



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Dpto. de Matemàtica Aplicada

Predicción del comportamiento de criptomonedas a partir
de métodos de inteligencia artificial

Trabajo Fin de Máster

Máster Universitario en Investigación Matemática

AUTOR/A: Ferrando Esteve, Aldo

Tutor/a: Conejero Casares, José Alberto

Cotutor/a: Milián Enrique, Carles

CURSO ACADÉMICO: 2022/2023



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



VNIVERSITAT
DE VALÈNCIA

investmat

Predicción del comportamiento de criptomonedas a partir de métodos de inteligencia artificial

Aldo Ferrando Esteve

Trabajo Final de Máster

Máster Universitario en
Investigación Matemática

Julio 2023

Tutores: J. Alberto Conejero
y Carles Milián

Dedicatoria

A mi familia.

Resumen

Una criptomoneda es un medio digital de intercambio que utiliza criptografía fuerte para asegurar las transacciones, controlar la creación de unidades adicionales y verificar la transferencia de activos usando tecnologías de registro distribuido. Las criptomonedas tienen control descentralizado, es decir, el control de cada moneda funciona a través de una base de datos descentralizada, usualmente una cadena de bloques “blockchain” que sirve como una base de datos de transacciones financieras pública. La predicción es una tarea común en los modelos de ciencia de datos que ayuda a las grandes corporaciones y gobiernos a planear, establecer objetivos y prevenir errores. En muchos modelos de predicción es casi más importante la detección del margen de error en las predicciones, así como la interpretación de los datos que nos ofrecen estos modelos. Con este trabajo tratamos de crear una herramienta para la inversión a corto plazo en criptomonedas a partir de datos financieros históricos del módulo yfinance de Yahoo.

Palabras clave: Criptomoneda, BlockChain, Python, Prophet, yfinance.

Índice general

1. Introducción.....	5
2. Criptomonedas	6
2.1 Métodos utilizados para predicción de señales financieras	7
2.2 Estado del arte sobre el problema de predicción de criptomonedas	8
3. Métodos de Python	12
3.1 Python.....	12
3.2 Google Colaboratory	13
3.3. Módulo yfinance Yahoo	14
3.4. Módulo Prophet Facebook	15
4. Análisis financiero	18
4.1 Análisis financiero de las gráficas obtenidas	18
4.2 Análisis de cambios debido a variables externas	20
4.3. Predicciones de cotizaciones de acciones en mercados bursátiles.....	21
4.4. Valoración de estrategias de inversión	23
5. Exponente de Hurst.....	25
6. Conclusiones	28
Referencias y bibliografía.....	29
ANEXO 1: Implementación en Prophet de la predicción de la cotización de criptomonedas	33
ANEXO 2: Modificaciones para la predicción de la cotización de cotizaciones bursátiles	34
ANEXO 3. Valoración de estrategias de inversión.....	34
ANEXO 4: Cálculo del exponente de Hurst.....	37

1. Introducción

Una criptomoneda es un activo digital negociable o una forma digital de dinero, construida sobre la tecnología de cadena de bloques que solo existe en línea. Las criptomonedas utilizan el cifrado para autenticar y proteger las transacciones, de ahí su nombre. Actualmente hay más de mil criptomonedas diferentes en el mundo.

Python es un lenguaje open source de propósito general, pero gracias al desarrollo de potentes librerías analíticas, procesamiento de datos y modelización predictiva se ha convertido en el principal lenguaje de programación utilizado para proyectos de Data Science, junto con R.

Utilizando el módulo `yfinance` podemos obtener datos financieros de Yahoo Finance utilizando Python. De esta manera, podemos recuperar información financiera de la empresa (por ejemplo, índices financieros), así como datos históricos del mercado.

La biblioteca Prophet es un módulo de código abierto diseñado para hacer pronósticos para conjuntos de datos de series temporales. Es fácil de usar y está diseñado para encontrar automáticamente un buen conjunto de hiperparámetros para el modelo en un esfuerzo por hacer pronósticos hábiles para datos con tendencias y estructura estacional.

En este trabajo utilizamos estas bibliotecas con Python para la obtención de gráficas de predicción de valores de las criptomonedas. Así, utilizando estos gráficos a corto plazo tenemos una herramienta para facilitar la inversión de criptomonedas. En particular, establecemos los siguientes objetivos en este trabajo.

La memoria está estructurada como se indica a continuación.

En el [Capítulo 2](#), introducimos brevemente las criptomonedas, haciendo referencia a la tecnología blockchain, explicando cómo la tecnología blockchain respalda a aquellas que se gestionan de manera distribuida. Además, revisamos algunos trabajos recientes sobre la predicción del comportamiento de la cotización de las criptomonedas. En el [Capítulo 3](#), se introducen las herramientas que utilizaremos basadas en Python que vamos a utilizar: los módulos Prophet de Meta y `yFinance` de Yahoo. En el [Capítulo 4](#) presentamos los resultados de los análisis del comportamiento de algunas criptomonedas y su aplicación a otras señales bursátiles. En el [Capítulo 5](#), se analiza el uso del exponente de Hurst a la hora de interpretar los resultados. Por último, en el [Capítulo 6](#), se establecen las conclusiones.

2. Criptomonedas

Una criptomoneda se define como cualquier forma de moneda que solo existe digitalmente, que por lo general no tiene una autoridad emisora o reguladora central, sino que utiliza un sistema descentralizado para registrar transacciones y gestionar la emisión de nuevas unidades, se basa en la criptografía para evitar la falsificación y las transacciones fraudulentas ([Catalini, Gans; 2018](#))

Las criptomonedas han ganado recientemente mucho interés por parte de inversores, bancos centrales y gobiernos de todo el mundo. La falta de cualquier forma de regulación política y su mercado lejos de ser “eficiente”, requieren nuevas formas de regulación en un futuro cercano. Desde un punto de vista econométrico, se ha encontrado que el proceso subyacente a la evolución de la volatilidad de las criptomonedas exhibe al mismo tiempo diferencias y similitudes con otras series temporales financieras, como las cotizaciones bursátiles o de intercambio de divisas ([Catania, Grassi, Ravazzolo; 2018](#)).

Los registros individuales de propiedad de criptomonedas se almacenan en un libro digital, que es una base de datos que utiliza criptografía fuerte para proteger los registros de transacciones, controlar la creación de monedas adicionales y verificar la transferencia de la propiedad de las monedas. A pesar de su nombre, las criptomonedas no se consideran necesariamente monedas en el sentido tradicional y, aunque se les ha aplicado diversos tratamientos categóricos, incluida la clasificación como productos básicos, valores y/o monedas, las criptomonedas son generalmente vistas en la práctica como una clase de activo distinto ([Chuen et al; 2017](#)).

Algunos esquemas criptográficos usan validadores para mantener la criptomoneda. En un modelo de prueba de participación, los propietarios ponen sus tokens como garantía. A cambio, obtienen autoridad sobre el token en proporción a la cantidad que apuestan. ([Barreiro-Gómez, Tembine; 2017](#)). En general, estos jugadores de token obtienen propiedad adicional en el token con el tiempo a través de tarifas de red, tokens recién acuñados u otros mecanismos de recompensa similares. Las criptomonedas no existen en forma física (como el papel moneda) y no las emite ninguna autoridad central. Las criptomonedas suelen utilizar un control descentralizado en lugar de una moneda digital de un banco central. Cuando una criptomoneda se acuña la emite un solo emisor, y en ese caso se considera centralizada. En cambio, cuando se implementa con control descentralizado, cada criptomoneda funciona a través de tecnología de registro distribuido, generalmente mediante una cadena de bloques, blockchain, que sirve como una base de datos de transacciones financieras públicas ([Tapsscott, Tapsscott; 2017](#)).

Una cadena de bloques es una lista creciente de registros, llamados bloques, que se vinculan mediante criptografía. Cada bloque contiene un hash criptográfico del bloque anterior, una marca de tiempo y datos de transacción (generalmente representados como un “hash tree”). La marca de tiempo prueba que los datos de la transacción existían cuando se publicó el bloque para ingresar

a su hash. Como cada bloque contiene información sobre el bloque anterior, forman una cadena, y cada bloque adicional refuerza a los anteriores. Por lo tanto, las cadenas de bloques son resistentes a la modificación de sus datos porque, una vez registrados, los datos de un bloque determinado no pueden modificarse retroactivamente sin alterar todos los bloques posteriores ([Mukhopadhyay et al 2017](#)),

Bitcoin, lanzado por primera vez como software de código abierto en 2009, es la primera criptomoneda descentralizada ([Nakamoto; 2018](#)). Desde el lanzamiento de bitcoin, se han creado muchas otras criptomonedas. En 2014 los cajeros automáticos de bitcoin comenzaron a surgir en todo el mundo, lo que permite a las personas cambiar efectivo por la criptomoneda, un pago digital seguro fuera de las instituciones financieras convencionales.

Según Jan Lansky ([2018](#)), una criptomoneda es un sistema que cumple seis condiciones:

- El sistema no requiere una autoridad central, su estado se mantiene a través del consenso distribuido.
- El sistema mantiene una descripción general de las unidades de criptomonedas y su propiedad.
- El sistema define si se pueden crear nuevas unidades de criptomonedas. Si esto es posible, el sistema define las circunstancias de su origen y cómo determinar la propiedad de estas nuevas unidades.
- La propiedad de las unidades de criptomonedas se puede probar exclusivamente criptográficamente.
- El sistema permite realizar transacciones en las que se cambia la propiedad de las unidades criptográficas. Una declaración de transacción solo puede ser emitida por una entidad que demuestre la propiedad actual de estas unidades.
- Si se introducen simultáneamente dos instrucciones distintas de cambio de titularidad de las mismas unidades criptográficas, el sistema realiza como máximo una de ellas.

2.1 Métodos utilizados para predicción de señales financieras

En la investigación financiera, los modelos de aprendizaje automático (*machine learning*) y de aprendizaje profundo (*deep learning*) se han convertido en herramientas importantes para el análisis de datos y la predicción de precios de criptomonedas. Los modelos tradicionales como los modelos de promedio móvil integrado autorregresivo (ARIMA) son utilizados por muchos analistas financieros y siguen siendo una opción popular para predecir precios a corto plazo. Sin embargo, los modelos de *deep learning*, como las redes neuronales recurrentes (RNN) y en particular las Long Short Term Memory (LSTM) ([Hocreicter, Schmidhuber; 1997](#)) y la Gate Recurrent Units (GRU) ([Cho et al, 2014](#)) han demostrado ser más efectivas para predecir y analizar los factores que afectan los precios de las criptomonedas a largo plazo.

Una de las principales ventajas de los modelos de deep learning es que son capaces de analizar grandes cantidades de datos de diferentes fuentes. Esto es especialmente útil en el análisis de criptomonedas, donde el volumen de datos es enorme y la información se encuentra en diferentes plataformas. Además, las redes neuronales son capaces de integrar diferentes fuentes de información y procesarlas para hacer predicciones sobre los precios futuros.

Estos modelos son mucho más adecuados que los modelos deterministas, dado que estos últimos asumen que los precios futuros dependen únicamente de los precios pasados y de las variables de entrada. Si bien es cierto que estos modelos se comportan bien a largo plazo, no suelen ser aplicables para la predicción de precios en tiempo real debido a la naturaleza estocástica de los mercados financieros. Las redes neuronales pueden solventar esta limitación mediante la introducción de pesos de manera estocástica o en los resultados de las funciones de transferencia

Además, mediante el procesamiento de lenguaje natural se pueden desarrollar técnicas automáticas para analizar el sentimiento en la información de las redes sociales y utilizarla para la inversión en mercados bursátiles ([Zhang, Skiena; 2010](#)). Además, la creciente base de usuarios de las redes sociales y el alto volumen de publicaciones también brindan información valiosa sobre el sentimiento para predecir la fluctuación del precio de criptomonedas por lo que algunos estudios utilizan redes neuronales de memoria a corto plazo (LSTM) ([Huang et al; 2021](#)).

