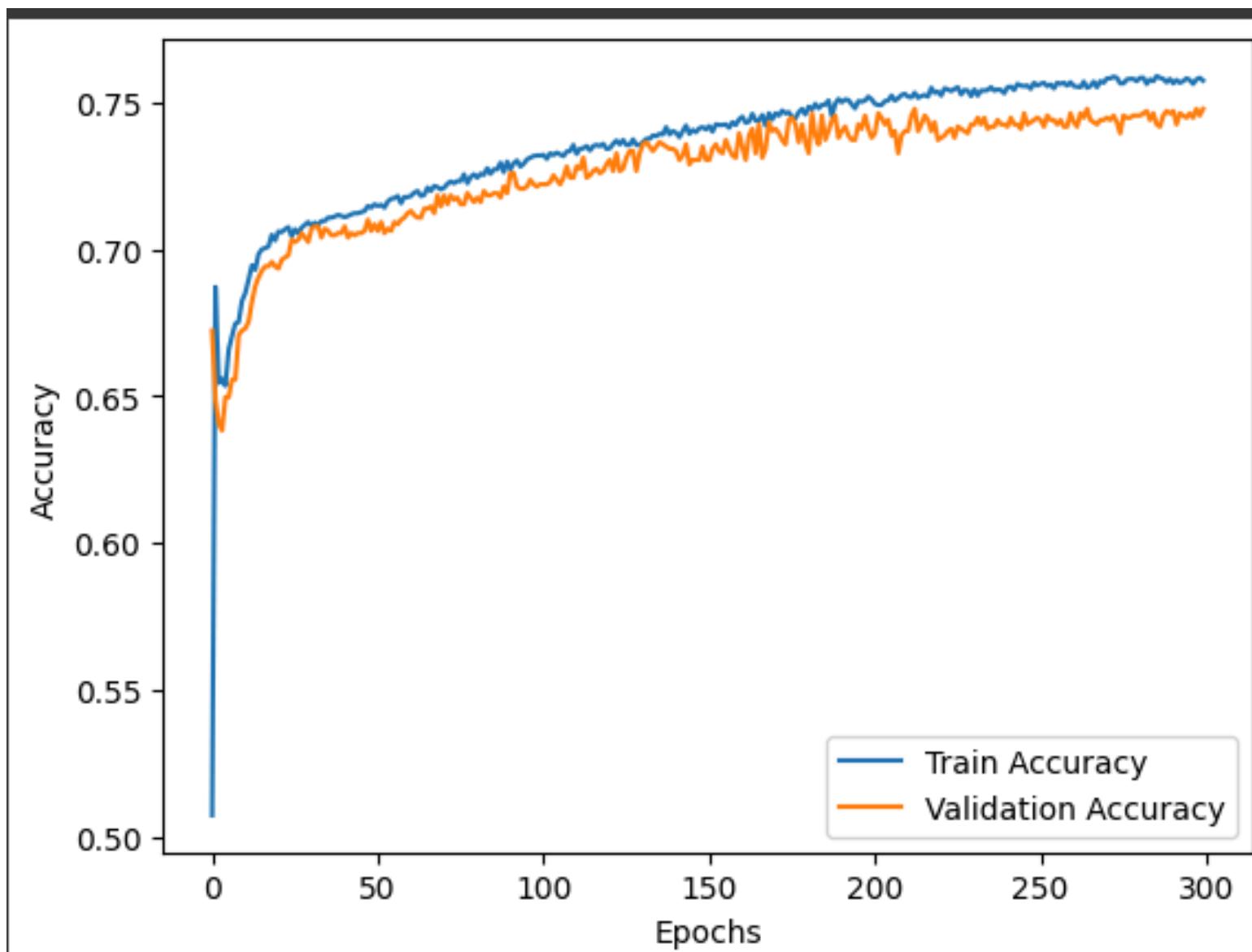
¿Qué información se utilizó?

- Indicadores nuevos
- Target: Señal de venta (1 si el precio sube  $\geq 1\%$  desde el mínimo entre 3 velas.

# SELL DL - SEÑALES DE VENTA BUY

¿Qué se realizó?

Modelo de redes neuronales densas para detectar oportunidades de venta



|              |      |      |      |      |
|--------------|------|------|------|------|
| 0            | 0.74 | 0.72 | 0.73 | 626  |
| 1            | 0.76 | 0.77 | 0.76 | 698  |
| accuracy     |      |      | 0.75 | 1324 |
| macro avg    | 0.75 | 0.75 | 0.75 | 1324 |
| weighted avg | 0.75 | 0.75 | 0.75 | 1324 |

¿Qué información se utilizó?

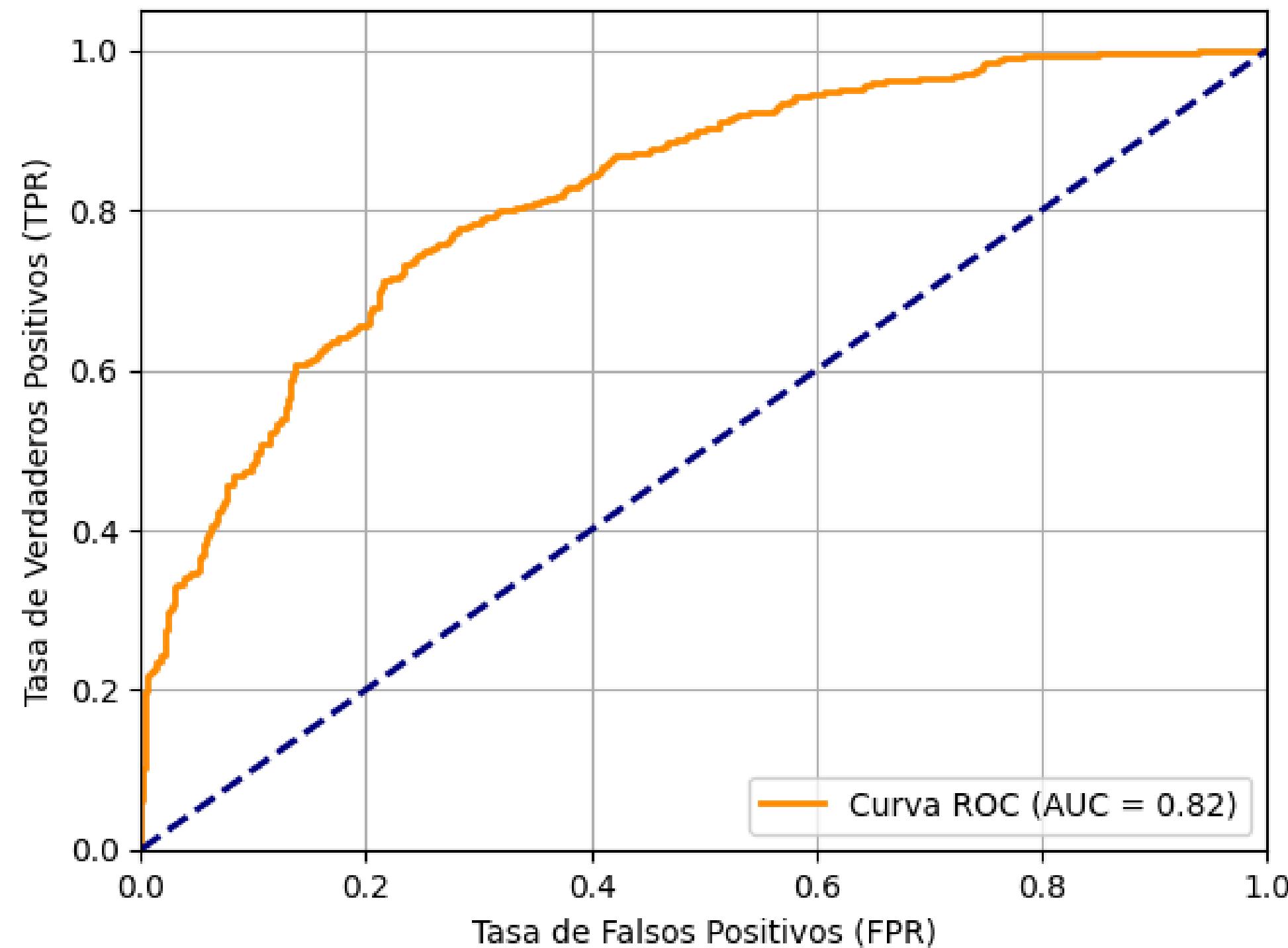
- Dataset, RSI, media móvil 5, media móvil 20
- Target: Señal de venta (1 si el precio baja  $\geq 2\%$  desde el máximo entre 5 velas.)

