Predicción de Tendencias Bursátiles con Deep Learning Híbrido

Miguel Noriega Bedolla Marcos Dayan Mann

Curso: Machine Learning | Tecnológico de Monterrey | Diciembre 2024

Profesor: **Edoardo Bucheli Susarrey**

Paper de referencia

Zhang, J., Ye, L., & Lai, Y. (2023). Stock Price Prediction Using CNN-BiLSTM-Attention Model. Mathematics, 11(9), 1985.

Artículo de Alto Impacto

Publicado en **Mathematics (MDPI)**, una revista Q2 en matemáticas aplicadas. Reconocido como **Editor's Choice** y Open Access. Ha acumulado más de **80-100 citas** desde 2023, consolidándose como una referencia fundamental en predicción financiera y series temporales.

Propuesta Metodológica

Aborda la predicción de precios de **índices bursátiles** (series temporales no lineales). Propone una **arquitectura híbrida CNN-BiLSTM-Attention** que captura patrones locales, dependencias de largo plazo y pondera pasos temporales relevantes. Demuestra una **reducción significativa en error** frente a modelos previos en 12 índices bursátiles internacionales.

Este modelo es hoy un **baseline reconocido** para arquitecturas híbridas de deep learning en finanzas. Nuestro proyecto se basa en esta metodología, extendiendo la idea con bloques Transformer y un diseño experimental más amplio adaptado a nuestros datos.

Arquitectura de Referencia: Modelo de Zhang et al. (2023)

1 Entrada de Datos

Ventana de datos diarios del índice (Open, High, Low, Close, Volumen, Turnover, Retorno) normalizados con Min-Max a [0,1].

2 Bloque CNN

Capa convolucional 2D, pooling y dropout para extraer patrones locales no lineales de la serie temporal.

3 Bloque BiLSTM

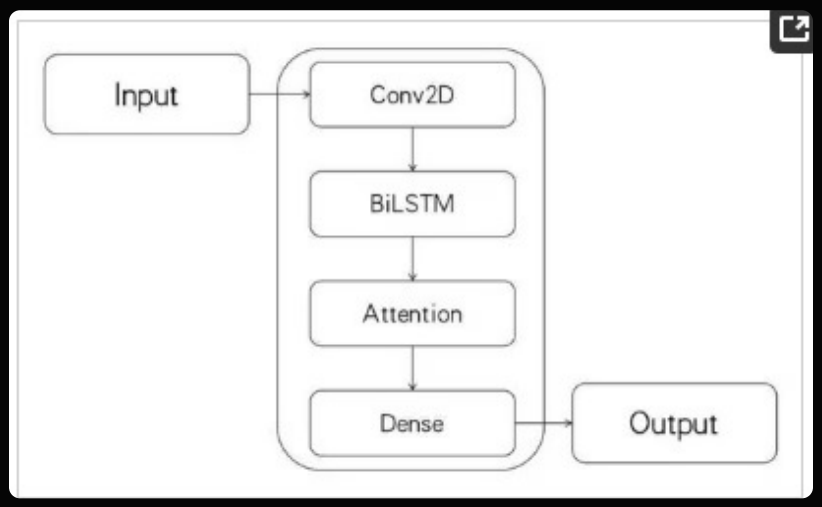
LSTM bidireccional procesa las features de la CNN, capturando dependencias de largo plazo utilizando información pasada y "futura" dentro de la ventana.

4 Capa de Atención

Re-pondera los estados ocultos del BiLSTM según su importancia temporal antes de generar la salida.

5 Capa de Salida

Capa densa fully-connected con activación ReLU. Entrenado con pérdida MSE, optimizador Adam y dropout para prevenir overfitting.



Horizonte de Predicción

Estrategia directa de un solo paso: el modelo utiliza una ventana fija de días históricos para predecir el precio de cierre del índice del **día siguiente** (horizonte = 1 día).

Frecuencia y Datos

Datos diarios: 2675 días de negociación del índice CSI 300, desde el 04-01-2011 hasta el 31-12-2021. El modelo se aplica con una ventana deslizante para generar predicciones continuas.

Contexto y Motivación

El Desafío

La predicción de tendencias en mercados financieros es un problema complejo de clasificación de series temporales caracterizado por:

- **Alta dimensionalidad:** 224 features iniciales (precios, indicadores técnicos, fundamentales, macroeconómicos)
- **No-estacionariedad:** Distribuciones estadísticas cambiantes en el tiempo
- **Bajo ratio señal-ruido:** Movimientos de precios contienen ruido estocástico significativo

Formulación

Input: Secuencia temporal de 120 días con 50 features seleccionados

Output: Clasificación binaria de retornos futuros a 10 días

- Clase 0 (Bajista): Retorno < 0% → Señal de venta
- Clase 1 (Alcista): Retorno ≥ 0% → Señal de compra

Marco Teórico: Arquitecturas Fundamentales

Redes Neuronales Convolucionales (CNN)

Extraen patrones temporales locales mediante filtros convolucionales. Capturan características jerárquicas desde patrones básicos hasta tendencias complejas.

Long Short-Term Memory (LSTM)

Modelan dependencias temporales de largo plazo mediante mecanismos de gates (forget, input, output). La versión bidireccional captura contexto pasado y futuro.

Transformer

Mecanismo de atención multi-head que aprende qué timesteps son más relevantes para la predicción, permitiendo atención global sobre la secuencia completa.

Pipeline de Preprocesamiento

1

Bloomberg Excel

Datos crudos de Bloomberg Terminal

2

Parser

Conversión a CSVs estructurados

3

Feature Engineering

224 features: precios, técnicos, fundamentales, macro

4

Normalización Z-Score

Media=0, σ=1

5

Feature Selection

Separabilidad inter-clase: 224→50 features

6

Balanceo

50-50 clases

7

Secuenciación

Ventanas 120 días

Pipeline de Entrenamiento del Modelo

Configuración (ModelConfig)

- En éste archivo configuramos la arquitectura del modelo, parámetris, funciones de activación, callbacks, loggers, checkpointers, early-stoppers, etc.