2.2 Estado del arte sobre el problema de predicción de criptomonedas

A continuación, revisamos algunos trabajos destacados sobre la predicción de cotizaciones de criptomonedas. En ([Rebane et al, 2018](#)), se compara el rendimiento del modelo de ARIMA con el de una red neuronal multicapa profunda recurrente construida a partir de Seq2Seq que utiliza una selección variada de tipos de entradas para predecir las cotizaciones de Bitcoin. El estudio encuentra que ambos modelos son efectivos para predecir los precios de las criptomonedas a medio y largo plazo, pero que las redes neuronales construidas superan ampliamente a los modelos ARIMA en términos de precisión. Como cabe esperar, al aumentar el número de capas ocultas y de neuronas generalmente mejora el rendimiento de los modelos. Además, el estudio encuentra que el rendimiento de ambos modelos se puede mejorar al incluir más datos históricos y al usar otras variables, como el volumen de *trading*. Sin embargo, los investigadores señalan que la precisión de predicción de ambos modelos está limitada por la volatilidad inherente y la imprevisibilidad de los mercados de criptomonedas y que en los periodos de mayor volatilidad, la precisión de las predicciones bajan.

Catania et al ([2018a](#)) probaron diferentes modelos estadísticos multivariantes autorregresivos vectoriales, a diferentes escalas temporales. Utilizaron modelos de regresión lineal sencillos, así como combinaciones de otros modelos. AR1 (Con un retardo), KS (Kolmogorov-Smirnov), DMA (desviación media absoluta) y DMS (diferencia mínima significativa). Los resultados de

este trabajo muestran grandes mejoras estadísticas significativas en el pronóstico puntual de Bitcoin y Ethereum cuando se utilizan combinaciones de modelos univariados y pronósticos de densidad para todas las criptomonedas cuando se basan en la selección de modelos multivariantes variables en el tiempo.

Siguiendo en la línea de modelos estadísticos, Catania et al ([2018b](#)) estudiaron la volatilidad condicional de las cuatro criptomonedas más negociadas en el momento: Bitcoin, Ethereum, Litecoin y Ripple. Se investiga el efecto de tener en cuenta la memoria larga en el proceso de volatilidad, así como su reacción asimétrica a valores pasados de la serie para predecir niveles de volatilidad. El conjunto de modelos que se consideran incluye el modelo GARCH de Bollerslev ([1986](#)) (M_1), el modelo Score Driven model with conditional Generalized Hyperbolic Skew Student's t (GHSKT) (M_2) junto con tres extensiones con: i) apalancamiento (M_3), ii) asimetría variable en el tiempo (M_4), y iii) integración fraccional en el proceso de volatilidad (M_5). Los resultados indican que M_1 es generalmente superado por el modelo más refinado Score Driven-GHSKT, M_2 . Las ganancias aumentan cuando el horizonte de pronóstico crece. Encontramos que, para Bitcoin, M_2 reporta mejores resultados que sus extensiones M_3 , M_4 y M_5 . Los hallazgos indicaron que los modelos de volatilidad más sofisticados que incluyen apalancamiento y sesgo variable en el tiempo pueden mejorar las predicciones de volatilidad en diferentes horizontes de pronóstico del 1% al 6% en comparación con alternativas más estándar. Las aplicaciones en optimizaciones de cartera, cobertura y fijación de precios de valores derivados, donde el modelado de volatilidad es de importancia primordial, pueden beneficiarse de estos hallazgos.

Derbentsev et al ([2019](#)) adaptaron los modelos estándar de árboles de regresión (*Binary Auto Regressive Tree*) y combinándolos con algoritmos de clasificación y modelos autorregresivos ARIMA para la predicción de cotizaciones a corto plazo de la cotización de las criptomonedas Bitcoin, Ethereum y Ripple. Para construir este modelo se utilizan 2 ramas, es decir, cada nodo tiene dos nodos hijos. El árbol final se elige a partir de estos nodos evaluando la información no solo de los nodos predictores que dividen la serie temporal en subconjuntos, sino también de aquellos que separan un determinado grupo de subconjuntos del conjunto, es decir, separan un subárbol del resto del árbol. La construcción de un árbol de regresión parte de un único valor (nodo raíz), que se define como la mediana de toda la serie temporal. El proceso de división se repite y se van variando las divisiones. Si en cada paso la división seleccionada mejora el modelo y es válida con una ganancia de información en términos de entropía, entonces se realiza esta división y se repite el procedimiento. Estos autores mostraron que este enfoque es más preciso que los modelos ARIMA-ARFIMA en el pronóstico de series temporales de criptomonedas tanto en los períodos de aumento lento como en los períodos de transición entre diferentes dinámicas. Las criptomonedas se negocian como acciones, brindan una alternativa factible al oro y se aprecian durante tiempos inciertos por lo que se puede suponer que sus precios están determinados

en parte por los índices bursátiles mundiales. Mallaldi y Deeriyah (2021) prueban esta hipótesis realizando un análisis de series temporales de rendimientos y volatilidades de Bitcoin y Ripple utilizando modelos autorregresivos. Mostraron que la caída de Bitcoin de 2018 podría haberse explicado utilizando estos métodos de series temporales y que los rendimientos de los mercados bursátiles mundiales y del oro no tienen un efecto causal en los rendimientos de Bitcoin, pero encontramos sí en el caso de los rendimientos de Ripple.

Dada la volatilidad de estas señales, en (Jay et al; 2020) se propone un enfoque diferente para predecir el precio de las criptomonedas utilizando redes neuronales estocásticas, argumentando que la volatilidad de los precios de las criptomonedas se puede capturar con mayor precisión modelando los procesos estocásticos subyacentes en lugar de los modelos deterministas que se usan comúnmente en los modelos financieros tradicionales. El modelo de red neuronal estocástica (SNN) propuesto utiliza una combinación de redes neuronales recurrentes (RNN) y procesos estocásticos para capturar los componentes deterministas y estocásticos de la dinámica de precios de las criptomonedas. Se utilizan datos históricos de precios y volumen de *trading* de Bitcoin, Ethereum y Litecoin para entrenar y validar su modelo SNN. Se consideraron arquitecturas de perceptrón multicapa y de LSTM. El modelo propuesto está basado en caminos aleatorios (*random walks*) induce aleatoriedad por capas en las activaciones de características observadas de las redes neuronales para simular la volatilidad del mercado y también se incluye una técnica para aprender el patrón de reacción del mercado. Los resultados de los experimentos muestran que el modelo SNN supera a varios otros modelos de última generación en términos de precisión y rentabilidad de la negociación.

En (Charandabi, Kamyar; 2021), los autores proporcionan una descripción general del uso de redes neuronales para la predicción de precios de criptomonedas y de los estudios recientes que han hecho uso de las mismas para predecir sus cotizaciones, revisando diferentes configuraciones, datos de entrada y algoritmos. También se analizan los desafíos asociados con la predicción de los precios de las criptomonedas, como la falta de datos y el hecho de que el mercado está muy influenciado por factores externos, como noticias, regulaciones gubernamentales, guerras, etc. Se analizaron las ventajas y desventajas de cada uno de los métodos presentados en términos de tiempo transcurrido, precisión de la predicción, error medio en porcentaje (MAPE) y R^2 , véase también los métodos analizados (Jaquart et al; 2021) presentando resultados en la misma línea. En general, se concluye que las ANN son una herramienta prometedora para predecir los precios de las criptomonedas, pero que se necesita más investigación para mejorar su precisión y confiabilidad, así como profundizar en ver qué otras fuentes de datos se pueden incorporar para mejorar estas predicciones.

Patel et al ([2021](#)) han utilizado métodos basados en deep learning para predecir los valores de precios de las criptomonedas Litecoin y Monero, con el fin de ayudar a las instituciones financieras a tomar mejores decisiones relacionadas con las inversiones en criptomonedas, dado que la alta volatilidad de los precios de criptomonedas provoca que los métodos tradicionales de análisis financieros no siempre son efectivos para predecir estos precios. En este trabajo proponen un *pipeline* de tres etapas: preprocesamiento de datos, extracción de características y predicción de precios. En la etapa de procesamiento previo de datos, recopilan los datos de cotización de las criptomonedas en bruto y posteriormente los procesan para eliminar el ruido y los valores atípicos. A continuación extraen algunas de los datos procesados mediante diversas técnicas, como transformadas wavelets y análisis de componentes principales (PCA). Por último, en la etapa de predicción de precios, combinan redes neuronales convolucionales (CNN) con unidades LSTM para predecir los precios futuros de las criptomonedas en función de las características extraídas. Los resultados muestran que el esquema propuesto predice los precios de estos valores con alta precisión, lo que revela que el esquema puede ser aplicable en varias predicciones de precios de criptomonedas, permitiendo tomar decisiones de inversión informadas y aumentar potencialmente las ganancias.

Como hemos comentado anteriormente, Huang et al ([2021](#)) combinaron datos de cotización de las criptomonedas y de redes sociales, con el fin de mejorar las predicciones. Utilizaron información de la red social Sina-Weibo y aplicaron algoritmos de análisis de sentimiento para identificar y extraer información subjetiva de las publicaciones. La arquitectura combina redes convolucionales con unidades LSTM. Una arquitectura mixta ha resultado muy eficiente en otros contextos, prediciendo la evolución de fenómenos de difusión anómala ([Garibo-i-Orts et al; 2021](#)) o de casos de COVID-19 ([Lozano et al, 2021](#)). Una vez preprocesados, los datos de las redes sociales relacionados con criptomonedas fueron convertidos en *embeddings* para alimentar las LSTMs. Compararon el rendimiento de su modelo LSTM con otros modelos tradicionales de aprendizaje automático, como máquinas de vector soporte (SVM) ([Tang.; 2013](#)) o Random Forests ([Breinman; 2001](#)) y descubrieron que el nuevo modelo superó a ambos modelos en términos de precisión y puntuación. Además, se desarrollaron las herramientas para capturar publicaciones de Sina Weibo y para describir la creación del diccionario de sentimientos cripto-específicos. Los experimentos realizados demostraron que el enfoque propuesto supera al modelo basado en autorregresión de última generación en un 18,5 % en precisión y un 15,4 % en recuperación, mostrando la conveniencia de añadir fuentes de información adicionales a los datos de cotización de las criptomonedas.

3. Métodos de Python

3.1 Python

Python es un lenguaje de programación multiparadigma, potente y fácil de aprender. Tiene estructuras de datos de alto nivel eficientes y un enfoque simple pero efectivo para la programación orientada a objetos. La sintaxis es elegante y la tipificación dinámica de Python, junto con su naturaleza interpretada, lo convierten en un lenguaje ideal para secuencias de comandos y desarrollo rápido de aplicaciones en muchas áreas en la mayoría de las plataformas.

En lugar de incorporar toda su funcionalidad en su núcleo, Python fue diseñado para ser altamente extensible a través de módulos. Esta modularidad compacta lo ha hecho particularmente popular como medio para agregar interfaces programables a las aplicaciones existentes. La visión de Van Rossum (1991) de un lenguaje central pequeño con una gran biblioteca estándar y un intérprete fácilmente extensible surgió de sus frustraciones con el lenguaje de programación ABC, que defendía el enfoque opuesto.

El intérprete de Python y la extensa biblioteca estándar están disponibles gratuitamente en forma de código fuente o binario para todas las plataformas principales desde el sitio web de (Python) y pueden distribuirse gratuitamente. El mismo sitio también contiene distribuciones y punteros a muchos módulos, programas y herramientas gratuitos de Python de terceros, además de documentación adicional. El intérprete de Python se amplía fácilmente con nuevas funciones y tipos de datos implementados en C o C++ (u otros lenguajes a los que se puede llamar desde C). Python también es adecuado como lenguaje de extensión para aplicaciones personalizables.

La programación orientada a objetos y la programación estructurada son totalmente compatibles, y muchas de sus características admiten la programación funcional y la programación orientada a eventos (incluso mediante metaprogramación y metaobjetos (“magic methods”). Muchos otros paradigmas son compatibles a través de extensiones, incluido el diseño por contrato (considerar los elementos de diseño como participantes de una relación similar al contrato de negocios) y la programación lógica.

