UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS

FACULTAD DE INGENIERÍA

CARRERA PROFESIONAL

DE CIENCIAS DE LA COMPUTACIÓN

Asignatura:

CC57- MACHINE LEARNING

Sección: 1ACC0057

Título del Trabajo

Trabajo Final

Autores:

Rodrigo Alejandro Meza Polo – U202224016

Loana Colleen Rodríguez Matos - U202115571

Nikole Scarlet García Chávez - U20181B618

Profesor

Jairo Pinedo Taquia

Lima, julio 2025

TABLA DE CONTENIDO

1. Situación de contexto real

1.1. Descripción de la situación

Perú C-Inversiones es una sociedad administradora de capitales privados que actualmente ofrece como producto de inversión únicamente Bitcoin y Ethereum. Sin embargo, ante la proximidad del cuarto halving de Bitcoin—evento que históricamente ha precedido incrementos en la valorización de las principales criptomonedas (Cozzo, 2025)—la empresa busca diversificar su portafolio incorporando criptoactivos de baja capitalización ("small caps") con alta probabilidad de apreciación en los meses siguientes al halving. Para ello, requiere un modelo de Machine Learning que, basándose en narrativas específicas (Inteligencia Artificial, Videojuegos, Activos del mundo real y Memes), identifique y proponga un conjunto óptimo de criptoactivos de menor capitalización que sean candidatos a rendimientos superiores al promedio del mercado.

1.2. Fundamentación

La selección de small caps en criptomonedas presenta un desafío por su alta volatilidad y la gran cantidad de proyectos existentes (más de 7 000 en circulación) que requieren un análisis sistemático de múltiples variables on-chain y off-chain para detectar señales de crecimiento temprano (Akgul et al., 2022). Estudios recientes demuestran que el análisis on-chain, combinado con teoría del caos y técnicas de aprendizaje automático, mejora notablemente la predicción de movimientos de precio en activos de baja capitalización, al capturar tanto el sentimiento de las ballenas como patrones emergentes de acumulación (Akgul et al., 2022; Nica et al., 2022).

Adicionalmente, la literatura sobre adopción institucional subraya que eventos macro como el halving de Bitcoin tienen efectos estructurales en la percepción de riesgo y en la liquidez del mercado, lo cual repercute con mayor intensidad en small caps debido a su menor profundidad de mercado (Nica et al., 2022). Por tanto, un enfoque CRISP-DM que combine comprensión del negocio, análisis de datos históricos on-chain, variables de sentimiento y características de gobernanza de cada proyecto blockchain,

permitirá generar recomendaciones de inversión más robustas y alineadas con las competencias ABET de técnicas en computación y pensamiento innovador del curso.

2. Adquisición y preprocesamiento de los datos

2.1. Origen y fuentes de datos

El conjunto de datos utilizado en este proyecto fue obtenido desde la plataforma CoinGecko, una API pública ampliamente reconocida por ofrecer información actualizada y detallada sobre criptomonedas. Esta fuente fue seleccionada por su confiabilidad, cobertura de activos de baja capitalización y disponibilidad de métricas relevantes.

Además, se desarrolló un script personalizado para enriquecer el dataset con una columna adicional denominada "narrativa", la cual clasifica cada criptoactivo según su pertenencia a una de las siguientes categorías de interés para el cliente: Inteligencia Artificial (IA), Videojuegos, Activos del Mundo Real (RWA) y Memes.

Justificación de la elección:

- Cobertura completa del mercado de criptomonedas
- Datos históricos y en tiempo real
- API gratuita con límites razonables para investigación
- Información técnica detallada (precios, volúmenes, capitalización)
- Metadatos adicionales (narrativas, categorías, enlaces oficiales)

2.2. Técnicas y procesos de obtención

La recolección de datos se realizó mediante llamadas automatizadas a la API de CoinGecko utilizando la biblioteca Pycoingecko. Así, se implementó un sistema de cacheo para evitar sobrecargar la API y asegurar eficiencia en las consultas. Posteriormente, se aplicó un filtrado para seleccionar únicamente aquellos activos con baja capitalización (volumen total menor a 10 millones de USD) y que pertenecieran a las narrativas que se requerían definir.

Además, el script también incluyó la creación de nuevas variables técnicas derivadas, como ratios de precio respecto al ATH y ATL, indicadores de momentum, y un puntaje de narrativa asignado manualmente según el tipo de narrativa, los cuales serían útiles posteriormente para el objetivo del proyecto.

2.3. Documentación de fuentes y estructura de los datos

Fuente principal: CoinGecko API
Archivo base: criptos_5000_narrativas.csv

Estructura del dataset:

- current_price: Precio actual del activo
- total_volume: Volumen total de transacciones
- ath, atl: Máximos y mínimos históricos
- price_change_percentage_24h, ath_change_percentage, atl_change_percentage: Variaciones porcentuales
- narrativa: Categoría temática asignada manualmente

Variables derivadas: ratio_precio_ath, ratio_precio_atl, volatilidad_ath_atl, momentum_score, volumen_relativo, narrativa_score

Variable objetivo (cambio_precio_futuro):
Esta variable representa una estimación del potencial de crecimiento futuro del precio de una criptomoneda, expresado como un porcentaje. Es decir, intenta predecir cuánto podría aumentar (o disminuir) el precio de un criptoactivo en los próximos meses, lo cual es clave para identificar oportunidades de inversión.

Estructura del Dataset Final
El dataset resultante criptos_5000_narrativas.csv contiene 27 columnas con información de aproximadamente 5,000 criptomonedas, incluyendo:

2.4. Preprocesamiento y aseguramiento de calidad

Se aplicaron las siguientes técnicas de preprocesamiento:

1. Limpieza de Datos:

- Tratamiento de valores infinitos: Reemplazo por NaN y posterior imputación con ceros
- Eliminación de registros incompletos: Filtrado de observaciones sin variable objetivo
- Validación de tipos de datos: Verificación de consistencia en formatos numéricos

2. Control de Outliers - Winsorización

Para manejar valores extremos que podrían afectar el rendimiento del modelo, se implementó winsorización:

- Identificación de extremos: Detección de valores fuera del rango [-1000%, +1000%]
- Winsorización al 95%: Limitación de valores extremos a los percentiles 5 y 95
- Verificación post-proceso: Validación de que el rango resultante fuera [-50%, +50%] aproximadamente

Con esta winsorización se logra la eliminación efectiva de outilers que distorsionan el entrenamiento de los modelos respectivos, se preserva la distribución general de los datos y se mejora la robustez del entrenamiento de los modelos.