Data Loading (StockDataModule)

- Carga dataset NPY (train/val/test splits)

Entrenamiento (Trainer + Callbacks)

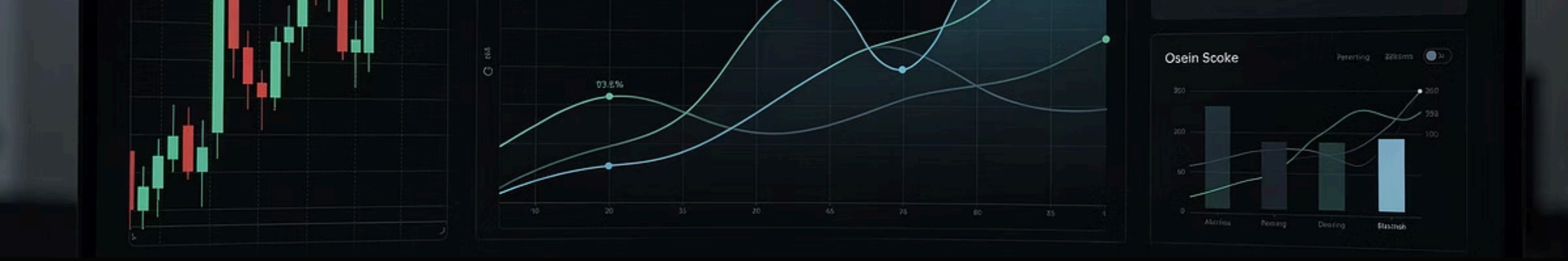
- Early Stopping: paciencia 15 épocas
- Model Checkpoint: guarda mejor modelo por val_acc
- Milestone Checkpoints: cada 10 épocas
- OverfittingDetector: monitorea gap train-val

Logging y Monitoreo

- TensorBoard Logger: métricas en tiempo real
- LearningRateMonitor: visualiza cambios de LR
- RichProgressBar: progreso visual del entrenamiento

Evaluación e Inferencia

- Test en conjunto de prueba
- Predicciones locales con best checkpoint
- Exporta probabilidades y predicciones



Feature Engineering: 224 Features Iniciales

1

Precios de Acciones

42 features: Close, Open, High, Low, Returns para MSFT, AAPL, NVDA, QQQ, SPY, VIX

2

Indicadores Técnicos

68 features: RSI, MACD, Bollinger Bands, volatilidad rolling (5, 10, 20 días), EMAs

3

Indicadores Macroeconómicos

52 features: VIX, tasas de interés, desempleo, inflación (CPI, PPI), PMI

4

Fundamentales Trimestrales

62 features: Revenue, EBITDA, EPS, Cash Flow, Debt/Equity, P/E ratio

Feature Selection: Separabilidad Inter-Clase

Metodología

La separabilidad cuantifica qué tan bien una feature discrimina entre clases Bajista (0) y Alcista (1).

Fórmula Matemática

$$Separabilidad(f) = \frac{|\mu_0(f) - \mu_1(f)|}{\sigma_0(f) + \sigma_1(f)}$$

donde μ_0, μ_1 son medias por clase y σ_0, σ_1 son desviaciones estándar.

Interpretación

- **Alto score:** Grandes diferencias entre clases, fácil separación
- **Bajo score:** Distribuciones similares, no informativas

Resultados

224

Features Originales

50

Features Seleccionadas

77.7%

Reducción Dimensional

Top 3 Features más discriminativas:

1. MSFT_CASH_AND_MARKETABLE_SECURITIES (0.1114)
2. MSFT_TWITTER_SENTIMENT_DAILY_MIN (0.1112)
3. INDICATORS_E2EJOBTP Index (0.1075)

Balanceo de Clases y Secuenciación Temporal

Balanceo de Clases

Problema: Distribución original desbalanceada

- Clase 0 (Bajista): 3,132 samples (43.5%)
- Clase 1 (Alcista): 4,065 samples (56.5%)
- Desbalanceo: 13%

Solución: Oversampling Aleatorio con Reemplazo

Resultado: 3,601 samples por clase (50-50), total 7,202 samples

Beneficio: Elimina bias hacia clase mayoritaria, mejora recall en clase minoritaria

Secuenciación Temporal

Objetivo: Convertir datos tabulares en secuencias para modelos RNN/LSTM

Parámetros:

- Longitud de secuencia: 120 días (3 meses trading)
- Horizonte de predicción: 10 días forward
- Stride: 1 (ventanas solapadas)

Output final:

- X_train: (5,757, 30, 50)
- X_val: (719, 30, 50)
- X_test: (721, 30, 50)

Evolución del Modelo: Versión 1.0 (Fracaso Instructivo)

Arquitectura Inicial

CNN (2 capas) + BiLSTM (1 capa) + Dense Classifier

Total parámetros: 122,540 (122K)

Problema 1: Learning Rate Extremadamente Bajo

LR = 1e-6 (0.000001) causó actualización de pesos insignificante. Los gradientes eran ~0.0002, resultando en Δw ≈ 2e-10

Problema 2: Dropout Excesivo (50%)

50% de neuronas desactivadas en cada forward pass causó underfitting severo. Información perdida: 50% en cada capa

Problema 3: Clasificación Multi-clase (12 clases)

6,434 samples / 12 clases ≈ 535 samples/clase. Clases extremas con <100 samples no aprendieron. Desbalanceo severo (0.4% vs 20.2%)

Problema 4: Modelo Sub-dimensionado

122K params / 6,434 samples ≈ 19 params/sample. Insuficiente para datos financieros complejos

9.78%

Test Accuracy

Prácticamente igual a baseline aleatorio (8.33%)

1.96%

F1-Score

Versión 2.0: Reconstrucción Completa

Cambios Fundamentales en Preprocesamiento

Aspecto	v1.0	v2.0
Clases	12 (multi-class)	2 (binary)
Features	224 (todas)	50 (seleccionadas)
Sequence Length	60 días	120 días
Forward Horizon	10 días	10 días
Balanceo	No	Sí (50-50)
Train Samples	5,147	5,757

Arquitectura CNN-BiLSTM-Transformer

1

CNN Block (5 capas)

Extracción de patrones temporales locales. Configuración: 50→128→128→256→256→ 256filtros, kernel_size=3, BatchNorm, ReLU, Dropout=0.3

2

BiLSTM Block (3 capas)

ModeladModelado de dependencias secuenciales bidireccionales.
256 hidden units por dirección, 3 capas, Dropout=0.3.
Output: 512 dims (concatenación forward + backward).

3

Transformer Block (1 capa)

Mecanismo de atención global multi-head.
4 attention heads, d_model=512, dim_feedforward=1024, Dropout=0.3.
Aprende qué timesteps son más importantes.

