

MÁSTER UNIVERSITARIO EN TECNOLOGÍAS DEL
SECTOR FINANCIERO: FINTECH
2023-2024

Trabajo Fin de Grado

**“Predicción de rentabilidad futura de
criptomonedas”**

Universidad Carlos III de Madrid
Changhao Wu

Tutor
Fernando Fernández Rebollo
Junio, 2024



Esta obra se encuentra sujeta a la licencia creative Commons
Reconocimiento - No comercial - Sin Obra Derivada

RESUMEN

El presente trabajo Fin de Máster tiene como objetivo realizar un estudio de la evolución del precio de las criptomonedas, concretamente de Bitcoin en un periodo de tiempo determinado. Para lograr este objetivo, se ha llevado a cabo un estudio exhaustivo que ha involucrado el uso del lenguaje de programación Python, un entorno de desarrollo Visual Studio y diversas librerías e herramientas con el propósito de análisis de datos, visualización, modelado y predicción de datos, todo ello guiado y estructurado según la metodología CRISP-DM, pasando por todos sus ciclos de desarrollo.

A través del estudio se han obtenido resultados significativos, que demuestra la complejidad de encontrar informaciones relevantes dentro de un mercado complejo. Este estudio se probará los distintos enfoques a la hora de crear un modelo predictivo para predecir las rentabilidades a futuro de las criptomonedas. Se dará un especial enfoque al uso estudio de redes de neuronas, específicamente LSTM para el estudio de series temporales.

Palabras clave

Análisis de datos, Análisis predictivo, Series temporales, Criptomonedas

ABSTRACT

The present Master's thesis aims to conduct a study on the evolution of cryptocurrency prices, specifically Bitcoin, over a determined period. To achieve this objective, an exhaustive study has been carried out involving the use of the Python programming language, the Visual Studio development environment, and various libraries and tools for the purposes of data analysis, visualization, modeling, and prediction. This study is guided and structured according to the CRISP-DM methodology, covering all its development cycles.

Significant results have been obtained through the study, demonstrating the complexity of finding relevant information within a complex market. This study will test different approaches in creating a predictive model to forecast the future returns of cryptocurrencies. Special focus will be given to the study of neural networks, specifically LSTM, for the analysis of time series

Key Words

Data Analysis, Predictive Analysis, Time Series, Cryptocurrencies

ÍNDICE DE CONTENIDOS

1. INTRODUCCIÓN.....	10
1.1. Motivación.....	10
1.2. Contexto.....	11
1.3. Objetivos.....	12
1.4. Metodología CRISP-DM.....	13
1. Comprensión del negocio.....	14
2. Comprensión de los datos.....	15
3. Preparación de los datos.....	16
4. Modelado.....	17
5. Evaluación.....	19
6. Despliegue.....	19
2. ESTADO DEL ARTE.....	20
2.1. Series temporales.....	20
2.2. Neurona Artificial.....	22
2.3. Redes neuronas.....	23
2.4. Red neuronal recurrente (RNN).....	24
2.5. Long-Short Term Memory (LSTM).....	25
2.6. Herramientas y framework.....	25
2.6.1. Python.....	25
2.6.2. Librería Python.....	26
3. CASO DE ESTUDIO.....	26
3.1. Comprensión del negocio.....	26
3.1.1. Digitalización del mercado financiero	26
3.1.2. Bitcoin y exchanges de criptomonedas.....	27
3.2. Objetivo de negocio.....	27
3.2.1. Objetivos técnicos.....	27
3.2. Comprensión de los datos.....	27
3.2.1. Recopilación de los datos.....	27
3.2.2. Características de los datos.....	28
4. ANÁLISIS DE LOS DATOS.....	28
4.1. Exploración inicial de los datos.....	28
4.1.1. Preparación del Dataframe.....	29
4.1.1.1. Fusión de los datos.....	29
4.1.1.2. Comprobación de datos nulos.....	30
4.1.2. Análisis exploratorio de datos.....	32
4.1.2.1. Tratamiento y fusión de datos.....	32
4.1.2.1. Visualización de precio cierre.....	33
Como se puede apreciar presenta cierta correlación entre ellas.....	34
4.1.2.2. Visualización de volumen.....	34

4.1.2.3. Visualización Media móvil.....	35
4.1.2.4. Visualización de evolución de rentabilidad por horas.....	37
4.1.2.5. Visualización de distribución de la rentabilidad.....	39
4.1.2.6. Correlación entre los precios.....	40
4.1.2.7. Correlación de precio de BTC entre ETH.....	41
4.1.2.8. Correlación y regresión entre criptomonedas.....	42
4.1.2.9. Matriz correlación de precio y rentabilidad.....	44
Con esto se puede sacar conclusión de que la evolución del precio y rentabilidad de las criptomonedas están altamente correlacionados.....	44
4.1.2.10. Comparación evolución precio de criptomonedas.....	44
4.2. Preparación de los datos y Selección de variables.....	46
4.3. Modelado.....	49
4.3.1. Modelado sin usar indicadores técnicos.....	49
4.3.2. Modelado usando indicadores técnicos.....	52
5. Despliegue.....	57
5.1. Selección de modelo.....	57
5.2. Despliegue en web.....	57
6. CONCLUSIONES Y TRABAJOS FUTUROS.....	59
6.1. Conclusiones.....	59
6.2. Trabajos futuros.....	59
7. ANEXO.....	60
8. BIBLIOGRAFÍA.....	61

INDICE DE FIGURAS

Figura 1.2 - Evolución precio bitcoin.....	8
Figura 1.4 - Grado de utilización de las distintas metodologías de minería de datos.....	10
Figura 1.4.5 - Ciclo CRISP-DM.....	11
Figura 1.4.5.1 - Ciclo Comprensión negocio.....	12
Figura 1.4.5.2 - Ciclo CRISP-DM Comprensión de datos.....	13
Figura 1.4.4.3 - Fase preparación de datos.....	14
Figura 1.4.5.4 - Fase modelado.....	15
Figura 2.1 - Gráfico media móvil.....	18
Figura 2.1.2 - Gráfico oscilador estocástico.....	19
Figura 2.1.3 - Gráfico volumen frente evolución del precio SP500.....	19
Figura 2.2 - Neurona artificial.....	20
Figura 2.3 - Red de neuronas.....	21
Figura 2.4 - Red de neurona recurrente.....	21
Figura 2.5 - LSTM de dos capas con tres entradas y salida.....	22
Figura 3.2.2 - Primeros registros del dataset en bruto.....	25
Figura 4.1.1 - Código fusión de datos.....	27
Figura 4.1.2 - Salida de ejecución del código fusión de datos.....	27
Figura 4.1.1.2 - Código de comprobación de datos nulos.....	28
Figura 4.1.1.3 - Salida de código de comprobación de datos nulos.....	28
Figura 4.1.2.1 - Código de tratamiento y fusión de datos.....	29
Figura 4.1.2.2 - Salida de código de tratamiento y fusión de datos.....	30
Figura 4.1.2.3 - Código función describe.....	30
Figura 4.1.2.4 - Salida de código función describe.....	30
Figura 4.1.2.5 - Código función describe.....	31
Figura 4.1.2.6 - Salida de código función describe.....	31
Figura 4.1.2.6 - Código visualización volumen.....	32
Figura 4.1.2.7 - Salida código visualización volumen.....	32
Figura 4.1.2.9 - Código visualización Media móvil.....	33
Figura 4.1.2.10 - Salida código visualización Media móvil.....	34
Figura 4.1.2.11 - Código visualización Rentabilidad.....	35
Figura 4.1.2.12 - Salida código visualización Rentabilidad.....	36
Figura 4.1.2.13 - Código distribución de Rentabilidad.....	36
Figura 4.1.2.14 - Salida código distribución de Rentabilidad.....	37
Figura 4.1.2.15 - Código Correlación entre precios de criptomonedas.....	38
Figura 4.1.2.16 - Salida código Correlación entre precios de criptomonedas.....	38
Figura 4.1.2.16 - Código correlación precio BTC-ETH.....	39
Figura 4.1.2.17 - Salida código correlación precio BTC-ETH.....	39
Figura 4.1.2.18 - Código correlación precio entre criptomonedas.....	40
Figura 4.1.2.19 - Salida código correlación precio entre criptomonedas.....	41

Figura 4.1.2.20 - Código matriz correlación precio entre criptomonedas.....	41
Figura 4.1.2.21- Salida código matriz correlación precio entre criptomonedas.....	41
Figura 4.1.2.22 - Código comparación de evolución de precios.....	42
Figura 4.1.2.23 - Salida código comparación de evolución de precios.....	43
Figura 4.2.1 - Código preparación de datos.....	44
Figura 4.2.2 - Código visualización media móvil y rsi.....	45
Figura 4.2.3 - Salida código visualización media móvil y rsi.....	46
Figura 4.3.1 - Código modelado.....	47
Figura 4.3.2 - Tabla de experimentaciones del entrenamiento sin indicadores técnicos.....	47
Figura 4.2.3 - Código visualización predicciones modelo.....	48
Figura 4.2.4 - Salida código visualización predicciones modelo.....	49
Figura 4.3.2.1 - Código modelado con indicadores técnicos.....	50
Figura 4.3.2.2 - Tabla de experimentaciones del entrenamiento con indicadores técnicos.....	51
Figura 4.3.2.3 - Código visualización predicciones modelo.....	53
Figura 4.3.2.4 - Salida código visualización predicciones modelo con indicadores técnicos.....	53
Figura 4.3.2.4 - Aplicación web con el modelo realizando predicciones en vivo.....	55

1.INTRODUCCIÓN

En este capítulo introductorio se presenta el contexto y la motivación detrás del TFG, así como los objetivos y las preguntas de investigación que se plantean para abordar el problema. Además, se hace referencia al marco regulador que enmarca el estudio, el método de trabajo utilizado para llevar a cabo la investigación y la organización del documento en sí. Este capítulo es fundamental para comprender la relevancia del tema y cómo se aborda en este trabajo de investigación.

1.1. Motivación

En los últimos años, las criptomonedas, y en particular Bitcoin, han emergido como una de las innovaciones más disruptivas en el ámbito financiero. Este fenómeno ha capturado la atención no solo de los inversores individuales, sino también de instituciones privadas [1]. A nivel global, la adopción de criptomonedas varía significativamente entre países, impulsada por diversos factores como la regulación, la infraestructura tecnológica y la situación económica local. Países como El Salvador han adoptado Bitcoin como moneda de curso legal, marcando un hito en la aceptación institucional de las criptomonedas [2]. Otros países, como Estados Unidos y varios miembros de la Unión Europea, están desarrollando marcos regulatorios que buscan equilibrar la innovación con la protección del consumidor. Esta disparidad en la adopción y regulación subraya la importancia de estudiar las criptomonedas desde múltiples perspectivas, considerando tanto los aspectos técnicos como los contextos socioeconómicos específicos [3].

Otro factor que está impulsando el interés y la adopción de Bitcoin es el desarrollo y aprobación de ETFs (Exchange-Traded Funds) basados en Bitcoin. Estos instrumentos financieros permiten a los inversores tradicionales acceder a Bitcoin sin tener que manejar directamente la criptomoneda, lo que reduce las barreras de entrada y aumenta la legitimidad de Bitcoin en los mercados financieros convencionales. La aprobación de ETFs de Bitcoin en diversos países ha sido un catalizador para una mayor adopción institucional y un aumento en el interés de los inversores [4].

La motivación principal de este Trabajo Fin de Máster radica en la creciente relevancia de las criptomonedas en el mercado financiero global y la complejidad de su dinámica de precios. A pesar de su popularidad, aún existe una considerable incertidumbre y falta de consenso sobre cómo predecir sus movimientos de manera efectiva. Este trabajo pretende contribuir al cuerpo de conocimiento existente, proporcionando insights valiosos a través de un enfoque metódico y científico.

1.2. Contexto

En el panorama económico actual, la rápida evolución de la tecnología y la digitalización está transformando fundamentalmente la forma en que se interactúa con el sistema financiero. Las criptomonedas, como Bitcoin, han surgido como una manifestación destacada de esta revolución, desafiando los paradigmas tradicionales de dinero y activos financieros.

La creación de Bitcoin en 2009 por una persona o grupo bajo el seudónimo de Satoshi Nakamoto marcó el inicio de una nueva era en la que las transacciones financieras pueden ser realizadas de manera descentralizada, sin la necesidad de intermediarios como bancos o gobiernos. La tecnología subyacente de las criptomonedas, blockchain, ha sido aclamada por su capacidad para garantizar la seguridad y la transparencia en las transacciones, al tiempo que preserva el anonimato y la privacidad de los usuarios [5].

A lo largo de los años, Bitcoin ha experimentado una trayectoria de crecimiento exponencial, atrayendo tanto el interés de inversores individuales como de instituciones financieras. La volatilidad característica de Bitcoin ha generado oportunidades significativas para obtener ganancias, pero también ha planteado desafíos en términos de gestión de riesgos y predictibilidad [6].



Figura 1.2 - Evolución precio bitcoin.

Además, la adopción de criptomonedas está influenciada por factores regulatorios y geopolíticos. La actitud de los gobiernos hacia las criptomonedas varía considerablemente en todo el mundo, desde la adopción entusiasta hasta la prohibición total. Esta incertidumbre

regulatoria puede tener un impacto significativo en el mercado y en la percepción pública de las criptomonedas como una clase de activo legítimo.

En este contexto dinámico y en constante evolución, es crucial comprender las fuerzas que impulsan el mercado de criptomonedas y explorar enfoques analíticos que puedan ayudar a los inversores y analistas a navegar por este nuevo paisaje financiero. Este Trabajo Fin de Máster se sitúa en la intersección de la tecnología, las finanzas y la ciencia de datos, con el objetivo de proporcionar una visión integral y basada en evidencia de la evolución del mercado de criptomonedas y las estrategias para analizar y predecir su comportamiento futuro.

1.3. Objetivos

A continuación, se expone el objetivo principal de este trabajo, seguido de una lista de objetivos específicos que se pretenden satisfacer:

Objetivo Principal: Estudiar y analizar la evolución de la rentabilidad futura del Bitcoin.

Con el fin de lograr el objetivo principal del trabajo, se han especificado también los siguientes objetivos complementarios:

- **Objetivo 1:** Estudio de exchanges de criptomonedas. Para la recopilación de datos se estudiarán los exchanges de criptomonedas existentes y las APIs que estos ofrecen para la extracción de los datos, con el fin de tener una fuente de información directa sobre Bitcoin y otras criptomonedas.
- **Objetivo 2:** Análisis de datos. Llevar a cabo un análisis exhaustivo de los datos recopilados. Se recurrirá a técnicas estadísticas y herramientas de minería de datos para explorar y examinar la información recopilada. El análisis de datos permitirá identificar patrones, tendencias y relaciones relevantes que ayudarán a responder las preguntas de investigación planteadas.
- **Objetivo 3:** Visualización de los datos. Presentar los resultados de manera clara y concisa. Se utilizarán gráficos, tablas y otros recursos visuales para comunicar de manera efectiva las conclusiones extraídas del análisis. La presentación del análisis de datos permitirá visualizar y comprender de forma más accesible los hallazgos relevantes obtenidos durante la investigación.
- **Objetivo 4:** Modelado. Utilizar los conocimientos adquiridos para realizar predicciones. Se emplearán técnicas de modelado y algoritmos de predicción para predecir rentabilidades y precios futuros de Bitcoin. Estas predicciones ayudarán a comprender el comportamiento futuro del fenómeno o sistema estudiado, y podrán servir de apoyo para la toma de decisiones en el contexto de investigación.
- **Objetivo 5:** Realización de un ciclo completo de la metodología CRISP-DM. La realización de un ciclo completo de la metodología CRISP-DM garantizará una estructura sistemática y una orientación precisa en el desarrollo de la investigación y el análisis de los datos.

1.4. Metodología CRISP-DM.

CRISP-DM (Cross-Industry Standard Process for Data Mining) proporciona un marco estructurado que guía el proceso de análisis de datos desde la comprensión del negocio hasta la implementación de soluciones, asegurando un enfoque metodológico coherente y riguroso en todas las etapas del proyecto.

Para llevar a cabo el desarrollo de un proyecto dentro del ámbito de la ciencia de datos, es necesario seguir una metodología. Las metodologías generalmente provienen de procedimientos estandarizados y de experiencias propias. Dentro de las metodologías existentes, CRISP-DM es la más popular según www.datascience-pm.com, basado en encuestas que se pueden visualizar en el siguiente gráfico [7]:

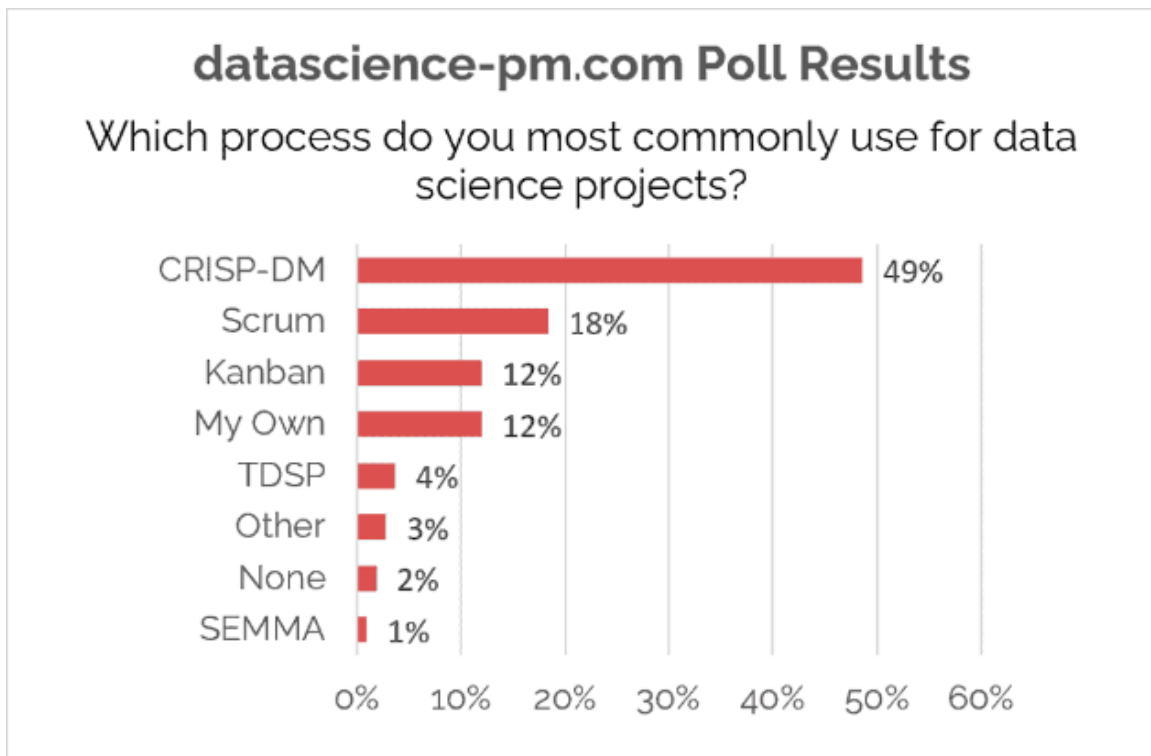


Figura 1.4 - Grado de utilización de las distintas metodologías de minería de datos.