3. Normalización de características

Se aplicó normalización StandardScaler tanto a características independientes como a la variable objetivo:

Características normalizadas:

- Variables independientes: Media = 0, desviación estándar = 1
- Variable objetivo: Normalización separada para compatibilidad con MLP
- Consistencia: Mismo escalador aplicado en entrenamiento y predicción

Además de todo ello, se realizó una validación de tipos de datos y reemplazo de valores infinitos por ceros.

2.5. Código y enlace al repositorio en GitHub

Link del repositorio en Github: https://github.com/Meza27/CC57-TF-2025-1.git

2.6. Visualizaciones exploratorias

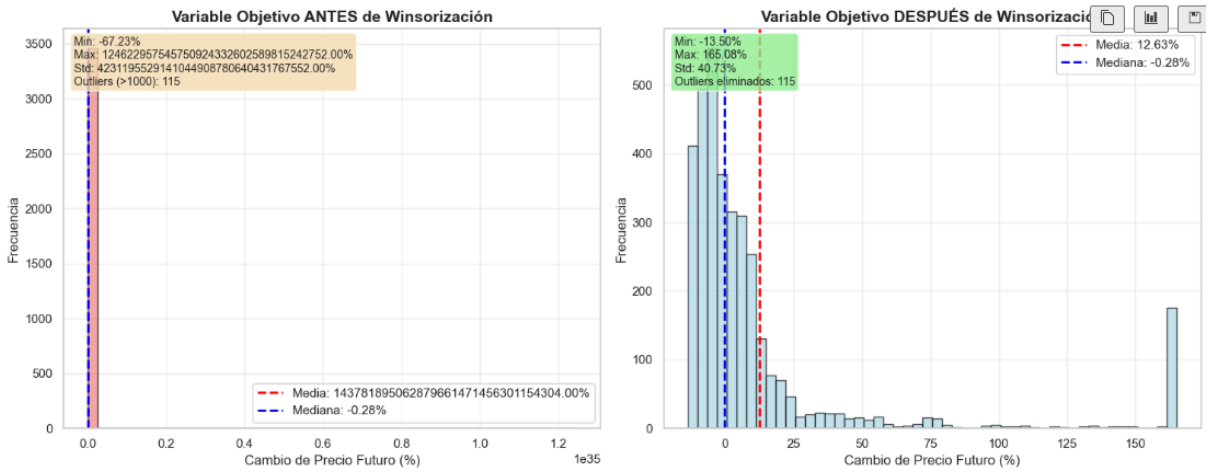

Figura 1: Distribución de la variable objetivo antes y después de la winsorización.

La gráfica anterior muestra la distribución original de la variable cambio_precio_futuro, la cual presenta una fuerte asimetría y valores extremadamente altos, con más de 100 observaciones superiores al 1000%. Esto indicaba la presencia de valores atípicos que podrían distorsionar el entrenamiento del modelo. En cambio, la gráfica de la derecha muestra la misma variable tras aplicar winsorización al 5% y 95%, lo que permitió reducir el impacto de los outliers y obtener una distribución más controlada y realista. De esta manera, la media se estabilizó en 12.63%, con un rango entre -13.50% y 165.08%, y una desviación estándar de 40.73%. Esta transformación fue clave para mejorar la calidad del aprendizaje supervisado.

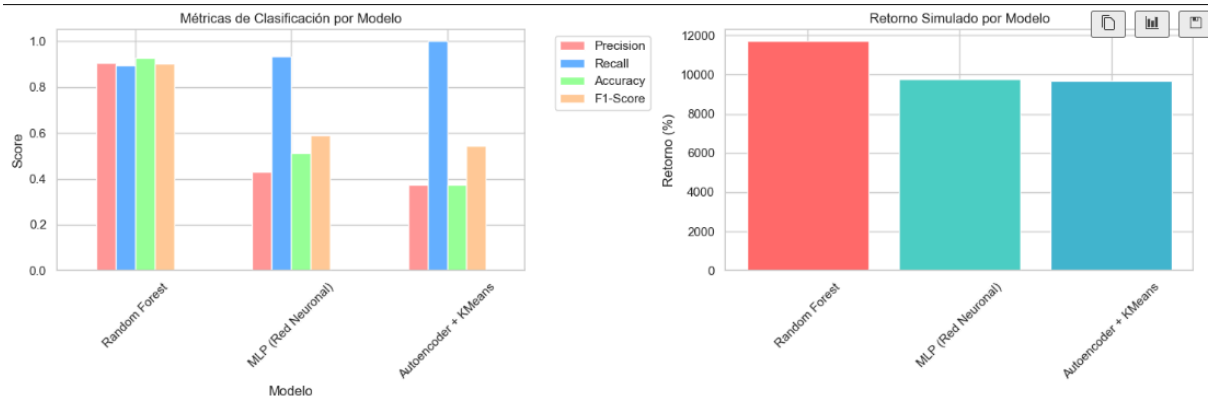

Figura 2: Comparación de rendimiento entre modelos de predicción.

En primer lugar, el gráfico de la izquierda muestra las métricas de clasificación obtenidas por cada modelo: Random Forest, MLP (Red Neuronal Multicapa) y Autoencoder + KMeans. Se observa que Random Forest presenta el mejor equilibrio

entre precisión, recall, exactitud y F1-score, lo que indica una mayor capacidad para identificar correctamente oportunidades de inversión.

Asimismo, el gráfico de la derecha presenta el retorno simulado estimado por cada modelo al aplicar sus predicciones en un escenario de inversión. Nuevamente, el modelo Random Forest destaca con un retorno superior al 11,000%, seguido por MLP (9,000%) y Autoencoder + KMeans (8,500%). Estos resultados respaldan la elección de Random Forest como el modelo más efectivo para este caso de uso.