Python usa escritura dinámica y una combinación de conteo de referencias y un *garbage collector* de detección de ciclos para la administración de memoria. Además, utiliza resolución dinámica de nombres, *late binding*, que vincula métodos y nombres de variables durante la ejecución del programa. Su diseño ofrece cierto soporte para la programación funcional como en el lenguaje de programación Lisp. Tiene funciones filter, map y reduce; listas de comprensiones, diccionarios, conjuntos y generadores de expresiones. La biblioteca estándar tiene dos módulos itertools y functools que implementan herramientas funcionales prestadas de Haskell y Standard ML.

Su filosofía central se resume en el documento (Peters, 2004) que incluye aforismos como:

- Bonito es mejor que feo.
- Explícito es mejor que implícito.
- Simple es mejor que Complejo.
- Complejo es mejor que complicado.
- La legibilidad cuenta.

Python se esfuerza por lograr una sintaxis y una gramática más simples y menos desordenadas, al mismo tiempo que brinda a los desarrolladores una opción en su metodología de codificación. En contraste con el lema de Perl "hay más de una forma de hacerlo", Python adopta la filosofía de "debería haber una, y preferiblemente solo una, forma obvia de hacerlo". Martelli (2007), miembro de la Python Software Foundation y autor del libro Python, escribió: "*Describir algo como 'inteligente' no se considera un cumplido en la cultura Python*".

Los desarrolladores de Python se esfuerzan por evitar la optimización prematura y rechazan parches en partes no críticas de la implementación de referencia de CPython que ofrecería aumentos marginales en la velocidad a costa de la claridad. Cuando la velocidad es importante, un programador de Python puede mover funciones de tiempo crítico a módulos de extensión escritos en lenguajes como C usando PyPy (intérprete y compilador JIT para el lenguaje Python, que se enfoca en la velocidad y eficiencia, y es 100% compatible con el intérprete original CPython). CPython traduce una secuencia de comandos de Python a C y realiza llamadas API de nivel C directas al intérprete de Python.

Los desarrolladores de Python buscan que sea divertido de usar. Esto se refleja en su nombre, un tributo al grupo de comedia británico Monty Python, y en ocasiones en enfoques lúdicos de tutoriales y materiales de referencia, como ejemplos que se refieren a SPAM y huevos (una referencia a un sketch de Monty Python) en su lugar del estándar foobar (variables metasintácticas utilizadas en informática para nombrar variables, funciones y comandos).

Un neologismo común en la comunidad de Python es pythonic, que tiene una amplia gama de significados relacionados con el estilo del programa. El código "Pythonic" puede usar bien los modismos de Python, ser natural o mostrar fluidez en el lenguaje, o ajustarse a la filosofía minimalista de Python y al énfasis en la legibilidad. El código que es difícil de entender o se lee como una transcripción aproximada de otro lenguaje de programación se denomina "no pythonic".

3.2 Google Colaboratory

Este proyecto se ha desarrollado en el entorno de trabajo Colab, también conocido como [Colaboratory](#) que permite programar y ejecutar Python en el navegador. Algunas de las ventajas obtenidas con la utilización de este IDE son que no requiere configuración, da acceso gratuito a

GPUs y permite compartir contenido fácilmente. Los cuadernos de Colab permiten combinar código ejecutable y texto enriquecido en un mismo documento. El cuaderno utilizado en este proyecto creado en Colab se almacena en la cuenta de Google Drive. Los cuadernos de Colab son fácilmente compartibles entre compañeros de trabajo o amigos, lo permite comentarlos o incluso editarlos.. Los cuadernos de Colab son cuadernos de Jupyter alojados en Colab. Para obtener más información sobre el proyecto Jupyter (2023). Con Colab, se puede aprovechar toda la potencia de las bibliotecas más populares de Python para analizar y visualizar datos, como las librería Pandas de análisis de datos (2023), la librería de cálculo científico NumPy (2023) y o la de visualización Matplotlib (2023) que se han utilizado en este proyecto. Se pueden importar proyectos propios a los cuadernos de Colab desde la cuenta de Google Drive, incluidas las hojas de cálculo, y también desde GitHub.

Con Colab, se pueden importar un conjunto de datos de imágenes, entrenar un clasificador de imágenes con dicho conjunto de datos y evaluar el modelo con tan solo usar unas pocas líneas de código como se demuestra en el data frame importado desde Yahoo de los datos históricos de las criptomonedas (2023a) y (2023b). Los cuadernos de Colab ejecutan código en los servidores en la nube de Google, lo que te permite aprovechar la potencia del hardware de Google, incluidas las GPU y TPU, independientemente de la potencia de tu equipo. Los principales proveedores de computación en la nube han adoptado Google's Colaboratory o herramientas derivadas como una interfaz frontal para los usuarios de la nube. Los ejemplos incluyen los cuadernos de Amazon SageMaker, Jupyter Notebook y el cuaderno de Azure de Microsoft.

3.3. Módulo yfinance Yahoo

Yfinance (2023b) es una herramienta de código abierto que utiliza las API disponibles públicamente de Yahoo y tiene fines educativos y de investigación.

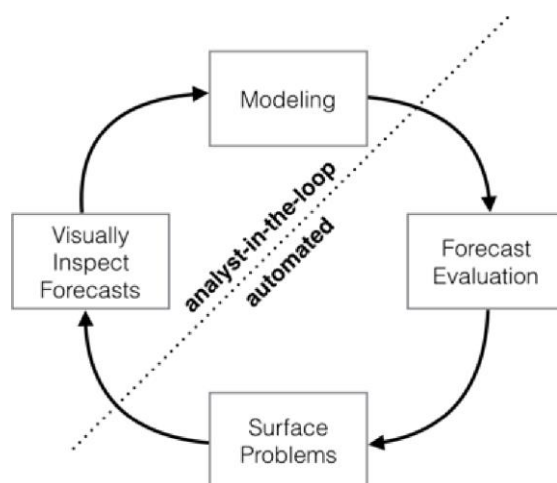
Symbol	Name	Price (Intraday)	Change	% Change	Market Cap v	Volume in Currency (Once 0.00 UTC)	Volume in Currency (24h)	Volume All Currencies (24h)	Circulating Supply	52 Week Range	Day Chart
BTC-USD	Bitcoin USD	40,810.75	+471.32	+1.17%	774,822B	32,967B	32,967B	32,967B	18,986M	28,883.62 - 68,789.83	
ETH-USD	Ethereum USD	2,765.24	+87.18	+3.26%	331,814B	15,403B	15,403B	15,403B	119,994M	1,880.37 - 4,891.79	
USDT-USD	Tether USD	1.0004	+0.0002	+0.0171%	80,155B	67,025B	67,025B	67,025B	80,122B	1.00 - 1.03	
BNB-USD	Binance Coin USD	384.66	+6.47	+1.71%	63,514B	1,671B	1,671B	1,671B	165,117M	213.82 - 880.85	
USDC-USD	USD Coin USD	0.999900	-0.000100	-0.011100%	52,460B	4,257B	4,257B	4,257B	52,466B	0.98 - 1.00	
XRP-USD	XRP USD	0.790600	-0.019700	-2.557400%	37,987B	2,307B	2,307B	2,307B	48,046B	0.46 - 1.90	
LUNA1-USD	Terra USD	88.37	-1.21	-1.35%	32,556B	2,512B	2,512B	2,512B	368,399M	3.94 - 154.58	
ADA-USD	Cardano USD	0.849400	+0.037100	+4.566500%	28,624B	1,209B	1,209B	1,209B	33,7B	0.78 - 3.10	
SOL-USD	Solana USD	87.16	+3.27	+3.90%	27,887B	2,204B	2,204B	2,204B	319,966M	12.37 - 280.08	
AVAX-USD	Avalanche USD	76.86	+6.65	+9.47%	20,496B	1,359B	1,359B	1,359B	266,676M	0.34 - 140.22	
HEX-USD	HEX USD	0.114400	+0.002600	+2.314300%	19,836B	14,591M	14,591M	14,591M	173,411B	0.01 - 0.98	
DOT-USD	Polkadot USD	19.00	+0.62	+3.38%	18,766B	1,077B	1,077B	1,077B	987,579M	10.42 - 58.00	
BUSD-USD	Binance USD USD	1.0001	+0.0011	+0.1056%	17,818B	5,997B	5,997B	5,997B	17,816B	1.00 - 1.03	
DOGE-USD	Dogecoin USD	0.117400	-0.003000	-2.578800%	15,578B	513,073M	513,073M	513,073M	132,671B	0.05 - 0.74	
UST-USD	TerraUSD USD	1.0033	+0.0021	+0.2147%	15,090B	566,255M	566,255M	566,255M	15,041B	0.92 - 1.04	
SHIB-USD	SHIBA INU USD	0.000000	0.000000	+1.771400%	12,328B	2,281B	2,281B	2,281B	549,063T	0.00 - 0.00	

Figura 1. Captura del módulo Yfinance.

No está afiliada, respaldada ni examinada por Yahoo, Inc. Ofrece una forma sencilla con subprocesos para descargar datos de mercado de finanzas de Yahoo! utilizando Python. Tiene una API es de código abierto (2023a) y de uso gratuito, es fácil de usar, devuelve datos como para ser convertidos en data frames con una granularidad de datos en minutos.

3.4. Módulo Prophet Facebook

Prophet es un software de código abierto lanzado por el equipo Core Data Science de Meta (antes Facebook). Prophet (2023) es una biblioteca Python¹/R para pronosticar datos de series temporales basado en un modelo aditivo en el que las tendencias no lineales se ajustan a la estacionalidad anual, semanal y diaria, además de los efectos de los días festivos. Funciona mejor con series temporales que tienen fuertes efectos estacionales y varias temporadas de datos históricos. Prophet es resistente a la falta de datos y los cambios en la tendencia y por lo general maneja bien los valores atípicos. Prophet se usa en muchas aplicaciones en Meta para producir pronósticos fiables para la planificación y el establecimiento de objetivos. Está basado en Stan (2023) que es una plataforma de última generación para el modelado y el cálculo estadísticos de alto rendimiento. Prophet permite obtener pronósticos razonable sobre datos desordenados sin esfuerzo manual de preprocesado. Además, es resistente a valores atípicos, falta de datos y cambios drásticos en su serie temporal. El procedimiento Prophet incluye muchas posibilidades para que los usuarios modifiquen y ajusten las predicciones. Se pueden usar parámetros interpretables por personas para mejorar su pronóstico agregando su conocimiento del dominio, compartiendo en este caso código con Stan. Para crear un gran número de pronósticos se requerirá de medios eficientes y automatizados para evaluar y comparar entre ellos, así como detectar cuándo es probable que tengan un desempeño deficiente.



¹<https://github.com/facebook/prophet/tree/main/python> y <https://github.com/facebook/prophet>

Figura 2: Vista esquemática del enfoque del analista para la previsión a escala.

El proceso comienza modelando la serie temporal utilizando una especificación flexible que tiene una interpretación humana directa para cada uno de los parámetros. Luego se producen pronósticos para este modelo y un conjunto de líneas de base razonables a través de una variedad de simulaciones históricas que pronostiquen las fechas y evalúen el desempeño del pronóstico. Cuando hay un bajo rendimiento u otros aspectos de los pronósticos se justifica la intervención humana y estos problemas potenciales son señalados para un analista humano en un orden de prioridad. El analista puede entonces inspeccionar el pronóstico y potencialmente ajustar el modelo basado en esta retroalimentación manualmente. Este es un modelo de pronóstico de series de tiempo diseñado para manejar las características comunes de series de tiempo de negocios. Es importante destacar que también está diseñado para tener parámetros ajustables sin conocer los detalles del modelo subyacente. Esto es necesario para que el analista ajuste efectivamente el modelo como se describe en la [Figura 2](#).