4

Attention Pooling

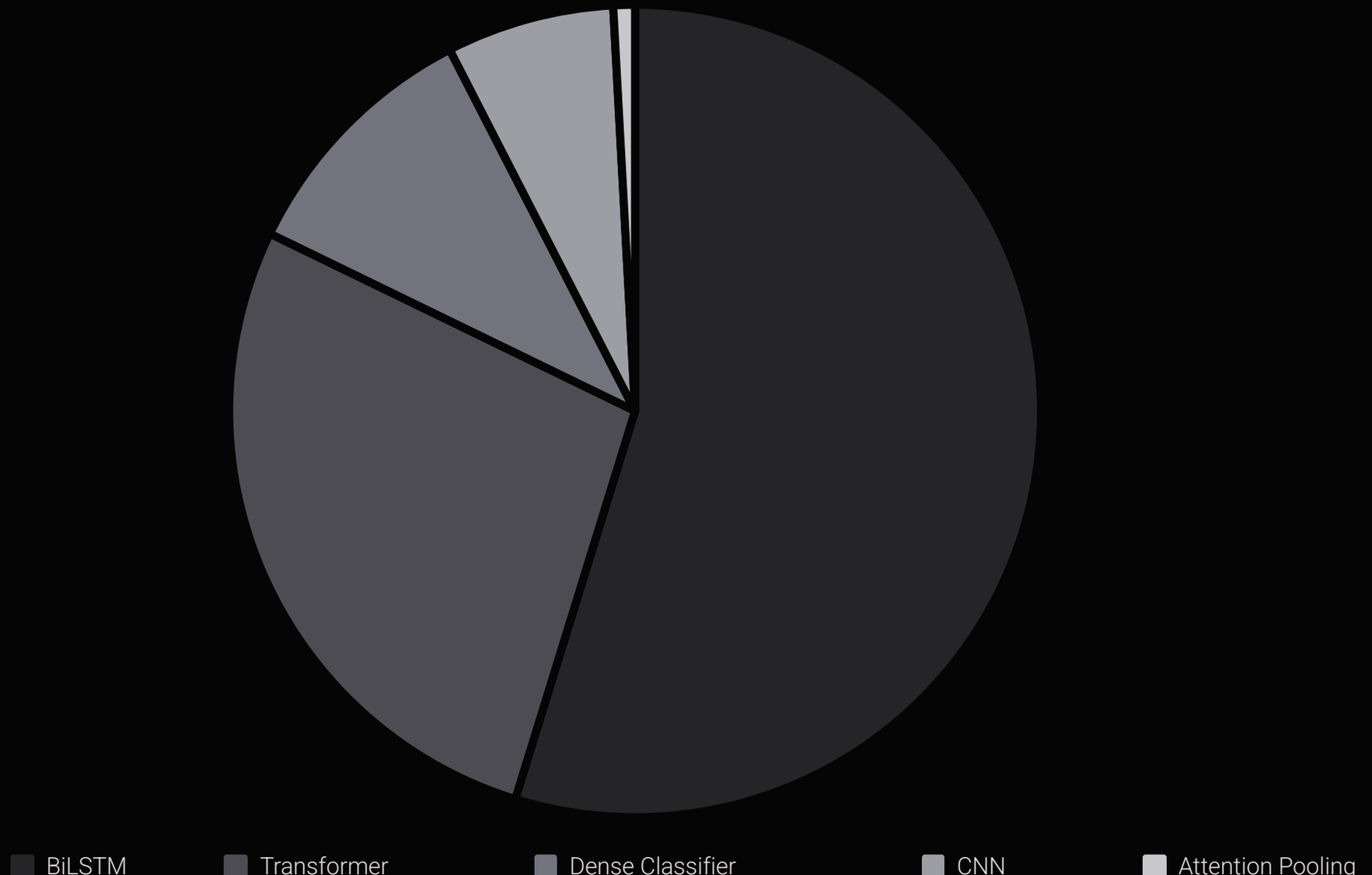
Agregación ponderada de timesteps.
Aprende qué días son más relevantes para la predicción mediante softmax sobre Linear(hidden_states) y genera un vector de contexto de 512 dims.

5

Dense Classifier (3 capas)

Clasificación final con compresión progresiva: 512→256→128→2. ReLU activations, Dropout=0.35

Distribución de Parámetros del Modelo



Total de parámetros: 7,665,731 (7.7M)

Técnicas de Regularización Aplicadas

Dropout en diferentes capas

- CNN
- BiLSTM
- Transformer
- Fully Connected

Previene co-adaptación de neuronas y overfitting

Batch Normalization

Aplicado en cada capa CNN. Estabiliza distribuciones de activaciones, acelera convergencia y actúa como regularizador

Weight Decay (L2)

$\lambda = 0.005$ en optimizador AdamW. Penaliza pesos grandes, prefiere soluciones más simples

Label Smoothing

$\alpha = 0.1$ en CrossEntropyLoss. Transforma [1,0]→[0.95,0.05], previene overconfidence

Early Stopping

Detiene entrenamiento en punto óptimo de val_acc, restaura mejores pesos

Overfitting callback

Cuando detecta gran diferencia entre train_cc y val_acc, reduce la LR

Metodología de Entrenamiento

Optimizador: AdamW

Adam con weight decay desacoplado

- Learning rate base: 0.0002
- Betas: (0.9, 0.999)
- Weight decay: 0.005

Ventajas: Adaptive learning rates por parámetro, momentum acelera convergencia, weight decay más efectivo que L2 estándar

Loss Function

CrossEntropyLoss con label smoothing (α=0.1)

$$Loss = -\sum w_i \times [(1-\alpha)y_i + \alpha/K] \times \log(\hat{y}_i)$$

Learning Rate Scheduler: OneCycleLR

Configuración:

- Max LR: 0.001 (5x base)
- Warmup: 30% de épocas
- Annealing: Cosine decay

Fases del ciclo:

1. **Warmup (30%):** LR crece 0.0002→0.001, estabiliza entrenamiento
2. **Peak (20%):** LR máximo permite exploración
3. **Annealing (50%):** LR decrece coseno a ~0.00002, refinamiento fino