# SELL

# AUC

# BUY

Curva ROC



# SELL

# TABLA DE COMPARACIÓN

# BUY

| MODELO            | presición   | Recall      | accuracy | F1          | AUC       |
|-------------------|-------------|-------------|----------|-------------|-----------|
| gradient boosting | 87 / 80     | 0.95 / 0.60 | 0.86     | 0.91 / 0.69 | no aplica |
| DNN               | 0.80 / 0.73 | 0.96/0.32   | 0.76     | 0.87 /0.45  | 0.81      |
| LSTM              | 0.75 / 0.0  | 0.90 / 0    | 0.75     | 0.85 / 0.0  | 0.5       |

# KMEANS - PCA / PARA AGRUPACIÓN DE DÍAS

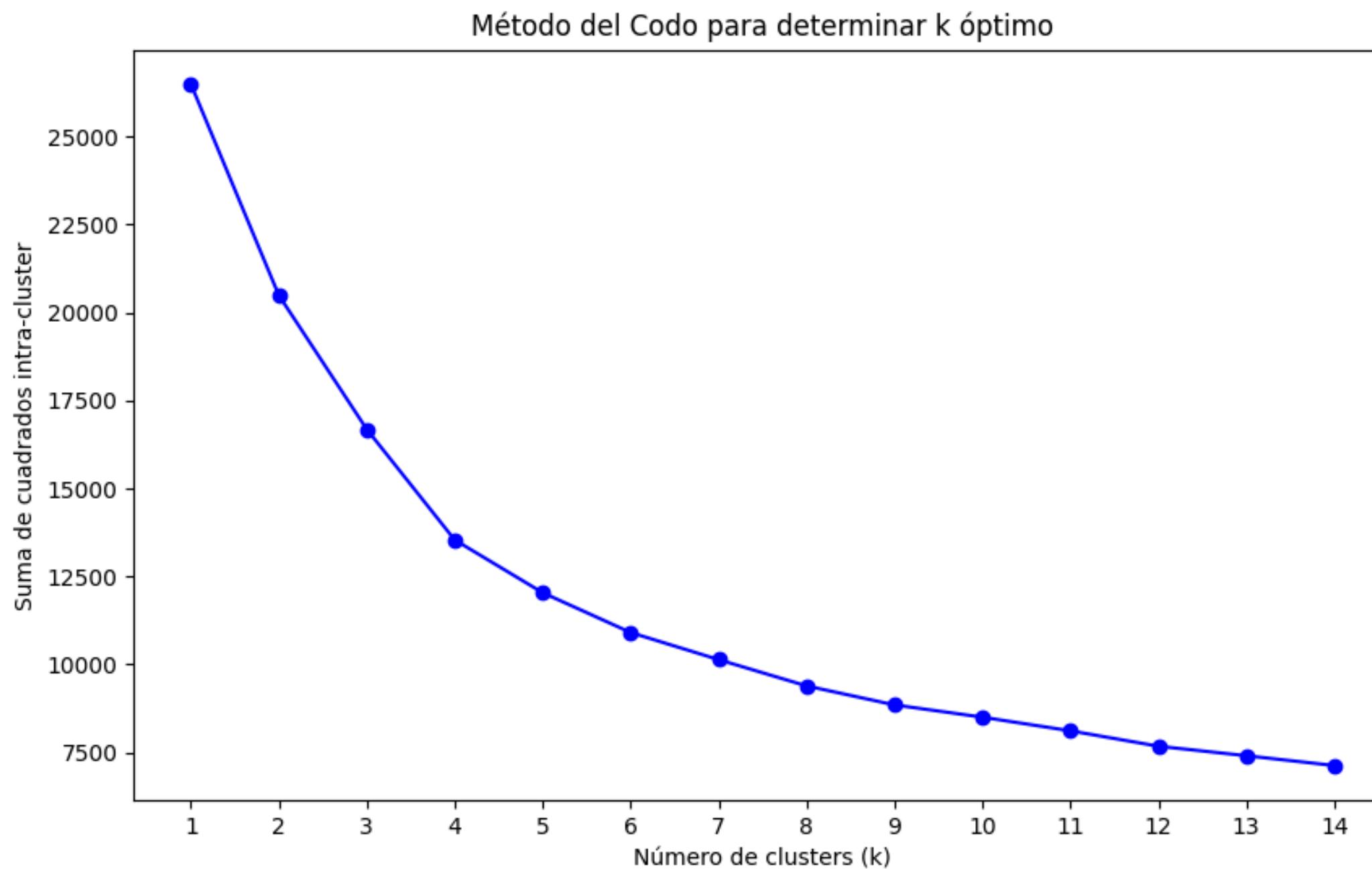
## Variables consideradas

- **Rango %:** Variación entre máximo y mínimo en porcentaje
- **Volumen:** Liquidez del día
- **Change %:** Variación porcentual entre apertura y cierre
- **RSI:** Índice de fuerza relativa