Para la implementación de Prophet se utiliza un modelo de serie temporal modular ([Harvey, Peters; 1990](#)) con tres componentes del modelo: tendencia, estacionalidad y días festivos. Se combinan en la siguiente ecuación:

$$y(t)=g(t)+s(t)+h(t)+ \epsilon_t$$

donde $g(t)$ es la función de tendencia que modela cambios no periódicos en el valor de la serie temporal, $s(t)$ representa cambios periódicos (por ejemplo, estacionalidad semanal y anual) y $h(t)$ representa los efectos de los días festivos que ocurren en horarios potencialmente irregulares durante uno o más días. El término de error ϵ_t representa cualquier cambio que no sea acomodado por el modelo asumiendo que es normalmente distribuido (término de ruido). Este planteamiento es similar a un modelo aditivo generalizado (GAM) ([Hastie, Tibshirani; 1987](#)). Estos modelos de regresión incorporan varios términos de regresión así como términos no lineales. Permiten además modelar la estacionalidad mediante un componente aditivo como con el suavizado exponencial ([Gardner; 1985](#)). La estacionalidad multiplicativa, donde el efecto estacional es un factor que multiplica $g(t)$, se puede lograr a través de una transformación logarítmica. La formulación GAM tiene la ventaja de que permite acomodar nuevos componentes según sea necesario, por ejemplo, cuando se identifica una nueva fuente de estacionalidad. Los GAM también se ajustan muy rápidamente, ya sea usando un algoritmo de backfitting ([Byrd et al; 1995](#)).

A la hora de predecir se puede considerar el problema como el de un *ajuste de curvas de manera automática*, más allá de considerar la estructura propia del proceso que conduce a la aplicación de la estructura temporal del fenómeno o a diferencia de los modelos como ARIMA, que buscan explicar los valores a partir de valores pasados propios para pronosticar valores futuros. Las formulaciones mediante ajustes de curvas proporcionan una serie de ventajas prácticas:

- Son modelos flexibles, puesto que podemos acomodar fácilmente la estacionalidad a múltiples períodos y dejar que el analista haga diferentes suposiciones acerca de las tendencias.
- A diferencia de los modelos ARIMA, las medidas no necesitan estar espaciadas regularmente y tampoco se necesita interpolar los valores faltantes ni eliminar los valores atípicos.
- El ajuste es muy rápido, lo que permite al analista explorar de forma interactiva muchas especificaciones del modelo, como podría ser mediante aplicaciones basadas en Shiny ([Chang et al. 2015](#)).
- El modelo de pronóstico tiene parámetros fácilmente interpretables que pueden ser cambiados por el analista para imponer suposiciones sobre la predicción. Además, los analistas suelen tener experiencia con la regresión y pueden ampliar fácilmente el modelo para incluir nuevas componentes.

La predicción automática tiene una larga historia, con muchos métodos adaptados a tipos específicos de series temporales ([Tashman, Leach; 1991](#)) y ([De Gooijer, Hyndman; 2006](#)). Prophet fue impulsado tanto por la naturaleza de las series temporales que se predicen en Facebook (tendencias por partes, estacionalidad múltiple, festivos a discreción del empleado), como por los desafíos que implica la previsión en escala.

En resumen, Prophet utiliza un modelo de regresión modular que a menudo funciona bien con parámetros por defecto y que permite a los analistas seleccionar los componentes que son relevantes para su problema de previsión y hacer fácilmente los ajustes necesarios. Además, permite medir y rastrear la precisión de la predicción y pronósticos que deberían verificarse manualmente para ayudar a los analistas a realizar mejoras incrementales. Este es un componente crítico que permite a los analistas identificar cuándo es necesario realizar ajustes en el modelo o cuando debería de ser necesario un modelo completamente diferente. Modelos simples y ajustables combinados con la monitorización de rendimiento a diferentes escalas permiten que una gran cantidad de analistas pronosticar un gran número y una variedad de series de tiempo.

La implementación básica del modelo se puede consultar en el [Anexo 1](#) . Para ello, es necesario instalar en primer lugar el módulo yfinance para la obtención de datos financieros de Yahoo. A continuación, necesitaremos importar las librerías necesarias para la ejecución del código, entre ellas yfinance y Prophet y establecer las gráficas a la hora de presentar los resultados.

4. Análisis financiero

En este capítulo analizaremos algunos de los resultados obtenidos en la predicción de comportamiento de las criptomonedas.

4.1 Análisis financiero de las gráficas obtenidas

En la [Figura 3](#), representamos las predicciones a partir de febrero de 2023 de la criptomoneda DOGE. Dogecoin código DOGE, símbolo Đ es una criptomoneda creada por los ingenieros de software Billy Markus y Jackson Palmer, quienes decidieron crear un sistema de pago como una "broma", burlándose de la salvaje especulación en criptomonedas en ese momento. Los valores reales se muestran con puntos negros, las tendencias de predicción se muestran en color azul y los intervalos de confianza en azul claro. Los resultados coinciden con la tendencia real mostrada, siendo la cotización de esta última de 0.063 dólares el día 1 de julio de 2023, no habiendo recomendado su compra, al haber examinado la tendencia de bajada.

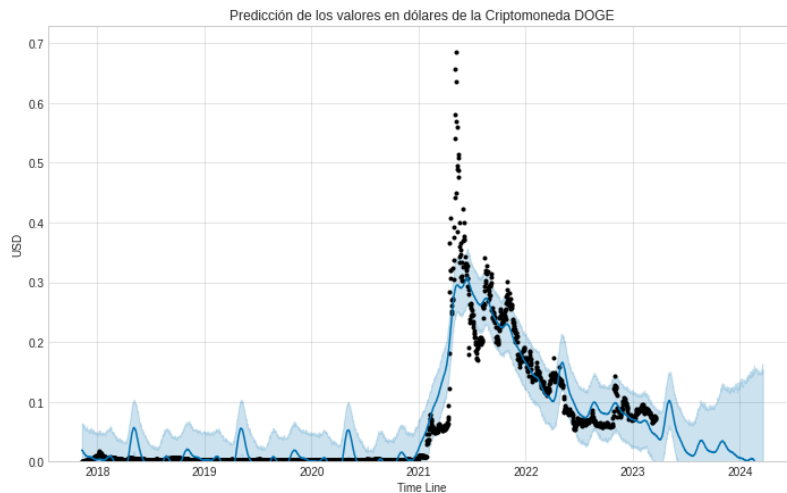


Figura 3: Gráfica de predicción de valor de la criptomoneda DOGE.

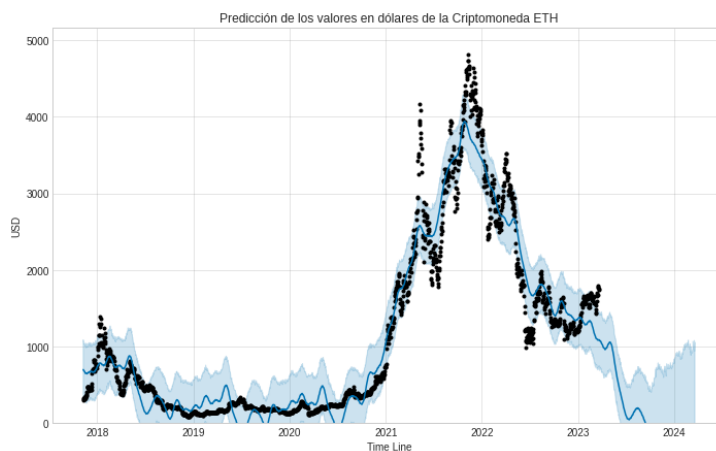


Figura 4: Gráfica de predicción de valor de la criptomoneda ETH.

En la [Figura 4](#), mostramos las predicciones de la criptomoneda Ether, nativa de la plataforma Ethereum, una cadena de bloques descentralizada de código abierto con funcionalidad de contrato inteligente. Entre las criptomonedas, Ether ocupa el segundo lugar después de Bitcoin en capitalización de mercado. Las predicciones auguraban la desaparición de Ether dadas las fuertes bajadas de la criptomoneda a finales de 2022. No se recomendaba su inversión, a pesar de que su cotización se recuperó y está en 1763 dólares el 1 de julio de 2023.

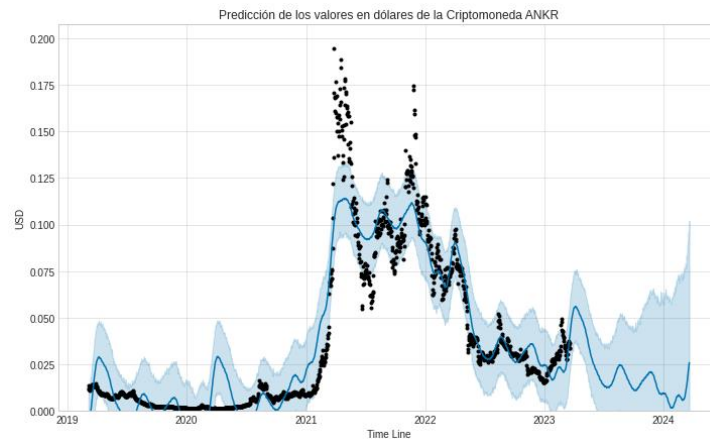


Figura 5: Gráfica de predicción de valor de la criptomoneda ANKR.

Ankr (ANKR) es una criptomoneda basada en Ethereum lanzada por Ankr, una empresa proveedora de infraestructura web3 en 2019. Su objetivo es permitir que los proveedores de recursos y los usuarios finales se conecten fácilmente a las tecnologías blockchain y las aplicaciones DeFi. En la [Figura 5](#), observamos la volatilidad de precios de esta moneda, aunque la tendencia es a la baja, puede ser utilizada en inversiones a corto plazo, comprando cuando el valor se encuentre en el pico de bajada y vendiendo antes de que el pico de subida indique descenso. No obstante, se muestra un comportamiento relativamente estable. Su cotización fue de 0.023 dólares el 1 de julio de 2023, siendo próxima a las predicciones realizadas.

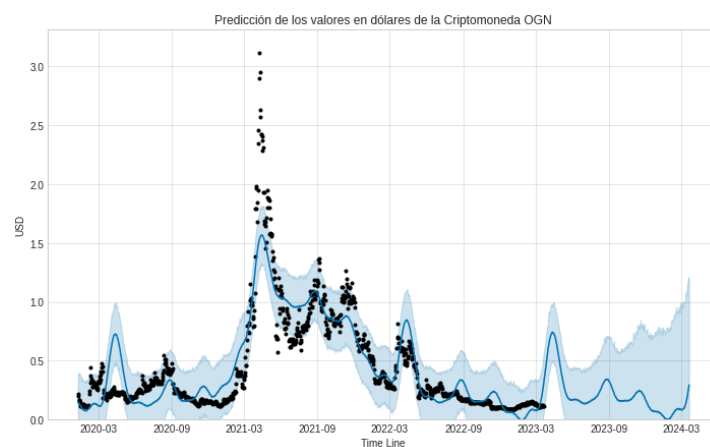
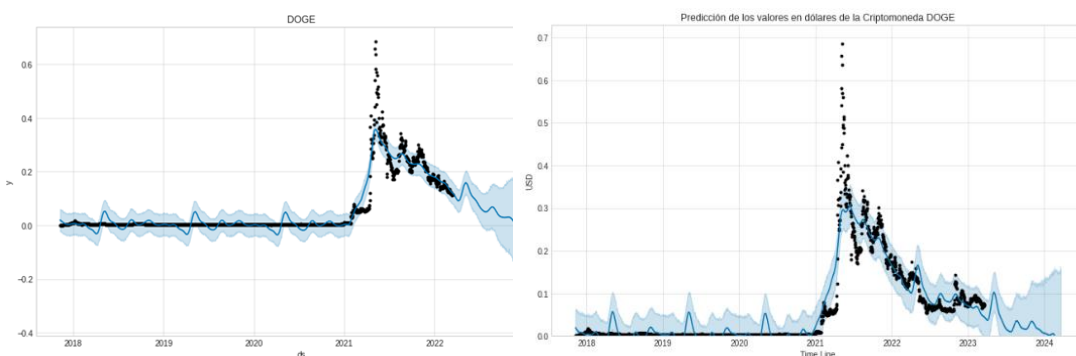


Figura 6: Gráfica de predicción de valor de la criptomoneda OGN.