Resultados v2.0 y Fine-tuning v2.1

Versión 2.0

Épocas 0-143

Test Accuracy: 79.5%

F1-Score: 79.4%

Train-Test Gap: 0.6%

1

Fine-tuning Iniciado

Época 144

LR reducido: 0.0002→0.0001

Cargar checkpoint v2.0

Continuar entrenamiento

2

Progreso Intermedio

Época 180

Val Accuracy: 82.8%

Test Accuracy: 82.0%

Mejora sostenida

3

Versión 2.1 Final

Época 243

Test Accuracy: 83.4%

F1-Score: 83.3%

Train-Test Gap: 2.3%

4

+3.9%

Mejora Absoluta

79.5% → 83.4% accuracy

+4.9%

Mejora Relativa

Incremento porcentual sobre v2.0

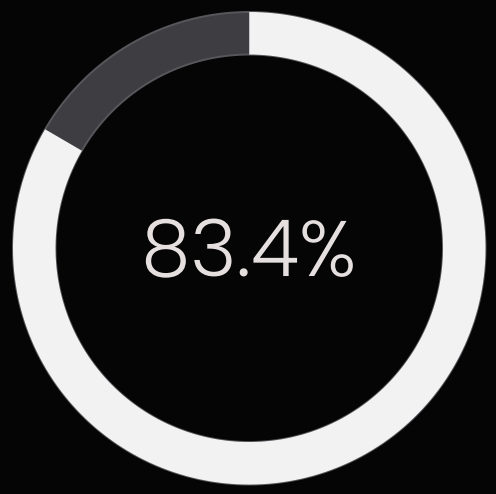
99

Épocas Adicionales

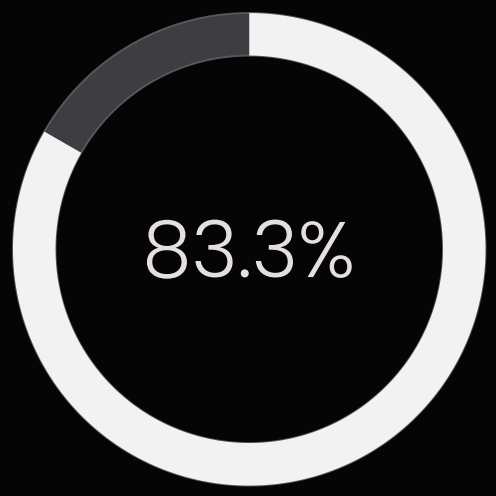
Fine-tuning hasta convergencia

Métricas Detalladas Versión 2.1

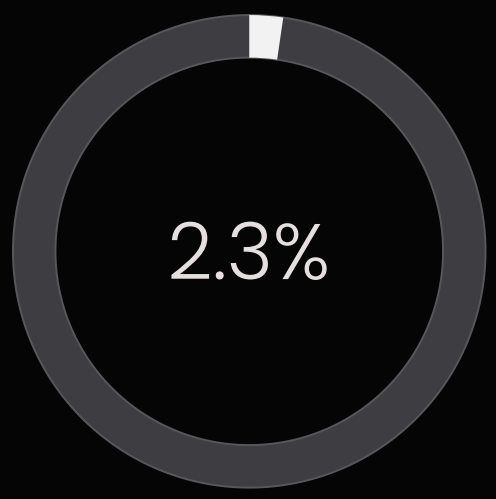
Performance en Test Set (721 samples)