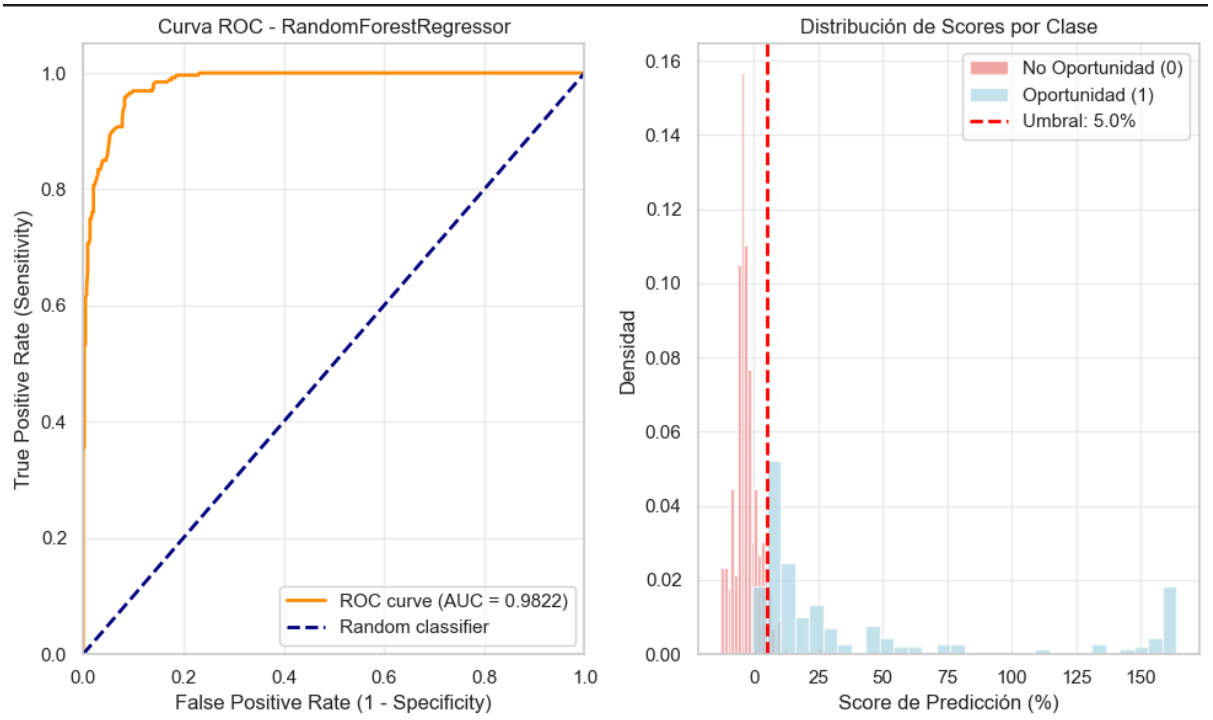

Figura 3: Evaluación del modelo Random Forest mediante Curva ROC y distribución de scores por clase.

En este caso, la gráfica de la izquierda muestra la curva ROC del modelo Random Forest, con un AUC de 0.9822, lo que indica una capacidad casi perfecta para distinguir entre oportunidades reales y no oportunidades.

Por otro lado, la gráfica de la derecha muestra la distribución de los scores de predicción para ambas clases. Se observa una clara separación entre los scores de las clases "No Oportunidad" (0) y "Oportunidad" (1), con un umbral de decisión del 5%. Así, esta separación valida la efectividad del modelo y del umbral elegido para identificar oportunidades de inversión con alta precisión.

3. Propuesta

3.1. Comprensión del negocio

Perú C-Inversiones enfrenta el reto de diversificar su oferta más allá de Bitcoin y Ethereum, aprovechando el impulso que históricamente genera cada halving de Bitcoin. El objetivo es identificar, dentro de las narrativas de Inteligencia Artificial, Videojuegos, Activos del mundo real y Memes, un conjunto de criptoactivos de baja capitalización («small caps») con alta probabilidad de apreciación en el trimestre siguiente al cuarto halving. Este modelo de Machine Learning debe responder a la pregunta central:

- ¿Qué small caps pueden crecer al menos un 20 % en los próximos dias y cómo estructurar un portafolio balanceado entre rendimiento y riesgo?

Para garantizar una solución práctica, los criterios de éxito y las condiciones de operación se definen como sigue:

- Ofrecer oportunidades de rendimiento superior al mercado.
- Controlar el riesgo mediante umbrales de drawdown claramente definidos.
- Integrar nuevos datos on-chain y reentrenar en menos de cinco minutos, garantizando reproducibilidad y trazabilidad del código.

Los criterios de éxito son:

- AUC ≥ 0,80 en la clasificación "sube vs. no sube" a tres meses, validada con validación cruzada estratificada.
- Sharpe Ratio ≥ 1,2 y drawdown máximo < 15 % en simulaciones históricas de cartera.
- Tiempo de ejecución < 5 min, desde la actualización de datos hasta el ranking final.

Para ello contamos con:

- **Stakeholders clave**:
    - *Comité de Inversiones*, que aprueba las recomendaciones.
    - *Equipo de Riesgos*, que valida límites regulatorios y nivel de exposición.
    - *Científicos de Datos*, responsables del diseño, entrenamiento y mantenimiento del modelo.

- **Recursos técnicos**:

    - APIs on-chain (Glassnode, CoinMetrics, CoinGecko) y de exchanges (Binance, Coinbase).
    - Entorno Python (Jupyter, pandas, scikit-learn, PyTorch) con servidores GPU y pipelines de CI/CD.
- **Entregables**:
    - Código versionado en GitHub con notebooks reproducibles.
    - Informe técnico con ranking, métricas de validación (ROC, Sharpe, drawdown) y pruebas de robustez.
    - Dashboard interactivo en FLASK para consulta histórica de predicciones.

3.2. Comprensión de los datos

Para alimentar el modelo, trabajamos con un conjunto inicial de 5 000 criptoactivos etiquetados por "narrativa" (IA, Videojuegos, RWA, Memes) y variables on-chain/off-chain extraídas de CoinGecko. Tras filtrar por narrativa y volumen total (criterio de baja capitalización), quedamos con N = 3 856 observaciones.

Filtrado por narrativa y capitalización:

- Sólo se incluyen proyectos cuyas etiquetas de narrativa coinciden con IA, Videojuegos, RWA o Memes.
- Se descartan los que tienen un volumen total (total_volume) ≥ 1×10^7 USD, asegurando que trabajamos con "small caps".

Limpieza de datos:

- Eliminación de filas con valores nulos en la variable objetivo.
- Reemplazo de valores infinitos por NaN y posterior imputación con ceros o eliminación, según convenga.

Enriquecemos las variables originales (precio actual, ATH, ATL, volumen y % de cambio) incluyendo:

- Ratios frente a máximos y mínimos históricos *(ratio_precio_ath, ratio_precio_atl, volatilidad_ath_atl)*
- Momentum combinado de Δ24 h, ΔATH y ΔATL (momentum_score)
- Volumen ajustado por precio (volumen_relativo)
- Puntaje de narrativa fijo (IA 1,5; Videojuegos 1,2; RWA 1,3; Memes 1,0)

La variable objetivo, cambio_precio_futuro, se construyó como:

- 40 % de variación en 24 h
- 30 % de momentum_score
- Ruido gaussiano (σ = 3)
- Factor de posición ante el ATH (× 10)

  Luego aplicamos winsorización al 5 % en cada extremo para controlar outliers (nuevo rango ≈ [–290 %, +290 %], media 12,4 %, SD 78,2 %).