El proceso CRISP-DM consta de seis fases principales, y cada una de ellas permite avanzar o retroceder dependiendo de los resultados de la fase actual. Es decir, puede no seguir un ciclo de vida lineal y pueden haber varias iteraciones antes de completar un ciclo completo. A continuación se muestra un diagrama en el que se puede observar el ciclo completo de CRISP-DM y las posibles transiciones entre ellas [8].

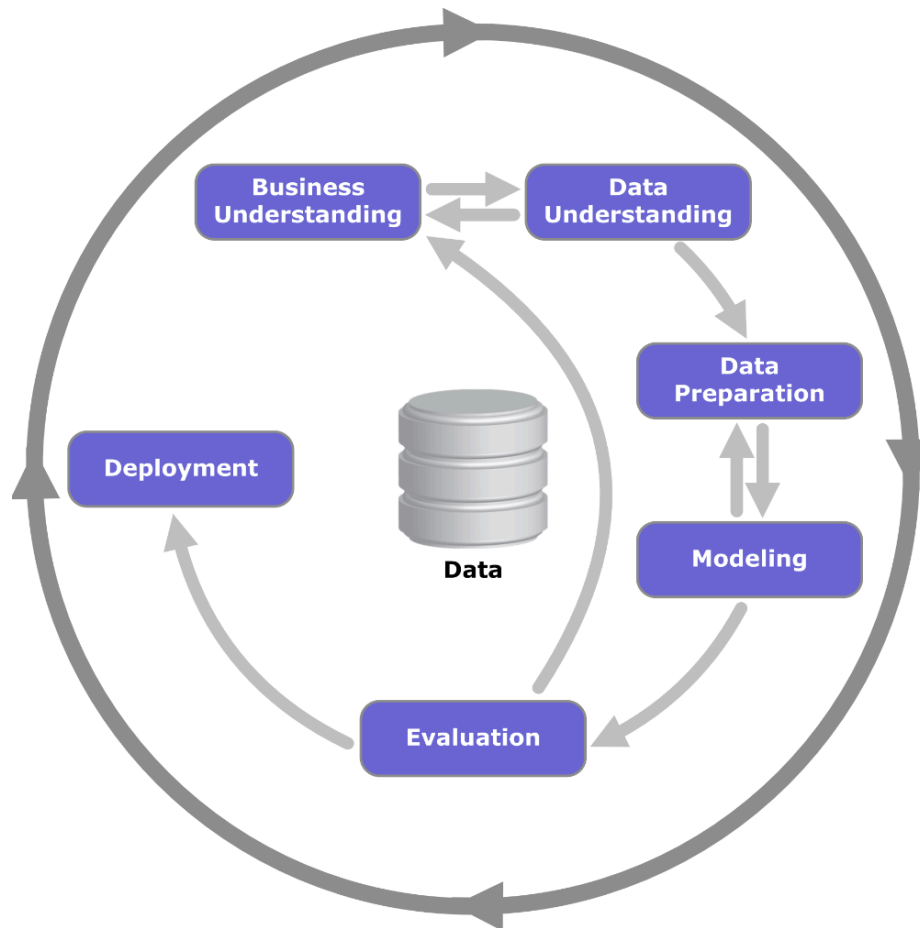


Figura 1.4.5 - Ciclo CRISP-DM.

En las siguientes secciones se introducirán en detalle las diferentes etapas del ciclo de vida de CRISP-DM.

1. Comprensión del negocio

En esta etapa se centra en el estudio y el entendimiento profundo de los objetivos y los requerimientos del proyecto. Se trata de una etapa fundamental y de extrema importancia para el desarrollo de las etapas posteriores. A pesar de ello, existe una tendencia general a infravalorar esta etapa inicial [9].

En esta fase inicial se encuentran las siguientes posibles tareas a realizar:

- **Determinar los objetivos del negocio:** Entender desde el punto de vista del negocio las necesidades del cliente final. A menudo, el cliente tiene múltiples objetivos y restricciones que deben equilibrarse adecuadamente.
- **Determinar la situación:** Evaluar el estado actual de la situación antes de comenzar con el proceso de minería de datos. Para ello, se deben considerar aspectos como el conocimiento previo disponible sobre el problema, la disponibilidad y la cantidad de

datos necesaria para abordar el problema, y la relación entre el coste y el beneficio de aplicar la minería de datos.

- **Determinar los objetivos de la minería de datos:** Además de definir los objetivos del negocio, también se deben definir los escenarios exitosos desde el punto de vista de la minería de datos.
- **Planificación del proyecto:** Determinar las tecnologías y herramientas a usar en cada fase del proyecto

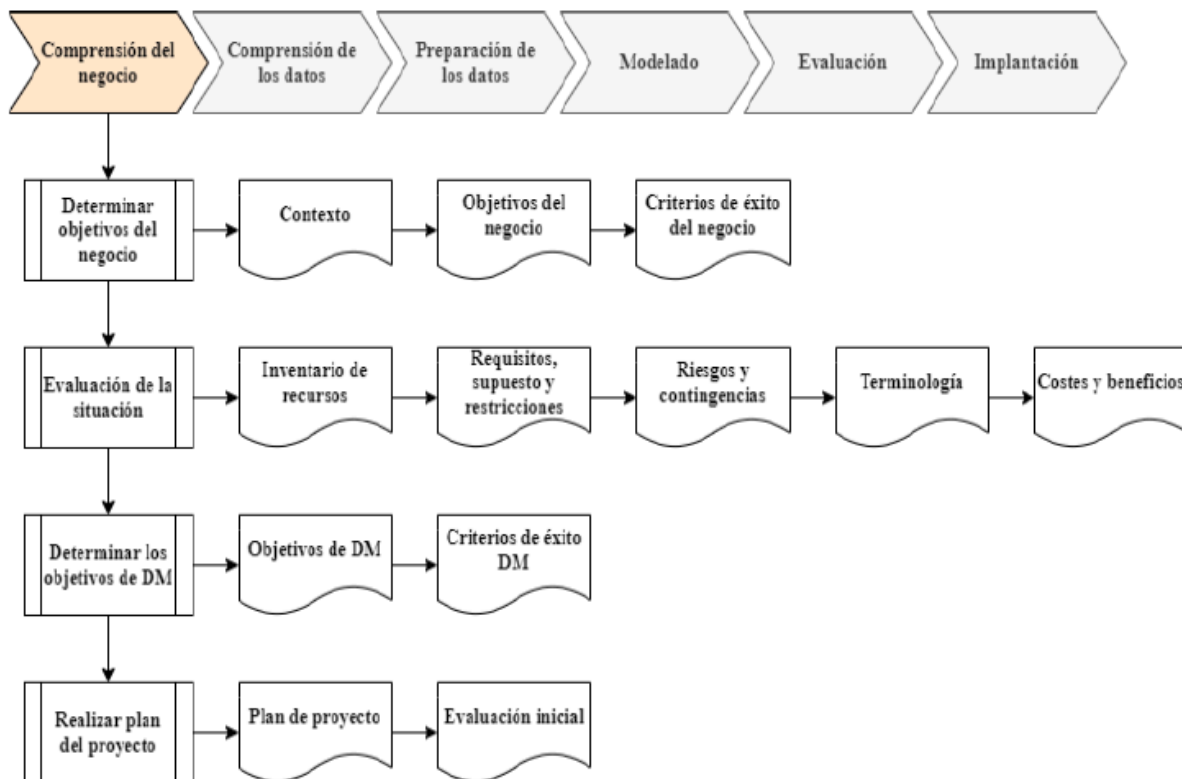


Figura 1.4.5.1 - Ciclo Comprensión negocio.

2. Comprensión de los datos.

Aquí se recopilan y exploran los datos disponibles relacionados con el problema en cuestión. Esto puede implicar la adquisición de datos de múltiples fuentes, la evaluación de la calidad de los datos y la identificación de posibles problemas o sesgos [10].

En esta fase se encuentran las siguientes posibles tareas a realizar:

- **Recolección de los datos necesarios:** Recopilar los datos relevantes para el proyecto.
- **Describir los datos:** Establecer el volumen de los datos e identificar el significado de cada campo.

- **Exploración de los datos:** Visualizar los datos e identificar posibles patrones.
- **Verificar la calidad de los datos:** Garantizar la completitud, verificar la consistencia y detectar valores nulos y valores fuera de rango.

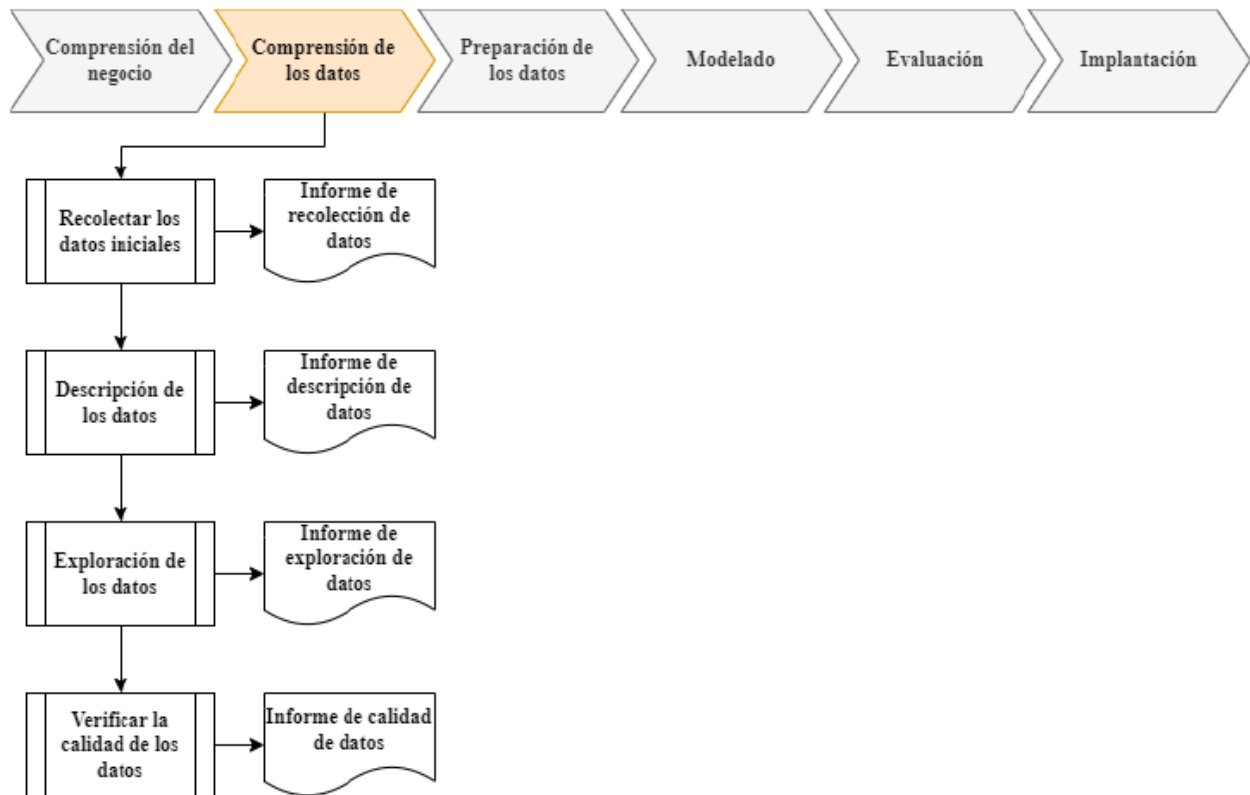


Figura 1.4.5.2 - Ciclo CRISP-DM Comprensión de datos.

3. Preparación de los datos.

La fase de preparación de datos, en general, lleva la mayor parte del tiempo. Se trata de adaptar los datos recolectados a las técnicas de minería de datos que se pretenden realizar a lo largo del proyecto [11].

En esta fase se encuentran las siguientes posibles tareas a realizar:

- **Selección de los datos:** Determinar los conjuntos de datos a utilizar y los conjuntos de datos a descartar.
- **Limpieza de los datos:** Esta tarea es complementaria a la anterior. Generalmente requiere un mayor esfuerzo y tiempo debido a las numerosas técnicas que existen para optimizar los datos. Se pueden usar técnicas como la normalización de los datos,

discretización de los datos, reducción de volúmenes, tratamiento de valores nulos, entre otras.

- **Construcción de los datos:** Derivar y agregar nuevos atributos que sean útiles basándose en los datos actuales.
- **Integrar los datos:** Crear nuevos datos mediante la combinación de datos de múltiples fuentes de información.
- **Formatear los datos:** Reformatear los datos si es necesario. Por ejemplo, convertir campos de valores string a campos de valores numéricos para realizar operaciones matemáticas.

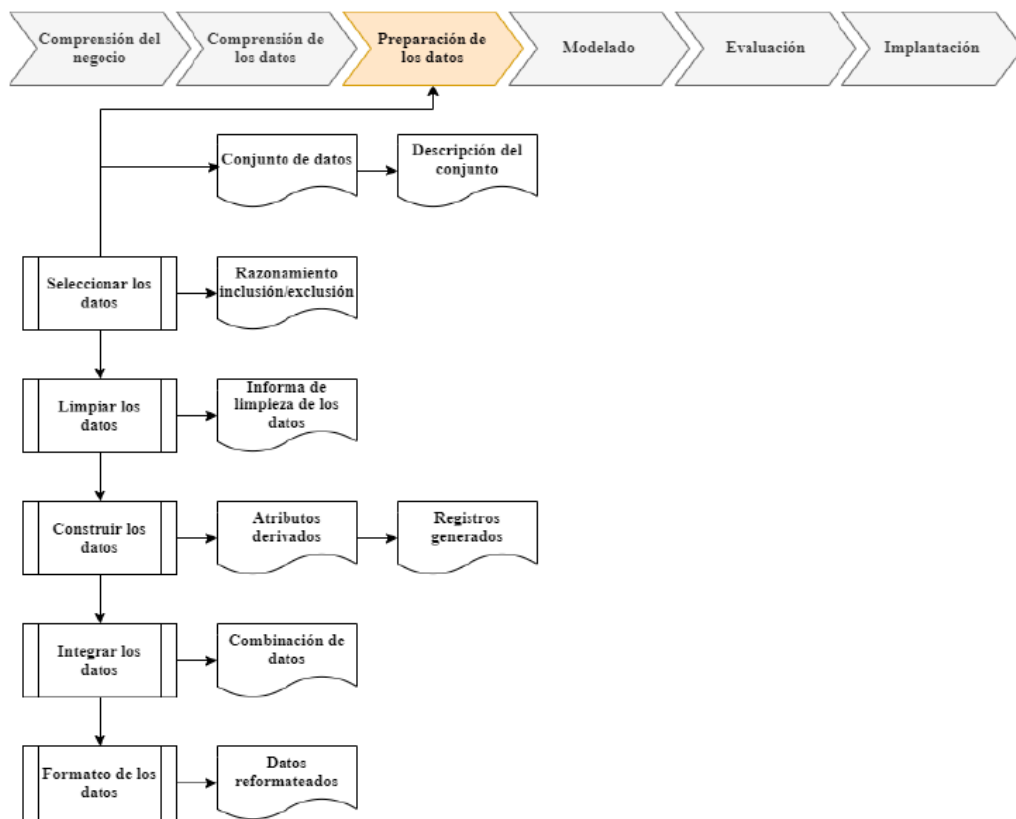


Figura 1.4.4.3 - Fase preparación de datos.

4. Modelado.

En esta etapa, se seleccionan y aplican técnicas de modelado para analizar los datos y extraer patrones o relaciones significativas. Se debe establecer una evaluación de las distintas técnicas y cuantificar el grado de adecuación de cada una de ellas. Se seleccionarán los parámetros

que se utilizan para el modelado dependiendo de las características de los datos y de la precisión que se quiera obtener [12].

En esta fase se encuentran las siguientes posibles tareas a realizar:

- **Selección de la técnica de modelado:** Determinar el algoritmo a probar, por ejemplo, regresión, redes neuronales, etc.
- **Generar plan de prueba:** Dependiendo del enfoque del modelado, se deberán dividir los datos para el entrenamiento, prueba y validación.
- **Construcción del modelo:** Ajustar óptimamente los hiper parámetros con el objetivo de minimizar la función de error.
- **Evaluar el modelo:** Interpretar los resultados en función del conocimiento del dominio.

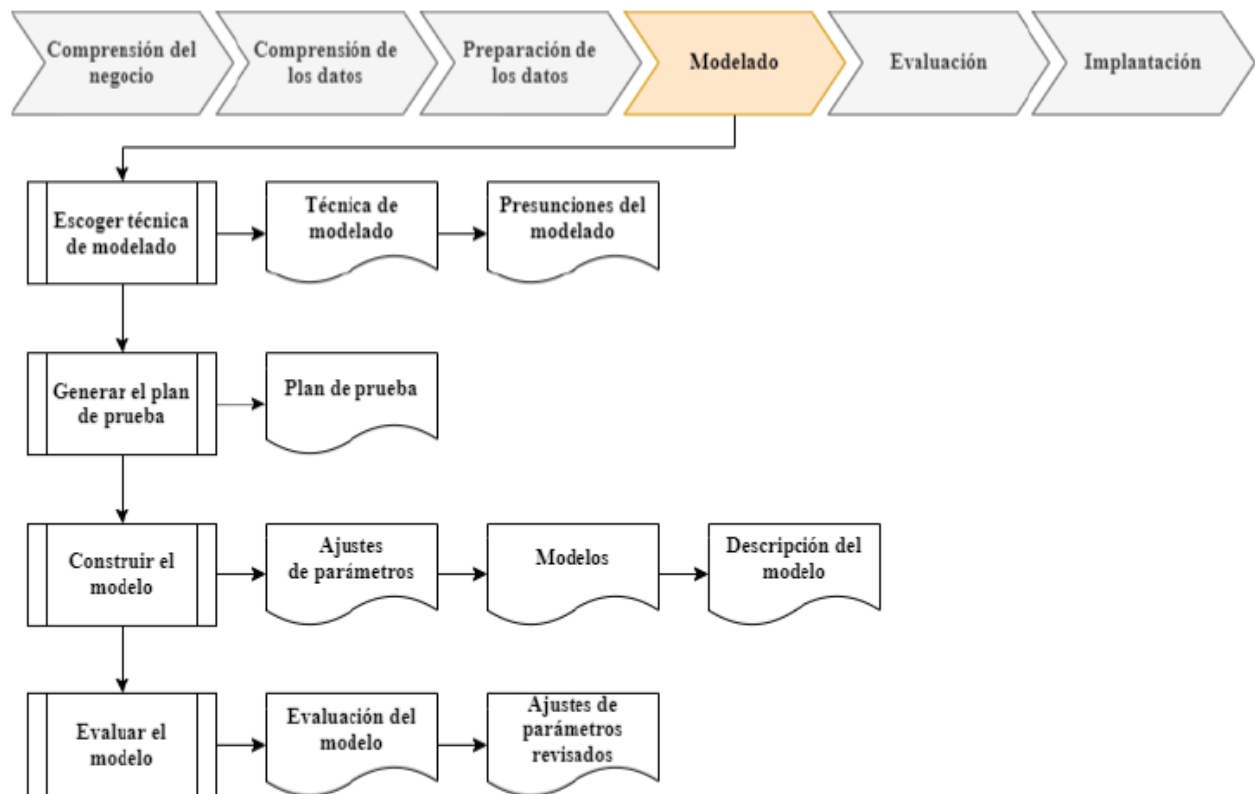


Figura 1.4.5.4 - Fase modelado.

