PROYECTO FINAL - SOLANA PREDICTOR

Applied Machine Learning (UNIT 25)

20 DE ENERO DE 2026

COMPUTER SCIENCE & AI
PABLO NICOLÁS SOTO IRAGO

Contenido

Introducción

Este documento recoge el trabajo integrado de las distintas entregas del curso Applied Machine Learning (MSMK): definición del problema, justificación teórica, experimentación en notebooks, análisis de resultados y despliegue de una aplicación web que expone un modelo predictivo sobre activos de la red Solana (por ejemplo: predicción de dirección de precio a corto plazo o predicción de return/valor). El objetivo es ofrecer una solución reproducible que pueda usarse como sistema de apoyo a la toma de decisiones por usuarios finales (traders, analistas) y que además permita recopilar feedback para re-entrenar el modelo. En esencia, el proyecto busca elaborar un programa que genere predicciones sobre el valor de Solana (SOL), proporcionando una herramienta práctica para la evaluación de oportunidades en el mercado de criptomonedas.

Definición del problema (Business Understanding)

Contexto

El ecosistema de criptomonedas es altamente volátil. Predicciones precisas del comportamiento a corto plazo (por ejemplo: próximas horas o días) pueden ayudar a planificar operaciones o a construir señales de trading. Este proyecto explora la predicción sobre el activo SOL (Solana) utilizando datos históricos de mercado y señales derivadas.

¿Qué problema resuelve el dataset?

El dataset aporta registros temporales (OHLCV: Open, High, Low, Close, Volume) de Solana junto con otras fuentes opcionales (book data, indicadores on-chain, sentimiento, datos macro). El objetivo es convertir esos datos en predicciones:

- Tarea de regresión: predecir el precio futuro (precio de cierre a t+1 día).
- Tarea de clasificación: predecir la dirección (sube/baja) en el horizonte T.

Hipótesis

Hipótesis principal: las características históricas del precio y volumen junto con indicadores técnicos permiten modelar, con utilidad práctica, la dirección del precio de Solana en horizontes cortos (1–7 días).

Hipótesis secundaria: modelos de Machine Learning (XGBoost/RandomForest) y modelos secuenciales (LSTM) capturan patrones complementarios; ensamblarlos mejora robustez.

Usuario final y caso de uso

Usuarios: traders cuantitativos, analistas de criptomonedas, investigadores académicos.

Valor: señales de predicción que se integran en un dashboard para la decisión humana, alertas automáticas de oportunidades y una base de datos de feedback para mejorar modelos.

Fundamentación teórica

Fundamentos relevantes

- Series temporales: autocorrelación, estacionalidad, heterocedasticidad.
- Aprendizaje supervisado: regresión y clasificación; overfitting y regularización.
- Redes neuronales recurrentes (LSTM): capacidad para modelar dependencias temporales.
- Árboles y ensamblados (RandomForest, XGBoost): manejo de no linealidad y variables mixtas.
- Métricas: MSE/RMSE/MAE para regresión; Accuracy, Precision, Recall, F1 y ROC-AUC para clasificación.

Justificación de algoritmos elegidos

- Baseline: regresión lineal y modelos logísticos para referencia y explicación.
- RandomForest/XGBoost: robustos ante outliers, pocas suposiciones, interpretabilidad parcial (importancias).
- LSTM / GRU: para capturar dinámicas temporales y patrones secuenciales en retornos.

- Métodos híbridos/ensemble: combinan ventajas (XGBoost sobre features estáticas + LSTM sobre series).

Relación con el ejercicio de la Red Neuronal en Excel

En Excel trabajamos el concepto de perceptrón: pesos, bias y error. En redes reales (pyTorch/Keras) esos conceptos se mantienen: la optimización ajusta pesos y bias minimizando la función de pérdida (MSE o cross-entropy). El análisis del error y la retropropagación son la versión ampliada de la iteración manual vista en Excel.