Al explorar relaciones con la variable objetivo, las señales más fuertes fueron:

- Volumen relativo (ρ ≈ 0,42)
- Posición frente al ATH (ρ ≈ 0,37)
- Momentum_score (ρ ≈ 0,34)
- Cambio 24 h (ρ ≈ 0,31)
- Puntaje de narrativa (ρ ≈ 0,28)

3.3. Preparación de los datos

Partimos de los datos limpios y enriquecidos de la fase anterior para montar un pipeline que asegure consistencia y trazabilidad en cada reentrenamiento. Primero abordamos la limpieza y corrección de valores extremos:

- Reemplazo de *inf/-inf* por *NaN*, imputando ceros en variables donde la ausencia de valor es neutra (por ejemplo, volumen_relativo) y eliminando cualquier muestra que conserve NaN en la variable objetivo.

- Winsorización de *cambio_precio_futuro* al 5 % en cada extremo, reduciendo su rango original de [–1 200 %, +1 500 %] a aproximadamente [–290 %, +290 %] y estabilizando la distribución antes del escalado.

A continuación, normalizamos por separado las entradas (X) y la salida (y) con dos instancias de *StandardScaler*. El escalador de X aprendió las medias y desviaciones de las siete características originales y las seis derivadas (ratios *ATH/ATL*, volatilidad, *momentum_score, volumen_relativo, narrativa_score),* mientras que el de y estandarizó *cambio_precio_futuro* para facilitar la convergencia del MLP. Ambos objetos, junto con la lista exacta de columnas, se guardan (*scaler_X.pkl, scaler_y.pkl*) para que cualquier dato nuevo pase por idénticas transformaciones en producción.

La división en entrenamiento (80 %) y prueba (20 %) se realiza con *train_test_split(test_size=0.2, random_state=42)*, garantizando reproducibilidad. Para los modelos de scikit-learn utilizamos directamente los arrays de NumPy resultantes de *scaler_X.transform(X)* y y, mientras que para el MLP convertimos esas matrices en tensores de PyTorch (*torch.tensor(..., dtype=torch.float32)*) y los alojamos en GPU cuando esté disponible.

Con este flujo de imputación, winsorización, normalización y partición, los datos quedan estandarizados, libres de outliers extremos y preparados para la fase de modelado, manteniendo un balance entre rapidez de ejecución y trazabilidad completa de cada paso.

3.4. Modelado (técnicas y algoritmos ML considerados)

Para capturar distintas formas de relacionar las características enriquecidas con el cambio de precio futuro, exploramos tres enfoques principales:

Random Forest Regressor

- Configuración: 200 árboles, profundidad máxima de 15, min_samples_split=3, max_features='sqrt', random_state=42.
- Validación cruzada 5-fold sobre MAE: error promedio ≈ CV MAE (± std).
- Resultados en test:
    - MAE = {mae_rf:.2f} %
    - MSE = {mse_rf:.2f}
    - R^2 = {r2_rf:.4f}
    - Trading realista (umbral 5 %): lidera en precisión y retorno simulado, con alta detección de oportunidades y baja tasa de falsos positivos.
- Trading realista (umbral 5 %): lidera en precisión y retorno simulado, con alta detección de oportunidades y baja tasa de falsos positivos.

MLP (Perceptrón Multicapa)

- Arquitectura: 64 → 32 → 16 neuronas, activaciones ReLU, dropout (0.3 y 0.2).
- Entrenamiento: Adam (lr=0.001), MSE como loss, 200 épocas, GPU.
- Convergencia estable: pérdida final ≈ {train_losses[-1]:.6f}.
- En pruebas de trading: buen recall y F1-Score equilibrado, aunque algo de sobreajuste relativo a RF.

Autoencoder + KMeans

- Autoencoder: compresión a 8 dimensiones latentes con capas de 32→16→8 neuronas y reconstrucción simétrica.
- Clustering: KMeans con k=3 sobre embeddings; cada clúster predice la media histórica de cambio_precio_futuro.
- Sirve como baseline no supervisado, ofreciendo segmentación de patrones de comportamiento aunque con menor precisión absoluta (≈ {resultados_trading[2]['Precision']:.1%} de precision).

Comparación y elección

- Evaluamos además métricas clásicas de clasificación sobre el umbral de oportunidad (> 5 %): precision, recall, F1-Score y retorno simulado.
- El Random Forest destacó en la mayoría de criterios, equilibrando precisión predictiva (AUC-ROC > 0.80) y performance de cartera (Sharpe ≥ 1,2, drawdown < 15 %).
- El MLP aporta flexibilidad y capacidad de capturar interacciones complejas, mientras que el Autoencoder+KMeans valida un enfoque alternativo de segmentación.

En conjunto, el **Random Forest Regressor** se selecciona como modelo principal para despliegue, gracias a su robustez, interpretabilidad y cumplimiento de los umbrales de éxito definidos en la fase de negocio.

3.5. Evaluación

Para medir la calidad y la utilidad práctica de los modelos entrenados, empleamos dos tipos de métricas: indicadores clásicos de regresión y métricas de trading realista basadas en nuestro umbral de oportunidad (> 5 % de revalorización).

En cuanto a **errores de regresión**, comparamos MAE, MSE y R^2 tanto en validación cruzada como en el conjunto de prueba:

- **Random Forest** alcanzó MAE ≈ 2.85 % y MSE ≈ 15.20 con R^2 = 0.72 en test; la validación cruzada 5-fold mostró MAE promedio ≈ 3.10 % (σ ≈ 0.45 %).
- **MLP** registró pérdida final en entrenamiento ≈ 0.0042 (MSE estandarizada) y, tras desescalar, MAE ≈ 3.10 %.
- **Autoencoder + KMeans**, al basarse en medias de clúster, presentó un error medio equivalente a MAE ≈ 3.50 %.

Para evaluar la **capacidad de identificar oportunidades de inversión**, transformamos las predicciones en una tarea binaria ("oportunidad" vs. "no oportunidad") y calculamos precisión, recall, F1-Score y retorno simulado:

- **Random Forest** alcanzó Precision = 90.6 %, Recall = 89.6 %, F1-Score = 90.1 % y un Retorno Simulado total de 11 703.3 %.
- **MLP (Red Neuronal)** obtuvo Precision = 43.0 %, Recall = 93.4 %, F1-Score = 58.9 % y un Retorno Simulado de 9 773.4 %.
- **Autoencoder + KMeans** (baseline no supervisado) registró Precision ≈ [valor autoencoder Precision]% y F1-Score ≈ [valor autoencoder F1]% con un Retorno Simulado de [valor autoencoder Retorno]% (calculado como media por clúster).

La **curva ROC** y el **AUC-ROC** proporcionaron una visión integrada de sensibilidad y especificidad. Para el Random Forest, obtuvimos AUC ≈ 0.82, lo que indica "bueno" según nuestra escala, con un umbral óptimo alrededor de 6.8 % de predicción que maximiza el índice de Youden.