5. Evaluación.

Se evalúan y validan los modelos desarrollados para determinar su eficacia y su capacidad para generalizar datos no vistos. Se llevan a cabo diversas actividades, como la evaluación del rendimiento de los modelos utilizando medidas de evaluación adecuadas, tales como tasas de error, precisión, sensibilidad, especificidad, entre otros. También se realiza una revisión exhaustiva de los resultados obtenidos en comparación con los criterios de éxito establecidos previamente en la fase de definición del proyecto [13].

En esta fase se encuentran las siguientes posibles tareas a realizar:

- **Evaluar los resultados:** Comprobar si los resultados se ajustan a los requerimientos del negocio.
- **Revisar el proceso:** Realizar una revisión completa del proceso de minería de datos con el fin de determinar si ha habido algún factor o tarea que se haya pasado por alto.
- **Determinar próximos pasos:** Decidir el despliegue del modelo o la realización de iteraciones adicionales. En casos extremos, se contempla la posibilidad de iniciar un nuevo proyecto.

6. Despliegue.

En esta última fase se implementan las soluciones desarrolladas en un entorno operativo y se comunican los resultados obtenidos. Esto puede implicar la creación de informes y visualizaciones para presentar los hallazgos del estudio, así como la integración del modelo predictivo en sistemas o aplicaciones relevantes [14].

En esta última fase se encuentran las siguientes tareas a realizar:

- **Plan de implantación:** Desarrollar y documentar el procedimiento para la implantación del modelo.
- **Plan de monitorización y de mantenimiento:** Una estrategia de mantenimiento bien planificada ayuda a evitar el uso incorrecto de los resultados de la minería de datos durante largos períodos de tiempo.
- **Producir informe final:** Documentar un resumen del proyecto en el que se detallen las experiencias y los resultados obtenidos. Dependiendo del plan de implementación, este informe puede ser una síntesis de las actividades realizadas y las lecciones aprendidas, o una presentación exhaustiva de los resultados obtenidos a través de la minería de datos.
- **Revisión del proyecto:** Evaluar los aciertos y desafíos del proyecto, identificando las áreas en las que se tuvo éxito y aquellas que necesitan mejorarse.

Al seguir la metodología CRISP-DM, se garantiza una aproximación sistemática y bien estructurada al estudio de la evolución del precio de las criptomonedas, lo que facilita la obtención de resultados sólidos y la toma de decisiones informadas basadas en evidencia.

2. ESTADO DEL ARTE

2.1. Series temporales

Una serie temporal es un conjunto de observaciones que se obtiene midiendo una variable única de manera regular a lo largo de un periodo de tiempo. Las series temporales no son independientes entre sí, durante el análisis debe tenerse en cuenta el orden temporal de las observaciones[15]. Cada observación de una serie temporal está asociada a un momento específico, lo que permite analizar cómo evoluciona una variable a lo largo del tiempo.

Un ejemplo común de las series temporales son el precio de las acciones en el mercado financiero o precio de las criptodivisas. La evolución de las series temporales presentan ciertos componentes que será detallada a continuación:

- **Tendencia:** Refleja el movimiento general a largo plazo en los datos, ya sea un aumento, una disminución o una estabilidad a lo largo del tiempo.
- **Estacionalidad** : Son patrones regulares y repetitivos que ocurren en intervalos específicos de tiempo, como estaciones del año, meses, días de la semana, etc.
- **Ciclo:** Fluctuaciones que ocurren en períodos más largos que la estacionalidad y que no son necesariamente regulares. Los ciclos pueden ser económicos o de otro tipo.
- **Irregularidad:** Variaciones aleatorias o residuales que no pueden ser explicadas por la tendencia, estacionalidad o ciclos.

En el ámbito financiero existen numerosos indicadores técnicos, modelos matemáticos y estadísticos, que ayudan a identificar la tendencia del precio en el mercado. A continuación se detalla algunos de los indicadores más utilizados: [16]:

- **Indicadores de tendencia:** movimientos prolongados de precio hacia alguna dirección durante un periodo de tiempo. Por ejemplo, la media móvil simple :

$$SMA_n(t) = \frac{Pr(t-n) + Pr(t-n+1) + \dots + Pr(t-1) + Pr(t)}{n} \quad (2.1)$$

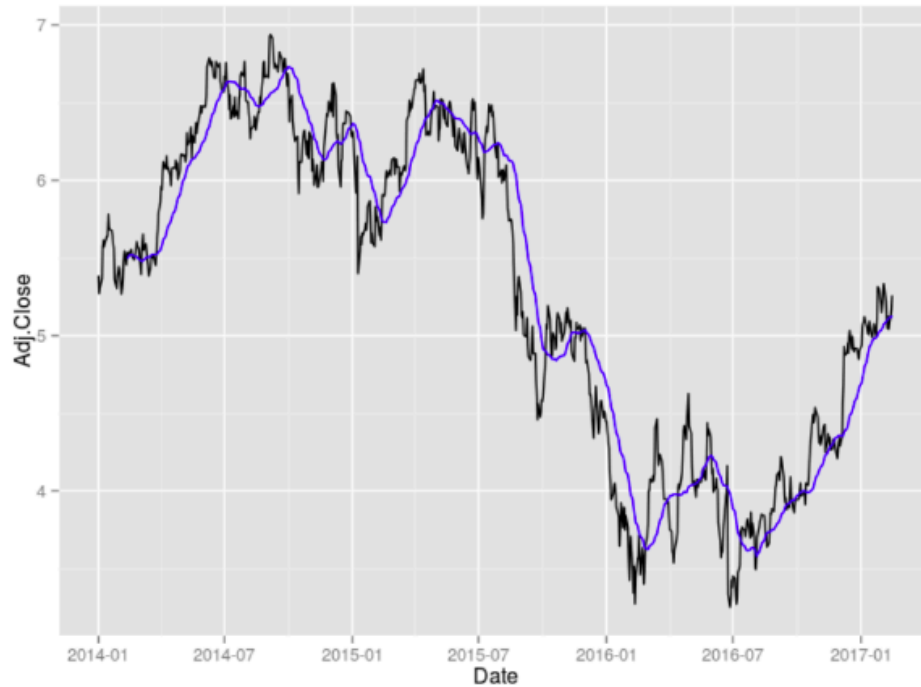


Figura 2.1 - Gráfico media móvil.

Media Móvil Exponencial (EMA):

$$EMA(t) = \alpha Pr(t) + (1 - \alpha) EMA(t - 1) \quad (2.2)$$

- **Indicadores de Momentum:** aceleración en los cambios de precio en un intervalo determinado. Por ejemplo, Oscilador Estocástico, que sitúa al precio respecto a el máximo y el mínimo de un intervalo de tiempo. Se utilizan media móviles sobre el oscilador para buscar señales técnicas.

$$SOsc(t) = 100 \times \frac{Pr(t) - Minn(t)}{Maxn(t) - Minn(t)} \quad (2.2)$$

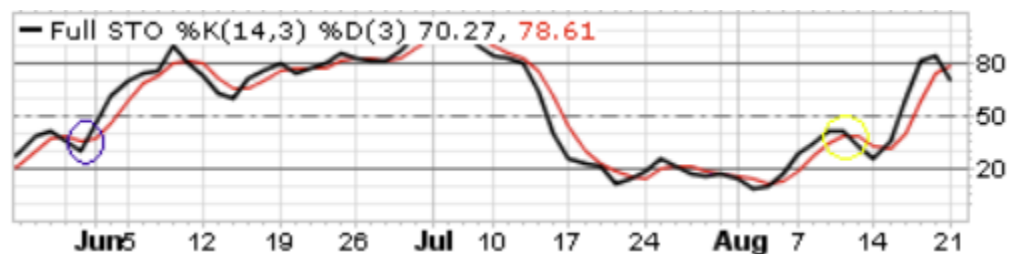


Figura 2.1.2 - Gráfico oscilador estocástico.

- **Indicadores de Volatilidad:** frecuencia e intensidad en la variación del precio a corto plazo. Por ejemplo:
 - Media del Rango Verdadero (ATR) : Mide el rango real respecto al precio de cierre anterior:

$$TR(t) = \max\{Max(t) - Min(t), Max(t) - Pr(t - 1), Pr(t - 1) - Min(t)\} \quad (2.3)$$

- **Indicadores de Volumen:** cantidad de transacciones de un activo y su relación respecto al precio en un intervalo de tiempo.

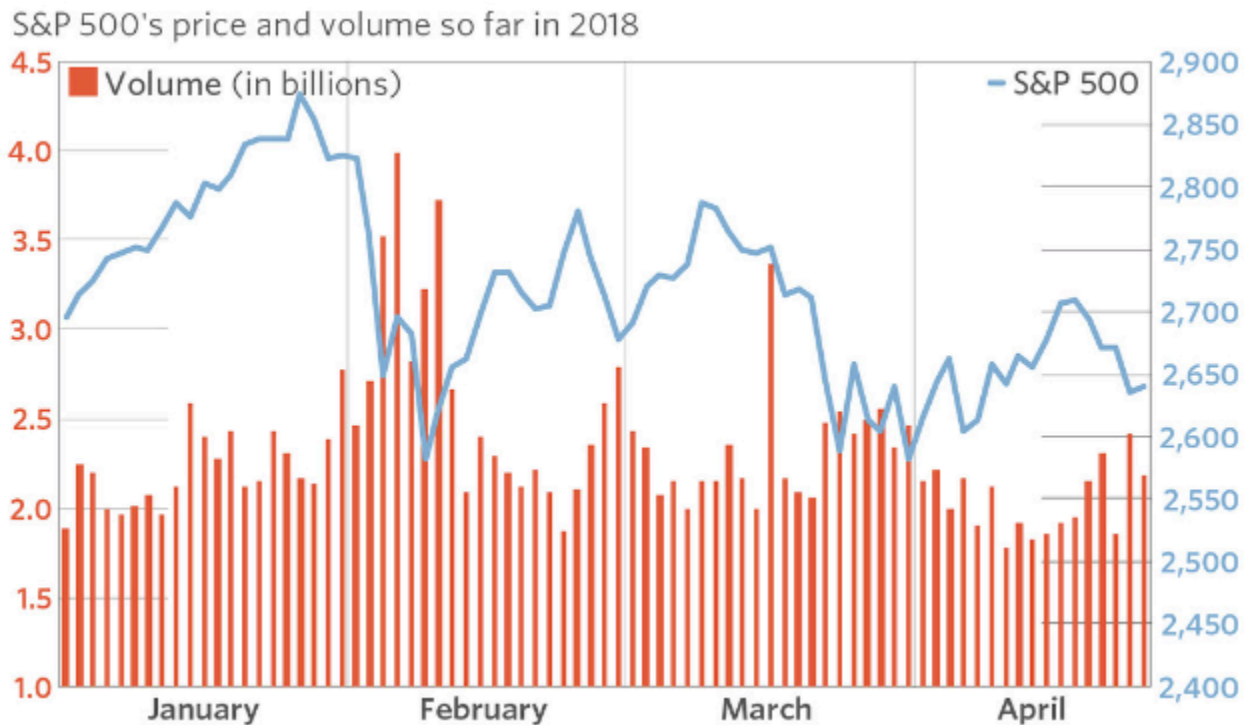


Figura 2.1.3 - Gráfico volumen frente evolución del precio SP500.

2.2. Neurona Artificial

Una neurona artificial es una unidad de procesamiento que imita la función de una neurona biológica. Cada neurona recibe una serie de entradas, las procesa y produce una salida. Una neurona artificial tiene los siguientes componentes[17]:

- **Entrada:** Son señales que pueden provenir de datos de entradas o las salidas de otras neuronas. Cada entrada tiene un peso asociado, que se ajusta durante el proceso de entrenamiento.

- **Pesos:** determina la importancia de cada entrada y es ajustando mediante el entrenamiento minimizando el error mediante el descenso de gradiente.
- **Función de activación:** Introduce no linealidad en el modelo, permitiendo que la red neuronal aprende y representa relaciones complejas. Por ejemplo, función de activación hiperbólica, unidad lineal rectificada, etc.
- **Bias:** Se trata de un término adicional a la suma ponderada de las entradas. Permite que la neurona modele mejor los datos al proporcionar una forma de ajustar la función de activación independientemente de las entradas.

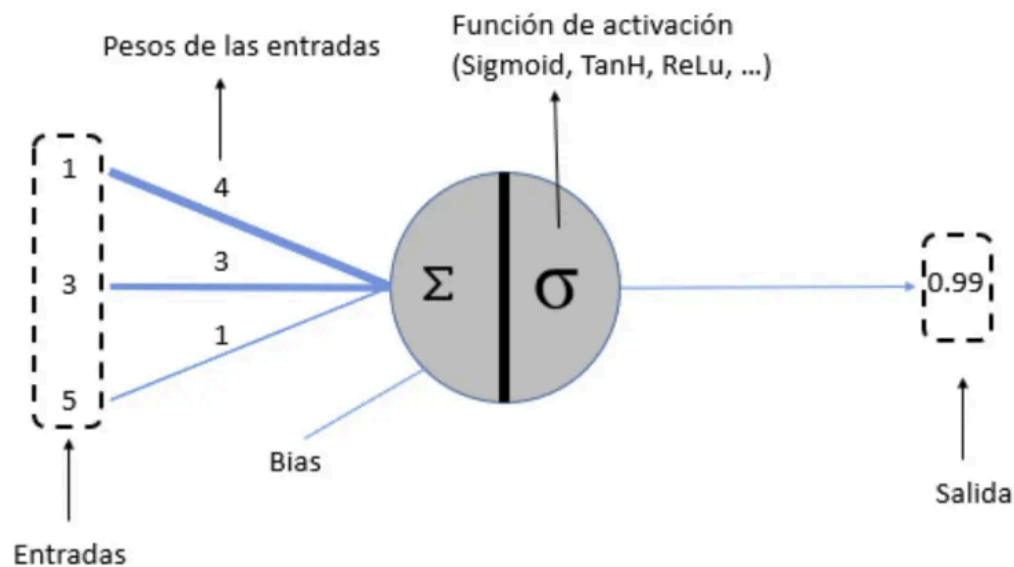


Figura 2.2 - Neurona artificial.

2.3. Redes neuronas

Las redes de neuronas artificiales son un grupo de neuronas interconectadas que transmiten señales entre sí. La información de entrada pasa por varias neuronas artificiales antes de producir la salida. [18]

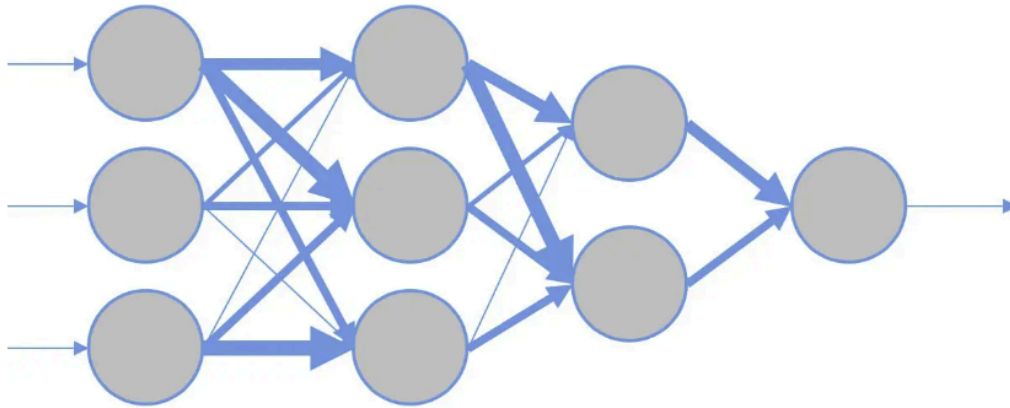


Figura 2.3 - Red de neuronas.

2.4. Red neuronal recurrente (RNN)

La red neuronal recurrente es un tipo de estructura de red neuronal artificial. Se caracteriza en que la salida de una neurona será utilizada como entrada de otra neurona en el mismo nivel de capa.[19]

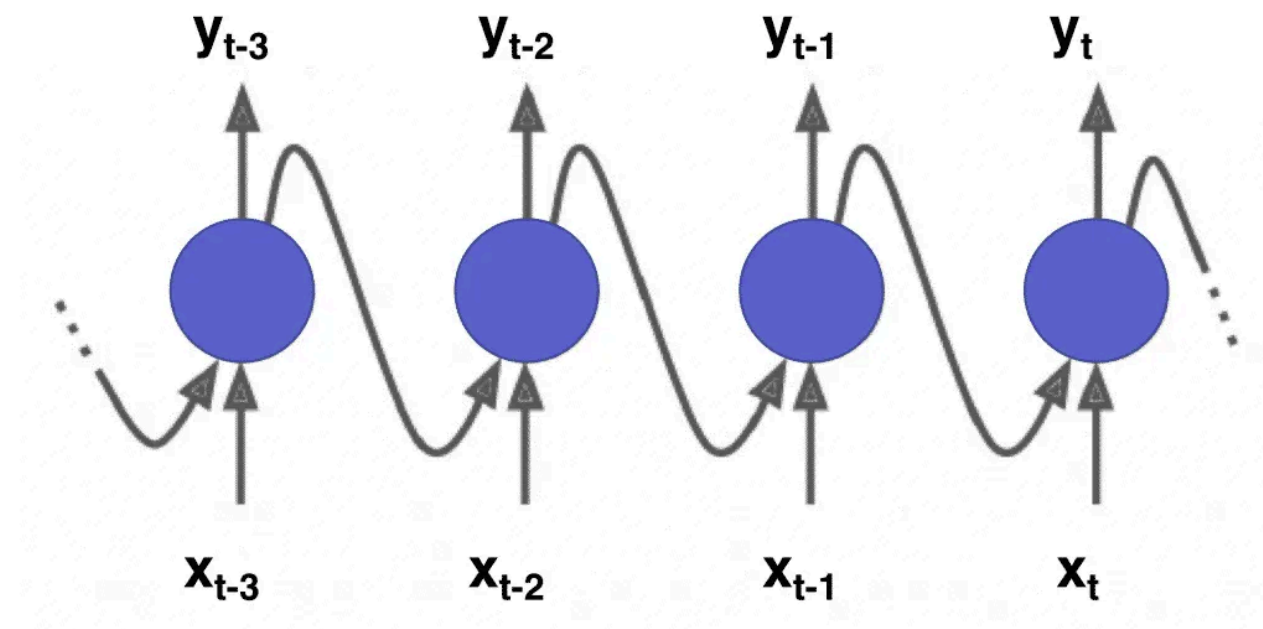


Figura 2.4 - Red de neurona recurrente.