Origin Protocol (OGN) es un token de Ethereum que impulsa la plataforma Origin, cuyo objetivo es impulsar los mercados descentralizados y de igual a igual. OGN se puede usar para replantear, gobernar y publicitar en la plataforma Origin. En la [Figura 6](#), analizamos las predicciones de las cotizaciones de esta criptomoneda detectamos picos de subida con tendencia a la baja, podría emplearse en una compra a corto plazo con inmediata venta en el pico de subida, pasado este periodo los picos de subida no obtienen mucho beneficio en venta a corto plazo ni a largo plazo. La cotización fue de 0.0734 dolares el 1 de julio de 2023, continuando la bajada mostrada desde inicio de año.

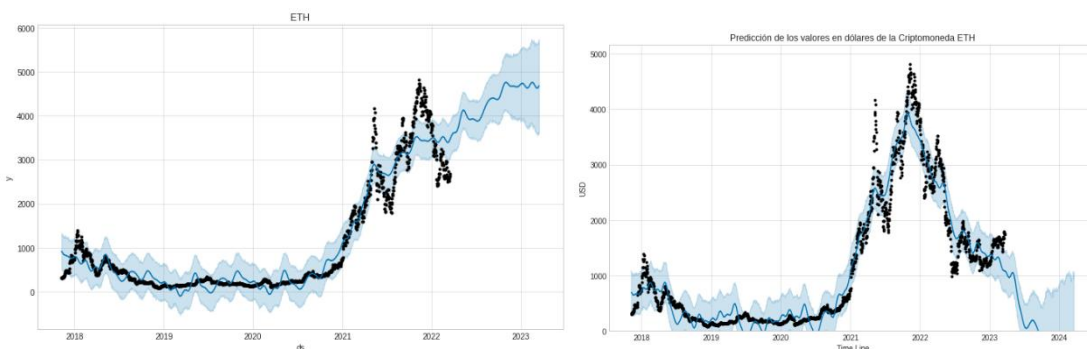
4.2 Análisis de cambios debido a variables externas

Pasamos a analizar los cambios en las predicciones, viendo las predicciones .



Figuras 7: Comparación de los cambios en la predicción de valor de la criptomoneda DOGE, a partir de datos hasta febrero de 2022 (izquierda) y febrero de 2023 (derecha).

En la [Figura 7](#), se observa que DOGE no ha sufrido cambios muy acentuados en su predicción ya que en este caso la gráfica nos muestra que la tendencia de esta criptomoneda era a la baja.



Figuras 8: Comparación de los cambios en la predicción de valor de la criptomoneda ETH, a partir de datos hasta febrero de 2022 (izquierda) y febrero de 2023 (derecha).

En la [Figura 8](#), observamos que ETH ha sufrido un cambio especialmente notable en su predicción, este gráfico nos muestra una tendencia de subida en el tiempo, pudiendo haber sido utilizada como inversión a largo plazo, pero la crisis de mayo de 2022 hizo que empezara a caer.

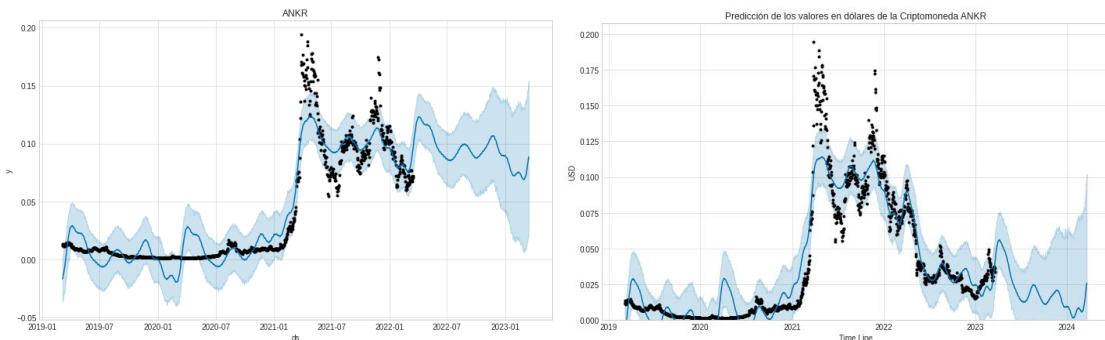


Figura 9: Comparación de los cambios en la predicción de valor de la criptomoneda ANKR, a partir de datos hasta febrero de 2022 (izquierda) y febrero de 2023 (derecha).

En la [Figura 9](#), observamos que las tendencias de predicción de ANKR de 2022 mostraban tendencias de subida y bajada variables en el tiempo, no obstante a partir de la crisis de primavera de 2022 las predicción se volvieron bajas aunque conduciendo a una cierta estabilidad.

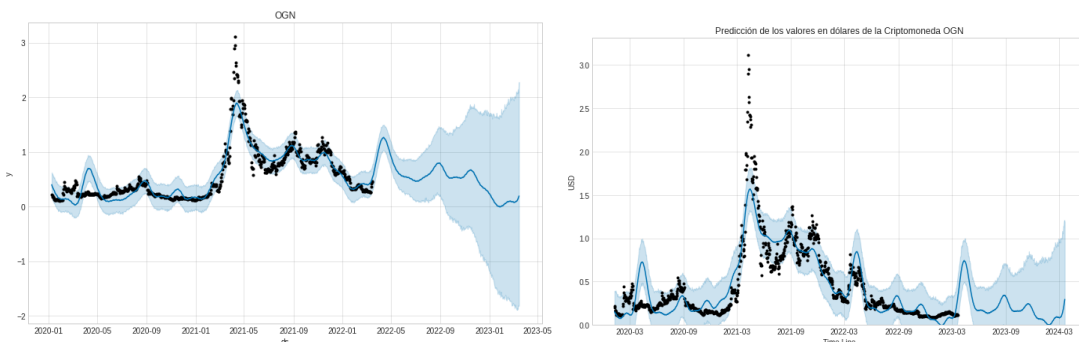


Figura 10: Comparación de los cambios en la predicción de valor de la criptomoneda OGN, a partir de datos hasta febrero de 2022 (izquierda) y febrero de 2023 (derecha).

En la [Figura 10](#), los gráficos nos indican que la criptomoneda OGN mostraba una ligera tendencia de bajada, con la predicción actual más acentuada en esta tendencia negativa, debería de ser utilizada en compra y venta a corto plazo al subir el valor de la misma.

4.3. Predicciones de cotizaciones de acciones en mercados bursátiles

Del mismo modo que utilizamos la predicción de valor de criptomonedas en base a sus precios históricos, se pueden predecir precios de compra para las acciones bursátiles del mercado de valores. Con estas gráficas los inversores pueden decidir comprar al alza y vender a corto o largo plazo según las tendencias de predicción. Mostramos algunos de estos ejemplos, junto con las

predicciones obtenidas en las [Figuras 11 y 12](#). Se destaca la parte más relevante del código en el Anexo 2.

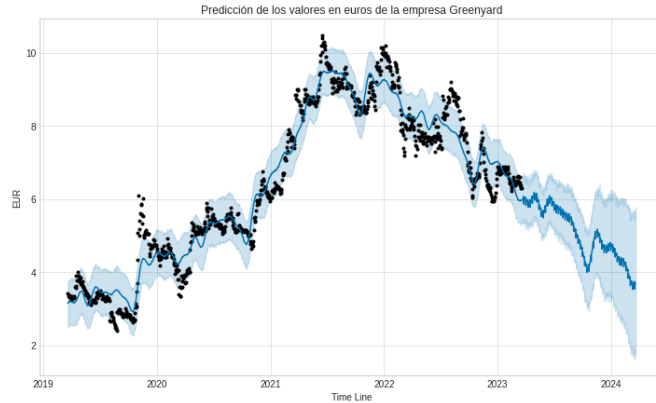


Figura 11: Gráfica de predicción de valor de acciones de la compañía Greenyard.

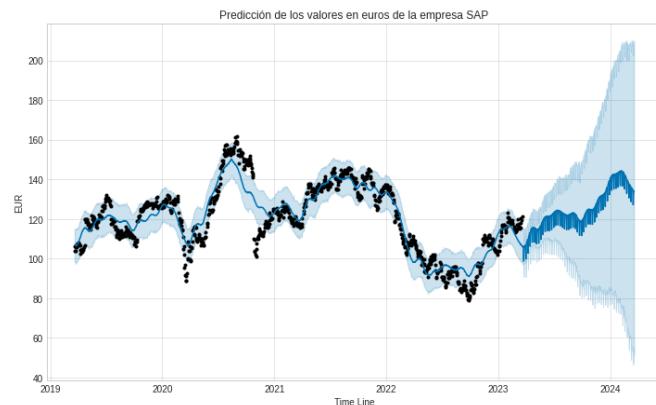


Figura 12: Gráfica de predicción de valor de acciones de la compañía SAP.

Antes de la pandemia, los criptoactivos como Bitcoin y Ether mostraban poca correlación con los principales índices bursátiles. Se pensaba que ayudaban a diversificar el riesgo y actuaban como cobertura frente a las oscilaciones de otras clases de activos.

Esto cambió después de las extraordinarias respuestas a la crisis del Banco Central Europeo y de la Reserva federal de USA a principios de 2020. Los precios de las criptomonedas y las acciones de Estados Unidos aumentaron por las fáciles condiciones financieras globales y un mayor interés por parte de los inversores. Con el paso del tiempo, los mercados financieros tradicionales y los criptomercados muestran una creciente interconexión entre las dos clases de activos que permite la transmisión de “Economic Shocks”, caídas del mercado de valores o rápida devaluación de una moneda que pueden desestabilizar los mercados financieros, como el sucedido en mayo de 2022.

La invasión rusa de Ucrania está ejerciendo presión sobre las acciones mundiales, elevando los precios del petróleo, avivando la inflación y desbaratando los planes de la Reserva Federal de USA para aumentar las tasas de interés. A medida que continúa la guerra, la volatilidad en los mercados financieros mundiales, incluidas las criptomonedas, experimentan grandes cambios. En el pasado los expertos financieros han defendido las criptomonedas como un activo no correlacionado con los mercados financieros tradicionales, habiendo observado recientemente que no es así, y que los efectos en unos se muestran en los otros.

4.4. Valoración de estrategias de inversión

A la hora de valora diferentes estrategias de inversión, podemos combinar las predicciones realizadas con estrategias como la de establecer patrones de una sola vela ([Orquín-Serrano, 2020](#)). A la hora de validarlo, estableceremos la premisa que si la estrategia no tiene poder predictor, entonces el promedio de las ganancias obtenidas será 0 (Hipótesis nula) y mayor que 0 en otro caso. Por tanto, el estadístico a validar es el retorno medio de las inversiones. Así como por el Teorema Central del Límite se espera una distribución normal de dichos promedios, la varianza de esta distribución depende, entre otras cosas, del número de distribuciones muestrales aleatorias que contribuyen con a dichos promedios. No obstante, pueden emplearse otras métricas de la estrategia, como el retorno absoluto de la estrategia o el cociente de retorno medio y desviación de retornos que ofrece una idea sobre el riesgo asumido para conseguir determinado nivel de beneficios. En la [Figura 13](#), mostramos las cotizaciones de Bitcoin. A partir de aquí establecemos predicciones con Prophet.



Figura 13. Cotizaciones de bitcoin desde octubre de 2010 a junio de 2023. El valor de referencia al inicio se referencia a 100 unidades.

En la [Figura 14](#), mostramos el resultado de una estrategia de inversión a corto plazo (diario) basada en operaciones realizadas tomando como referencia las predicciones obtenidas mediante Prophet para el día siguiente. Se puede observar cómo esta estrategia hubiera permitido mantener

el valor de la inversión inicial hasta julio de 2023, a pesar de las bajadas sufridas en la cotización. Además, se observa que los retornos por las predicciones siempre se mantienen por encima de las cotizaciones reales.