# KMEANS - PCA

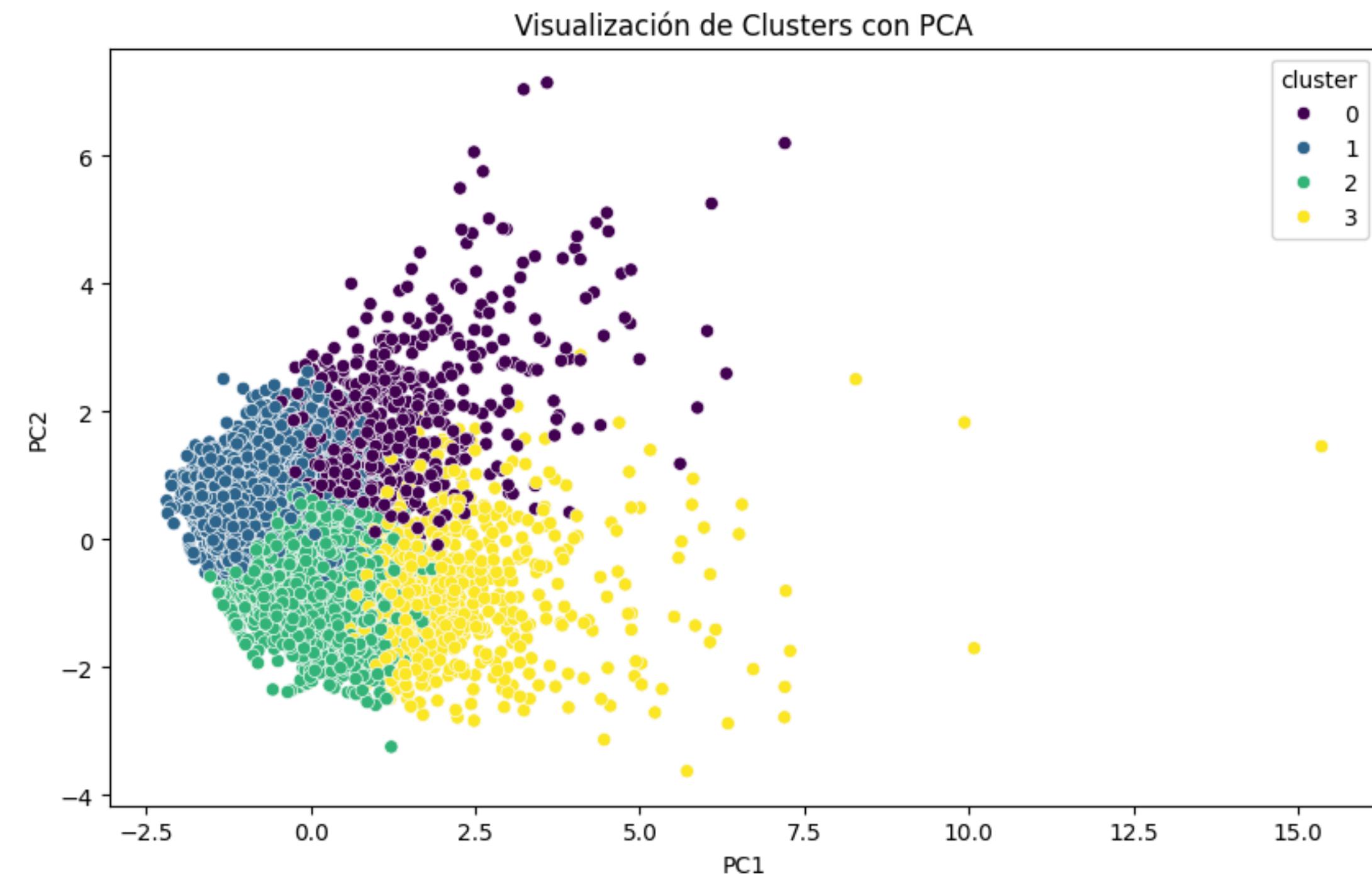
## Método del codo - elbow method



Determinamos el número óptimo de clústers, en este caso elegimos  $k=4$

# KMEANS - PCA

## Visualización 2D con PCA



# KMEANS - PCA

## ¿Qué podríamos inferir?

| cluster | volume       | range_pct | change_pct | RSI       |
|---------|--------------|-----------|------------|-----------|
| 0       | 1.666238e+08 | 6.129560  | 3.851819   | 56.650112 |
| 1       | 7.567157e+07 | 1.921177  | 0.272387   | 70.423125 |
| 2       | 7.066219e+07 | 2.313074  | -0.233140  | 41.054777 |
| 3       | 1.963408e+08 | 6.033213  | -3.167154  | 45.301149 |

- **Clúster 0:** Día alcista fuerte con impulso
- **Clúster 1:** Día de continuación suave
- **Clúster 2:** Corrección o bajada leve
- **Clúster 3:** Ventas masivas

Recordemos:

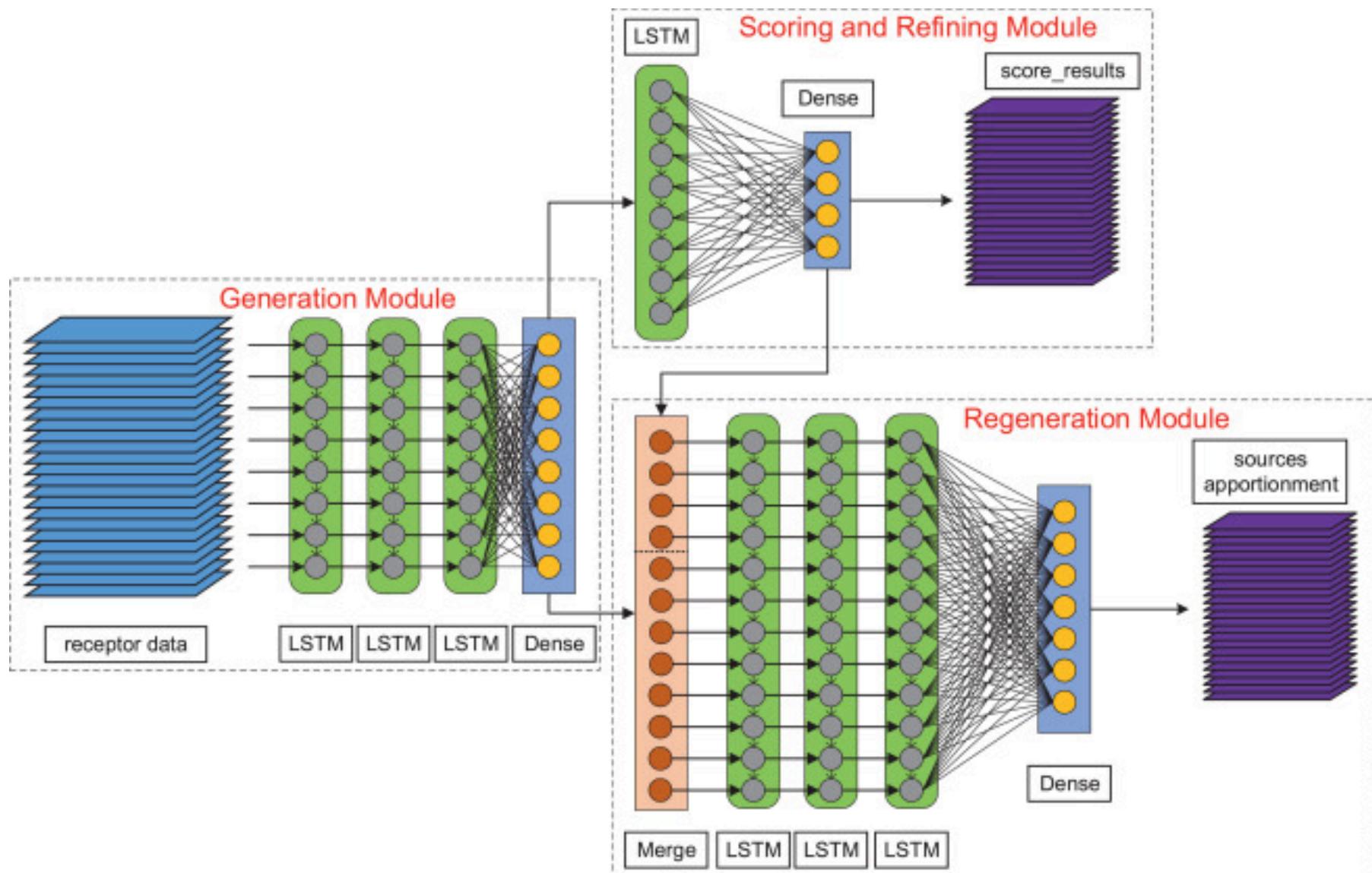
Sobrecomprado (RSI > 70) Sobrevenido (RSI < 30)