Ingeniería y análisis de datos

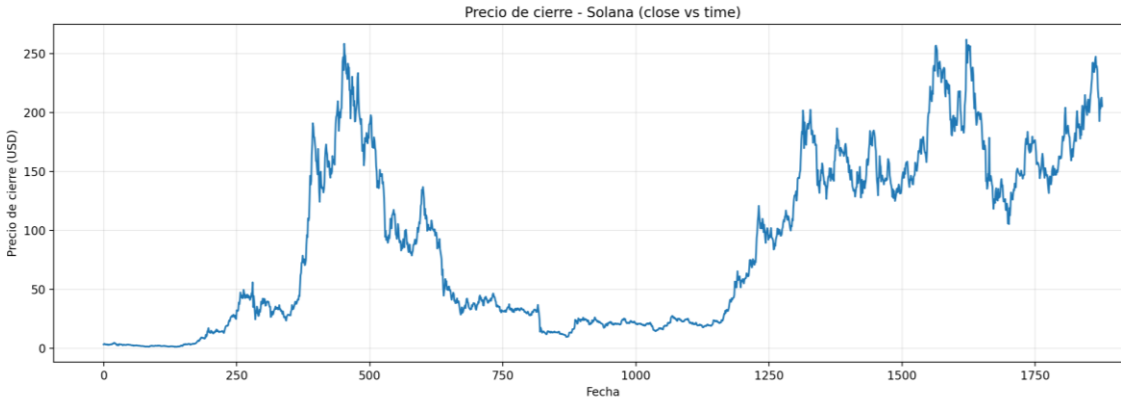
Descripción del dataset

- Fuente: API exchange (ej. Binance/Coinbase), Kaggle o dump histórico.

- Campos típicos: timestamp, open, high, low, close, volume, trades, spread.
- Periodicidad: minuto/5min/1h/día. En este proyecto se usó periodicidad diaria (ejemplo), agrupar y re-muestrear si procede.