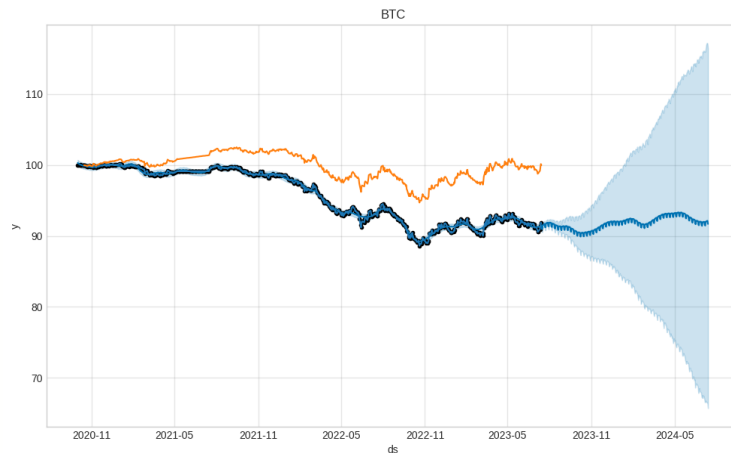


Figura 14: Cotizaciones de bitcoin desde octubre de 2010 a junio de 2023 (azul y negro), junto con el resultado de una estrategia de inversión predicciones a diario calculadas con Prophet (naranja) y los intervalos de confianza de las predicciones a partir de julio de 2023 (azul claro), junto con la región de incertidumbre (azul claro).

Por último, a partir de las predicciones diarias, se puede establecer una estrategia de inversión a corto plazo (valor de cotización al final del día siguiente). La validez de esta o de cualquier otra estrategia puede ser estimada a partir del cálculo de Kolmogorov-Smirnov, viendo si dicha estrategia tiene un efecto positivo o no. En la [Figura 15](#), el cálculo de este estadístico muestra que sí que hay diferencias significativas. Se puede ver el código en el [ANEXO 3](#).

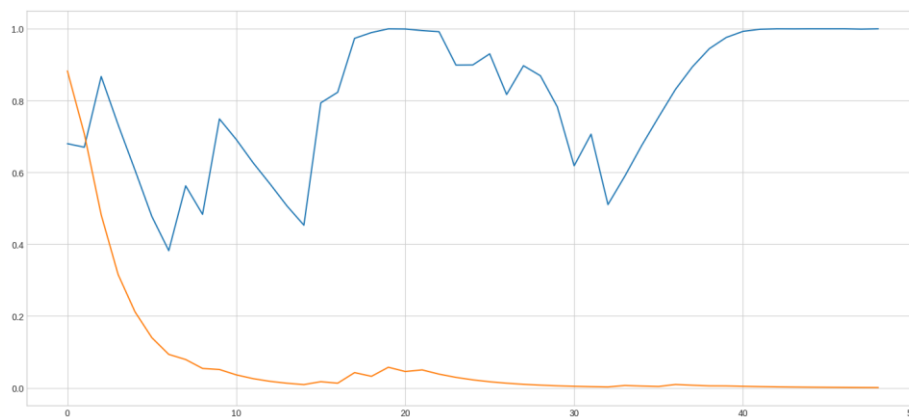


Figura 15. Cálculo del estadístico de Kolmogorov Smirnov para las series temporales real (azul) y de los beneficios esperados (naranja).

5. Exponente de Hurst

Al analizar series financieras, se tiene de alguna manera presente el comportamiento histórico. Es interesante considerar si la serie temporal sigue un proceso aleatorio tipo random walk, si se dará un “mean-reverting” (término financiero para la suposición de que el precio de un activo tenderá a converger al precio promedio a lo largo del tiempo) o si habrá una cierta tendencia (al alza o a la baja). En el mundo de las finanzas, muchos “traders” elaboran estrategias basadas en el momento y la reversión de la media.

El exponente de Hurst ([Hurst, 1951](#)) es una medida de memoria a largo plazo (memoria inactiva) de una serie temporal donde medimos la cantidad por la cual una serie dada se desvía de un camino aleatorio. El valor escalar muestra la tendencia relativa de una serie temporal a agruparse en una dirección particular (patrón de tendencia tanto en forma positiva como negativa) o retroceder fuertemente a la media (patrón de reversión a la media) y resulta de utilidad en el análisis de señales financieras, como se puede ver en ([Carbone et al, 2004](#)).

Los distintos valores del exponente de Hurst oscilan siempre entre 0 y 1.

- $H < 0.5$ – Reversión a la media. Un valor más cercano a 0 significa que hay un patrón fuerte del proceso de reversión a la media. En situaciones prácticas, significa que un valor alto será seguido por un valor bajo y viceversa.
- $H = 0.5$ – Camino aleatorio. Esto significa que puede ir en cualquier dirección y no hay una deducción clara posible de los parámetros dados.
- $H > 0.5$ – Serie de tendencia. Un valor más cercano a 1 significa que el patrón de tendencia es fuerte y la tendencia continuará. Generalmente, significa que un valor alto será seguido por un valor más alto.

En la [Figura 15](#), vemos algunos ejemplos de series temporales de los casos de reversión a la media (mean reverting), camino aleatorio (random walk) y serie de tendencia (trending)

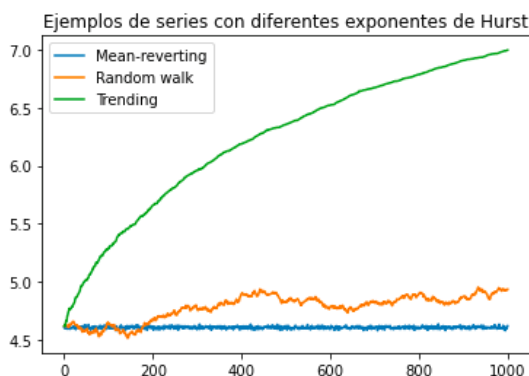


Figura 16: Ejemplos de series temporales generada con diferentes exponentes de Hurst

En la [Tabla 1](#), mostramos los exponentes de Hurst calculados para las series del ejemplo de la [Figura 16](#), con diferentes intervalos de tiempo. Aunque con diferencias, los resultados de los exponentes coinciden con el exponente de Hurst calculado.

	Lag 5	Lag 10	Lag 20	Lag 100
Ej. Mean reverting	-0.0104	0.0014	0.0030	-0.0024
Ej. Random Walk	0.4979	0.4828	0.4546	0.4549
Ej. Trending	0.7255	0.7643	0.7803	0.8169

Tabla 1. Cálculo de exponentes de Hurst para diferentes series temporales.

Como aplicación, vamos a calcular el exponente de Hurst de la serie de cotizaciones de Bitcoin desde Octubre de 2010, véase la [Figura 17](#). Para ellos es necesario calcular la desviación estándar de las diferencias entre la serie y sus versiones rezagadas. Para establecer posible rango, estimaremos el exponente de Hurst como una pendiente de la gráfica logarítmica del número de retrasos de las variables versus las desviaciones estándar mencionadas.

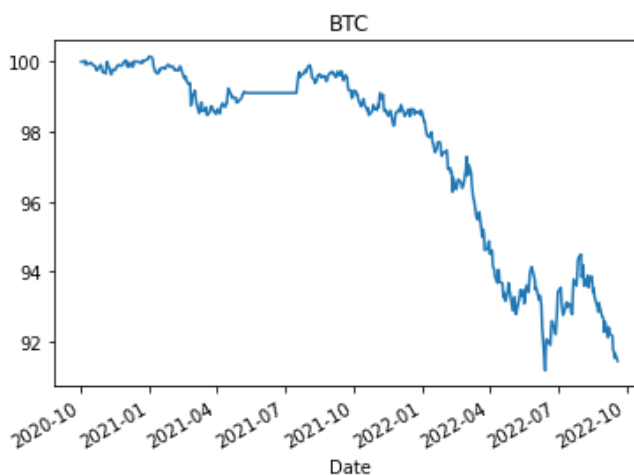


Figura 17: Gráfica de historial de valor de la criptomoneda Bitcoin.

Los resultados que obtenemos para los diferentes intervalos son 0.4368 (lag=5), 0.4585 (lag=10) y 0.51608 (lag=20). Se observa que a medida que aumenta el espaciado, el comportamiento se aproxima a una camino aleatorio, aunque hay una cierta tendencia de reversión a la media. La reversión a la media asume que propiedades como el rendimiento de las acciones y la volatilidad volverán a su promedio a largo plazo con el tiempo. Matemáticamente, tal serie temporal se denomina proceso de Ornstein-Uhlenbeck. En tales estrategias, los inversores intentan ganar dinero asumiendo que

después de algunos eventos extremos (ya sean positivos o negativos), el precio de las acciones volverá al patrón a largo plazo.

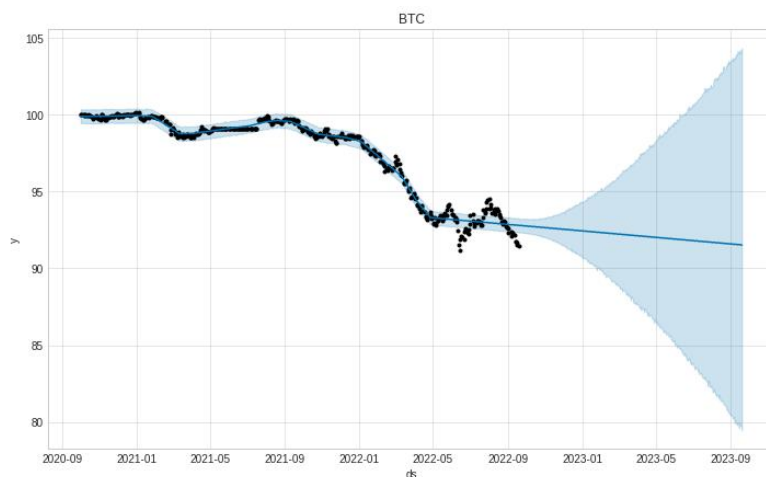


Figura 18: Gráfica de cambios de predicción de valor de la criptomoneda BTC.

Observando los resultados de predicción de Prophet para Bitcoin de la [Figura 18](#), se observan que las predicciones se mantienen estables a la baja, por lo que aplicar considerar que el comportamiento es mean reverting no es mala estrategia, asumiendo que puede haber en algún momento cambios bruscos de precio en los valores de bitcoin.

6. Conclusiones

En este último capítulo definiremos las conclusiones extraídas tras el desarrollo de cada uno de los apartados de este proyecto, derivadas de las investigaciones y análisis realizados en los mismos.

Hemos propuesto un modelo para la generación de predicciones de valor de criptomonedas con ayuda de las librerías `yfinance` de Yahoo y `Prophet` de Facebook. Para el desarrollo de este programa se ha creado un `DataFrame` con los datos históricos de las criptomonedas seleccionadas extraídos del mercado de finanzas de Yahoo. La librería `Prophet` se ha utilizado para pronosticar las cotizaciones de las criptomonedas mediante un modelo aditivo en el que las tendencias no lineales se ajustan a la estacionalidad. Con la implementación de este programa se ha demostrado la viabilidad del proyecto utilizando diferentes tecnologías para el diseño de esta app de negocios.

Esta herramienta puede ser utilizada tanto por profesionales de inversión en mercados financieros como por usuarios con menos experiencia en activos bursátiles ya que las conclusiones gráficas que se extraen son muy intuitivas. El analista financiero usuario de este programa podrá personalizar de manera intuitiva los parámetros de suavizado para la estacionalidad y así ajustar las predicciones a los ciclos históricos, realizando las predicciones durante varias épocas del año y comparando varios tipos de criptomonedas, puede llegarse a mejores conclusiones para la inversión en las mismas.

Finalmente, la funcionalidad principal de este programa es la de predecir de manera fiable las fluctuaciones de precios que afectan a las criptomonedas, teniendo siempre en cuenta mediante análisis financiero que los factores externos, que afectan también a otro tipo de valores bursátiles y activos, pueden afectar de manera cuantitativa a la desviación del valor en comparación a la predicción realizada.