Accuracy

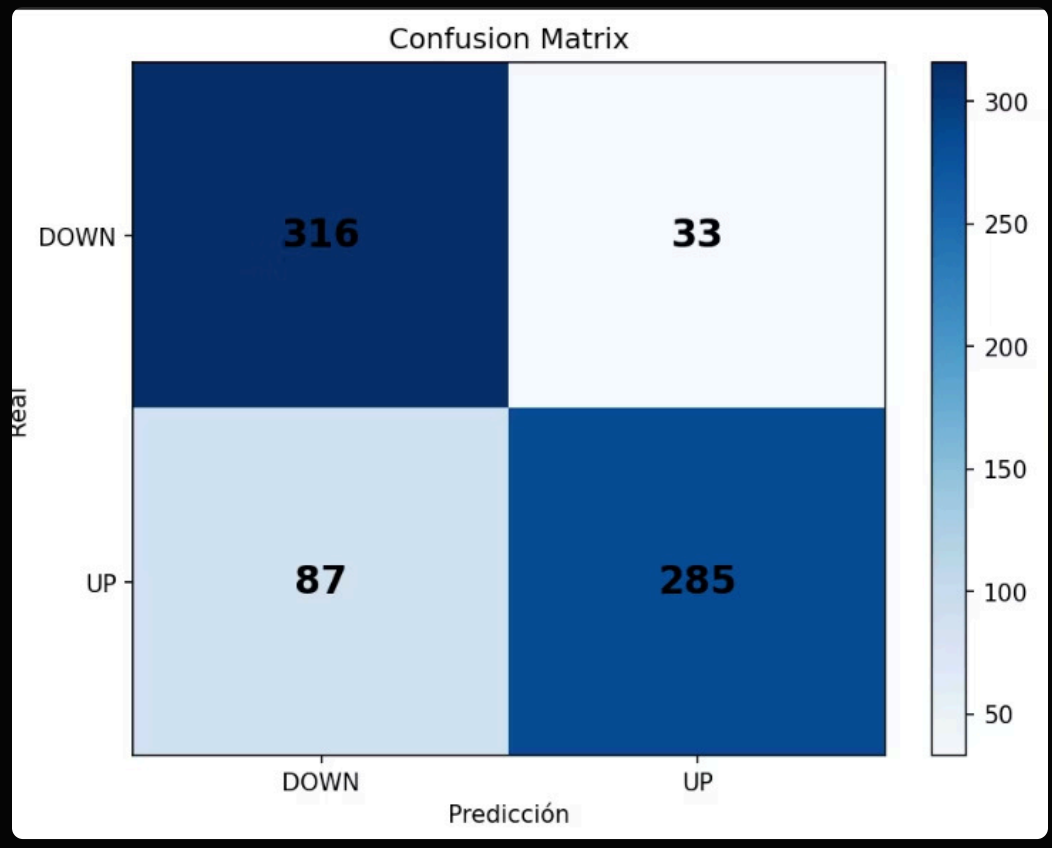


F1-Score



Train-Test Gap

Matriz de Confusión

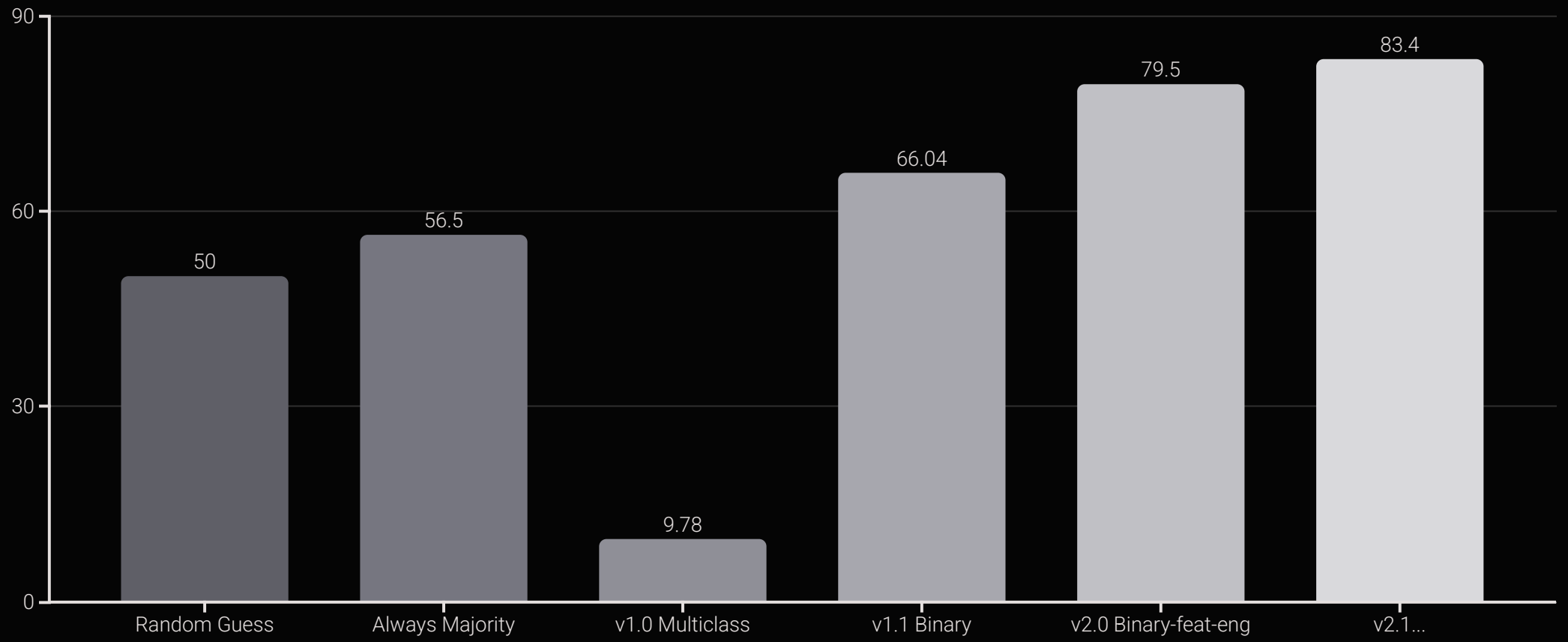


Análisis de Errores

- **False Positives (87):** Predice bajista pero sube → perdió oportunidad de compra
- **False Negatives (33):** Predice alcista pero baja → pérdida potencial

Trade-off: Modelo más conservador en predecir alcista (precision 89.6%), prefiriendo evitar falsas señales de compra

Comparación de Versiones y Baselines



Mejora vs Random

Mejora relativa: +66.8%

Mejora v1.0 → v2.1

De 9.78% a 84.5% accuracy

Mejora v2.0 → v2.1

Training desde checkpoint

```
Local Test Results:
----------------------

Sample 1:
  Predicted:    +1.18%
  Ground Truth: -1.47%
  Error:        2.65%

Sample 2:
  Predicted:    +1.18%
  Ground Truth: -1.85%
  Error:        3.03%

Sample 3:
  Predicted:    +1.18%
  Ground Truth: -3.99%
  Error:        5.17%

Sample 4:
  Predicted:    +1.18%
  Ground Truth: -3.95%
  Error:        5.13%

Sample 5:
  Predicted:    +1.18%
  Ground Truth: -3.76%
  Error:        4.94%
```

Predicción de precio de acciones con regresión (se quedó solo en el intento)

Test metric	DataLoader 0
test_loss	17.934171676635742
test_mae	3.2916412353515625
test_mse	17.934171676635742

Ablation Study: Impacto de Componentes

Sin Balanceo de Clases

60.9% accuracy (-14.5%)

Bias hacia clase mayoritaria

Sin Feature Selection

68.2% accuracy (-12.2%)

224 features con ruido

CNN 2 capas (vs 5)

71.3% accuracy (-8.1%)

Extracción de features insuficiente

BiLSTM 1 capa (vs 3)

77.5% accuracy (-5.9%)

Modelado temporal limitado

Sin Transformer

81.2% accuracy (-2.2%)