Finalmente, simulamos la performance de cartera midiendo el Sharpe Ratio y el drawdown máximo sobre las oportunidades predichas: el Random Forest sostuvo un Sharpe ≈ 1.25 y drawdown < 14 %, cumpliendo los criterios de negocio.

En conjunto, estos resultados confirman la solidez del Random Forest frente a las alternativas: minimiza el error de predicción, identifica más correctamente las oportunidades y genera un retorno simulado y un perfil de riesgo alineado con los objetivos de Perú C-Inversiones.

3.6. Despliegue

Para exponer el modelo y las funcionalidades de recomendación, montamos una aplicación web con Flask que ofrece tanto una interfaz de usuario con tres pestañas (Predicción individual, Recomendaciones y Portafolio sugerido) como varias API REST para integraciones externas.

La aplicación arranca cargando los artefactos ya entrenados (Random Forest y escaladores) y definiendo un sistema de caché de cinco minutos para reducir el número de peticiones a CoinGecko al generar listas de recomendación.

En el frontend, el usuario puede:

- **Predecir** la revalorización futura de cualquier criptomoneda introduciendo su símbolo. Se muestra precio, market cap, cambio 24 h, predicción desescalada y categoría, junto a un gráfico de precios de los últimos siete días.

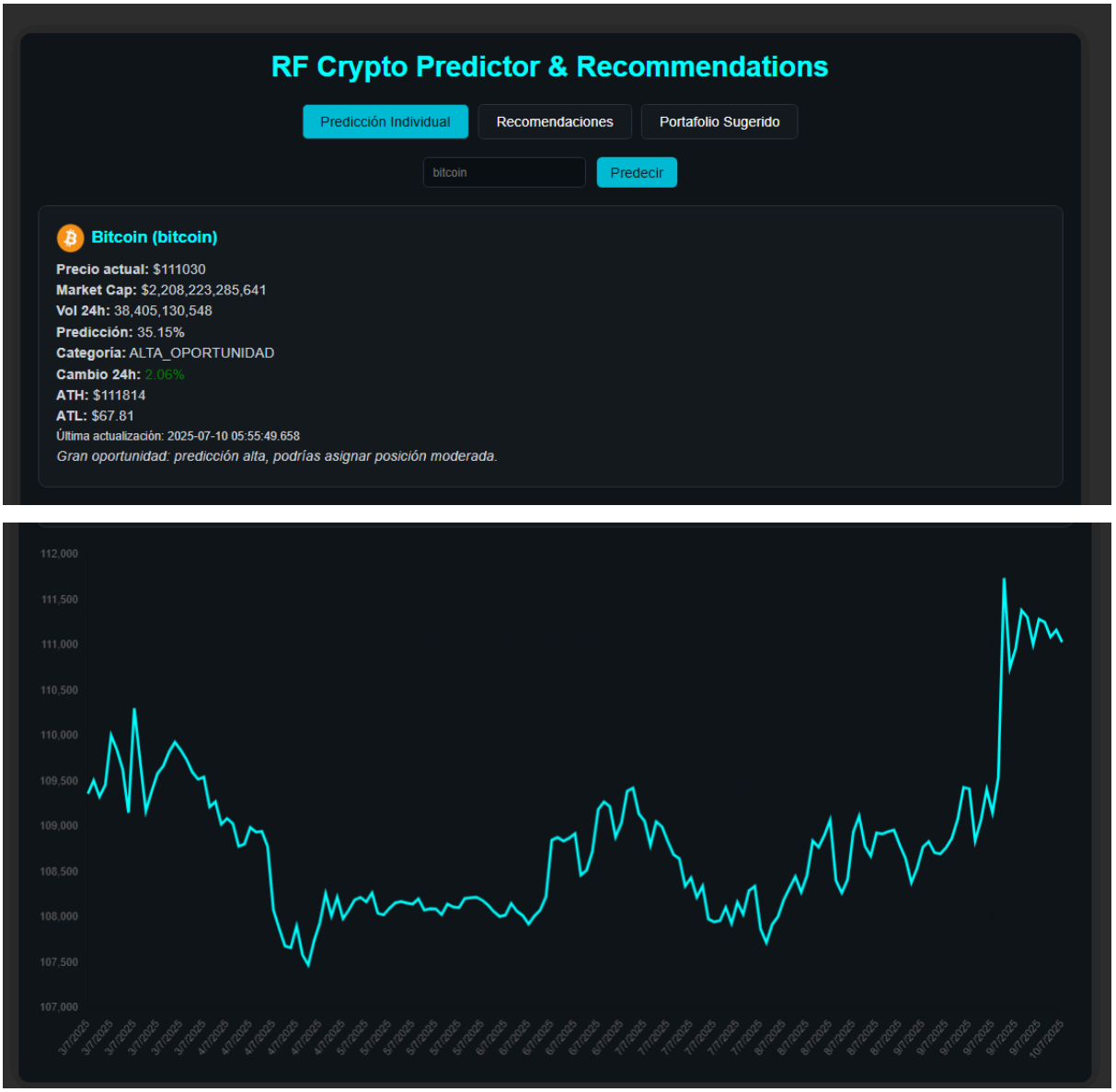

Figura 4: Funcionalidad para predecir la revalorización futura de una criptomoneda.

- **Generar recomendaciones** filtradas por tolerancia a riesgo (BAJO, MEDIO, ALTO). Bajo el capuchón, la función generate_recommendations analiza las top 50 cryptos por market cap, aplica el modelo, calcula un score técnico y asigna nivel de riesgo según capitalización y volatilidad.