La red neuronal recurrente tiene la limitación de los gradientes que se desvanecen y explotan . Se produce cuando los gradiente se vuelven extremadamente grandes o pequeños que dificultan el proceso de entrenamiento. Dicho problema es solucionado con la nueva arquitectura denominada Long-Short Term Memory (LSTM) que será detallada en la siguiente sección.

2.5. Long-Short Term Memory (LSTM)

Long-Short Term Memory (LSTM) es una arquitectura variante de las redes de neuronas recurrentes, están diseñadas específicamente para solucionar la limitación de los gradientes que se desvanecen y explotan, permitiendo el aprendizaje de dependencias a largo plazo en secuencia de datos.[19]

Un ejemplo de arquitectura de LSTM con dos capas, 3 entradas y 3 salidas es mostrado en el siguiente gráfico:

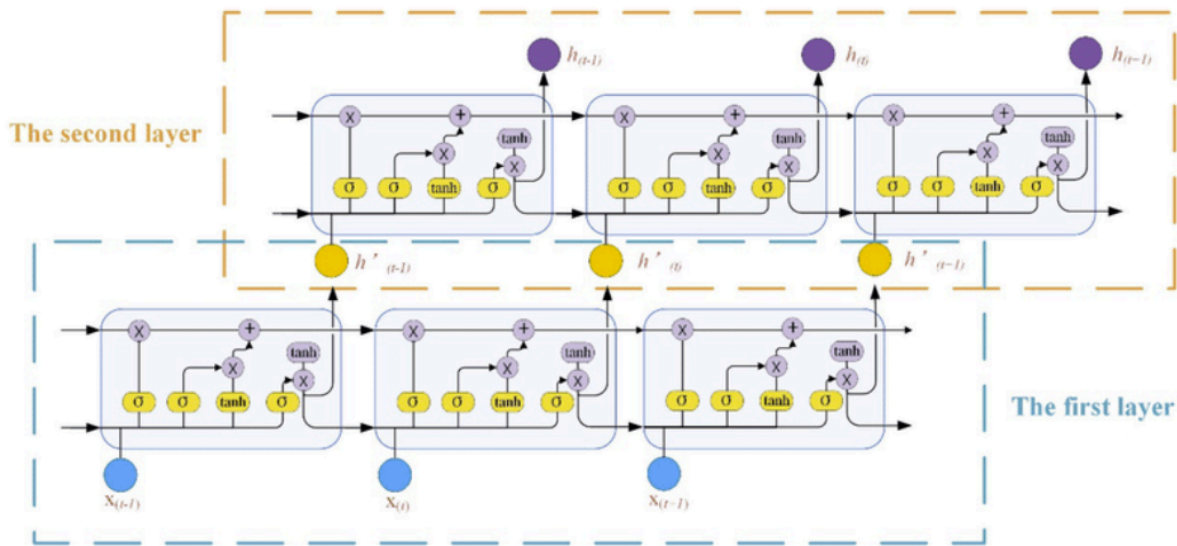


Figura 2.5 - LSTM de dos capas con tres entradas y salida.

2.6. Herramientas y framework

2.6.1. Python

Python es uno de los lenguajes de programación más populares en la actualidad debido a una serie de características particulares que han facilitado trabajar en campos en auge como la inteligencia artificial, el big data y el machine learning, entre otros [21]. Algunas de las características que hacen que Python sea tan popular son [22]:

- Este lenguaje no está especializado en un área específica, se puede utilizar para desarrollar cualquier tipo de programa.
- Multiplataforma: Python puede ser ejecutado en cualquier plataforma para la que exista un intérprete, su diseño no está centrado en una plataforma única.
- Sintaxis clara: la sintaxis de este lenguaje es simple y clara comparada con otros lenguajes comúnmente usados, Python no requiere de bloques de código delimitados

por caracteres como llaves o palabras claves. En Python se delimita usando una notación identidad mediante tabulaciones.

2.6.2. Librería Python.

Las principales bibliotecas utilizadas en la realización de este trabajo han sido:

- **Numpy:** Es una biblioteca esencial en el ecosistema de Python, reconocida por su capacidad de manipular y analizar arreglos multidimensionales de manera eficiente [39].
- **Pandas:** Es una biblioteca de Python ampliamente reconocida en el ámbito del análisis y manipulación de datos. Su enfoque se centra en proporcionar estructuras de datos y herramientas de alto rendimiento para llevar a cabo tareas habituales de limpieza, transformación y análisis de datos de manera eficiente[23].
- **Matplotlib:** Es una biblioteca ampliamente reconocida en el campo de la visualización de datos. Su principal objetivo es proporcionar herramientas flexibles y de alto nivel para la creación de gráficos estáticos, interactivos y personalizados[24].
- **Seaborn:** Es una biblioteca de visualización de datos que se basa en Matplotlib y está especialmente diseñada para crear gráficos estadísticos atractivos y con un alto nivel de información[25].
- **Scikit-learn (Sklearn):** Biblioteca de aprendizaje automático de código abierto y ampliamente utilizada en Python. Proporciona una amplia gama de algoritmos y Sergio Tapia Jorge 41 herramientas eficientes para realizar tareas de aprendizaje automático, como clasificación, regresión, agrupamiento y selección de características [26].
- **Dash:** Marco de trabajo que permite crear aplicaciones web y cuadro de mandos interactivos a través del lenguaje python.

3. CASO DE ESTUDIO

En el presente capítulo, se introduce en detalle el caso de estudio enmarcado en este Trabajo de Fin de Grado, se determina el problema a abordar, se establecen los principales objetivos y se plantean preguntas de investigación.

3.1. Comprensión del negocio.

El objetivo de esta sección es comprender el contexto y los objetivos del trabajo, así como las preguntas de negocio que se abordarán.

3.1.1. Digitalización del mercado financiero .

El proceso de digitalización de los mercados financieros empezó en el año 1969, donde la bolsa de Nueva York introdujeron sistemas que permitían realizar órdenes electrónicamente. Esta iniciativa provocó la expansión y la adopción de sistemas de trading electrónicos. Tras la

digitalización completa de la bolsa en principio de los años 2000, surgió el concepto de trading de alta frecuencia y trading algorítmico, donde las firmas privadas usaban algoritmos y computadores de altas prestaciones para realizar órdenes de manera autónoma [27].

3.1.2. Bitcoin y exchanges de criptomonedas.

En el año 2008 se creó la primera criptomoneda, Bitcoin, que utiliza criptografía para asegurar transacciones y verificar la transferencia de activos. La criptomoneda Bitcoin se caracteriza por la descentralización mediante la tecnología blockchain, seguridad de transacciones, transparencia de la transacciones, inmutabilidad de las transacciones.

En la actualidad para realizar compra y venta de criptomonedas, la forma más común para realizar dichas acciones es a través de los exchanges de criptomonedas. Generalmente los exchanges de criptomonedas ofrecen interfaces de programación (API) para ofrecer la opción de trading algorítmico para sus usuarios.

3.2. Objetivo de negocio.

Este trabajo tiene como objetivo principal la utilización de los datos históricos de Bitcoin para realizar predicciones sobre rentabilidades o precios a futuro de bitcoin.

3.2.1. Objetivos técnicos.

A continuación, se identifican los objetivos técnicos alineados con los objetivos de negocio.

- Extracción de los datos históricos del exchange de criptomonedas Binance.
- Aplicar técnicas de visualización de datos para identificar patrones.
- Modelado con LSTM.
- Despliegue de modelo en una aplicación Web, alimentado con datos vivos del exchange de criptomonedas Binance.

3.2. Compresión de los datos.

En esta fase se explicará el proceso de extracción de datos y se estudiarán los campos disponibles de los datos. Concretamente se extraerá el precio histórico de bitcoin y se procederá al análisis de esta.

3.2.1. Recopilación de los datos.

La fuente de datos proviene del exchange Binance “<https://www.binance.com/en/landing/data>”. En la propia página web de Binance te permite descargar datos de series temporales de cualquier criptomoneda. Concretamente se extraerá datos en bruto entre las fechas de 09/2019

a 04/2024 con una granularidad de una hora de las criptomonedas BTC/USDT, DOT/USDT, ETH/USDT y AVAX/USDT.

3.2.2. Características de los datos.

A continuación se muestra un ejemplo de los cuatro primeros registros del data frame bruto:

	open_time	open	high	low	close	volume	close_time	quote_volume	count	taker_buy_volume	taker_buy_quote_volume	ignore
0	1567296000000	9588.74	9630.00	9579.34	9617.06	546.960415	1567299599999	5.257943e+06	7428	261.131656	2.510480e+06	0
1	1567299600000	9617.06	9642.58	9612.08	9614.99	453.105152	1567303199999	4.362498e+06	5789	258.764082	2.491295e+06	0
2	1567303200000	9614.99	9616.00	9595.76	9605.78	458.004517	1567306799999	4.400751e+06	5485	321.355630	3.088171e+06	0
3	1567306800000	9606.12	9624.01	9605.78	9623.02	633.588749	1567310399999	6.092024e+06	5500	446.853542	4.296649e+06	0
4	1567310400000	9623.02	9630.21	9605.60	9614.46	542.260896	1567313999999	5.216076e+06	5536	368.470786	3.544535e+06	0

Figura 3.2.2 - Primeros registros del dataset en bruto

Cada uno de los campos tiene valores numéricos y con los siguientes significados:

- **Open_time:** Tiempo de apertura de la vela en formato de tiempo Unix.
- **Open:** Precio de apertura de la vela.
- **High:** Precio más alto de la vela.
- **Low:** Precio más bajo de la vela.
- **Close:** Precio de cierre.
- **volume:** Volumen de operaciones.
- **Close_time:** tiempo de cierre.
- **Count:** Número de operaciones.
- **taker_buy_volume:** volumen de operaciones que ha drenado liquidez.

4. ANÁLISIS DE LOS DATOS.

En el presente capítulo, se presenta en detalle el proceso seguido durante el desarrollo del caso de estudio enmarcado en este Trabajo de Fin de Grado. Este proceso se ha basado en una simplificación de la metodología CRISP-DM (Cross-Industry Standard Process for Data Mining), siguiendo sus fases principales, pero omitiendo algunas subtarefas debido a que esta metodología está pensada para ser seguida por un equipo de trabajo con diversos perfiles complementarios con los que no se ha contado para la realización de este trabajo.

4.1. Exploración inicial de los datos.

Para una mejor comprensión y familiarización con los datos que se van a usar durante el trabajo, se ha llevado a cabo una exploración inicial. Antes de comenzar con ella, se han realizado diferentes transformaciones en el Data Frame con el fin de consolidar los datos y mejorar su calidad.

4.1.1. Preparación del Dataframe.

Antes de comenzar con el análisis, se han realizado diferentes transformaciones en Dataframe con el fin de su posterior visualización y uso.

4.1.1.1. Fusión de los datos.

Los datos brutos extraídos se encuentran en la carpeta adjunta “./data” donde se encuentran cuatro carpetas con sus correspondientes ficheros .csv y cada una de ellas representa datos de series temporales mensuales con granularidad de 1 hora.

Para facilitar su uso se han fusionado los ficheros correspondientes a su criptomoneda en un único fichero .csv para su posterior uso. A continuación se detalla el proceso en código:

```
Python
import pandas as pd
import glob
import matplotlib.pyplot as plt
# Definir la dirección de los datos .csv
csv_files_path = './data/BTC_USDT/*.csv'

# Crear lista de ficheros .csv
csv_files = glob.glob(csv_files_path)

# Definir nuevos nombres a las columnas de los datos
column_names = ['open_time', 'open', 'high', 'low', 'close', 'volume',
                'close_time', 'quote_volume', 'count', 'taker_buy_volume',
                'taker_buy_quote_volume', 'ignore']

# Fusionar y concatenar los contenidos de los 56 .csv
dataframes = []
for file in csv_files:
    df = pd.read_csv(file, header=None, names=column_names)
    dataframes.append(df)
combined_df = pd.concat(dataframes, ignore_index=True)

# Guardar el fichero
combined_df.to_csv('./processed_data/combined_data/combined_bitcoin_data.csv',
index=False)

# Mostrar los primeros valores
combined_df.head()
```

Figura 4.1.1 - Código fusión de datos.

Unset

	open_time	open	high	low	close	volume	\
0	1567296000000	9588.74	9630.00	9579.34	9617.06	546.960415	
1	1567299600000	9617.06	9642.58	9612.08	9614.99	453.105152	
2	1567303200000	9614.99	9616.00	9595.76	9605.78	458.004517	
3	1567306800000	9606.12	9624.01	9605.78	9623.02	633.588749	
4	1567310400000	9623.02	9630.21	9605.60	9614.46	542.260896	

	close_time	quote_volume	count	taker_buy_volume	\
0	1567299599999	5.257943e+06	7428	261.131656	
1	1567303199999	4.362498e+06	5789	258.764082	
2	1567306799999	4.400751e+06	5485	321.355630	
3	1567310399999	6.092024e+06	5500	446.853542	
4	1567313999999	5.216076e+06	5536	368.470786	

	taker_buy_quote_volume	ignore
0	2.510480e+06	0
1	2.491295e+06	0
2	3.088171e+06	0
3	4.296649e+06	0
4	3.544535e+06	0

Figura 4.1.2 - Salida de ejecución del código fusión de datos.

Este proceso se aplica cuatro veces para cada conjunto de datos de criptomonedas que disponemos, concretamente conjunto de datos BTC/USDT, DOT/USDT, ETH/USDT y AVAX/USDT.

4.1.1.2. Comprobación de datos nulos.

Los precios de las acciones del mercado financiero generalmente tienen valores nulos en los fines de semana y días festivos. Esto se debe a que la bolsa cierra en dichas fechas. Para tratar valores nulos hay diversas aproximaciones, por ejemplo, tomar el precio del día anterior o la media de los precios entre el día anterior y el día siguiente. Generalmente para el precio de las criptomonedas no existen dichos problemas, ya que los exchanges de criptomonedas no cierra, pero si que se puede contemplar casos de que los servidores de los exchanges de criptomonedas sufran un alto tráfico y que provoque pausa de servicio temporal, como es el

caso del exchange Coinbase en Jueves 5 de Marzo de 2024 cuando bitcoin llegó a su nuevo máximo histórico de precio [28].

Procederemos a comprobar valores nulos en los datos en el siguiente código detallado:

```
Python
import pandas as pd
combined_csv_path = './processed_data/combined_data/combined_bitcoin_data.csv'
combined_df = pd.read_csv(combined_csv_path)

null_values = combined_df.isnull().sum()
print("Null values in each column:")
print(null_values)
```

Figura 4.1.1.2 - Código de comprobación de datos nulos.

```
Unset
Null values in each column:
open_time          0
open               0
high              0
low               0
close             0
volume            0
close_time        0
quote_volume      0
count             0
taker_buy_volume  0
taker_buy_quote_volume 0
ignore            0
```

Figura 4.1.1.3 - Salida de código de comprobación de datos nulos.

Como se puede ver no hay valores nulos. Por tanto, no será necesario tratar dichos valores y para el resto de datos tampoco tiene valores nulos.

4.1.2. Análisis exploratorio de datos.

Tras recolectar y entender los datos que se dispone, en esta sección se procede a analizar el precio de cierre de los datos recopilados. En esta sección se utilizan las librerías pandas, numpy, matplotlib y seaborn para el tratamiento y visualización de datos.

4.1.2.1. Tratamiento y fusión de datos.

```
Python
# Lista de criptomonedas que disponemos
cripto_list = ['BTC', 'AVAX', 'DOT', 'ETH']

# Lectura de datos local
BTC = pd.read_csv('./processed_data/combined_data/combined_bitcoin_data.csv')
AVAX = pd.read_csv('./processed_data/combined_data/combined_avax_data.csv')
DOT = pd.read_csv('./processed_data/combined_data/combined_dot_data.csv')
ETH = pd.read_csv('./processed_data/combined_data/combined_eth_data.csv')

# Guardar datos en una lista
cripto = [BTC, AVAX, DOT, ETH]
cripto_name = ["BTC", "AVAX", "DOT", "ETH"]
for c in cripto:
    c['open_time'] = pd.to_datetime(c['open_time'], unit='ms')
    c.set_index('open_time', inplace=True)

# Filtrar por fecha
start = '2020-09'
end = '2024-04'
for i in range(len(cripto)):
    cripto[i]=cripto[i][start:end]

for cripto_c, com_name in zip(cripto, cripto_name):
    cripto_c["cripto_name"] = com_name

# Concatenar los datos
df = pd.concat(cripto, axis=0)

df.tail(10)
```

Figura 4.1.2.1 - Código de tratamiento y fusión de datos.

	open	high	low	close	volume	close_time	quote_volume	count	taker_buy_volume	taker_buy_quote_volume	ignore	cripto_name
open_time												
2024-04-30 14:00:00	2993.60	3037.42	2985.01	3020.38	24778.4721	1714489199999	7.455754e+07	62215	12328.1627	3.709404e+07	0	ETH
2024-04-30 15:00:00	3020.38	3029.00	2985.16	2994.12	26680.0330	1714492799999	8.004333e+07	61643	12233.6121	3.670742e+07	0	ETH
2024-04-30 16:00:00	2994.11	2998.60	2945.00	2986.99	37975.1894	1714496399999	1.128950e+08	84519	18307.8533	5.443547e+07	0	ETH
2024-04-30 17:00:00	2987.00	2988.00	2975.07	2985.36	40111.6000	1714499999999	5.707040e+07	40025	8742.1402	2.510235e+07	0	ETH

Figura 4.1.2.2 - Salida de código de tratamiento y fusión de datos.

La función describe genera estadística descriptiva que permite tener una información más completa de los datos.

```
Python
BTC.describe()
```

Figura 4.1.2.3 - Código función describe.

	open	high	low	close	volume	close_time	quote_volume	count	taker_buy_volume	taker_buy_quote_volume	ignore
count	40861.000000	40861.000000	40861.000000	40861.000000	40861.000000	4.086100e+04	4.086100e+04	4.086100e+04	40861.000000	4.086100e+04	40861.0
mean	29356.240981	29496.310845	29208.306316	29357.492164	3724.749298	1.640942e+12	9.597291e+07	8.334776e+04	1842.674989	4.750147e+07	0.0
std	16978.841096	17069.156319	16882.312624	16979.266188	4846.474858	4.250014e+10	1.158856e+08	1.012882e+05	2417.305955	5.780768e+07	0.0
min	4131.230000	4610.000000	3782.130000	4130.640000	0.000000	1.567300e+12	0.000000e+00	0.000000e+00	0.000000	0.000000e+00	0.0
25%	13576.140000	13638.290000	13506.280000	13584.110000	1204.707170	1.604131e+12	2.503904e+07	2.697100e+04	588.745684	1.205467e+07	0.0
50%	27303.950000	27407.420000	27200.240000	27304.400000	2093.060814	1.640970e+12	5.948544e+07	4.728000e+04	1028.098150	2.914025e+07	0.0
75%	42054.540000	42237.240000	41851.500000	42057.260000	4180.188240	1.677744e+12	1.237612e+08	9.338600e+04	2061.455590	6.143485e+07	0.0
max	73577.350000	73777.000000	73215.140000	73577.350000	137207.188600	1.714522e+12	3.005634e+09	1.442097e+06	68611.450390	1.502977e+09	0.0

Figura 4.1.2.4 - Salida de código función describe.