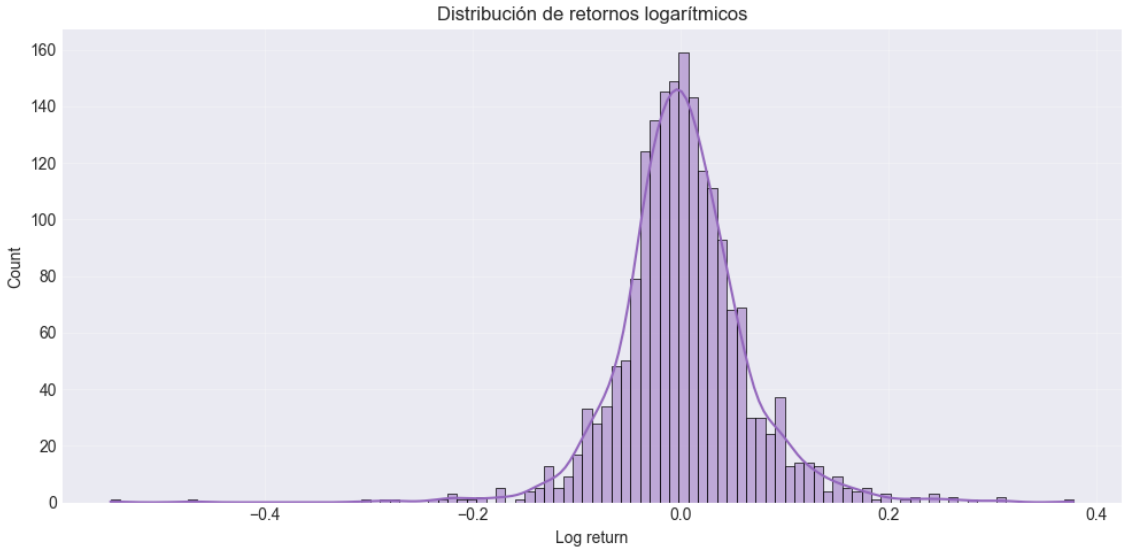
EDA — Las 2-3 gráficas clave

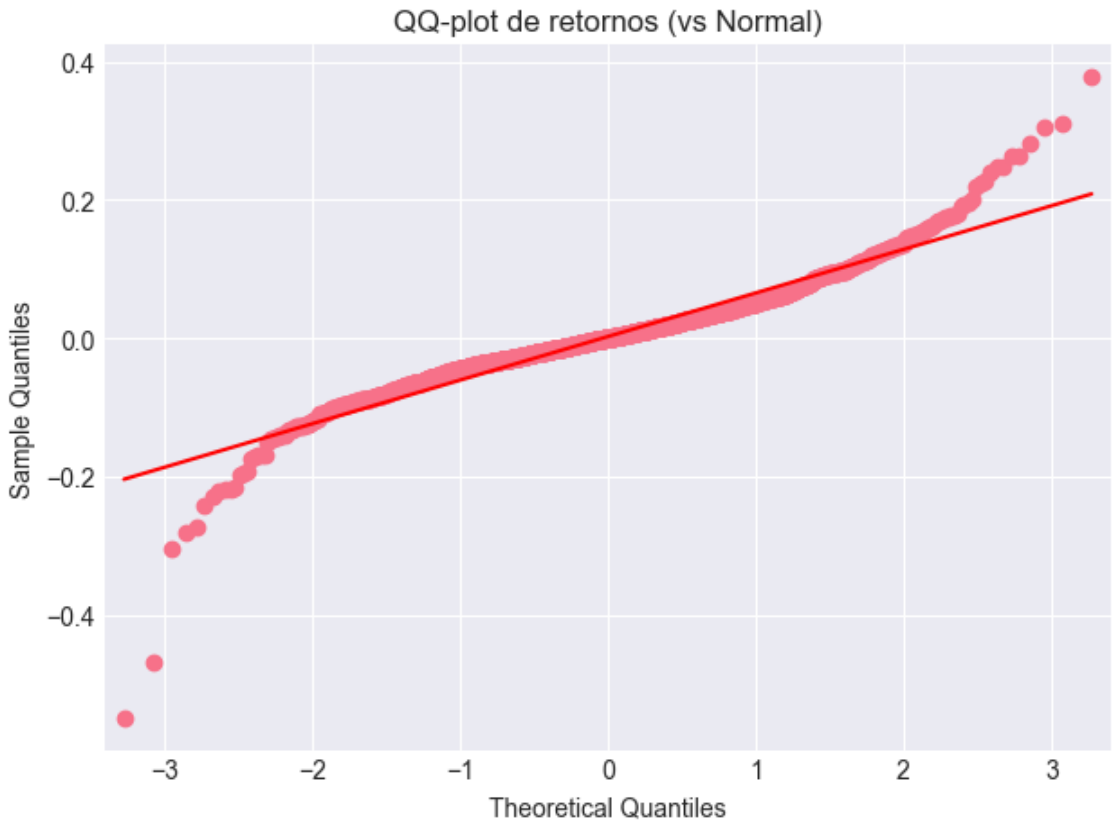
Serie temporal del precio de cierre (close vs time)



Explica tendencia, volatilidad y eventos extremos.

Distribución de retornos (histograma) y QQ-plot





Muestra asimetría y cola pesada -> implicaciones para métricas y pérdida.

Matriz de correlación entre features (incluye indicadores técnicos)

Identifica redundancias y colinealidad; ayuda a seleccionar features.

Preprocesamiento: limpieza y transformaciones

- Manejo de valores faltantes: forward-fill o eliminación según proporción.

- Reindexado por timestamp y resampling (si la fuente tiene irregularidades).

- Outlier handling: winsorizing o clipping en percentiles (1%, 99%).

- Transformaciones: log(price) o log-returns = ln(p_t / p_{t-1}) para estabilizar varianza.

- Escalado: StandardScaler o RobustScaler (preferible con outliers).

Diseño experimental y particionado temporal

Particionado temporal (no aleatorio): train / validation / test por fechas (ej. 70/15/15).

Validación: TimeSeriesSplit o walk-forward validation para simular despliegue real.

Métricas diferentes según tarea: RMSE/MAE para regresión; Accuracy/F1/ROC-AUC para clasificación de dirección.

Modelos probados y configuración

- Baseline: Persistence model (p_t = p_{t-1}), regresión lineal.
- Tree-based: RandomForestRegressor / RandomForestClassifier.
- Deep Learning: LSTM (Keras) para predicción secuencial.
- Ensembles: promedio ponderado o stacking.