# SELL

# LSTM+ PYTORCH CUDA

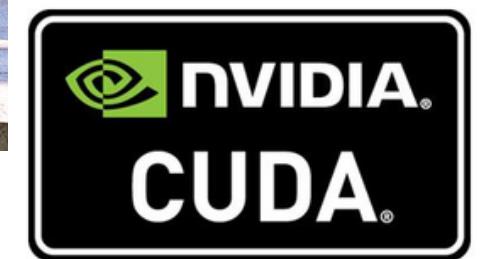
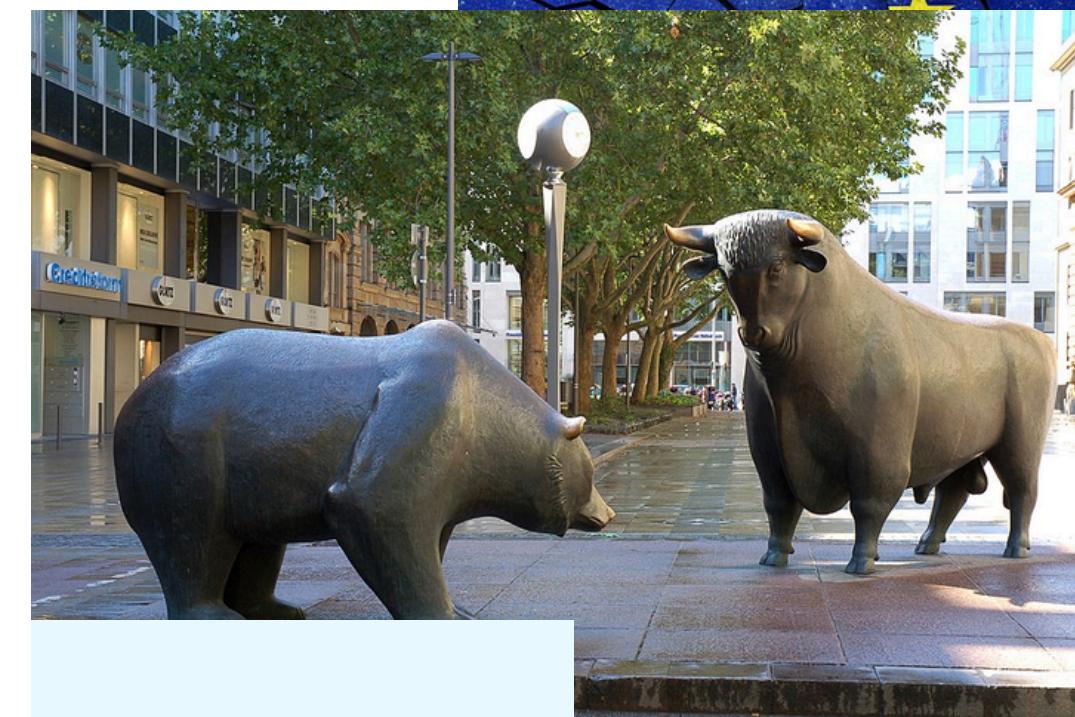
## ¿De que se trata?

# BUY



- CAMBIO EN LA CONFIGURACIÓN DE CAPAS
- USO DE DATASET DE 1 MINUTO
- AÑADIR MAS INDICADORES
- CAMBIAR ESTRATEGIA DE TRADING PARA IGUALAR CLASES

PyTorch





# SELL

# LSTM+ PYTORCH CUDA

## Diferencias al anterior

| Característica     | Código 1             | Código 2               |
|--------------------|----------------------|------------------------|
| Capas LSTM         | 3 (512→256→128)      | 2 (64→32)              |
| Capas Densas       | 128→64→1             | 16→1                   |
| Penalización de FP | BCE ponderado manual | ClassWeight automático |
| Activación final   | Logits + sigmoid     | Sigmoid directa        |

| Aspecto   | Código 1 (PyTorch) | Código 2 (Keras)   |
|-----------|--------------------|--------------------|
| Framework | PyTorch            | TensorFlow + Keras |

| Indicador     | Código 1                   | Código 2                  |
|---------------|----------------------------|---------------------------|
| Cantidad      | Más de 20                  | Solo 6                    |
| Ejemplos      | MACD, RSI, TEMA, TRIX, ATR | MACD, RSI, Bollinger, EMA |
| Volumen usado | No                         | No                        |

Cambia el enfoque. La prioridad es la precisión.