Como se puede ver, tras filtrar por fechas entre 09/2020 y 04/24 tenemos en total 40861 de registros para cada criptomoneda.

4.1.2.1. Visualización de precio cierre.

A continuación se procede a detallar el código para el cálculo y visualización de la evolución del precio de cierre de las criptomonedas.

```
Python
# Visualizamos precio histórico de cada criptomoneda.
plt.figure(figsize=(15, 10))
plt.subplots_adjust(top=1.25, bottom=1.2)

for i, company in enumerate(cripto, 1):
    plt.subplot(2, 2, i)
    company['close'].plot()
    plt.ylabel('close')
    plt.xlabel(None)
```

```
plt.title(f"Precio cierre de {cripto_list[i - 1]}")
plt.tight_layout()
```

Figura 4.1.2.5 - Código función describe.

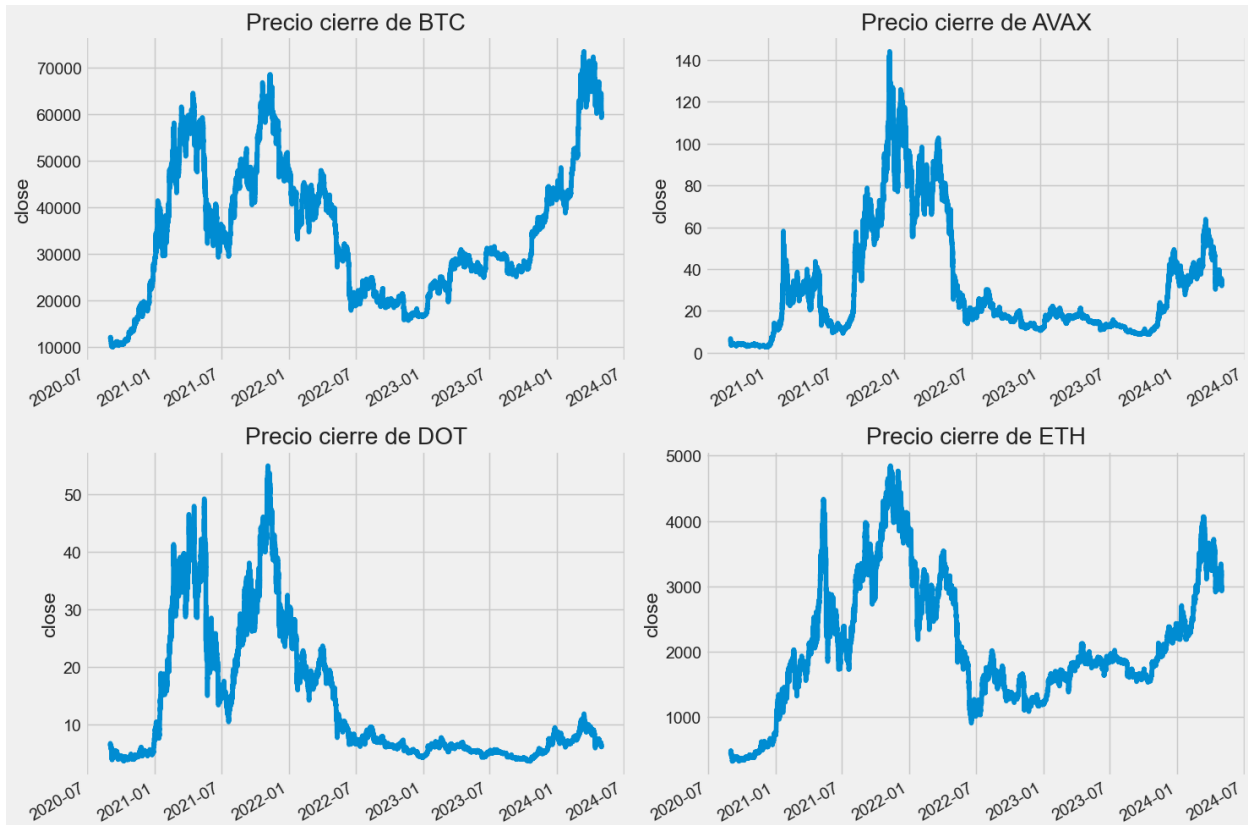


Figura 4.1.2.6 - Salida de código función describe.

Como se puede apreciar presenta cierta correlación entre ellas.

4.1.2.2. Visualización de volumen.

A continuación se procede a detallar el código para el cálculo y visualización del volumen de las criptomonedas.

```
Python
# Visualización de volumen
plt.figure(figsize=(15, 10))
plt.subplots_adjust(top=1.25, bottom=1.2)
start = '2023-09'
```



```

end = '2024-04'
for i, company in enumerate(cripto, 1):
    plt.subplot(2, 2, i)
    company[start:end]['volume'].plot()
    plt.ylabel('volume')
    plt.xlabel(None)
    plt.title(f"Sales volume for {cripto_list[i - 1]}")

plt.tight_layout()

```

Figura 4.1.2.6 - Código visualización volumen.

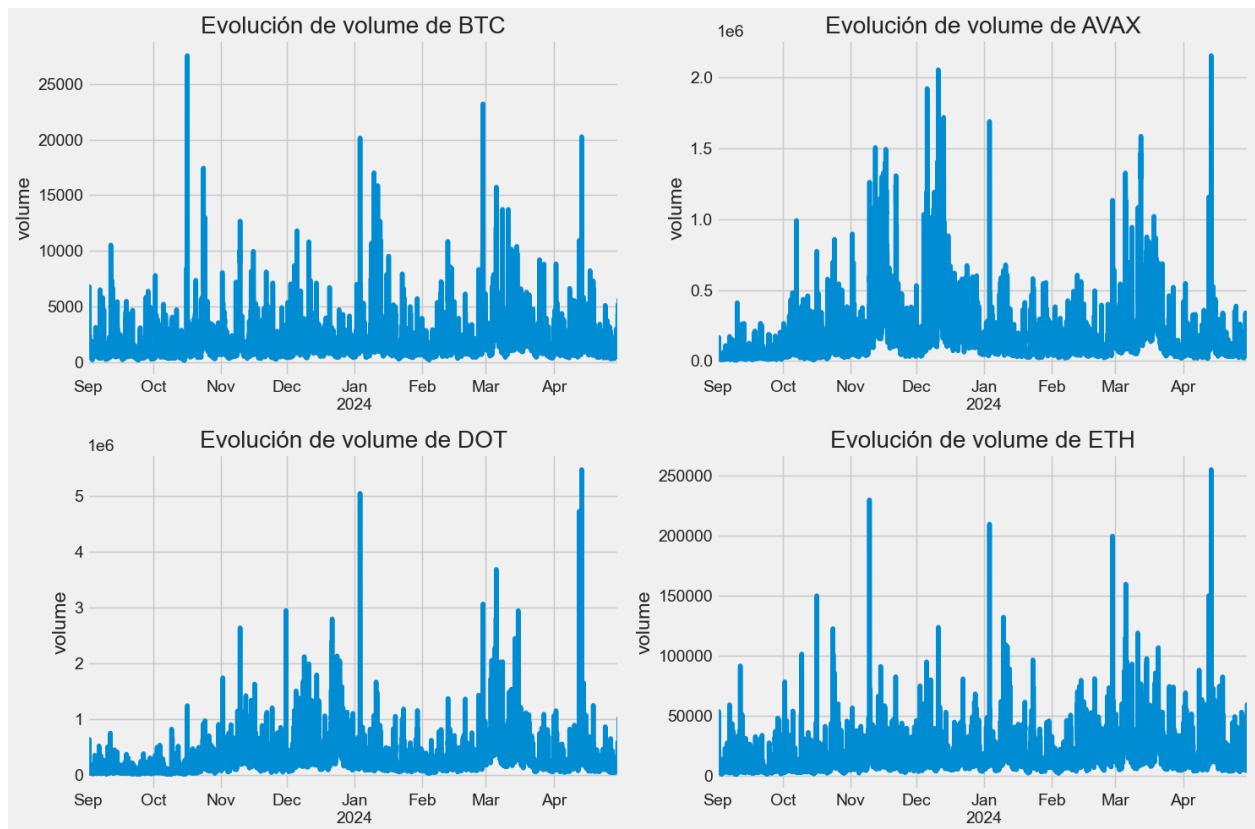


Figura 4.1.2.7 - Salida código visualización volumen.

4.1.2.3. Visualización Media móvil.

A continuación se procede a detallar el código para el cálculo y visualización de la media móvil con ventana temporal de 10, 20 y 50.

```

Python
ma_day = [10, 20, 50]
start = '2024-04-15'
end = '2024-04-30'
for ma in ma_day:
    for company in crypto:
        column_name = f"MA for {ma} hour"
        company[column_name] = company['close'].rolling(ma).mean()

fig, axes = plt.subplots(nrows=2, ncols=2)
fig.set_figheight(10)
fig.set_figwidth(15)

cripto[0][start:end][['close', 'MA de 10 horas', 'MA de 20 horas', 'MA de 50
horas']].plot(ax=axes[0,0])
axes[0,0].set_title('BTC')

cripto[1][start:end][['close', 'MA de 10 horas', 'MA de 20 horas', 'MA de 50
horas']].plot(ax=axes[0,1])
axes[0,1].set_title('AVAX')

cripto[2][start:end][['close', 'MA de 10 horas', 'MA de 20 horas', 'MA de 50
horas']].plot(ax=axes[1,0])
axes[1,0].set_title('DOT')

cripto[3][start:end][['close', 'MA de 10 horas', 'MA de 20 horas', 'MA de 50
horas']].plot(ax=axes[1,1])
axes[1,1].set_title('ETH')

fig.tight_layout()

```

Figura 4.1.2.9 - Código visualización Media móvil.

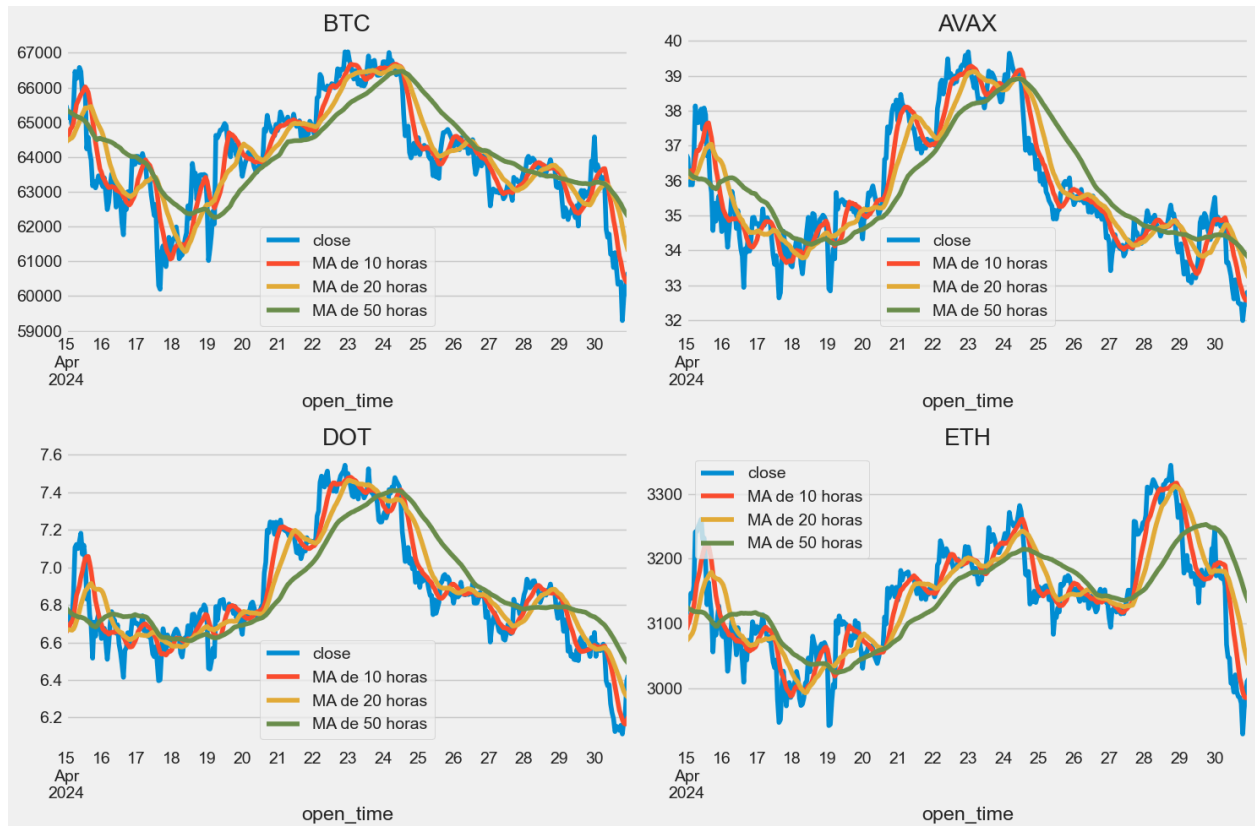


Figura 4.1.2.10 - Salida código visualización Media móvil.

Como se puede apreciar, las medias móviles entre distintas criptomonedas presentan cierta correlación.

4.1.2.4. Visualización de evolución de rentabilidad por horas.

A continuación se procede a detallar el código para el cálculo y visualización de la rentabilidad por horas.

```
Python
start = '2024-04-15'
end = '2024-04-30'
for company in crypto:
    company['Rentabilidad'] = company['close'].pct_change()

fig, axes = plt.subplots(nrows=2, ncols=2)
fig.set_figheight(10)
fig.set_figwidth(15)
```

```

cripto[0][start:end]['Rentabilidad'].plot(ax=axes[0,0], legend=True,
linestyle='--', marker='o')
axes[0,0].set_title('BTC')

cripto[1][start:end]['Rentabilidad'].plot(ax=axes[0,1], legend=True,
linestyle='--', marker='o')
axes[0,1].set_title('AVAX')

cripto[2][start:end]['Rentabilidad'].plot(ax=axes[1,0], legend=True,
linestyle='--', marker='o')
axes[1,0].set_title('DOT')

cripto[3][start:end]['Rentabilidad'].plot(ax=axes[1,1], legend=True,
linestyle='--', marker='o')
axes[1,1].set_title('ETH')

fig.tight_layout()

```

Figura 4.1.2.11 - Código visualización Rentabilidad.

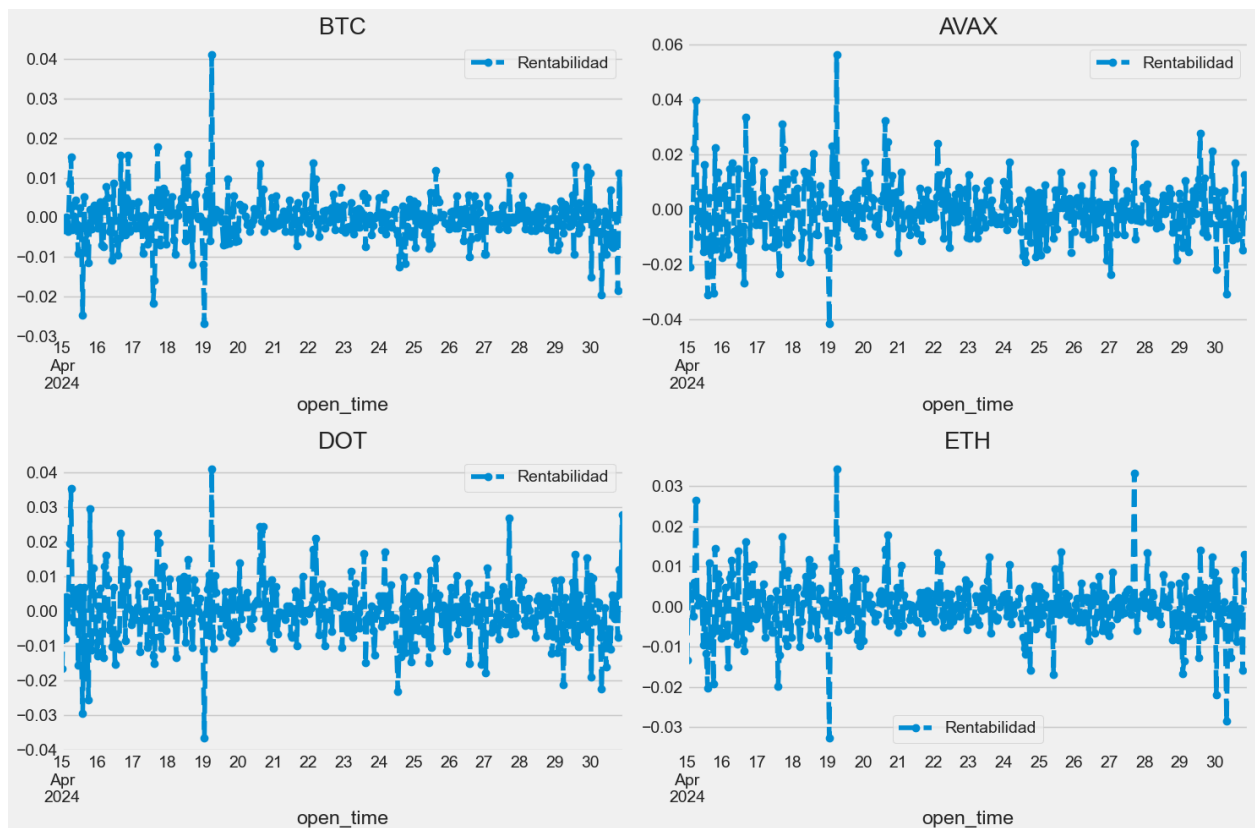


Figura 4.1.2.12 - Salida código visualización Rentabilidad.

4.1.2.5. Visualización de distribución de la rentabilidad.

A continuación se procede a detallar el código para el cálculo y visualización de la distribución de la rentabilidad.

```
Python
plt.figure(figsize=(12, 9))

for i, company in enumerate(cripto, 1):
    plt.subplot(2, 2, i)
    company['Rentabilidad'].hist(bins=50)
    plt.xlabel('Rentabilidad')
    plt.ylabel('Counts')
    plt.title(f'{cripto_name[i - 1]}')

plt.tight_layout()
```

Figura 4.1.2.13 - Código distribución de Rentabilidad.