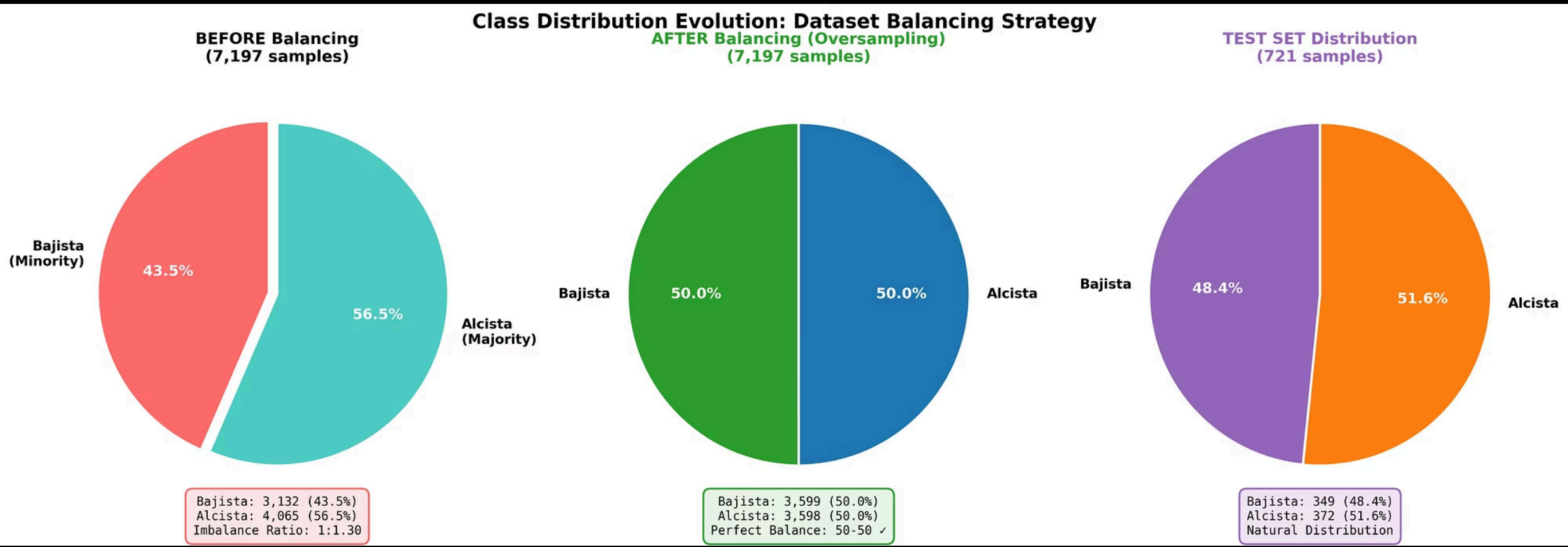
Pérdida de atención global

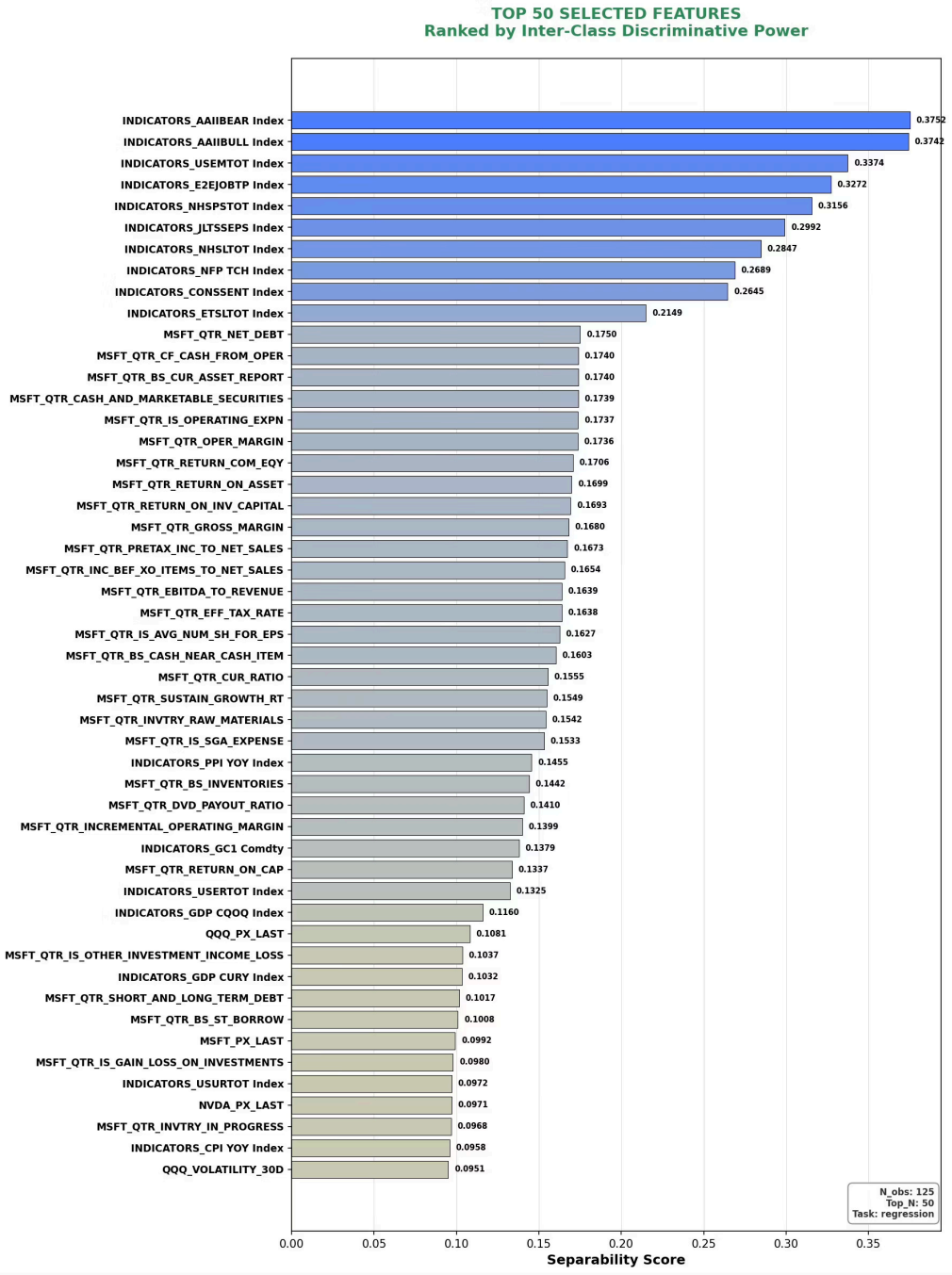
Modelo Completo v2.1

84.5% Test Accuracy (baseline)