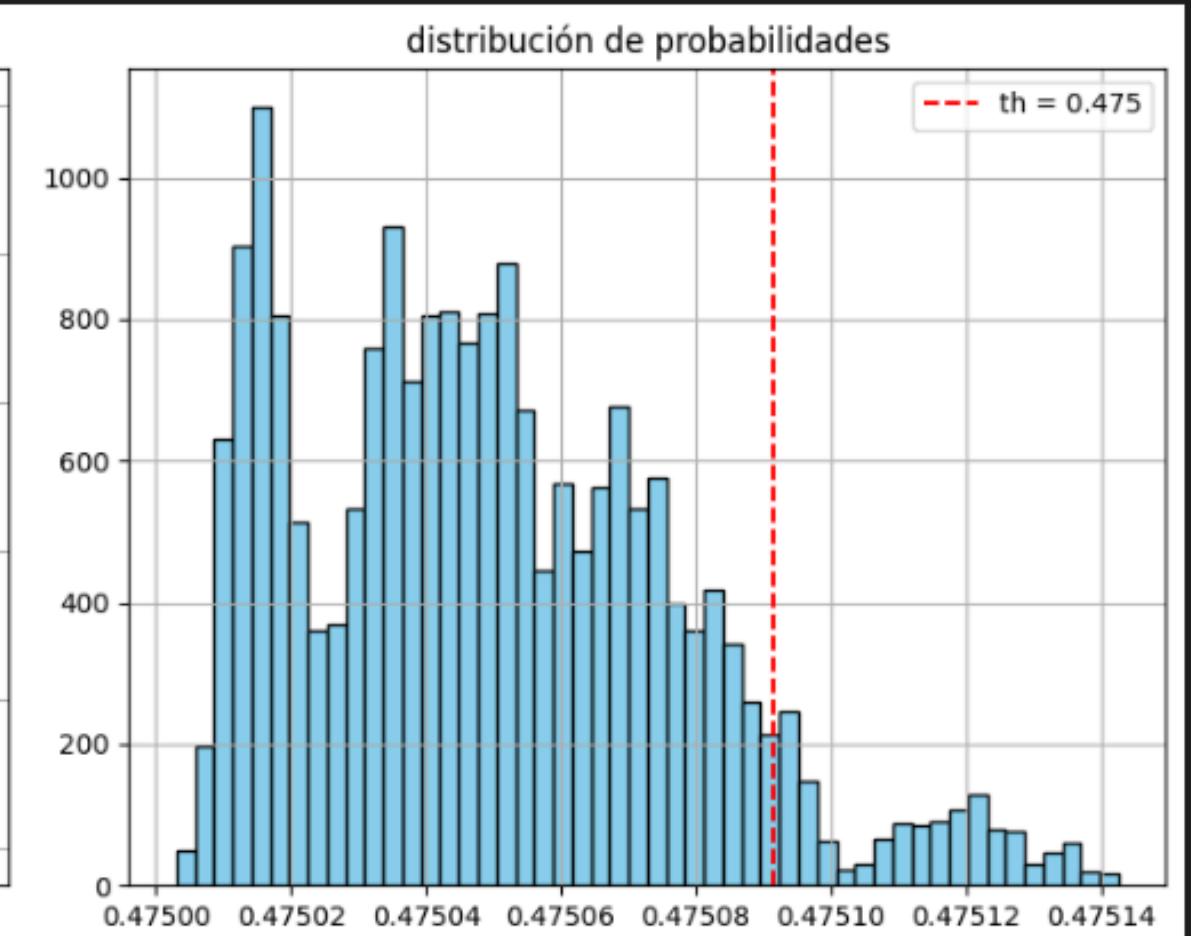
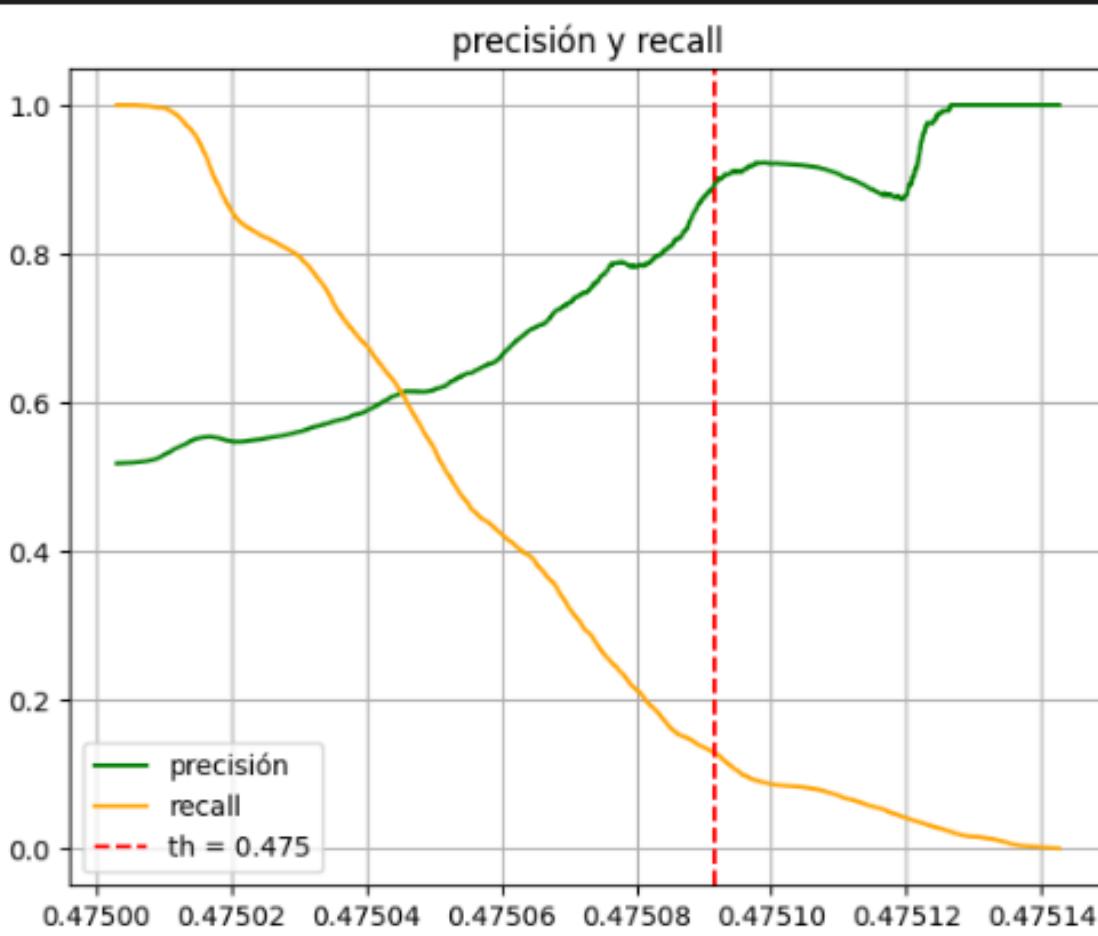
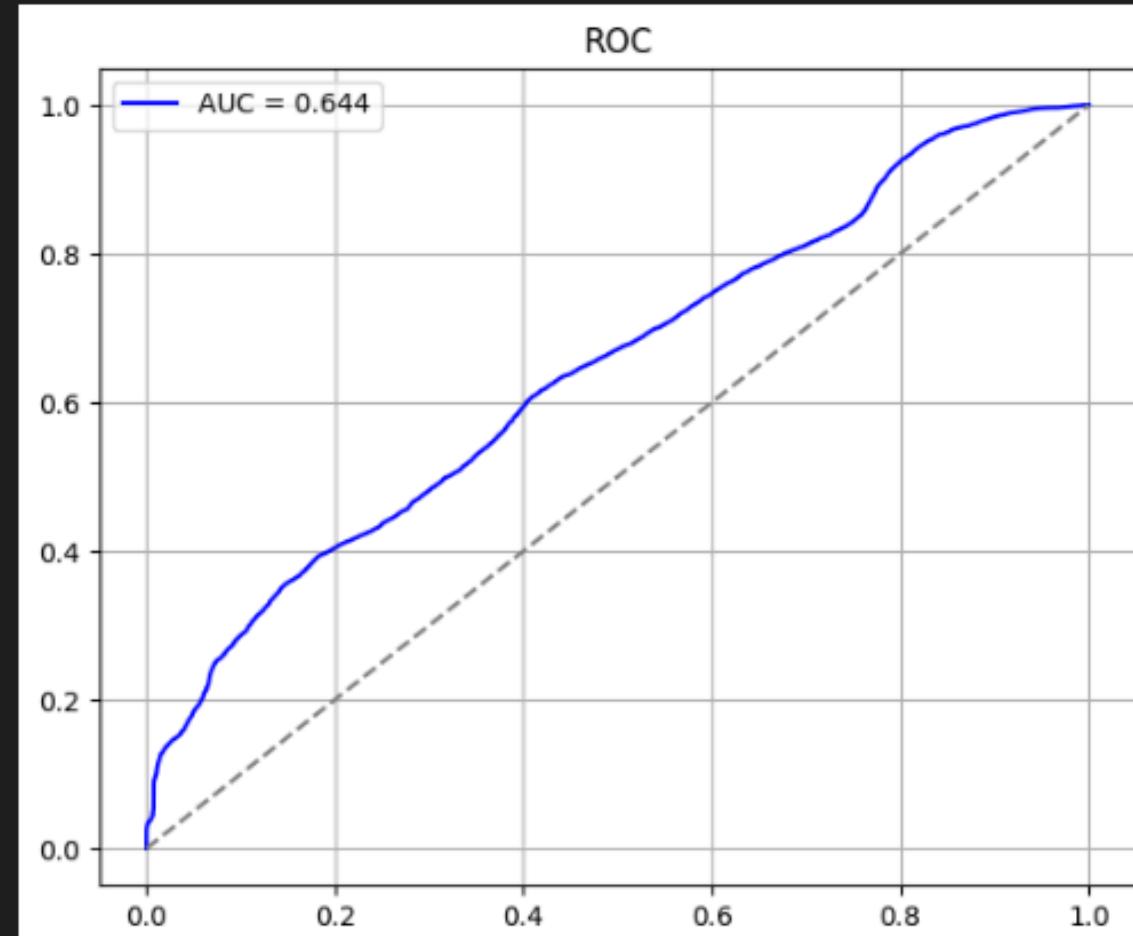
# LSTM+ PYTORCH CUDA