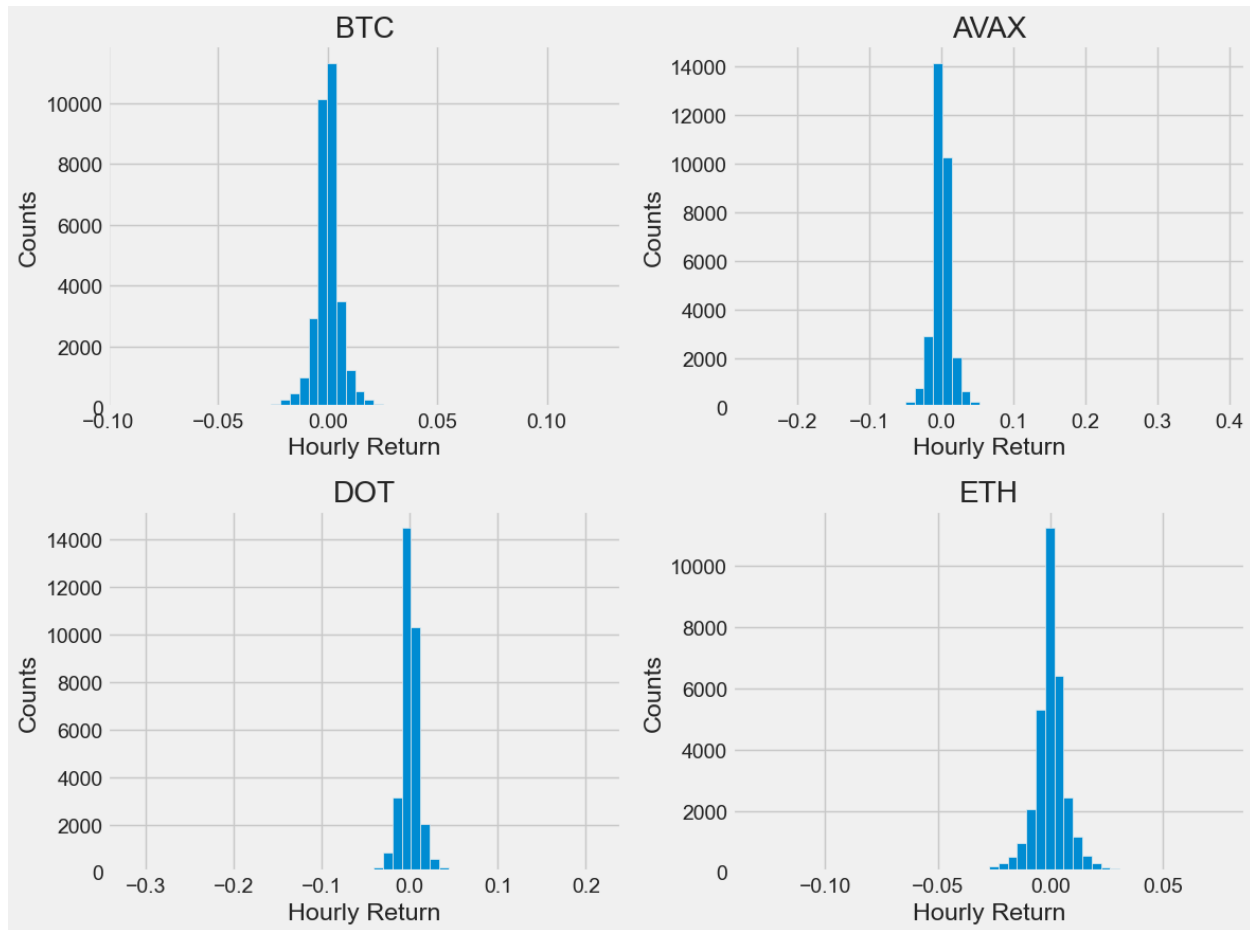


Figura 4.1.2.14 - Salida código distribución de Rentabilidad.

4.1.2.6. Correlación entre los precios.

A continuación se procede a detallar el código para el cálculo y visualización de la correlación del precio entre las criptomonedas.

```
Python
closing_prices = df.pivot_table(values='close', index=df.index,
                                columns='cripto_name')

# Calcular matriz de correlación
correlation_matrix = closing_prices.corr()

plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Matriz correlación')
plt.show()
```

Figura 4.1.2.15 - Código Correlación entre precios de criptomonedas.

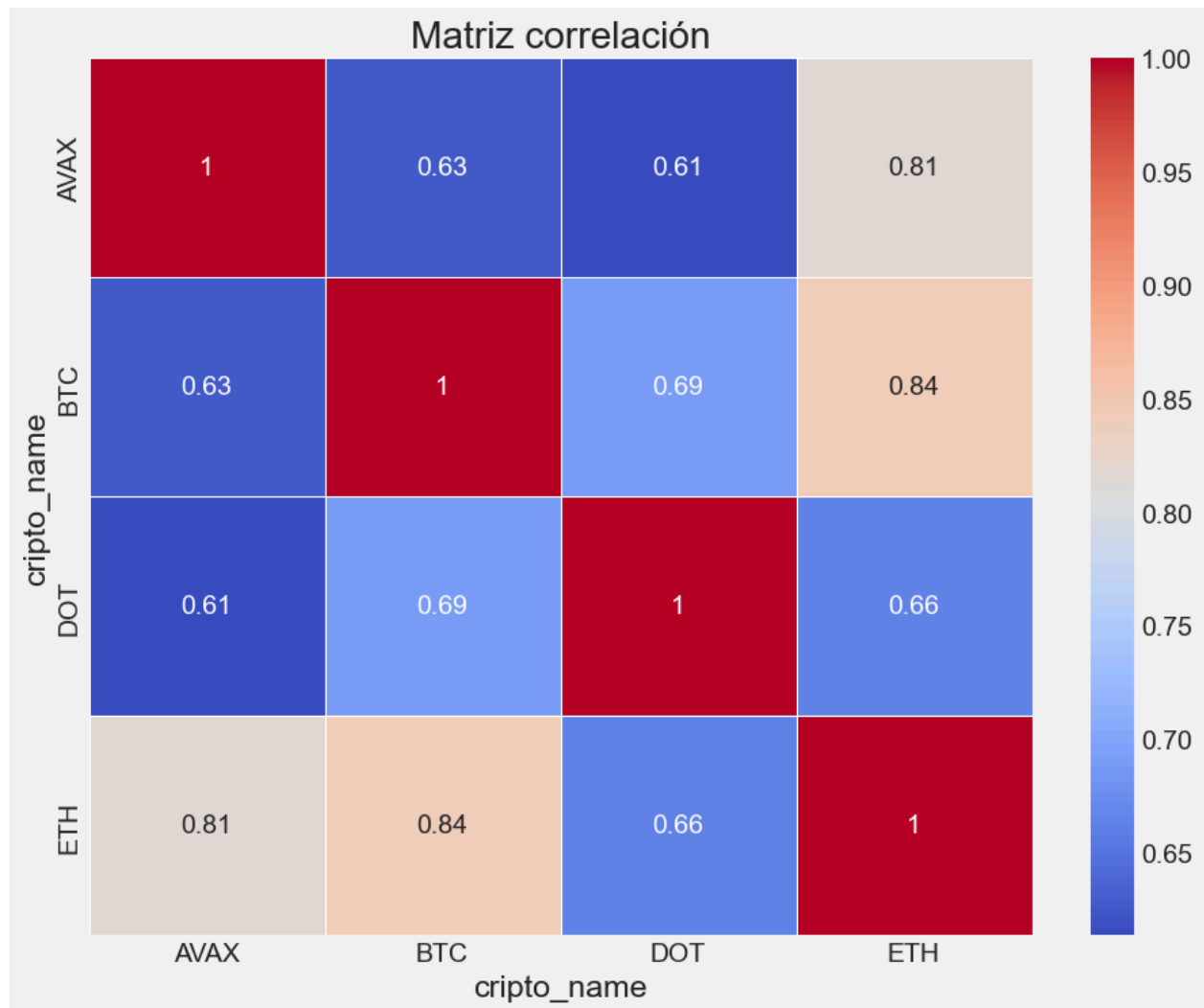


Figura 4.1.2.16 - Salida código Correlación entre precios de criptomonedas.

Como se puede observar, los precios de las criptomonedas están linealmente correlacionados.

4.1.2.7. Correlación de precio de BTC entre ETH.

A continuación se procede a detallar el código para el cálculo y visualización del gráfico de dispersión entre los precios de bitcoin y ethereum.

Python

```
# We'll use jointplot to compare the daily returns of Google and Microsoft
sns.jointplot(x='BTC', y='ETH', data=tech_rets, kind='scatter')
```

Figura 4.1.2.16 - Código correlación precio BTC-ETH.

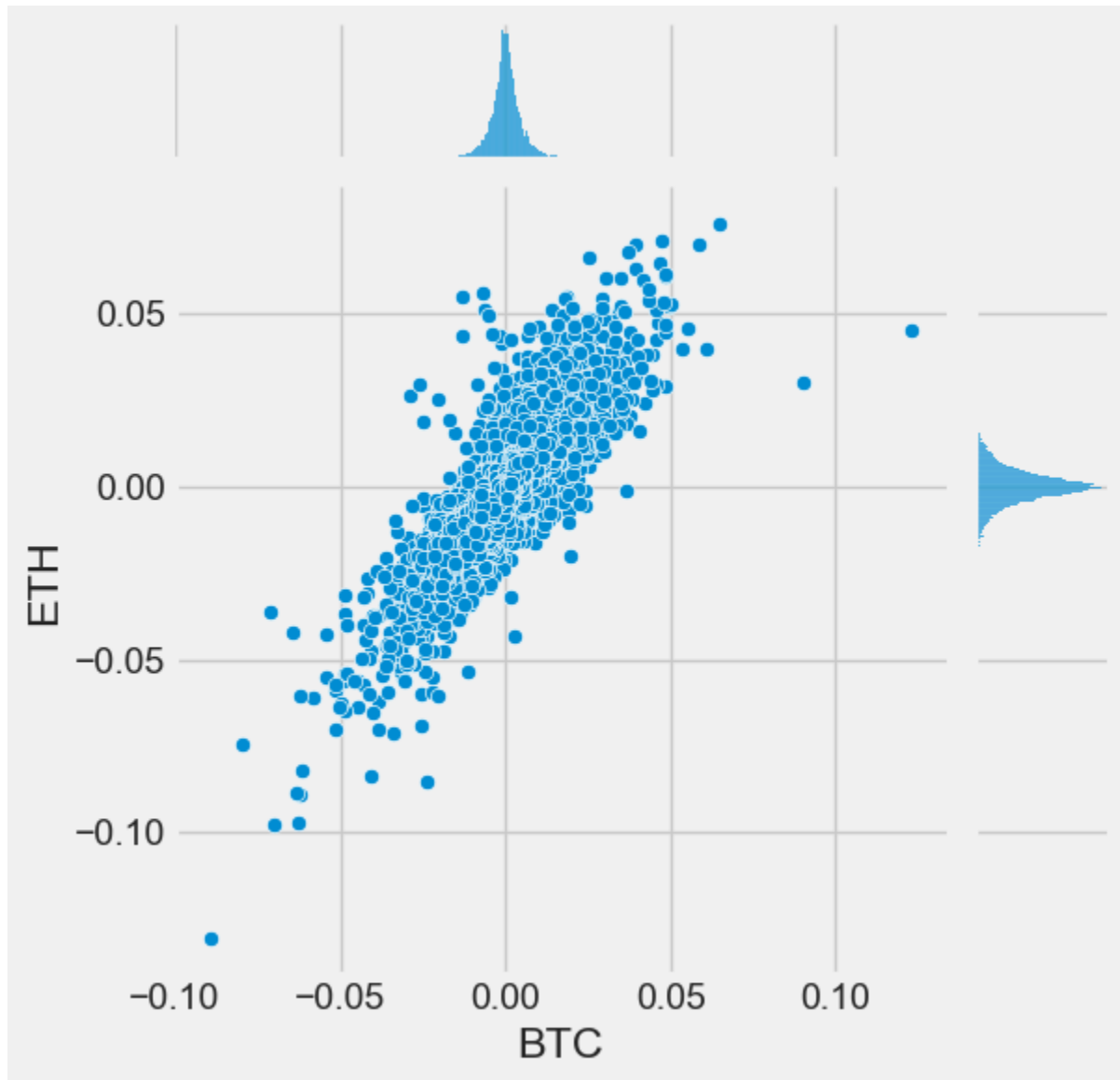


Figura 4.1.2.17 - Salida código correlación precio BTC-ETH.

Se puede observar que la concentración de puntos se puede ajustar a una función de regresión lineal. En la siguiente sección visualizamos la regresión entre los precios de las criptomonedas.

4.1.2.8. Correlación y regresión entre criptomonedas.

A continuación se procede a detallar el código para el cálculo y visualización de la correlación y regresión entre todas las criptomonedas.


```
Python  
sns.pairplot(tech_rets, kind='reg')
```

Figura 4.1.2.18 - Código correlación precio entre criptomonedas.

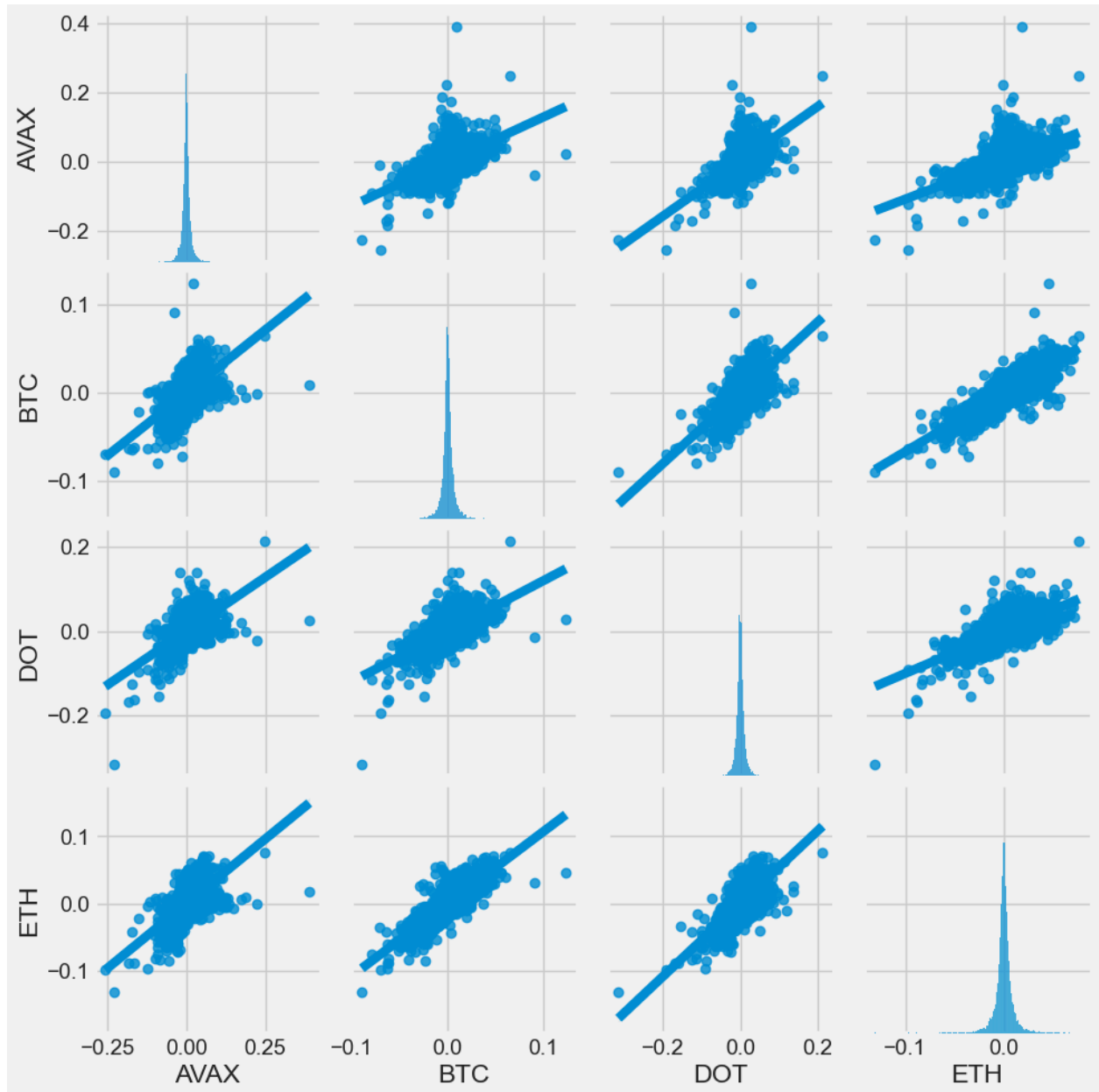


Figura 4.1.2.19 - Salida código correlación precio entre criptomonedas.

4.1.2.9. Matriz correlación de precio y rentabilidad.

A continuación se procede a detallar el código para el cálculo y visualización de la matriz de correlación entre el precio y rentabilidad de las criptomonedas..

```
Python
plt.figure(figsize=(12, 10))
plt.subplot(2, 2, 1)
sns.heatmap(tech_rets.corr(), annot=True, cmap='summer')
plt.title('Correlation of stock return')

plt.subplot(2, 2, 2)
sns.heatmap(closing_prices.corr(), annot=True, cmap='summer')
plt.title('Correlation of stock closing price')
```

Figura 4.1.2.20 - Código matriz correlación precio entre criptomonedas.

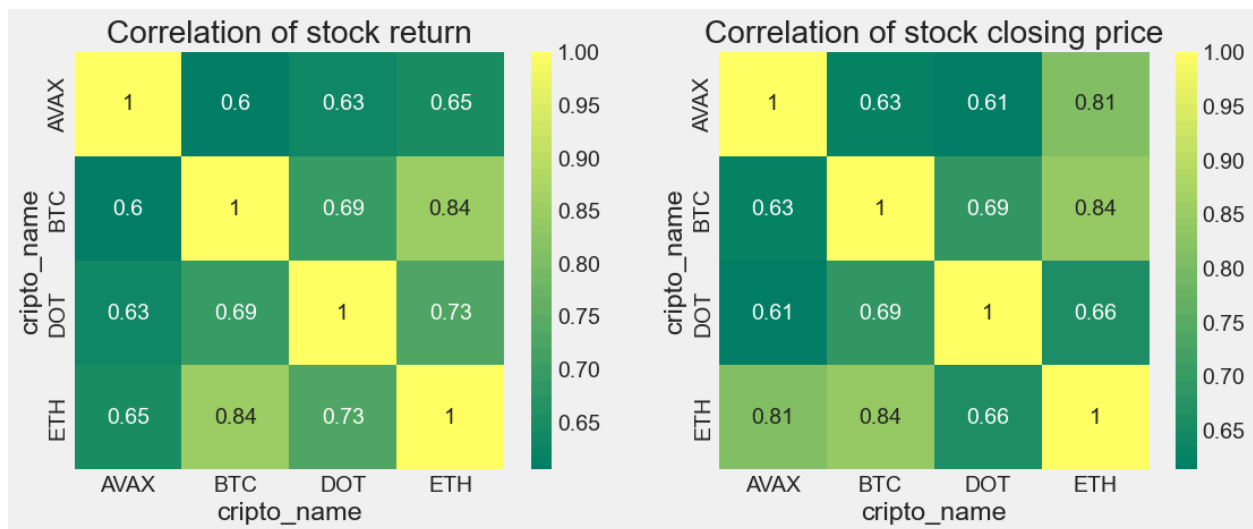


Figura 4.1.2.21- Salida código matriz correlación precio entre criptomonedas.