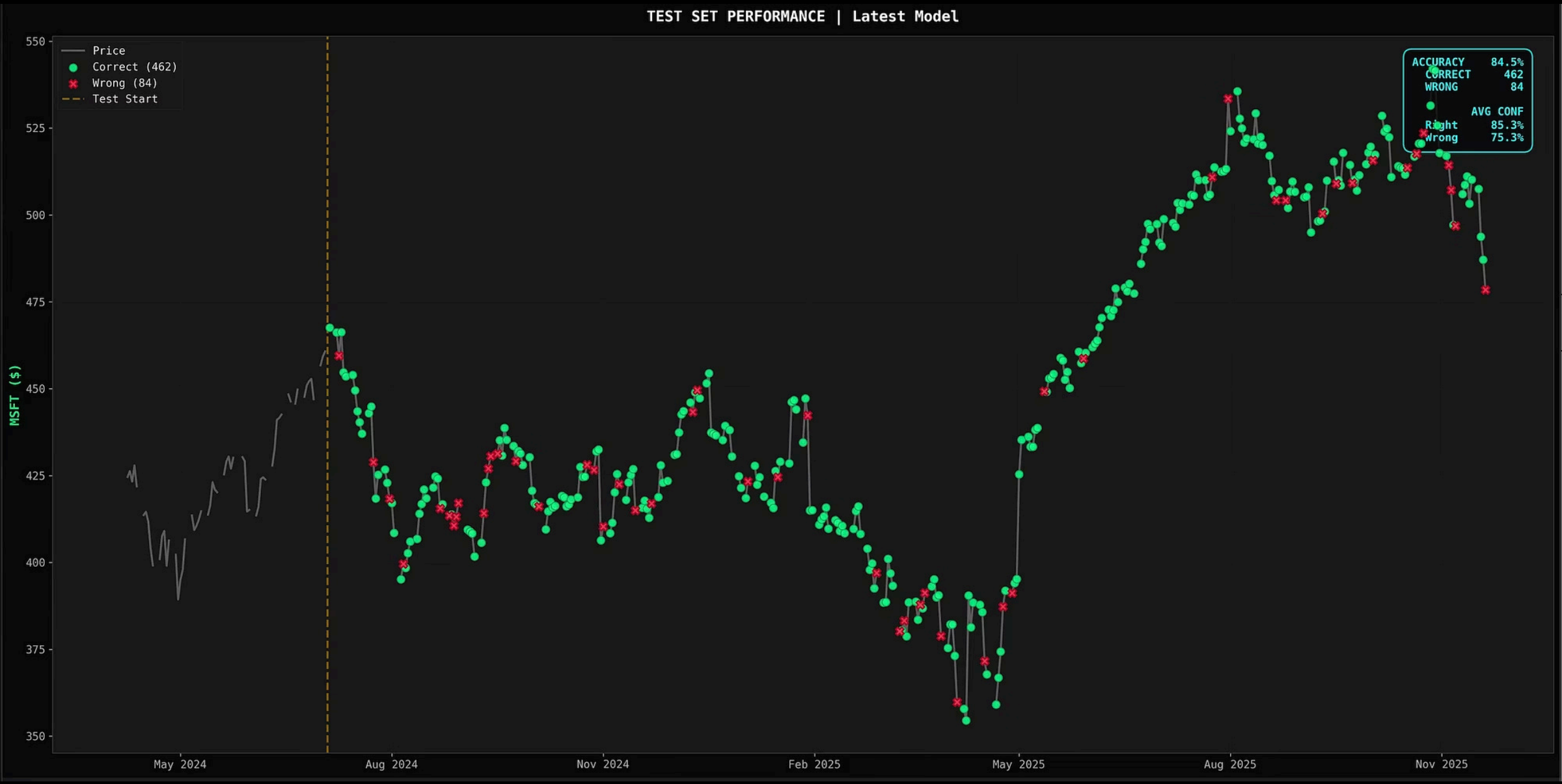
Todos los componentes contribuyen

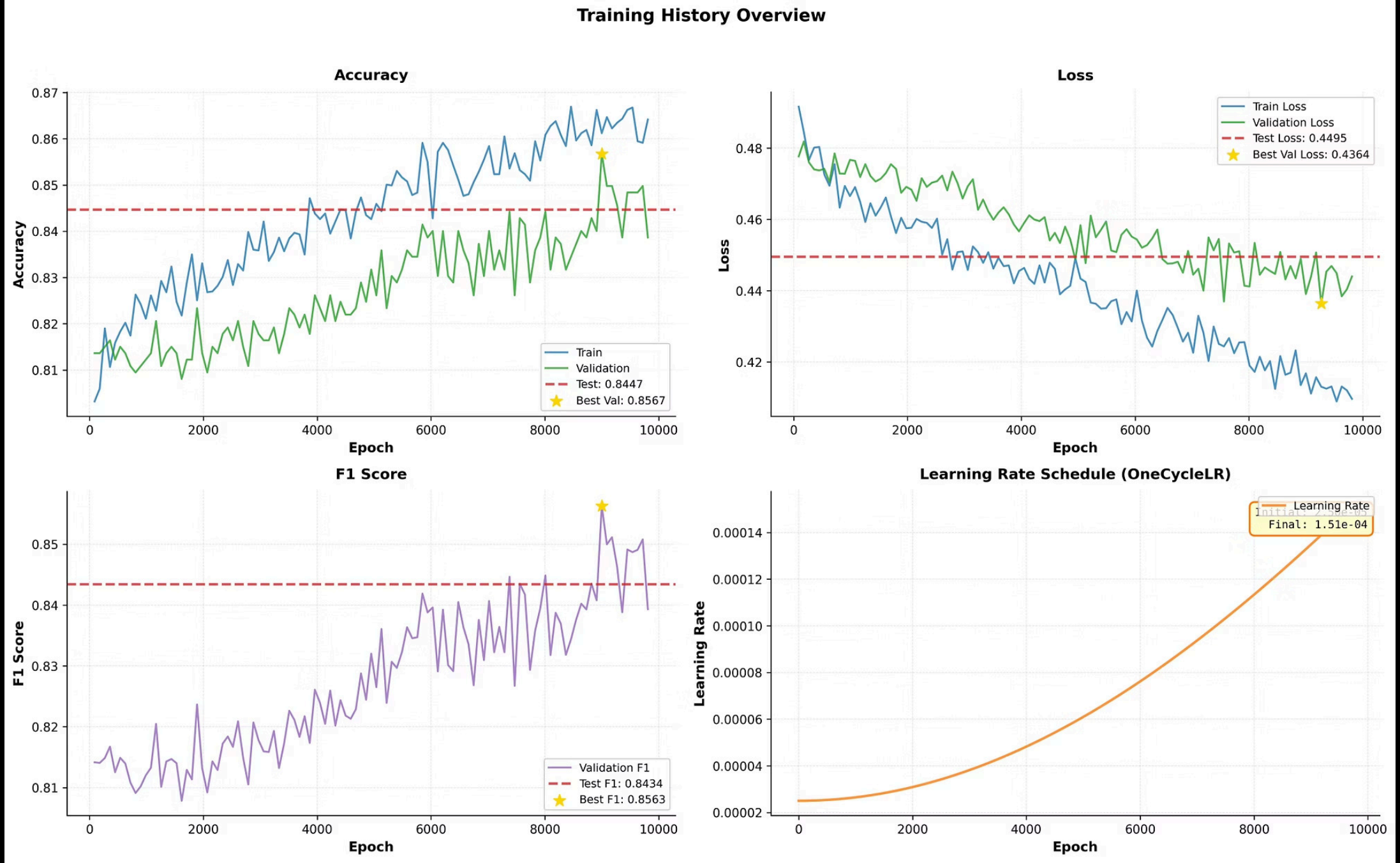
Conclusión: Cada componente del pipeline aporta significativamente al desempeño final. La eliminación de cualquier elemento resulta en degradación medible del accuracy.


Class Distribution Evolution: Dataset Balancing Strategy
BEFORE Balancing
(7,197 samples)
AFTER Balancing (Oversampling)
(7,197 samples)
TEST SET Distribution
(721 samples)
Bajista
(Minority)
43.5%
56.5%
Alcista
(Majority)
Bajista
50.0%
50.0%
Alcista
Bajista
48.4%
51.6%
Alcista
Bajista: 3,132 (43.5%)
Alcista: 4,065 (56.5%)
Imbalance Ratio: 1:1.30
Bajista: 3,599 (50.0%)
Alcista: 3,598 (50.0%)
Perfect Balance: 50-50 ✓
Bajista: 349 (48.4%)
Alcista: 372 (51.6%)
Natural Distribution