Referencias y bibliografía

1. J. Abraham et al. "Cryptocurrency price prediction using tweet volumes and sentiment analysis." *SMU Data Science Review* 1.3 (2018).
2. J. Barreiro-Gomez and H. Tembine. Blockchain token economics: A mean-field-type game perspective. *IEEE Access* 7, 64603-64613 (2019). <https://doi.org/10.1109/ACCESS.2019.2917517>
3. T. Bollerslev. Generalized autoregressive conditional heteroskedasticity. *J. Econometrics* 31(3), 307-327 (1986). [https://doi.org/10.1016/0304-4076\(86\)90063-1](https://doi.org/10.1016/0304-4076(86)90063-1)
4. L. Breiman, Random forests. *Machine Learning* 45, 5–32 (2001). <https://doi.org/10.1023/A:1010933404324>
5. R.H. Byrd et al. A limited memory algorithm for bound constrained optimization." *SIAM J. Sci. Comput.* 16(5) 1190-1208 (1995).
6. A. Carbone, G. Castelli, and H.E. Stanley. Time-dependent Hurst exponent in financial time series. *Physica A* 344(1-2) 267-271 (2004). <https://doi.org/10.1016/j.physa.2004.06.130>
7. C. Catalini and J.S. Gans. Initial coin offerings and the value of crypto tokens. No. w24418. National Bureau of Economic Research (2018). https://www.nber.org/system/files/working_papers/w24418/revisions/w24418.rev0.pdf
8. L. Catania, S. Grassi, and F. Ravazzolo. Forecasting cryptocurrencies financial time series. (2018). Norwegian Business School. Working paper. <http://hdl.handle.net/11250/2489408>
9. L. Catania, S. Grassi, and F. Ravazzolo. Predicting the Volatility of Cryptocurrency Time-Series. In: Corazza, M., Durbán, M., Grané, A., Perna, C., Sibillo, M. (eds) *Mathematical and Statistical Methods for Actuarial Sciences and Finance*, 203-207. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-89824-7_37
10. W. Chang et al. Package ‘shiny’. See <http://citeseerx.ist.psu.edu/viewdoc/download> (2015).
11. S.E. Charandabi and K. Kamyar. Prediction of cryptocurrency price index using artificial neural networks: A survey of the literature. *Eur. Res. Manag. Bus. Econ.* 6(6), 17–20 (2021). <https://doi.org/10.24018/ejbmr.2021.6.6.1138>
12. K. Cho et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014). <https://arxiv.org/abs/1406.1078>
13. J.G. De Gooijer and R.J. Hyndman. 25 years of time series forecasting. *Int. J. Forecast.* 22(3) 443-473, (2006). <https://doi.org/10.1016/j.ijforecast.2006.01.001>

14. V. Derbentsev et al. Forecasting cryptocurrency prices time series using machine learning approach. SHS Web of Conferences. Vol. 65. EDP Sciences (2019).
<https://doi.org/10.1051/shsconf/20196502001>
15. Google Colaboratory: https://colab.research.google.com/?utm_source=scs-index
16. A.C. Harvey and S. Peters. Estimation procedures for structural time series models. J. Forecasting 9(2), 89-108 (1990). <https://doi.org/10.1002/for.3980090203>
17. S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural Comput. 9(8) 1735-1780 (1997). <http://dx.doi.org/10.1162/neco.1997.9.8.1735>
18. E.S. Gardner Jr. Exponential smoothing: The state of the art. J. Forecasting 4.1 (1985): 1-28 (1985). <https://doi.org/10.1002/for.3980040103>
19. Ò. Garibo-i-Ort et al. Efficient recurrent neural network methods for anomalously diffusing single particle short and noisy trajectories." J. Phys. A: Math. Theor. 54(50) 504002, (2021).
20. T. Hastie and R. Tibshirani. Generalized additive models: some applications. J. Am. Stat. Assoc. 82(398), 371-386 (1987).
21. X. Huang et al. LSTM based sentiment analysis for cryptocurrency prediction. Database Systems for Advanced Applications: 26th International Conference, DASFAA 2021, Taipei, Taiwan, April 11–14, 2021. Lecture Notes in Computer Science Proceedings, Part III 26 (2021). https://doi.org/10.1007/978-3-030-73200-4_47
22. H.E. Hurst. Long-term storage capacity of reservoirs. Trans. Am. Soc. Civ. Eng. 116.1 (1951): 770-799.
23. P. Jaquart, D. Dann, and C. Weinhardt. Short-term bitcoin market prediction via machine learning. J. Financ. Data Sci. 7, 45-66 (2021). <https://doi.org/10.1016/j.jfds.2021.03.001>
24. P. Jay, V. Kalariya, P. Parmar, S. Tanwar, N. Kumar and M. Alazab. Stochastic neural networks for cryptocurrency price prediction. IEEE Access, 8, 82804-82818 (2020).
<https://doi.org/10.1109/ACCESS.2020.2990659>
25. Jupyter <https://jupyter.org/> Acceso 1 de Julio de 2023.
26. J. Lansky. Possible state approaches to cryptocurrencies. J.Syst. Integr. 9.1 (2018): 19.
27. D.K.C. Lee, L. Guo, and Y. Wang. Cryptocurrency: A new investment opportunity?. J. Altern. Invest. 20.3 (2017): 16-40. <http://dx.doi.org/10.2139/ssrn.2994097>
28. M.A. Lozano et al. Open data science to fight COVID-19: winning the 500k XPRIZE pandemic response challenge. Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Cham: Springer International Publishing, 384-399 (2021). https://doi.org/10.1007/978-3-030-86514-6_24
29. R.K. Malladi, P.L. Dheeriyaa. Time series analysis of cryptocurrency returns and volatilities. J. Econ. Finan. 45, 75–94 (2021). <https://doi.org/10.1007/s12197-020-09526-4>

30. A. Martelli. Python. Guía de referencia. tr: Gorjón Salvador, Bruno (1 edición). Anaya Multimedia-Anaya Interactiva. ISBN 978-84-415-2317-3. (2007)
31. Matplotlib <https://matplotlib.org/> Acceso 1 de Julio de 2023.
32. U. Mukhopadhyay, A. Skjellum, O. Hambolu, J. Oakley, L. Yu and R. Brooks. A brief survey of Cryptocurrency systems. 2016 14th Annual Conference on Privacy, Security and Trust (PST), Auckland, New Zealand, 745-752 (2016) <https://doi.org/10.1109/PST.2016.7906988>.
33. S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. <http://www.bitcoin.org/bitcoin.pdf> (2008).
34. Numpy <https://numpy.org/>
35. I. Orquín-Serrano, I. Predictive power of adaptive candlestick patterns in Forex Market. EurUSD Case. Mathematics 2020, 8(802) (2020).. <https://doi.org/10.3390/math8050802>
36. Pandas. <https://pandas.pydata.org/> Acceso 1 de Julio de 2023.
37. M.M. Patel et al. A deep learning-based cryptocurrency price prediction scheme for financial institutions. J. Infer. Secur. Appl. 55 (2020): 102583. <https://doi.org/10.1016/j.jisa.2020.102583>
38. T. Peters, Pep 20—the zen of python. Python enhancement proposal (2004). <https://peps.python.org/pep-0020/>
39. Prophet. Procedimiento de predicción automático: <https://pypi.org/project/fbprophet/> Acceso 1 de Julio de 2023.
40. Python: <https://www.python.org> Acceso 1 de Julio de 2023.
41. J. Rebane, I. Karlsson, P. Papapetrou, and S. Denic. Seq2Seq RNNs and ARIMA models for cryptocurrency prediction: A comparative study. Proceedings of SIGKDD Workshop on Fintech (SIGKDD Fintech'18). Presented at the SIGKDD Fintech'18, London, UK, August 19-23, 2018. <http://urn.kb.se/resolve?urn=urn:nbn:se:su:diva-161409>
42. Stan. <https://mc-stan.org/> Acceso 1 de Julio de 2023.
43. Y. Tang. Deep learning using linear support vector machines. arXiv preprint arXiv:1306.0239 (2013). <https://arxiv.org/abs/1306.0239>
44. L.J. Tashman and M.L. Leach. Automatic forecasting software: A survey and evaluation. Int. J. Forecast. 7(2), 209-230 (1991). [https://doi.org/10.1016/0169-2070\(91\)90055-Z](https://doi.org/10.1016/0169-2070(91)90055-Z)
45. D. Tapscott, Don and A. Tapscott. La revolución blockchain. Descubre cómo esta nueva tecnología transformará la economía global. Ediciones Deusto (2017).
46. G. Van Rossum. Python 0 (1991).
47. Yahoo! Finance's API: <https://pypi.org/project/yfinance/> Acceso 1 de Julio de 2023.
48. Yahoo! Finance's <https://finance.yahoo.com/> Acceso 1 de Julio de 2023.

49. W. Zhang and S. Skiena. Trading strategies to exploit blog and news sentiment. Proceedings of the international AAAI conference on Web and Social Media. 4(1) (2010). <https://doi.org/10.1609/icwsm.v4i1.14075>

ANEXO 1: Implementación en Prophet de la predicción de la cotización de criptomonedas

```
>>> from fbprophet import Profeta

>>> m = Profeta()

>>> m.fit(df) # df es un pandas.DataFrame con las columnas 'y' y 'ds'

>>> future = m.make_future_dataframe(períodos = 365)

>>> m.predecir(futuro)

pip install yfinance
import pandas as pd
import yfinance as yf
import numpy as np
import matplotlib.pyplot as plt
from prophet import Prophet
from prophet.plot import plot

plt.style.use('seaborn-whitegrid')
def make_forecast(ticker, days, historical='max'):
    # extrae datos de yahoo finanzas
    stock_data = yf.Ticker(ticker)

    historical_data = stock_data.history(historical, auto_adjust=True)

    # crea un nuevo dataframe para las fechas (ds) y los precios (y)
    df = pd.DataFrame()

    df['ds'] = historical_data.index.values
    df['y'] = historical_data['Close'].values
    df.tail()

    # crea un modelo Prophet con los datos
    m = Prophet(daily_seasonality=False)
    m.fit(df)

    future = m.make_future_dataframe(days, freq='D')

    forecast = m.predict(future)

    m.plot(forecast)

    return forecast

assets_ = []
assets = []
for i in range(10):
    assets_ = pd.read_html('https://en.wikipedia.org/wiki/List_of_cryptocurrencies')[
i]
    assets.extend((assets_.Symbol).tolist())
print(assets)
['BTC,[2] XBT, ₿', 'LTC, Ł', 'NMC', 'PPC', 'DOGE, XDG, Ð', 'GRC', 'XPM', 'XRP', 'NXT', 'AUR', 'DASH', 'NEO', 'MZC', 'XMR', 'TIT', 'XVG',
'XLM', 'VTC', 'ETH, ƒ', 'ETC', 'Nano', 'USDT', 'FIRO', 'ZEC', 'BCH[59]', 'EOS', 'ADA, ₺', 'TRX', nan, 'ALGO', 'AVAX', 'SHIB', 'DESO',
'SAFEMOON', 'ICP']

# coins = ['DOGE', 'ETH', 'ANKR', 'OGN', 'BTC', 'LTC', 'NMC', 'PPC', 'GRC', 'XPM', '
XRP', 'NXT', 'AUR', 'DASH', 'NEO', 'XMR', 'XVG', 'XLM', 'VTC', 'ETC', 'USDT', 'FIRO
', 'ZEC', 'BCH', 'EOS', 'ADA', 'TRX', 'ALGO', 'AVAX', 'DESO', 'SAFEMOON', 'ICP']
coins = ['DOGE', 'ETH', 'ANKR', 'OGN']
for coin in coins:
    cryptocoin = coin+"-USD"
    make_forecast(cryptocoin, 365)
    plt.title('Predicción de los valores en dólares de la Criptomoneda '+coin)
    plt.ylabel('USD')
    plt.xlabel('Time Line')
    plt.ylim(ymin=0)
```