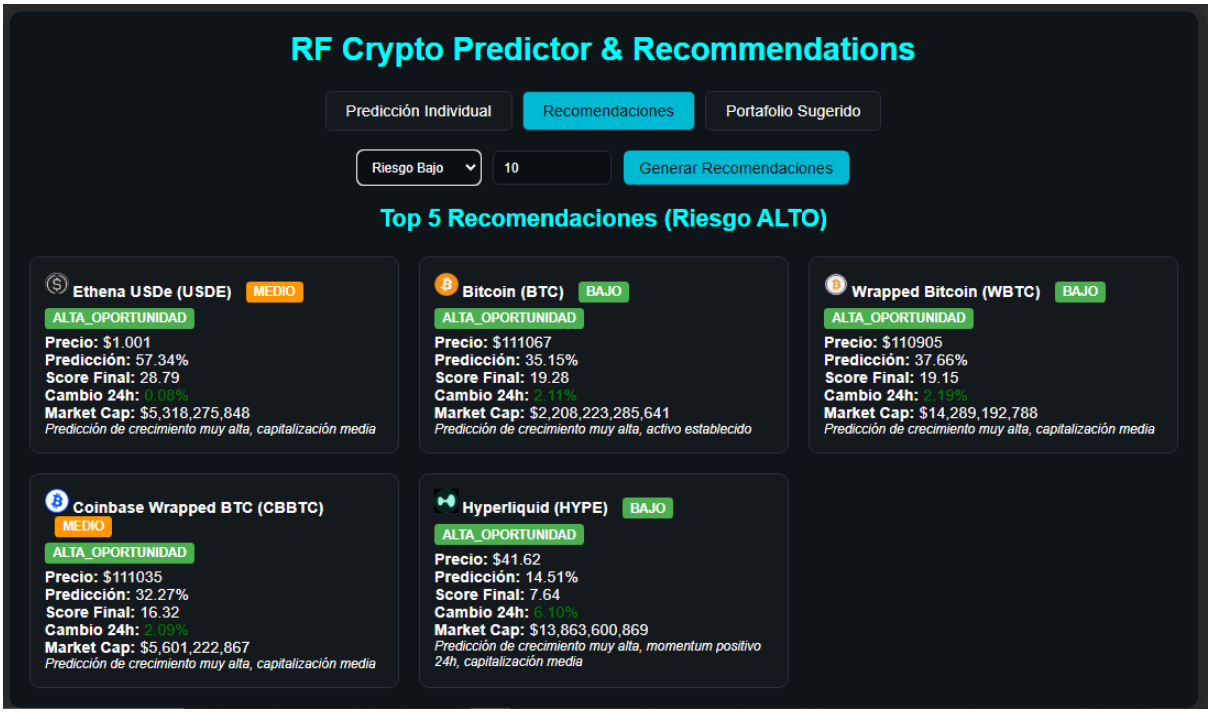

Figura 5: Funcionalidad para filtrar de acuerdo a una cantidad y un riesgo específico.

- **Recibir un portafolio diversificado**, asignando porcentajes del presupuesto (40 % a alta oportunidad, 35 % a moderada, 25 % a baja) y mostrando para cada activo la inversión sugerida, cantidad de unidades y porcentaje del total.

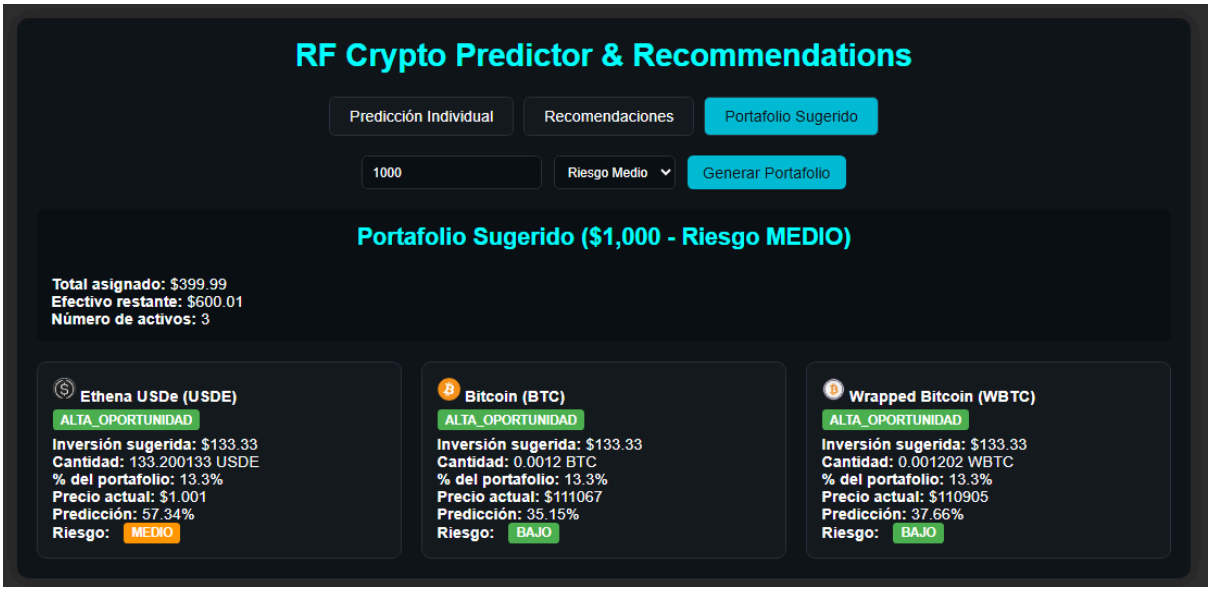

Figura 6: Funcionalidad para generar un portafolio sugerido de acuerdo a un riesgo específico

En paralelo, las API ofrecen:

- /api/predict-crypto (POST): devuelve JSON con predicción y categoría para un símbolo dado.
- /api/recommendations (GET): lista de recomendaciones según tolerancia al riesgo y límite de activos.
- /api/portfolio (POST): sugerencia de cartera basada en presupuesto y perfil de riesgo, con totales de inversión y efectivo restante.

Con esta arquitectura, logramos una solución:

- **Escalable**, gracias al caché y a la separación clara entre lógica de negocio y presentación.
- **Reproducible**, al fijar semillas y versionar escaladores.
- **Rápida**, cumpliendo el requisito de ranking en menos de 5 min tras ingesta de datos on-chain.

3.7. Cronograma de trabajo

El proyecto sigue un ciclo de semanas, asignando a cada fase tareas específicas para cumplir los objetivos de la metodología CRISP-DM y las necesidades de Perú C-Inversiones:

Semana	Actividad Clave
1	• Comprensión del negocio: objetivos, criterios, stakeholders. • Adquisición inicial de datos on-chain y off-chain; filtrado por narrativa y baja capitalización.
2	• Limpieza y preprocesamiento: manejo de nulos/infinitos, winsorización de la variable objetivo • Ingeniería de características: ratios ATH/ATL, momentum_score, volumen_relativo, narrativa_score.
	• Modelado y validación: entrenamiento de Random Forest, MLP y Autoencoder+KMeans;

3	búsqueda de hiperparámetros. • Evaluación con métricas de regresión (MAE, R^2, AUC-ROC) y trading (Precision, Recall, F1, Sharpe, drawdown).
4	• Despliegue: serialización de modelos y escaladores; desarrollo de API Flask e interfaz web; configuración de caché. • Pruebas finales, ajustes y documentación: tests de robustez, redacción del informe y preparación de la exposición.

4. Ingeniería de características

4.1. Selección de variables

Para el desarrollo del modelo de predicción, se seleccionaron inicialmente las siguientes variables numéricas provenientes del dataset original de CoinGecko:

- current_price: Precio actual del activo.
- total_volume: Volumen total de transacciones.
- ath: Precio máximo histórico.
- atl: Precio mínimo histórico.
- price_change_percentage_24h: Variación porcentual en las últimas 24 horas.
- ath_change_percentage: Variación respecto al ATH.
- atl_change_percentage: Variación respecto al ATL.