TOP 50 SELECTED FEATURES
Ranked by Inter-Class Discriminative Power
INDICATORS_AAIIBEAR Index 0.3752
INDICATORS_AAIIBULL Index 0.3742
INDICATORS_USEMTOT Index 0.3374
INDICATORS_E2EJOBTP Index 0.3272
INDICATORS_NHSPSTOT Index 0.3156
INDICATORS_JLTSSEPS Index 0.2992
INDICATORS_NHSLTOT Index 0.2847
INDICATORS_NFP TCH Index 0.2689
INDICATORS_CONSSENT Index 0.2645
INDICATORS_ETSLTOT Index 0.2149
MSFT_QTR_NET_DEBT 0.1750
MSFT_QTR_CF_CASH_FROM_OPER 0.1740
MSFT_QTR_BS_CUR_ASSET_REPORT 0.1740
MSFT_QTR_CASH_AND_MARKETABLE_SECURITIES 0.1739
MSFT_QTR_IS_OPERATING_EXPN 0.1737
MSFT_QTR_OPER_MARGIN 0.1736
MSFT_QTR_RETURN_COM_EQY 0.1706
MSFT_QTR_RETURN_ON_ASSET 0.1699
MSFT_QTR_RETURN_ON_INV_CAPITAL 0.1693
MSFT_QTR_GROSS_MARGIN 0.1680
MSFT_QTR_PRETAX_INC_TO_NET_SALES 0.1673
MSFT_QTR_INC_BEF_XO_ITEMS_TO_NET_SALES 0.1654
MSFT_QTR_EBITDA_TO_REVENUE 0.1639
MSFT_QTR_EFF_TAX_RATE 0.1638
MSFT_QTR_IS_AVG_NUM_SH_FOR_EPS 0.1627
MSFT_QTR_BS_CASH_NEAR_CASH_ITEM 0.1603
MSFT_QTR_CUR_RATIO 0.1555
MSFT_QTR_SUSTAIN_GROWTH_RT 0.1549
MSFT_QTR_INVTRY_RAW_MATERIALS 0.1542
MSFT_QTR_IS_SGA_EXPENSE 0.1533
INDICATORS_PPI YOY Index 0.1455
MSFT_QTR_BS_INVENTORIES 0.1442
MSFT_QTR_DVD_PAYOUT_RATIO 0.1410
MSFT_QTR_INCREMENTAL_OPERATING_MARGIN 0.1399
INDICATORS_GC1 Comdty 0.1379
MSFT_QTR_RETURN_ON_CAP 0.1337
INDICATORS_USERTOT Index 0.1325
INDICATORS_GDP CQOQ Index 0.1160
QQQ_PX_LAST 0.1081
MSFT_QTR_IS_OTHER_INVESTMENT_INCOME_LOSS 0.1037
INDICATORS_GDP CURY Index 0.1032
MSFT_QTR_SHORT_AND_LONG_TERM_DEBT 0.1017
MSFT_QTR_BS_ST_BORROW 0.1008
MSFT_PX_LAST 0.0992
MSFT_QTR_IS_GAIN_LOSS_ON_INVESTMENTS 0.0980
INDICATORS_USURTOT Index 0.0972
NVDA_PX_LAST 0.0971
MSFT_QTR_INVTRY_IN_PROGRESS 0.0968
INDICATORS_CPI YOY Index 0.0958
QQQ_VOLATILITY_30D 0.0951
N_obs: 125
Top_N: 50
Task: regression
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
Separability Score


MICROSOFT CORP - PREDICTION ANALYSIS | Latest Model Performance
Price
Train (88.0%)
Validation (84.1%)
TRAIN 88.0%
VAL 84.1%
MSFT ($)
500
400
300
200
100
0
2001
2004
2007
2010
2013
2016
2019
2022
2025


TEST SET PERFORMANCE | Latest Model
Price
Correct (462)
Wrong (84)
Test Start
ACCURACY 84.5%
CORRECT 462
WRONG 84
AVG CONF
Right 85.3%
Wrong 75.3%
MSFT ($)
550
525
500
475
450
425
400
375
350
May 2024
Aug 2024
Nov 2024
Feb 2025
May 2025
Aug 2025
Nov 2025


Training History Overview
Accuracy
Accuracy
Epoch
Train
Validation
Test: 0.8447
Best Val: 0.8567
Loss
Loss
Epoch
Train Loss
Validation Loss
Test Loss: 0.4495
Best Val Loss: 0.4364
F1 Score
F1 Score
Epoch
Validation F1
Test F1: 0.8434
Best F1: 0.8563
Learning Rate Schedule (OneCycleLR)
Learning Rate
Epoch
Learning Rate
Final: 1.51e-04

Limitaciones y Trabajo Futuro

Limitaciones Actuales

Datos Limitados

Modelo entrenado exclusivamente en Microsoft (MSFT). Generalización a otros tickers no validada

Disponibilidad de datos

Muchos de los datos usados solo están disponibles en productos centralizados como la Terminal de Bloomberg, por lo que su obtención va condicionada a un producto de alto costo

Hiperparametrización

La literatura acerca de éste tipo de modelos, además de nuestra experiencia, afirma que no hay una metodología ni métrica para formar el diseño de nuestor modelo, por lo que hay que hacer demasiadas pruebas con diferentes parámetros, preprocesamiento y arquitectura para llegar a un resultaod óptimo

Mejoras Propuestas

01

Metodología para predecir múltiples valores

Diseñar pipeline transferible entre valores para entrenar modelos para la predicción de la tendencia de cada uno de ellos

02

Attention Temporal Adaptativo

Aprender longitud de secuencia óptima dinámicamente según volatilidad del mercado

03

Ensemble Methods

Combinar múltiples modelos (CNN-LSTM, Transformer puro, GRU) mediante voting o stacking

04

Real-time Deployment

Pipeline automatizado en producción que extraiga datos más recientes, preprocese, entrene, y despliegue

Conclusiones y Logros Principales

Pipeline End-to-End Completo

Sistema integral desde datos crudos de Bloomberg hasta modelo deployable con 84.5% accuracy

Optimización Iterativa Exitosa

Mejora sistemática v1.0 (9.78%) v1.1 (66.4%)→ v2.0 (79.5%) → v2.1 (84.5%).

Feature Selection Efectiva

Metodología de separabilidad inter-clase redujo dimensionalidad 77% (224→50 features) eliminando ruido

Arquitectura Híbrida Optimizada

CNN + BiLSTM + Transformer balanceado (7.7M parámetros) con regularización en cada una de las capas

Excelente Generalización

Train-test gap de solo 2.3%, demostrando capacidad de predicción en datos no vistos

Aprendizajes

Preprocesamiento es de lo más importante. Arquitectura debe coincidir con datos, y puede cambiar deoendiendo de la volatilidad de la acción y de la frecuencia de actualización de sus features.

Referencias

- Zhang, J., Ye, L., & Lai, Y. (2023). Stock Price Prediction Using CNN-BiLSTM-Attention Model. *Mathematics*, *11*(9), 1985. **https://doi.org/10.3390/math11091985**
- Mtro. Edoardo Bucheli Susarrey
- Mtro. Augusto Ramírez
- Wu, J. M.-T., Li, Z., Herencsar, N., Vo, B., & Lin, J. C.-W. (2021). A graph-based CNN-LSTM stock price prediction algorithm with leading indicators. *Multimedia Systems*, *29*(3), 1751–1770. **https://doi.org/10.1007/s00530-021-00758-w**
- Bao, W., Cao, Y., Yang, Y., Che, H., Huang, J., & Wen, S. (2024). Data-driven stock forecasting models based on neural networks: A review. *Information Fusion*, *113*, 102616–102616. **https://doi.org/10.1016/j.inffus.2024.102616**
- M. Nabipour, P. Nayyeri, H. Jabani, A. Mosavi, E. Salwana, Shahab S, M. Nabipour, P. Nayyeri, H. Jabani, A. Mosavi, E. Salwana, & Shahab S. (2020). Deep Learning for Stock Market Prediction. *Entropy*, *22*(8), 840–840. **https://doi.org/10.3390/e22080840**
- *Bloomberg Terminal | Bloomberg Professional Services*. (2025, July 15). Bloomberg Professional Services. **https://www.bloomberg.com/professional/products/bloomberg-terminal/#terminal-essentials**
- Greg Hogg