ANEXO 2: Modificaciones para la predicción de la cotización de cotizaciones

bursátiles

```
assets_ = []
assets = []
for i in range(10):
    assets_ = pd.read_html('https://en.wikipedia.org/wiki/List_of_S%26P_500_companies')
    assets.extend((assets_.Symbol).tolist())
print(assets)
```

```
['MMM', 'AOS', 'ABT', 'ABBV', 'ABMD', 'ACN', 'ATVI', 'ADM', 'ADBE', 'ADP', 'AAP', 'AES', 'AFL', 'A', 'AIG', 'APD', 'AKAM', 'ALK', 'ALB', 'ARE',
```

```
plt.title('Predicción de los valores en euros de la empresa Greenyard')
plt.ylabel('EUR')
plt.xlabel('Time Line')
```

```
make_forecast('SAP', 365, '4y')
plt.title('Predicción de los valores en euros de la empresa SAP')
plt.ylabel('EUR')
plt.xlabel('Time Line')
```

ANEXO 3. Valoración de estrategias de inversión

```
#Descarga del dataset.
assets_ = pd.read_html('https://en.wikipedia.org/wiki/List_of_S%26P_500_companies')
[0]
# assets = (assets_.Symbol).tolist()
# indice = np.random.randint(0, len(assets))
# instrument = assets[indice]
instrument = 'BTC'
data = yf.download(instrument, start='2008-01-01', as_panel=False)
# empresa = assets_.loc[indice, 'Security']
# print(f'Hemos descargado los datos históricos de la empresa {empresa}.')
print(f'Hemos descargado los datos históricos de la Criptomoneda {instrument}.')
Hemos descargado los datos históricos de la Criptomoneda BTC.

# Seguimiento gráfico de los datos históricos de la Criptomoneda BTC.
plt.plot(data['Adj Close'])
plt.title(instrument)

# *****
# Cálculo de los parámetros definitorios de la estrategia de trading modificando
los periodos
periods = [1, 5, 10]
# *****
periodos = [f'EMA{p}' for p in range(len(periods))]
for instr in [instrument]:
    for p in range(len(periods)):
        data[f'EMA_{instr}_{periods[p]}'] = ta.trend.ema_indicator(data['Adj Close'],
periods[p], fillna = False)
data.dropna(inplace=True)
base = len(periods)
for instr in [instrument]:
    all_EMAs = data[[f'EMA_{instr}_EMA{e}' for e in range(len(periods))]]
    for e in range(len(periods)):
        all_EMAs[f'{instr}_EMA{e}'] = None
    for i in tqdm.tqdm(range(len(all_EMAs))):
        for e in range(len(periods)):
            all_EMAs.at[all_EMAs.index[i], f'{instr}_EMA{e}'] = sorted(range(len(per
iods)), key=lambda k: all_EMAs.iloc[i][k])[e]
    data = pd.concat([data, all_EMAs], axis = 1)
    for e in range(len(periods)):
        if e==0:
            data[f'Fase_num_base_3_{instr}'] = data[f'{instr}_EMA0']
        else:
```

```

        data[f'Fase_num_base_3_{instr}'] = data[f'Fase_num_base_3_{instr}'] + (
data[f'{instr}_EMA(e)']*(base**e)
data[f'Fase_num_base_3_{instr}'] = data[f'Fase_num_base_3_{instr}'].astype('float')
data[f'Fase_num_base_3_{instr}'] = np.round(data[f'Fase_num_base_3_{instr}']/np.max
(data[f'Fase_num_base_3_{instr}'].unique()),6)
data['Ordinal'] = np.arange(1,data.shape[0]+1)
valor_inicial=data['Adj Close'][0]
data['Dif_Adj_Close'] = np.round(data['Adj Close'].diff(1),15)
data.dropna(inplace=True)
retorno_medio_diario_base = np.round(np.mean(data.Dif_Adj_Close),15)
data['Dif_Adj_Close_Norm'] = np.round(data['Dif_Adj_Close'] - retorno_medio_diario_
base,15)

plt.plot(data.Dif_Adj_Close.cumsum()+valor_inicial)
plt.plot(data.Dif_Adj_Close_Norm.cumsum()+valor_inicial)

#Cálculo del backtest de la estrategia
posibles_fases = data[f'Fase_num_base_3_{instr}'].unique()

direccion = 'long'
# *****
fase_clave_long_entry = posibles_fases[3]#np.max(data[f'Fase_num_base_3_{instr}'].u
nique()) # fase bajista, B3
fase_clave_long_exit = posibles_fases[3]#np.max(data[f'Fase_num_base_3_{instr}'].un
ique())
# *****

if direccion == 'long':
    if data.iloc[0][f'Fase_num_base_3_{instr}']==fase_clave_long_entry: # si comenz
amos en una zona en que debería haber una compra abierta, saltar a la siguiente, po
rque esta primera no hemos entrado y no cuenta completa
        data = data.iloc[data[data[f'Fase_num_base_3_{instr}']==fase_clave_long_ent
ry].iloc[0].Ordinal-1:]
        if data.iloc[-
1][f'Fase_num_base_3_{instr}']==fase_clave_long_exit: # si la data acaba en una ope
ración abierta, restringir la data hasta que no haya operaciones abiertas
            data = data.iloc[:int(data[np.logical_not(data[f'Fase_num_base_3_{instr}']
==fase_clave_long_entry)].iloc[-1].Ordinal-1)]

            data['Entry_signal_long'] = np.where(np.logical_and((data[f'Fase_num_base_3_{ins
tr}']==fase_clave_long_entry,np.logical_not(data[f'Fase_num_base_3_{instr}'].shift(
1)==fase_clave_long_entry)),1,0)
            data['Entries'] = data.Entry_signal_long
            data['Exit_signal_long'] = np.where(np.logical_and((data[f'Fase_num_base_3_{inst
r}'].shift(1)==fase_clave_long_exit,np.logical_not(data[f'Fase_num_base_3_{instr}']
==fase_clave_long_exit)),-1,0)
            data['Exits'] = data.Exit_signal_long
            ops_long = np.count_nonzero(data.Entry_signal_long)
            if np.logical_not(len(data[data.Exit_signal_long==
1])==ops_long): # si no se cierran todas las ops que se abren
                data.at[data.index[-1],'Exit_signal_long'] = -1
                assert len(data[data.Entry_signal_long==1].index) == len(data[data.Exit_signal_
long==-1].index)
            OPS_LONG = pd.DataFrame(columns = ['Apertura op', 'Cierre op', 'Return', 'Equit
y'])
            OPS_LONG['Apertura op'] = data[data.Entry_signal_long==1].index
            OPS_LONG['Cierre op'] = data[data.Exit_signal_long==-1].index

            for op in tqdm.tqdm(range(ops_long)):
                OPS_LONG.at[op, 'Duración'] = OPS_LONG.loc[op, 'Cierre op']- OPS_LONG.loc[op
p, 'Apertura op']
                OPS_LONG.at[op, 'Duracion_str'] = str(OPS_LONG.at[op, 'Duración'])
                OPS_LONG.at[op, 'Duración días'] = int(re.findall('\d{1,3}', OPS_LONG.at[op
, 'Duracion_str'])[0])
                OPS_LONG.at[op, 'Return'] = data.loc[OPS_LONG.loc[op, 'Apertura op']:OPS_LO
NG.loc[op, 'Cierre op']][1:]['Dif_Adj_Close_Norm'].sum()
                OPS_LONG['Equity'] = OPS_LONG.Return.cumsum()
                retorno_medio_estrategia = np.mean(OPS_LONG.Return)
                plt.plot(OPS_LONG.Equity, 'o-')

# *****

```

```

Cálculo de MonteCarlo para la obtención de las distribuciones muestrales aleatorias
OPS_LONG.dropna(inplace=True)
# *****
Nsim = 1000
# *****
data_mix = copy.copy(data)
sims = pd.DataFrame(index = range(Nsim), columns = ['Final_return', 'Mean_return', '
Std_return', 'Sharpe_Ratio'])
numeros = data['Dif_Adj_Close_Norm'].tolist()
for n in tqdm.tqdm(range(Nsim)):
    random.shuffle(numeros)
    data_mix['Dif_Adj_Close_Norm'] = numeros
    for op in range(ops_long):
        OPS_LONG.at[op, f'Return_{n}'] = data_mix.loc[OPS_LONG.loc[op, 'Apertura op']:0
PS_LONG.loc[op, 'Cierre op']][1:]['Dif_Adj_Close_Norm'].sum()
        sims.at[n, 'Final_return'] = OPS_LONG[f'Return_{n}'].cumsum().iloc[-1]
        sims.at[n, 'Mean_return'] = np.mean(OPS_LONG[f'Return_{n}'])
        sims.at[n, 'Std_return'] = np.std(OPS_LONG[f'Return_{n}'])
        sims.at[n, 'Sharpe_Ratio'] = sims.at[n, 'Mean_return'] / sims.at[n, 'Std_return']

plt.hist(OPS_LONG[f'Return_0'], bins=100, alpha=0.5)
plt.hist(OPS_LONG[f'Return_1'], bins=100, alpha=0.5)

# Se utiliza el modelo de predicción Prophet y se compara con la predicción del
Teorema Central del límite
make_forecast(instrument, 365)
plt.plot(data.Dif_Adj_Close.cumsum()+valor_inicial)
plt.plot(data.Dif_Adj_Close_Norm.cumsum()+valor_inicial)
plt.title(instrument)

from scipy import stats
np.random.seed(12345678)

rvs1 = OPS_LONG['Return_0']
rvs2 = OPS_LONG['Return_20']
stats.ks_2samp(rvs1, rvs2)

KstestResult(statistic=0.053125, pvalue=0.7580724560481628)

for i in range(Nsim):
    plt.plot(OPS_LONG[f'Return_{i}'].cumsum().values.astype('float'))

rvs = sims.Final_return.values.astype('float')
stats.kstest(stats.norm.rvs, 'norm', N=Nsim)
KstestResult(statistic=0.02225365386872158, pvalue=0.6963534811493965)

# Kolmogorov-
Smirnov ks_2samp con las posibles parejas resultado > 0.5 son iguales y aplicable t
eorema central del limite
return_num_array1 = []
return_num_array2 = []
ksarray = []
ksarraynorm = []
for index, row in OPS_LONG.iterrows():
    return_num1 = 'Return_'+ str(index)
    return_num_array1.insert(index, row[return_num1])
    for i, row in OPS_LONG.iterrows():
        if i < index:
            continue
            return_num2 = 'Return_'+ str(i)
            return_num_array2.insert(i, row[return_num2])
        ksvalue = stats.ks_2samp(return_num_array1, return_num_array2)
        ksarray.insert(index, ksvalue[1])
        ksnormvalue = stats.kstest(return_num_array1, stats.norm.cdf)
        ksarraynorm.insert(index, ksnormvalue[1])
steps = np.linspace(0,1,100)
plt.plot(ksarray)
plt.plot(ksarraynorm)

```

ANEXO 4: Cálculo del exponente de Hurst

```

import yfinance as yf
import numpy as np
import pandas as pd

spy_df = yf.download("BTC",
                    start="2010-01-01",
                    end="2022-10-31",
                    progress=False)

spy_df["Adj Close"].plot(title="BTC");
sample_size = 1000
scalar = 100

np.random.seed(114)
df = pd.DataFrame(data={"Mean-
reverting": np.log(np.random.randn(sample_size) + scalar),
                    "Camino aleatorio": np.log(np.cumsum(np.random.randn(sample
_size)) + scalar),
                    "Trending": np.log(np.cumsum(np.random.randn(sample_size) +
1) + scalar)})

df.plot(title="Serie temporal generada");
def get_hurst_exponent(time_series, max_lag=20):
    """Devuelve al exponente Hurst de una serie temporal"""

    lags = range(2, max_lag)

    # varianzas de las diferencias rezagadas
    tau = [np.std(np.subtract(time_series[lag:], time_series[:-
lag])) for lag in lags]

    # calcula la pendiente de la gráfica logarítmica
    reg = np.polyfit(np.log(lags), np.log(tau), 1)

    return reg[0]
for lag in [5, 10, 20, 100, 300, 500]:
    print(f"Hurst exponents with {lag} lags ----")
    for column in df.columns:
        print(f"{column}: {get_hurst_exponent(df[column].values, lag):.4f}")

make_forecast('BTC', 365, '4y')
plt.title('BTC')

```