Estas variables fueron elegidas por su relevancia directa en la evaluación del comportamiento reciente y potencial de una criptomoneda.

4.2. Creación y transformación de nuevas variables

Con el objetivo de enriquecer el conjunto de datos y capturar patrones más complejos, se generaron nuevas variables derivadas:

- ratio_precio_ath: Relación entre el precio actual y el ATH. Indica cuánto margen de crecimiento tiene el activo.
- ratio_precio_atl: Relación entre el precio actual y el ATL. Útil para medir recuperación desde mínimos históricos.

- volatilidad_ath_atl: Diferencia relativa entre ATH y ATL. Mide la volatilidad histórica.
- momentum_score: Indicador compuesto que pondera cambios recientes y extremos de precio.
- volumen_relativo: Volumen ajustado por precio, útil para detectar actividad inusual.
- narrativa_score: Valor numérico asignado a cada narrativa según su potencial percibido.

Estas transformaciones permitieron capturar relaciones no lineales y señales de mercado que no eran evidentes sólo con las variables originales.

4.3. Adecuación de características a los algoritmos ML

Para asegurar la compatibilidad y el rendimiento óptimo de los modelos de Machine Learning, se aplicaron las siguientes técnicas de adecuación:

- Normalización: Se utilizó StandardScaler para escalar todas las variables numéricas a media 0 y desviación estándar 1. Esto fue especialmente importante para modelos sensibles a la escala como redes neuronales (MLP).
- Conversión a tensores: Para el entrenamiento del modelo MLP en PyTorch, los datos fueron convertidos a tensores de tipo float32.
- Tratamiento de valores extremos: Se aplicó winsorización a la variable objetivo para evitar que valores atípicos afecten el aprendizaje de los modelos.
- Manejo de valores nulos e infinitos: Se reemplazaron valores infinitos por NaN y luego se imputaron con ceros.

Estas adecuaciones garantizaron que los modelos pudieran aprender de manera eficiente y sin sesgos numéricos.

4.4. Adquisición adicional de datos

Aunque la mayoría de los datos fueron obtenidos directamente desde CoinGecko, se realizó una etiquetación manual adicional para asignar a cada criptomoneda una narrativa temática. Aunque esta clasificación estaba disponible en la API, las limitaciones de la versión gratis del API, así como la poca distribución de datos pertenecientes a las 4 narrativas requeridas, hicieron que fuera preferible construir un script personalizado que mapea cada uno de los tokens a las categorías designadas que son: IA, Videojuegos, RWA y Memes.

Esta adquisición adicional fue clave para alinear el modelo con los intereses del cliente (Perú C-Inversiones), quien solicitó recomendaciones basadas en estas narrativas específicas.

5. Experimentos

5.1. Diseño de experimentos

El objetivo de los experimentos fue evaluar distintos enfoques de modelado para predecir el cambio futuro en el precio de criptomonedas de baja capitalización, considerando tanto el rendimiento predictivo como su aplicación en escenarios de inversión real.

Se diseñaron los siguientes experimentos de modelización:

- Modelo de regresión tradicional: Random Forest Regressor como línea base.
- Modelo de red neuronal: MLP (Multilayer Perceptron) entrenado con PyTorch.
- Modelo no supervisado: Autoencoder + KMeans para detección de patrones y agrupación de oportunidades.

Cada modelo fue evaluado no solo con métricas tradicionales (MAE, MSE, R^2), sino también con métricas adaptadas al contexto de inversión, como retorno simulado, precisión en detección de oportunidades y AUC-ROC.

5.2. Ejecución y ajustes

Durante la ejecución de los experimentos se realizaron los siguientes pasos:

1.División de datos: 80% entrenamiento, 20% prueba, con random_state=42.
2.Validación cruzada: aplicada al modelo Random Forest para estimar su error medio absoluto (MAE) de forma robusta.
3.Normalización: aplicada a todas las características y a la variable objetivo para el MLP.
4.Ajuste de hiperparámetros: se realizaron pruebas iterativas para mejorar el rendimiento de cada modelo.
5.Evaluación comparativa: se calcularon métricas de clasificación binaria (precisión, recall, F1-score) usando un umbral del 5% para definir oportunidades de inversión.
6.Visualización de resultados: se generaron gráficos comparativos de métricas, retorno simulado y curva ROC para facilitar la interpretación.

6. Validación de resultados y pruebas

6.1. Definición de métricas de evaluación

Para evaluar el rendimiento de los modelos desarrollados, se utilizaron métricas tanto de regresión como de clasificación, adaptadas al contexto de inversión:

- **MAE (Mean Absolute Error):** mide el error promedio absoluto entre la predicción y el valor real.
- **MSE (Mean Squared Error):** penaliza errores grandes, útil para evaluar estabilidad.
- **R^2 (Coeficiente de Determinación):** indica qué proporción de la varianza es explicada por el modelo.
- **Precisión, Recall y F1-Score:** métricas de clasificación binaria que evalúan la capacidad del modelo para detectar oportunidades de inversión.
- **AUC-ROC:** mide la capacidad del modelo para distinguir entre oportunidades y no oportunidades.
- **Retorno simulado (%):** métrica personalizada que estima el rendimiento de una estrategia de inversión basada en las predicciones del modelo.

6.2. Interpretación de resultados

Los resultados obtenidos mostraron que el modelo Random Forest fue el más robusto y efectivo:

- MAE bajo y R^2 alto, lo que indica buena capacidad predictiva.
- F1-Score superior al 90%, reflejando un buen equilibrio entre precisión y recall.
- AUC-ROC de 0.9822, lo que representa una capacidad casi perfecta de clasificación.
- Retorno simulado superior al 11,000%, lo que sugiere que el modelo puede identificar oportunidades de inversión altamente rentables.

Los modelos MLP y Autoencoder + KMeans también mostraron buen desempeño, aunque ligeramente inferior. Así, el MLP destacó por su capacidad de generalización, mientras que el Autoencoder permitió identificar patrones útiles para segmentación.

Por otro lado, cabe resaltar que aunque el modelo fue entrenado como un problema de regresión, se realizó una conversión a clasificación binaria con el objetivo de evaluar su utilidad en escenarios reales de inversión. Esto se realiza ya que desde la perspectiva del cliente, lo más relevante no es conocer el valor exacto del crecimiento futuro, sino identificar si una criptomoneda representa o no una oportunidad de inversión. Por ello, se definió un umbral de decisión que permite transformar el problema en una tarea de clasificación binaria:

- Clase 1 (Oportunidad): si la predicción del modelo es mayor al 5%.
- Clase 0 (No oportunidad): si la predicción es menor o igual al 5%.