Ideas Futuras y Mejoras

El desarrollo actual sienta las bases para un sistema predictivo funcional. Sin embargo, el horizonte evolutivo del proyecto apunta hacia la creación de una plataforma de análisis mucho más poderosa, flexible e intuitiva. Las siguientes ideas delinean la dirección futura:

Visión

Plataforma de Análisis Inteligente y adaptativa

La mejora principal consiste en evolucionar la aplicación web actual hacia una interfaz completamente dinámica y gobernada por lenguaje natural. La visión es una plataforma web que funcione como un analista cuantitativo conversacional, capaz de reconfigurarse sobre la marcha según las necesidades expresadas por el usuario.

Interfaz Centrada en el Prompt:

La página no presentaría un dashboard fijo de gráficos y métricas. En su lugar, la interfaz principal consistiría en una barra de texto prominente (similar a los buscadores o asistentes de IA modernos), donde el usuario puede introducir su petición o pregunta en lenguaje natural.

Reorganización Dinámica de la Interfaz:

Tras recibir un prompt, la plataforma interpretaría la intención del usuario y reorganizaría completamente el layout y los componentes de la página para servir esa petición de manera óptima. Por ejemplo:

- Prompt: *"Muéstrame la correlación histórica entre Bitcoin y Solana en el último año, el gráfico de velas semanal de SOL con RSI y MACD, y las predicciones de mis modelos para los próximos 3 días."*
- Respuesta de la Web: La página se regeneraría automáticamente para mostrar: 1) Un gráfico de líneas con la correlación móvil, 2) Un widget con un gráfico

OHLC interactivo superpuesto con los indicadores técnicos solicitados, y 3) Un panel con las predicciones de los modelos de clasificación y regresión desplegados.

Base de Datos Multi-Activo en Tiempo Real:

Para sustentar esta flexibilidad, el backend se ampliaría para ingerir, procesar y almacenar datos en tiempo real (OHLCV, ordenes, on-chain) de todas las criptomonedas más importantes (Bitcoin, Ethereum, principales altcoins). Un sistema de caché y procesamiento optimizado permitiría generar gráficos y calcular métricas complejas en segundos, independientemente del activo o ventana temporal solicitada.

Capacidades Analíticas Extendidas:

La plataforma podría integrar módulos para:

- Análisis Comparativo: "Compara el drawdown de SOL frente a ETH en el último ciclo bajista."
- Backtesting de Estrategias Simples: "Simula una estrategia de compra cuando el RSI de SOL está bajo 30 y el volumen es un 50% superior al promedio de 20 días."
- Síntesis de Contexto: "Resume las noticias y el sentimiento en redes sociales para Cardano de la última semana y superpón el precio."

Este enfoque transformaría la herramienta de un sistema predictivo estático en un entorno de descubrimiento y análisis interactivo, reduciendo la fricción entre la pregunta del usuario y la visualización de datos, y permitiendo una exploración de mercado más rápida y profunda. El modelo predictivo de Solana actual pasaría a ser uno de los muchos "agentes analíticos" disponibles dentro de este ecosistema flexible.

Anexos

Repositorio con el trabajo completo (sin despliegue de la App):
https://github.com/PabloNSI/solana-predictor.git

App (Repositorio en GitHub, se despliega en local):

https://github.com/PabloNSI/solana-predictor-app

Bibliografía y referencias

Se a utilizado IA para realizar este proyecto.

- Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow.

- Chollet, F. (2018). Deep Learning with Python.

- Brownlee, J. (2018). Machine Learning Mastery series (varios artículos sobre series temporales).

- Documentación XGBoost / LightGBM / sklearn / Keras.

- Recursos on-chain & APIs: Binance API docs, CoinGecko API.

“Todo el trabajo debe estar respaldado por una investigación y referenciado a lo largo del texto mediante el sistema de referencia de Harvard y se deberá incluir una bibliografía en el mismo formato. El uso incorrecto de las referencias puede dar lugar a plagio si no se aplica correctamente.”