## RESULTADOS LSTM

TH = 0.475 | precisión = 0.891 | recall = 0.129 | f1 = 0.226 | acc = 0.541 | auc = 0.644

matriz de confusión:

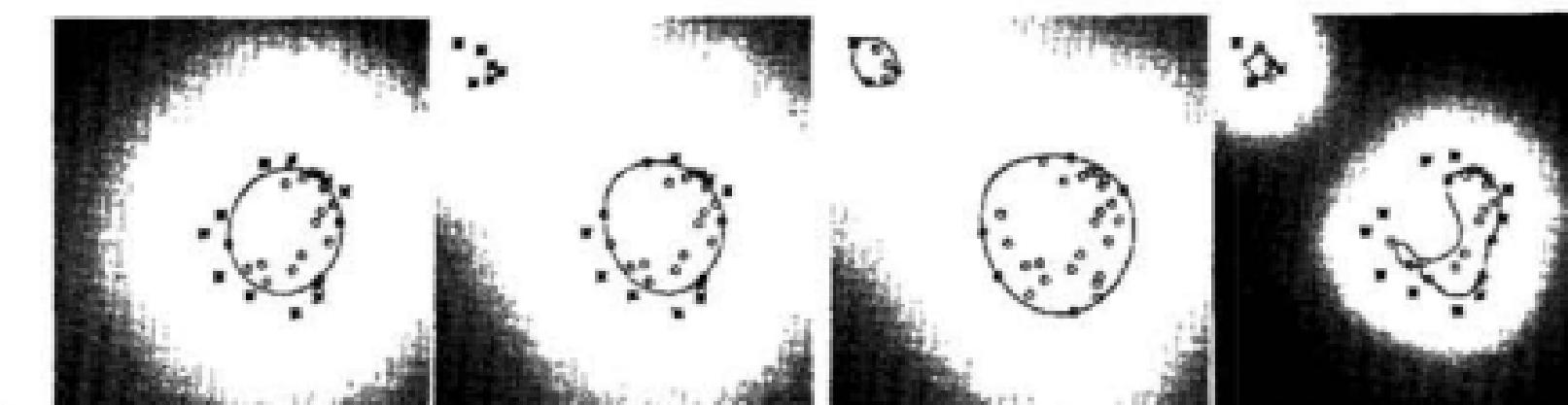
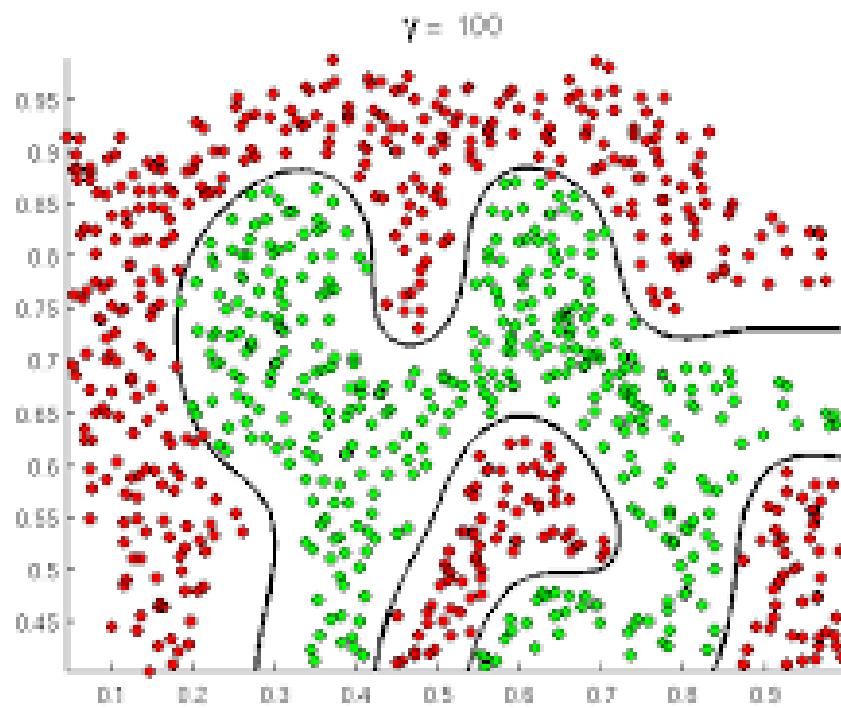
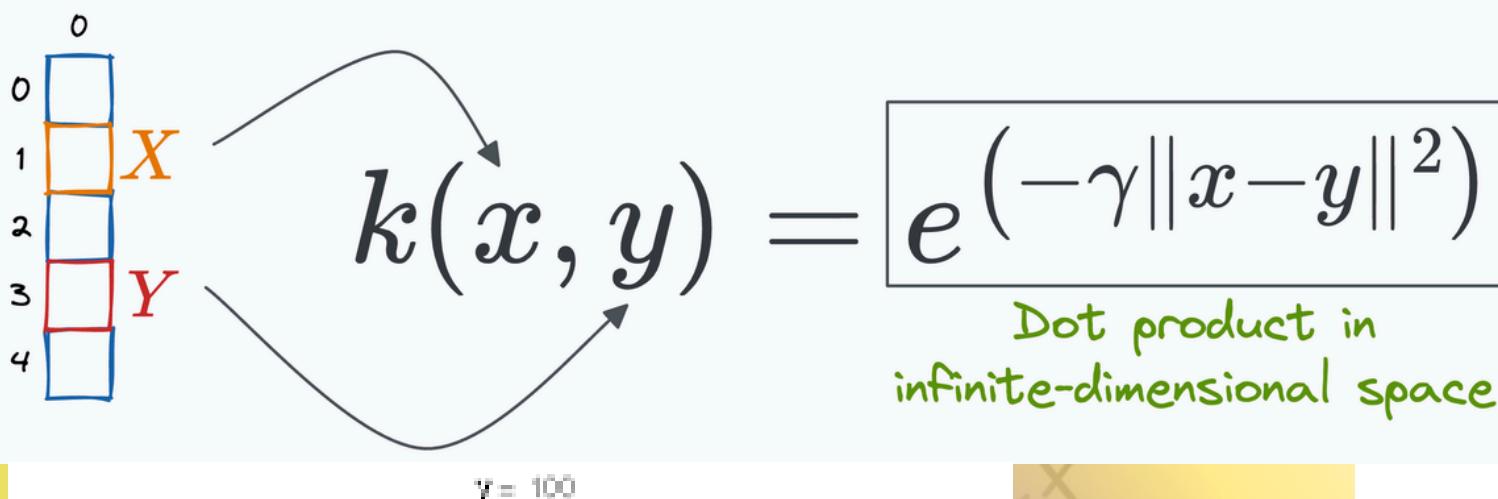
```
[[9407 163]
 [8954 1332]]
```