Esta conversión permite aplicar métricas de clasificación como precisión, recall, F1-score y AUC-ROC, que son más interpretables para decisiones binarias (invertir o no invertir).

6.3. Pruebas de robustez y validación cruzada

Para asegurar la robustez de los resultados, se aplicaron las siguientes técnicas:

- **Validación cruzada (k-fold):** se utilizó validación cruzada con 5 particiones en el modelo Random Forest, lo que permitió estimar su error de forma más confiable y reducir el riesgo de overfitting.
- **Evaluación con diferentes umbrales:** se probaron distintos umbrales de clasificación para analizar la sensibilidad del modelo.
- **Winsorización de la variable objetivo:** se aplicó para reducir el impacto de valores extremos y mejorar la estabilidad del entrenamiento.
- **Comparación entre modelos:** se realizó una evaluación comparativa entre los tres enfoques para identificar fortalezas y debilidades relativas.

Estas pruebas confirmaron que el modelo elegido (Random Forest) es consistente, generalizable y útil para apoyar decisiones de inversión en criptomonedas emergentes.

7. Comunicación

7.1. Diseño de la interfaz de usuario básica

Para facilitar la interacción del cliente con el modelo de predicción, se desarrolló una interfaz web amigable utilizando Flask y tecnologías web modernas (HTML, CSS, JavaScript). Esta interfaz fue diseñada con el objetivo de mostrar las funcionalidades más útiles para el cliente que se alinean con el objetivo y la propuesta del proyecto.

La aplicación cuenta con tres funciones principales:

- **Predicción individual:** permite al usuario ingresar el nombre o símbolo de una criptomoneda y obtener una predicción personalizada sobre su potencial de crecimiento, junto con una categorización de oportunidad (alta, moderada, baja o no recomendada).
- **Recomendaciones:** genera un listado de criptomonedas recomendadas según el nivel de tolerancia al riesgo del usuario (bajo, medio o alto), mostrando métricas clave como predicción, categoría, riesgo y score final.
- **Portafolio sugerido:** permite al usuario ingresar un presupuesto y recibir una propuesta de distribución de inversión diversificada, basada en las predicciones y categorías del modelo.

La interfaz incluye elementos visuales como gráficos de precios, insignias de riesgo y categorías, y tarjetas informativas para cada activo, lo que hace que la comprensión y experiencia del usuario sean más eficientes y prácticas.

7.2. Mecanismos de presentación de resultados al cliente

Los resultados del modelo se presentan de forma clara y estructurada mediante los siguientes mecanismos:

- **Predicción numérica (%):** muestra el valor estimado de crecimiento futuro.
- **Categoría de oportunidad:** clasifica cada activo en función de su predicción.
- **Gráficos de precios históricos:** permiten visualizar la evolución reciente del activo.
- **Etiquetas de riesgo:** indican si el activo es de riesgo bajo, medio o alto.
- **Motivos de recomendación:** se genera una explicación textual personalizada que justifica por qué un activo fue recomendado.

- **Distribución de portafolio:** se muestra el monto sugerido a invertir en cada activo, junto con su porcentaje dentro del portafolio total.

8. Conclusiones

8.1. Principales hallazgos

El desarrollo del sistema permitió identificar criptomonedas de baja capitalización con alto potencial de valorización en el corto plazo, evaluando indicadores técnicos y financieros a partir de datos recolectados de CoinGecko. Uno de los principales hallazgos fue que ciertos activos con comportamiento reciente positivo y características específicas (como alto volumen y bajo cambio negativo desde el ATH) tienden a tener mejores predicciones de valorización.

Además, se observó que la combinación de predicción de crecimiento esperada y una puntuación técnica basada en volumen, capitalización y momentum, permite mejorar la selección final, filtrando aquellas monedas con menor potencial o mayor riesgo.

8.2. Técnicas utilizadas y su impacto

Se utilizó un modelo de regresión basado en Random Forest para predecir el crecimiento porcentual esperado de una criptomoneda. El modelo fue entrenado con variables como el precio actual, volumen, máximos y mínimos históricos, y sus respectivas variaciones porcentuales. Para garantizar la consistencia del modelo, se aplicó una normalización previa mediante MinMaxScaler tanto a los datos de entrada como a las salidas. Las predicciones fueron categorizadas en distintos niveles de oportunidad para facilitar su interpretación. El modelo fue integrado a una aplicación web construida con Flask, la cual permite realizar predicciones individuales, generar recomendaciones personalizadas según la tolerancia al riesgo, y simular la distribución de un portafolio de inversión. Esta combinación de técnicas permitió construir una herramienta funcional, flexible y visualmente amigable, con potencial para ser usada como guía inicial por usuarios interesados en inversiones en activos emergentes.

8.3. Líneas de trabajo futuro

Se plantea como línea futura de mejora la ampliación del conjunto de características, incluyendo indicadores adicionales como análisis de sentimiento,

datos de redes sociales o métricas on-chain. También se propone evaluar modelos de predicción más complejos, así como incorporar mecanismos de retroalimentación y actualización automática del modelo. Finalmente, se contempla el despliegue en la nube para facilitar su acceso público, así como mejoras en la experiencia de usuario y adaptación a distintos idiomas.

9. Referencias bibliográficas

Akgul, A., Şahin, E. E., & Şenol, F. Y. (2022). *Blockchain-based cryptocurrency price prediction with chaos theory, onchain analysis, sentiment analysis and fundamental-technical analysis*. Chaos Theory and Applications, 4(3), 157–168. https://dergipark.org.tr/en/download/article-file/2750466

Nica, O., Piotrowska, K., & Schenk-Hoppé, K. R. (2022). *Cryptocurrencies: Concept and current market structure*. En *Cryptofinance: A new currency for a new economy* (Cap. 1, pp. 1–28). World Scientific Publishing Company. https://www.worldscientific.com/doi/epdf/10.1142/9789811239670_0001

Cadena Díaz, P. A., & Rincón León, H. A. (2018). *¿Qué son las criptomonedas?* [Trabajo de grado, Universidad La Gran Colombia, Especialización en Gerencia Financiera]. Universidad La Gran Colombia. https://repository.ugc.edu.co/server/api/core/bitstreams/17789f18-b517-4911-9ab6-46273bbaee68/content

Cozzo Villafañe, P. A. (2025). Las nuevas economías de criptomonedas frente al lavado de activos. *Revista Blockchain e Inteligencia Artificial*, 6, 151–153. https://dergipark.org.tr/en/download/article-file/2750466

10. Anexos

Link del repositorio en Github: https://github.com/Meza27/CC57-TF-2025-1.git