Con esto se puede sacar conclusión de que la evolución del precio y rentabilidad de las criptomonedas están altamente correlacionados.

4.1.2.10. Comparación evolución precio de criptomonedas.

A continuación se procede a detallar el código para la visualización para comparar la evolución de precio entre las criptomonedas para ver cómo se comporta la correlación entre ellas.

Python

```
# Normalizar datos entre 0 y 1 basando en precio máximo histórico y mínimo histórico
for i in range(len(crypto)):
    min_price = crypto[i]['close'].min()
    max_price = crypto[i]['close'].max()
    crypto[i]['normalized_close'] = (crypto[i]['close'] - min_price) / (max_price - min_price)

df = pd.concat(crypto, axis=0)
plt.figure(figsize=(14, 8))
for name in crypto_name:
    plt.plot(df[df['cripto_name'] == name].index, df[df['cripto_name'] == name]['normalized_close'], label=name, linewidth=0.8)

plt.title('Normalized Cryptocurrency Prices Over Time')
plt.xlabel('Date')
plt.ylabel('Normalized Price')
plt.legend()
plt.show()
```

Figura 4.1.2.22 - Código comparación de evolución de precios.

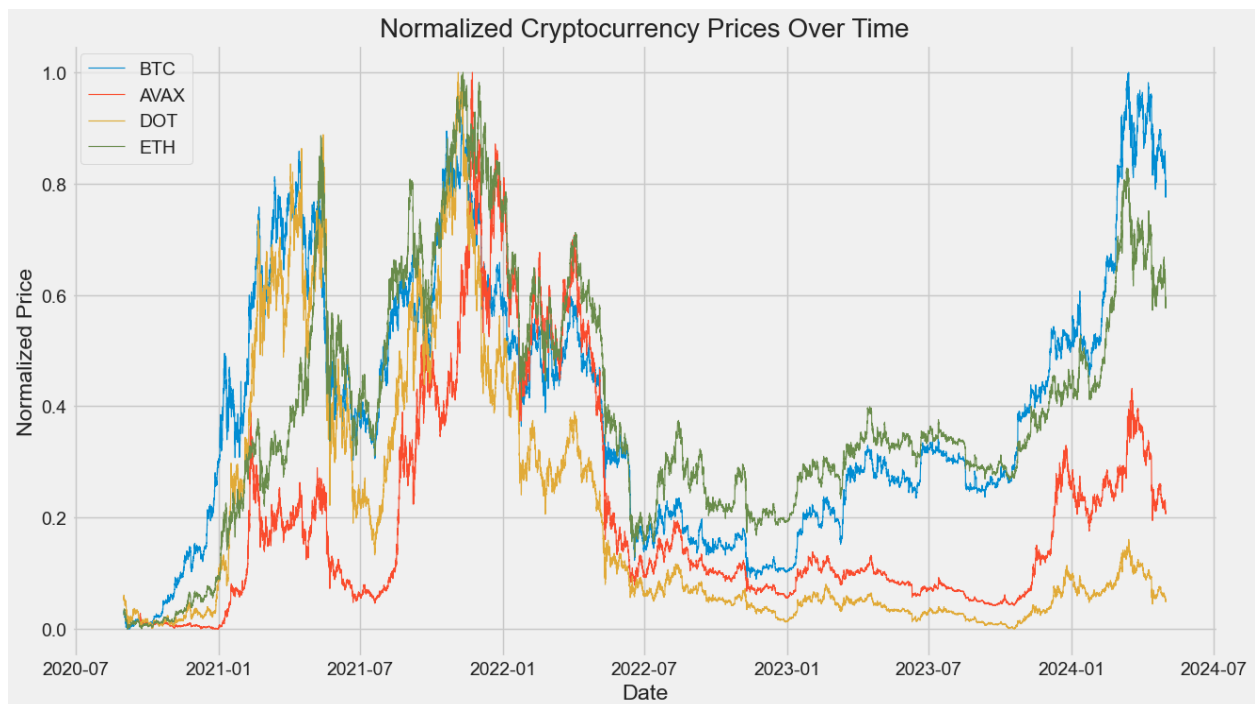


Figura 4.1.2.23 - Salida código comparación de evolución de precios.

4.2. Preparación de los datos y Selección de variables.

En esta sección, se realiza una preparación de los datos con el objetivo de obtener un conjunto de datos preparados y coherentes que sean aptos para el modelado.

Como se ha visualizado en la sección anterior en la figura 4.1.2.9. y 4.1.2.10. ,los precios de las criptomonedas están altamente correlacionados. Por tanto para el entrenamiento del modelo sólo se ha seleccionado el dataset de bitcoin, ya que este último tiene una mayo capitalización y los precios de las criptomonedas de una menor capitalización, generalmente, dependen del precio del Bitcoin [29].

De cara a los campos a utilizar, en esta iteración, se utilizará el precio de cierre de Bitcoin y indicadores derivados del precio de cierre como media móvil (20, 50, 100) y oscilador estocástico (20, 50, 100). A continuación se detalla en código el proceso tomado para la agregación de dichos datos:

Python

```
# Lectura de dataset
combined_df = pd.read_csv('./processed_data/combined_bitcoin_data.csv')

# Ventanas temporales a añadir a cada dataset.
moving_averages_window_size = [20,50,100]
rsi_window_size = [20,50,100]

# Fijar la fecha como índice
combined_df['open_time'] = pd.to_datetime(combined_df['open_time'], unit='ms')
combined_df.set_index('open_time', inplace=True)

# Calculo de media móvil
for n in moving_averages_window_size:
    combined_df['moving_average_{}'.format(n)] =
combined_df['close'].rolling(window=n).mean()

# Calculo oscilador estocástico
def calculate_rsi(data, window):
    delta = data.diff(1)
    gain = (delta.where(delta > 0, 0)).rolling(window=window).mean()
    loss = (-delta.where(delta < 0, 0)).rolling(window=window).mean()
    rs = gain / loss
    rsi = 100 - (100 / (1 + rs))
```

```

    return rsi

for n in rsi_window_size:
    combined_df['rsi_{}'.format(n)] = calculate_rsi(combined_df['close'], n)

# Guardar datos agregados en carpeta local.
combined_df.to_csv('./processed_data/aggregated_bitcoin_data.csv')

```

Figura 4.2.1 - Código preparación de datos.

A continuación se realiza una visualización de media móvil frente a oscilador estocástico.

```

Python
start = '2022-08-01'
end = '2022-09-01'
combined_df = combined_df[start:end]

plt.figure(figsize=(14, 14))

plt.subplot(2, 1, 1)
plt.plot(combined_df['close'], label='Closing Price')
for n in moving_averages_window_size:
    plt.plot(combined_df['moving_average_{}'.format(n)], label='{}-Day Moving
Average'.format(n))
plt.xlabel('Date')
plt.ylabel('Price')
plt.title('Bitcoin Closing Price and Moving Averages')
plt.legend()

plt.subplot(2, 1, 2)
for n in rsi_window_size:
    plt.plot(combined_df['rsi_{}'.format(n)], label='{}- RSI'.format(n))
plt.axhline(30, linestyle='--', alpha=0.5, color='red')
plt.axhline(70, linestyle='--', alpha=0.5, color='green')
plt.xlabel('Date')
plt.ylabel('RSI')
plt.title('Bitcoin Relative Strength Index (RSI)')
plt.legend()

```

```
plt.tight_layout()
plt.show()
```

Figura 4.2.2 - Código visualización media móvil y rsi.

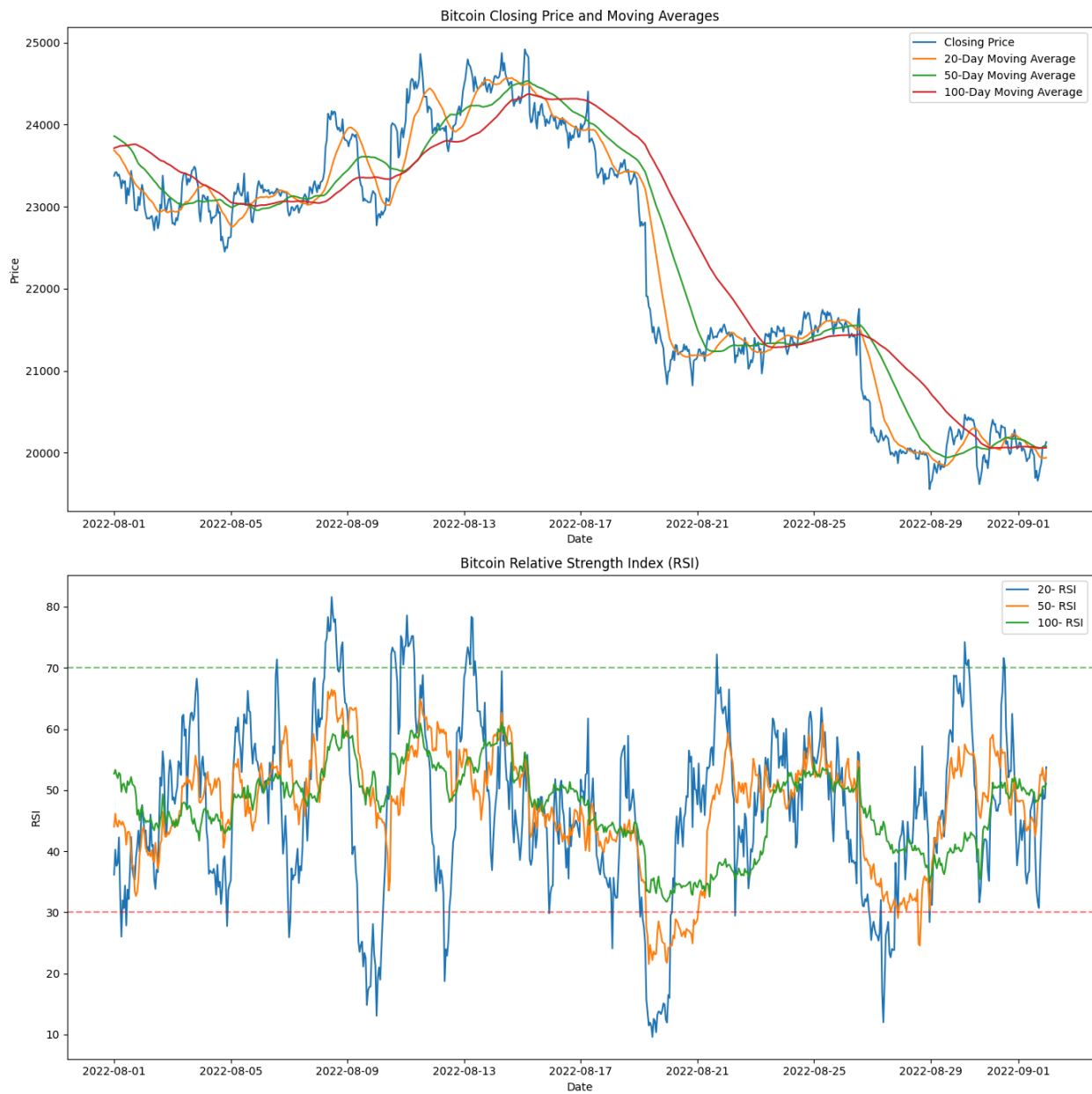


Figura 4.2.3 - Salida código visualización media móvil y rsi.

4.3. Modelado.

En esta sección se procede a desarrollar un modelo predictivo mediante el uso de redes de neuronas artificiales, concretamente LSTM. En esta fase se estudiará el uso de distintos grupos de datos para el modelado y los resultados de estos.

4.3.1. Modelado sin usar indicadores técnicos.

En este primer modelo se tomará el precio de cierre con una ventana temporal de 60 horas como entrada para el modelo LSTM. Por tanto, el modelo tiene 60 entradas y una salida, ambas de valores numérico y continuo.

A continuación se detalla en código los procesos tomados:

Python

```
# Leemos nuestro dato procesado de la sección anterior
data =
pd.read_csv('./processed_data/aggregated_data/aggregated_bitcoin_data.csv')
data = data[['open_time', 'close']]
data['open_time'] = pd.to_datetime(data['open_time'])
data.set_index('open_time', inplace=True)

# Preprocesamos dato, escalamos el datos entre 0-1
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(data)

# Escalamos el datos entre 0-1
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(dataset)

# Preparamos los datos en secuencias de ventana temporal 60
def create_sequences(data, time_steps=60):
    sequences = []
    labels = []
    for i in range(len(data) - time_steps):
        sequences.append(data[i:i+time_steps])
        labels.append(data[i+time_steps])
    return np.array(sequences), np.array(labels)

time_steps = 60
X, y = create_sequences(scaled_data, time_steps)
```

```

# Dividimos los datos de entrenamiento en train y test
train_size = int(len(X) * 0.8)
X_train, X_test = X[:train_size], X[train_size:]
y_train, y_test = y[:train_size], y[train_size:]

# Construimos nuestro modelo
model = Sequential()
model.add(LSTM(50, return_sequences=True, input_shape=(time_steps, 1)))
model.add(LSTM(50, return_sequences=False))
model.add(Dense(25))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mean_squared_error')
model.fit(X_train, y_train, batch_size=5, epochs=10)

```

Figura 4.3.1 - Código modelado.

A este punto el objetivo es minimizar el error mediante descenso gradiente y ajustar hiper parámetros y diferentes configuraciones del modelo para conseguir un error mínimo. Se han realizado las siguientes experimentaciones:

Capa 1	Capa 2	Capa 3	Capa 4	batch	epoch	loss	MSE
LTSM 60	Densa 1	-	-	5	5	4.92e-03	134537
LTSM 128	LTSM 60	Densa 1	-	5	5	3.15e-03	132685
LTSM 128	LTSM 60	Densa 25	Densa 1	5	5	2.88e-03	93452
LTSM 128	LTSM 60	Densa 25	Densa 1	5	10	6.21e-04	68536
LTSM 128	LTSM 60	Densa 25	Densa 1	10	10	4.48e-04	64809

Figura 4.3.2 - Tabla de experimentaciones del entrenamiento sin indicadores técnicos.

Se ha quedado con la última configuración por tener menor error cuadrático medio. A continuación se procede mostrar el resultado gráficamente con los datos de validación y la predicción a 60 horas hacia futuro tomando como entradas la predicción previa.

Python

```
# Predecimos el siguiente precio de los datos de entrenamiento
predictions = model.predict(X_test)
predictions = scaler.inverse_transform(predictions)

# Predecimos 60 secuencia hacia futuro
future_predictions = []
current_sequence = X_test[0]
for _ in range(len(predictions)):
    next_prediction = model.predict(current_sequence.reshape(1, time_steps, 1))
    future_predictions.append(next_prediction[0])
    current_sequence = np.append(current_sequence[1:], next_prediction, axis=0)
future_predictions = scaler.inverse_transform(future_predictions)

# Mostramos gráficamente
train = data[:train_size+time_steps]
valid = data[train_size+time_steps:]
valid['Predictions'] = predictions
valid['Future_forecast'] = future_predictions
plt.figure(figsize=(16,8))
plt.title('LSTM Model')
plt.xlabel('Date')
plt.ylabel('Close Price USD')
plt.plot(train['close'], label='Train')
plt.plot(valid[['close', 'Predictions', 'Future_forecast']],
label=['Validation', 'Predictions', 'Future_forecast'])
plt.legend(loc='lower right')
plt.show()
```

Figura 4.2.3 - Código visualización predicciones modelo.



Figura 4.2.4 - Salida código visualización predicciones modelo.

Como se puede apreciar la predicción de la serie de 60 precios a futuro no se ajusta al precio real de la validación.

4.3.2. Modelado usando indicadores técnicos.

En este segundo modelo se tomará una ventana temporal de 60 horas, pero a diferencia del primer modelo se añadirán los valores de oscilador estocástico y media móvil de 20, 50 y 60.

A continuación se detalla en código los procesos tomados:

```
Python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense

# Cargamos datos
data =
pd.read_csv('./processed_data/aggregated_data/aggregated_bitcoin_data.csv')[-20
00:]
```

```

data = data[['open_time', 'close', 'moving_average_20', 'moving_average_50',
'moving_average_100', 'rsi_20', 'rsi_50', 'rsi_100']]
data['open_time'] = pd.to_datetime(data['open_time'])
data.set_index('open_time', inplace=True)

# Preprocesamos los datos
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(data)

# Creamos secuencia de 60
def create_sequences(data, time_steps=60):
    sequences = []
    labels = []
    for i in range(len(data) - time_steps):
        sequences.append(data[i:i+time_steps])
        labels.append(data[i+time_steps][0])
    return np.array(sequences), np.array(labels)
time_steps = 60
X, y = create_sequences(scaled_data, time_steps)

# División de datos en entrenamiento y validación
train_size = int(len(X) * 0.8)
X_train, X_test = X[:train_size], X[train_size:]
y_train, y_test = y[:train_size], y[train_size:]

# Contrucción de modelo
model = Sequential()
model.add(LSTM(50, return_sequences=True, input_shape=(time_steps,
scaled_data.shape[1])))
model.add(LSTM(50, return_sequences=False))
model.add(Dense(25))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mean_squared_error')
model.fit(X_train, y_train, batch_size=10, epochs=10)

```

Figura 4.3.2.1 - Código modelado con indicadores técnicos.

A este punto el objetivo es minimizar el error mediante descenso gradiente y ajustar hiper parámetros y diferentes configuraciones del modelo para conseguir un error mínimo. Se han realizado las siguientes experimentaciones:

Capa 1	Capa 2	Capa 3	Capa 4	batch	epoch	loss	MSE
--------	--------	--------	--------	-------	-------	------	-----

LTSM 60	Densa 1	-	-	5	5	4.92e-03	151627
LTSM 128	LTSM 60	Densa 1	-	5	5	3.15e-03	125311
LTSM 128	LTSM 60	Densa 25	Densa 1	5	5	2.88e-03	92331
LTSM 128	LTSM 60	Densa 25	Densa 1	5	10	6.21e-04	68351
LTSM 128	LTSM 60	Densa 25	Densa 1	10	10	4.48e-04	44809

Figura 4.3.2.2 - Tabla de experimentaciones del entrenamiento con indicadores técnicos.