To separate the data set from the origin, we solve the following quadratic program:

$$\min_{w \in F, \xi \in \mathbb{R}^l, \rho \in \mathbb{R}} \frac{1}{2} \|w\|^2 + \frac{1}{\nu l} \sum_i \xi_i - \rho \quad (3)$$

$$\text{subject to } (w \cdot \Phi(\mathbf{x}_i)) \geq \rho - \xi_i, \quad \xi_i \geq 0. \quad (4)$$



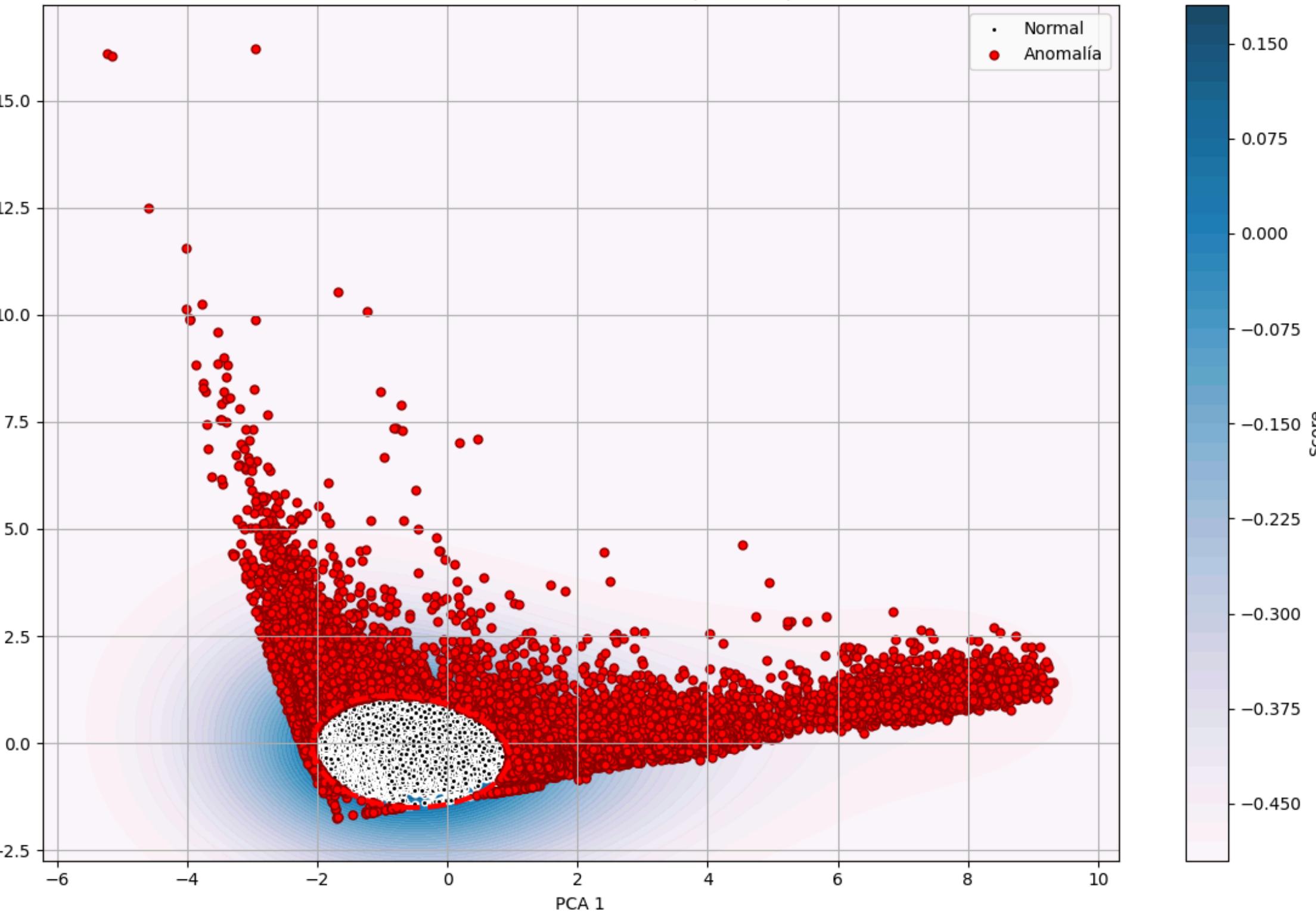
|                        |            |            |            |            |
|------------------------|------------|------------|------------|------------|
| $\nu, \text{width } c$ | 0.5, 0.5   | 0.5, 0.5   | 0.1, 0.5   | 0.5, 0.1   |
| frac. SVs/OLs          | 0.54, 0.43 | 0.59, 0.47 | 0.24, 0.03 | 0.65, 0.38 |
| margin $\rho/\ w\ $    | 0.84       | 0.70       | 0.62       | 0.48       |

Figure 1: *First two pictures*: A single-class SVM applied to two toy problems;  $\nu = c = 0.5$ , domain:  $[-1, 1]^2$ . Note how in both cases, at least a fraction of  $\nu$  of all examples is in the estimated region (cf. table). The large value of  $\nu$  causes the additional data points in the upper left corner to have almost no influence on the decision function. For smaller values of  $\nu$ , such as 0.1 (*third picture*), the points cannot be ignored anymore. Alternatively, one can force the algorithm to take these ‘outliers’ into account by changing the kernel width (2): in the *fourth picture*, using  $c = 0.1$ ,  $\nu = 0.5$ , the data is effectively analyzed on a different length scale which leads the algorithm to consider the outliers as meaningful points.

# OCSVM

## Implementación y resultados

Visualización PSA con GPU (OCSVM)



Anomalías detectadas: 17396 de 46739

No tenemos métricas de precisión porque se hizo el enfoque hacia la detección de valores atípicos, pero este modelo es especialmente terrible para ser entrenado por estrategia.

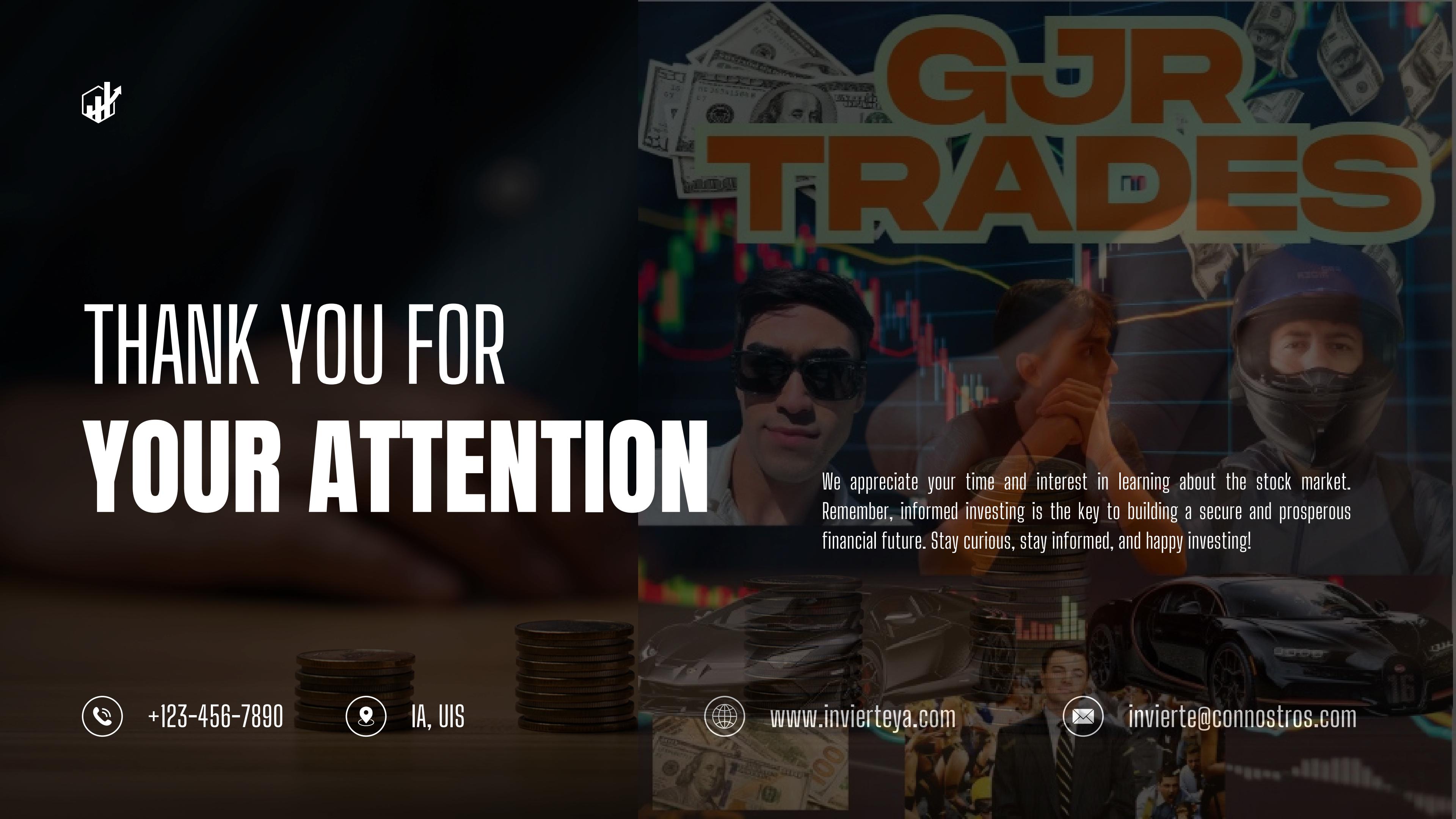
**LINK COLAB :**

**[https://colab.research.google.com/drive/1az2tR6L8Elck6eGg7wLsRoBcGj5IJM\\_Y?](https://colab.research.google.com/drive/1az2tR6L8Elck6eGg7wLsRoBcGj5IJM_Y?usp=sharing)**

**usp=sharing**



# THANK YOU FOR YOUR ATTENTION



We appreciate your time and interest in learning about the stock market. Remember, informed investing is the key to building a secure and prosperous financial future. Stay curious, stay informed, and happy investing!



+123-456-7890



IA, UIS



[www.invierteaya.com](http://www.invierteaya.com)



[invierte@connostros.com](mailto:invierte@connostros.com)