Se ha quedado con la última configuración por tener menor error cuadrático medio. A continuación se procede mostrar el resultado gráficamente con los datos de validación y la predicción a 60 horas hacia futuro tomando como entradas la predicción previa.

```

Python
# Crear Predicciones
predictions = model.predict(X_test)
scaler2 = MinMaxScaler(feature_range=(0, 1))
close_data = data[['close']]
scaled_data = scaler2.fit_transform(close_data)
predictions = scaler2.inverse_transform(predictions)

# Calculo de oscilador estocástico
def calculate_rsi(prices, period=14):
    gains = []
    losses = []
    for i in range(1, len(prices)):
        diff = prices[i] - prices[i - 1]
        if diff > 0:
            gains.append(diff)
            losses.append(0)
        else:
            gains.append(0)
            losses.append(abs(diff))
    avg_gain = sum(gains[:period]) / period
    avg_loss = sum(losses[:period]) / period
    rs = avg_gain / avg_loss
    rsi = 100 - (100 / (1 + rs))
    return rsi

# Calculo de la siguiente secuencia para predicción
def calcular_next_sequence(close_prices):
    rsi_20 = calculate_rsi(close_prices, 20)

```

```

    rsi_50 = calculate_rsi(close_prices, 50)
    rsi_100 = calculate_rsi(close_prices, 100)
    mv_20 = sum(close_prices[-20:]) / 20
    mv_50 = sum(close_prices[-50:]) / 50
    mv_100 = sum(close_prices[-100:]) / 100
    sequence =
[close_prices[-1][0],mv_20[0],mv_50[0],mv_100[0],rsi_20,rsi_50,rsi_100[0]]
    return sequence

# Predecir hacia futuro
X_test = X_test[:200]
future_predictions = []
current_sequence = X_test[0]
for _ in range(len(predictions)):
    next_prediction = model.predict(current_sequence.reshape(1, time_steps,
X_train.shape[2]))
    future_predictions.append(next_prediction[0])
    close_prices = []
    for n in current_sequence[1:]:
        close_prices.append(n[0])
    close_prices.append(next_prediction[0])
    sequence =calcular_next_sequence(close_prices)
    current_sequence = np.append(current_sequence[1:], [sequence], axis=0)
future_predictions = scaler2.inverse_transform(future_predictions)

# Mostramos gráficamente
train = data[:train_size+time_steps]
valid = data[train_size+time_steps:]
valid['Predictions'] = predictions
valid['Future_forecast'] = future_predictions
train = train[-120:]
valid = valid[:60]
plt.figure(figsize=(16,8))
plt.title('LSTM Model')
plt.xlabel('Date')
plt.ylabel('Close Price USD')
plt.plot(train['close'], label='Train')
plt.plot(valid[['close', 'Predictions','Future_forecast']],
label=['Validation', 'Predictions','Future_forecast'])
plt.legend(loc='lower right')
plt.show()

```

Figura 4.3.2.3 - Código visualización predicciones modelo.

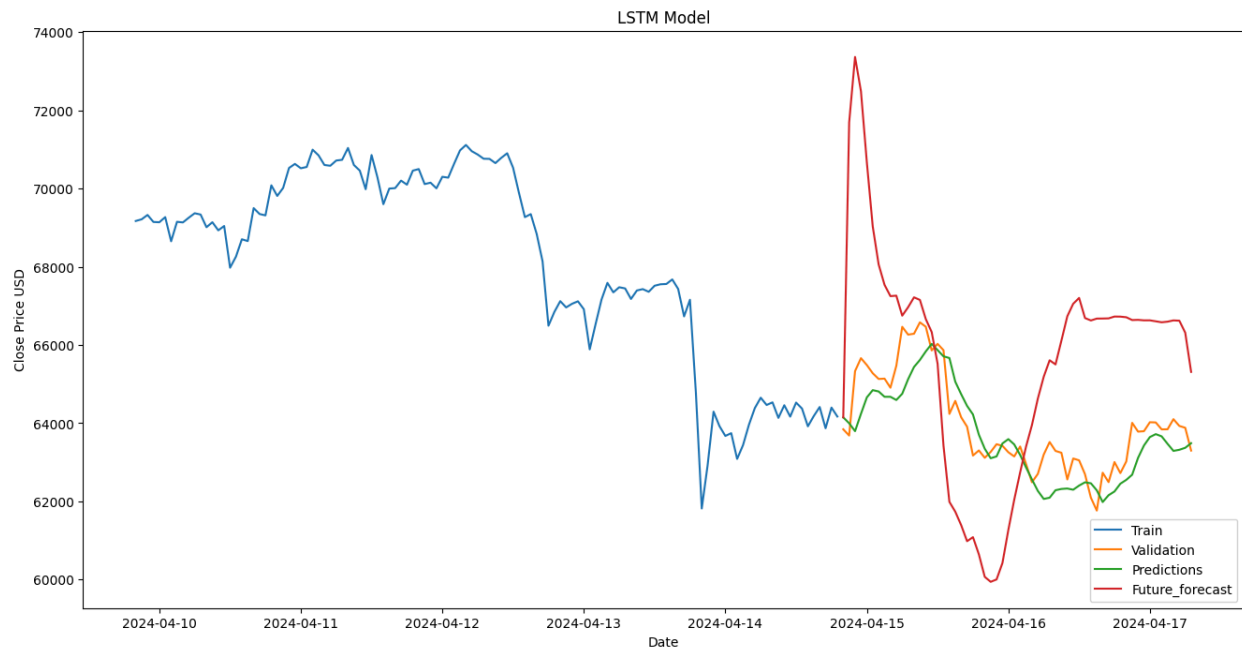


Figura 4.3.2.4 - Salida código visualización predicciones modelo con indicadores técnicos.

Como se puede apreciar, al usar indicadores técnicos como entrada para el entrenamiento se ajusta mejor al precio de validación en comparación con el modelo sin el uso de los indicadores técnicos.

5. Despliegue.

En esta sección se seleccionará el modelo adecuado para el despliegue en una página web. Los datos serán extraídos del exchange de criptomonedas Binance y se realizará predicciones en vivo.

5.1. Selección de modelo.

El criterio de selección se basará en el error cuadrático medio y el comportamiento gráfico de los modelos desarrollados. Como se ha estudiado en la sección anterior se aprecia que al entrenar el modelo sin indicadores técnicos presenta un error cuadrático medio de 64809 del mejor resultado conseguido frente a 44809 del modelo entrenado con indicadores técnicos.

Por otra parte, al visualizar el modelo gráficamente se puede apreciar en la figura 4.3.1 tiene comportamiento similar a una media móvil y en el caso del modelo con indicador técnico en la figura 4.3.2 se asemeja más al movimiento del precio de la criptomoneda.

Por tanto, en la siguiente sección procederemos al despliegue del modelo entrenado con indicador técnico para la predicción en vivo del precio.

5.2. Despliegue en web.

Para el despliegue del modelo se extrae datos en vivo del exchange de criptomonedas Binance cada minuto y se procede a la actualización del precio gráficamente en una interfaz web utilizando la biblioteca Dash. El código fuente de dicha aplicación web se puede encontrar en el fichero adjunto “./DASHAPP.py”, una vez instalado las dependencias requerida solo es necesario ejecutar dicho código y se creará un servicio web en un puerto local, accediendo a dicho servicio desde cualquier navegador se mostrará una interfaz se se puede apreciar en la siguiente figura:

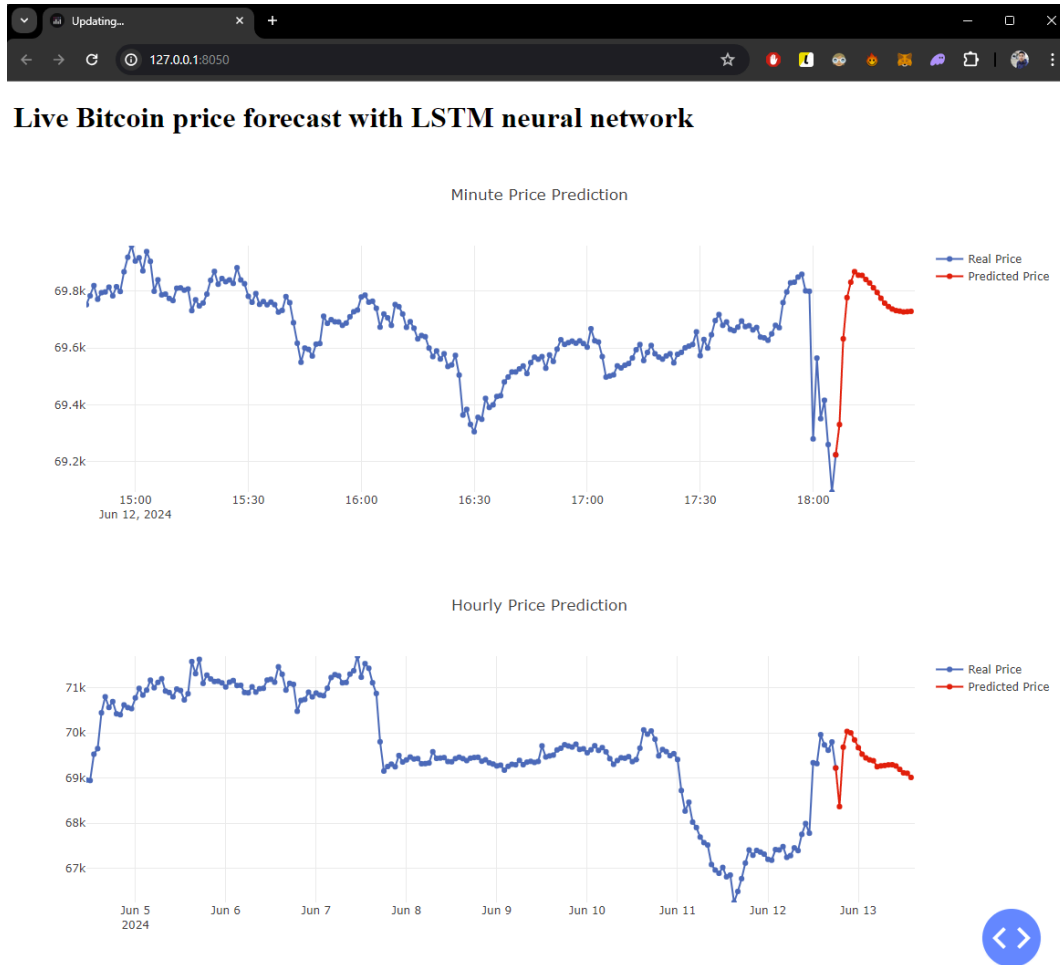


Figura 4.3.2.4 - Aplicación web con el modelo realizando predicciones en vivo.

Por fines didácticos también se ha incluido también las predicciones en minutos utilizado el mismo modelo entrenado con el propósito de ver la actualización del precio y la predicción en vivo. Dicho modelo es entrenado con series temporales de granularidad por horas, por tanto, las predicciones en minutos tendrá menor precisión que las de predicciones por horas.

6. CONCLUSIONES Y TRABAJOS FUTUROS.

6.1. Conclusiones.

En este trabajo fin de máster se ha llevado a cabo un análisis del precio a través de los datos históricos del precio de Bitcoin. Con fines académicos se ha conseguido el objetivo principal del estudio de la metodología de desarrollo CRISP-DM y realizar el ciclo completo de CRISP-DM para el problema de predicción de precios de Bitcoin.

6.2. Trabajos futuros.

El desarrollo de este trabajo es solo el paso inicial de un proyecto que puede ser mucho mayor, como se define en la metodología CRISP-DM, las fases de un proyecto de este tipo pueden repetirse y mejorarse de forma iterativa constantemente, durante el desarrollo de este trabajo, se ha completado una primera iteración de esta metodología. A continuación se presenta los pasos futuros que se puede tomar:

- Realizar una nueva iteración de esta investigación con una cantidad de datos mayor. Y realizar más iteraciones probando distintos conjuntos de datos e indicadores técnicos.
- Realizar predicciones de rentabilidad en lugar del precio.
- El precio de esta puede ser influido por diversos factores, por ejemplo, las noticias. Se puede ampliar el trabajo añadiendo un modelo que analice el sentimiento del mercado para las noticias de twitter.

7. ANEXO

En esta sección se explican los ficheros adjuntos del trabajo:

- **Carpeta “./data”** : Contiene datos de las cuatro criptomonedas elegidas para el desarrollo del trabajo. Está dividida en cuatro subcarpetas correspondientes al precio histórico de las cuatro criptomonedas en formato .csv.
- **Carpeta “./processed_data”**: Contiene los datos tratados, limpiados, fusionados y agregados.
- **“1_data_merge.ipynb”**: Código para fusión de datos.
- **“2_data_exploration.ipynb”**: Código para realizar análisis exploratorio de los datos.
- **“3_data_aggregation.ipynb”**: Código para agregar datos como media móvil y oscilador estocástico.
- **“4_modelling_without_technical_indicator.ipynb”**: Código para el modelado sin indicadores técnico.
- **“5_modelling_with_technical_indicator.ipynb”**: Código para el modelado con indicadores técnicos.
- **“Modelo_con_indicador_tecnico.h5”** : Mejor modelo entrenado con indicador técnico.
- **“Modelo_sin_indicador_tecnico.h5”**: Mejor modelo entrenado sin indicador técnico.
- **“DASHAPP.py”**: Aplicación web hecho en DASH para el despliegue del modelo en vivo.

8. BIBLIOGRAFÍA.

[1] «Crypto: rise in institutional interest» KPMG, 10 06 2022. [En línea]. Available: <https://kpmg.com/xx/en/home/insights/2022/12/crypto-rise-in-institutional-interest.html>. [Último acceso: 06 06 2024].

[2] «El Salvador cumple dos años de la adopción del bitcoin como moneda de curso legal» autor, 07 09 2023. [En línea]. Available: <https://efe.com/economia/2023-09-07/el-salvador-cumple-dos-anos-de-la-adopcion-del-bitcoin-como-moneda-de-curso-legal/>. [Último acceso: 02 06 2024].

[3] «How are Cryptocurrencies Regulated in the U.S. and the EU?» Dow Jones, 10 01 2023. [En línea]. Available: <https://www.dowjones.com/professional/risk/glossary/cryptocurrency/us-eu-regulation/>. [Último acceso: 06 06 2024].

[4] «Much anticipated, the US SEC approves bitcoin ETFs» Liz Lumley, 06 01 2023. [En línea]. Available: <https://www.thebanker.com/Much-anticipated-the-US-SEC-approves-bitcoin-ETFs-1705393382>. [Último acceso: 06 06 2024]

[5] «Bitcoin Documentation» Bitcoin. [En línea]. Available: <https://bitcoin.org/en/bitcoin-for-individuals>. [Último acceso: 06 06 2024]

[6] «Bitcoin's Price History» Investopedia, 26 03 2024. [En línea]. Available: link. [Último acceso: 06 06 2024]

[7] «CRISP-DM» www.datascience-pm.com. [En línea]. Available: <https://www.datascience-pm.com/>. [Último acceso: 06 06 2024]

[8] «Cross Industry Standard Process for Data Mining» Wikipedia, 10 01 2023. [En línea]. Available: https://es.wikipedia.org/wiki/Cross_Industry_Standard_Process_for_Data_Mining. [Último acceso: 06 06 2024]

[9] «Business Understanding» datascience-pm, 28 04 2024. [En línea]. Available: https://www.datascience-pm.com/crisp-dm-2/#I_Business_Understanding. [Último acceso: 06 06 2024]

[10] «Data Understanding» datascience-pm, 28 04 2024. [En línea]. Available: https://www.datascience-pm.com/crisp-dm-2/#II_Data_Understanding. [Último acceso: 06 06 2024]

[11] «Data processing» datascience-pm, 28 04 2024. [En línea]. Available: https://www.datascience-pm.com/crisp-dm-2/#I_Data_Processing. [Último acceso: 06 06 2024]

[12] «Modeling» datascience-pm, 28 04 2024. [En línea]. Available: https://www.datascience-pm.com/crisp-dm-2/#l_Modeling. [Último acceso: 06 06 2024]

[13] «Evaluation» datascience-pm, 28 04 2024. [En línea]. Available: https://www.datascience-pm.com/crisp-dm-2/#l_Evaluation. [Último acceso: 06 06 2024]

[14] «Deployment» datascience-pm, 28 04 2024. [En línea]. Available: https://www.datascience-pm.com/crisp-dm-2/#l_Deployment. [Último acceso: 06 06 2024]

[15] «Introducción a las series temporales» IBM, 23 08 2023. [En línea]. Available: <https://www.ibm.com/docs/es/spss-statistics/saas?topic=forecasting-introduction-time-series>. [Último acceso: 06 06 2024]

[16] «Análisis de Datos en el Sector Financiero » Fernando Fernandez, 05 02 2021.

[17] «Artificial neuron» Wikipedia, 12 01 2021. [En línea]. Available: https://en.wikipedia.org/wiki/Artificial_neuron. [Último acceso: 06 06 2024]

[18] «Red neuronal artificial» Wikipedia, 10 01 2021. [En línea]. Available: https://es.wikipedia.org/wiki/Red_neuronal_artificial. [Último acceso: 06 06 2024]

[19] «Recurrent Neural Networks (RNNs), Clearly Explained!!!» StatQuest, 10 01 2023. [En línea]. Available: https://www.youtube.com/watch?v=AsNTP8Kwu80&ab_channel=StatQuestwithJoshStarmer. [Último acceso: 06 06 2024]

[20] «Recurrent Neural Networks (RNNs), Clearly Explained!!!» StatQuest, 10 01 2023. [En línea]. Available: <https://youtu.be/YCzL96nL7j0?si=2uD3pT89qs0kfCRz>. [Último acceso: 06 06 2024]

[21] Santander Universidades, «Python: qué es y por qué deberías aprender a utilizarlo,» 09 04 2021. [En línea]. Disponible en: <https://www.becassantander.com/es/blog/python-que-es.html>. [Último acceso: 01 05 2023].

[22] «Qué es Python,» 19 11 2003. [En línea]. Available: <https://desarrolloweb.com/articulos/1325.php>. [Último acceso: 01 05 2023].

[23] «pandas,» [En línea]. Disponible en: <https://pandas.pydata.org/>. [Último acceso: 13 06 2023].

[24] «matplotlib,» [En línea]. Disponible en: <https://matplotlib.org/>. [Último acceso: 13 06 2023].

[25] «seaborn,» [En línea]. Disponible en: <https://seaborn.pydata.org/>. [Último acceso: 13 06 2023].

[26] «scikit-learn,» [En línea]. Disponible en: <https://scikit-learn.org/stable/>. [Último acceso: 13 06 2023].

[27] «The establishment of NASDAQ as the first electronic stock market is detailed in various financial history texts» *Electronic Exchanges*.

[28] «Coinbase crashed as Bitcoin soared to record high» Sam Shedden, 07 03 2024. [En línea]. Available: <https://readwrite.com/coinbase-crashed-as-bitcoin-soared-to-record-high/>. [Último acceso: 06 06 2024]

[29] «Dependent on BTC: Why Altcoins Follow Bitcoin?» Swapzone, 08 09 2023. [En línea]. Available: <https://medium.com/coinmonks/dependent-on-btc-why-altcoins-follow-bitcoin-2023-update-ec5bdf981cd7>. [Último acceso: 06 06 2024]

[30] «Dash documentation» Dash. [En línea]. Available: <https://dash.plotly.com/>. [Último acceso: 06 06 2024]