PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA

PRONÓSTICO DEL RANGO DEL PRECIO DEL ETHER EMPLEANDO ANÁLISIS DE SENTIMIENTO COMO VARIABLE EXÓGENA

Tesis para obtener el título profesional de Ingeniero Industrial

AUTOR

Carhuachin Gamboa, Paulo Cesar

ASESOR:

Rodríguez Anticona, Miguel Ángel

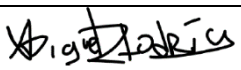
Lima, septiembre, 2024

Informe de Similitud

Yo, Miguel Ángel Rodríguez Anticona
docente de la Facultad de Ciencias e Ingeniería de la Pontificia Universidad Católica del Perú, asesor(a) de la tesis/el trabajo de investigación titulado Pronóstico del rango del precio del Ether empleando análisis de sentimiento como variable exógena del autor Paulo César Carhuachin Gamboa dejo constancia de lo siguiente:

- El mencionado documento tiene un índice de puntuación de similitud de 18% Así lo consigna el reporte de similitud emitido por el software *Turnitin* el 21/08/2024.
- He revisado con detalle dicho reporte y la Tesis o Trabajo de Suficiencia Profesional, y no se advierte indicios de plagio.
- Las citas a otros autores y sus respectivas referencias cumplen con las pautas académicas.

Lugar y fecha: Lima 21 Agosto 2024

Apellidos y nombres del asesor / de la asesora: Rodríguez Anticona Miguel Ángel	
DNI:46402997	Firma 
ORCID: https://orcid.org/0000-0003-2522-3422	

Resumen

En el año 2009, surgió la primera criptomoneda conocida como Bitcoin. Sin embargo, no fue sino hasta el año 2020, con el inicio de la pandemia global, que este tipo de activos digitales despertó interés en un segmento de la población. Esto se debió a que las restricciones impuestas para prevenir la propagación del COVID-19 hicieron que los inmigrantes buscaran alternativas a los métodos tradicionales para enviar remesas, ya que estos se volvieron inaccesibles. Como resultado, las criptomonedas se convirtieron en una opción viable para la transferencia de valor. Además, provocó una disminución de los empleos físicos, dejando a muchas personas desempleadas y con tiempo libre. Esto incentivó la exploración de nuevas formas de generar ingresos. Como consecuencia, inversores *retail* comenzaron a considerar las criptomonedas como una vía para preservar su capital en un entorno marcado por la inflación o la falta de confianza en las políticas gubernamentales. Otro grupo de individuos percibe estos activos digitales como una oportunidad para aumentar su patrimonio, dado que, a diferencia de las acciones, las criptomonedas carecen de un valor intrínseco y se basan principalmente en especulaciones de mercado. El presente estudio se enfoca en determinar el rango del precio del Ether con el fin de poder proporcionar un modelo cuantitativo que complemente el análisis técnico. Se emplearán métodos tradicionales de pronóstico de series de tiempo, así como modelos de ML que incluirán la variable exógena sentimiento del mercado. La obtención de esta variable se llevará a cabo a través de la extracción de los titulares de noticias de la plataforma financiera Investing mediante web scraping. La intención es evaluar si el sentimiento transmitido por estos titulares de noticias tiene un impacto significativo en el valor de la criptomoneda. Finalmente, se presentará un nuevo enfoque conocido como predicción conformal que permite cuantificar la incertidumbre y proporcionar un intervalo de confianza a

los pronósticos realizados por el modelo de ML. Se concluye que la predicción conformal robustece el modelo de ML seleccionado.

“All models are wrong, but some are useful”

George Box

INDICE GENERAL

Pág.

INDICE DE TABLAS

INDICE DE FIGURAS

INTRODUCCIÓN

Las primeras cuatro décadas del Internet trajeron el correo electrónico, la web, redes sociales, la web móvil, *big data*, *cloud computing* y el Internet de las cosas. Ello redujo las barreras para el acceso a la información y permitió colaborar e intercambiar información en tiempo real desde distintos espacios geográficos. Además, introdujo nuevos medios de comunicación, formas de venta y entretenimiento.

Sin embargo, todavía no podemos confiar los unos a los otros para realizar transacciones en línea, sin la necesidad de un intermediario como un banco o el gobierno ya que estos mismos recogen nuestra información e invaden nuestra privacidad para obtener beneficios comerciales o seguridad nacional.

Incluso con el Internet, según Don Tapscott y Alex Tapscott (2016), su estructura de costos excluye alrededor de 2,500 millones de personas del sistema financiero mundial. A pesar de las promesas de un mundo con igualdad de condiciones, los beneficios económicos y políticos han demostrado ser asimétricos. La tecnología no crea prosperidad más de lo que destruye la privacidad de las personas que la utilizan. No obstante, en los últimos años el término *blockchain* se ha puesto de moda. Según Molano (2019), "el *blockchain* es un conjunto de tecnologías que permiten la transferencia de un valor o activo de un lugar u otro, sin ayuda de terceros". Asimismo, Don Tapscott y Alex Tapscott (2016), complementan dicha definición pues señalan que las *blockchains* permiten enviar dinero de forma directa y segura, sin pasar por un banco o compañía de tarjetas de crédito. En lugar de hablar del Internet de las cosas, ahora se trata del internet del valor o dinero pues la información se mueve a través del mundo de manera instantánea, sin embargo, la transferencia de dinero de un país a otro es caro, lento y poco fiable. De acuerdo con Ripple (2017), "cada año alrededor del mundo se realizan pagos transfronterizos valorizados en 180 billones de dólares cuyo costo asociado es de más de 1,7 billones". El presente estudio se va a centrar en el primer aplicativo desarrollado con la tecnología *blockchain*, el bitcoin o de forma más general, las criptomonedas. Las criptomonedas cada vez están ganando aceptación por parte las personas. A pesar de países como China que se oponen a esta ya que prohibió las transacciones de criptomonedas (Bloomberg News, 2021). Personas influyentes como Jerome Powell, presidente del Sistema de la Reserva Federal, ha declarado que Estados Unidos no tiene ninguna intención de prohibir el bitcoin y las criptomonedas (McShane, 2021). Por su parte, Laurence Fink, Chairman y CEO de Blackrock, y Jamie Dimon, Chairman y CEO de JP Morgan están apostando por

estas. A nivel nacional, de acuerdo con Google trends, en los últimos 5 años ha aumentado el interés por el término criptomonedas (Ver Figura1).

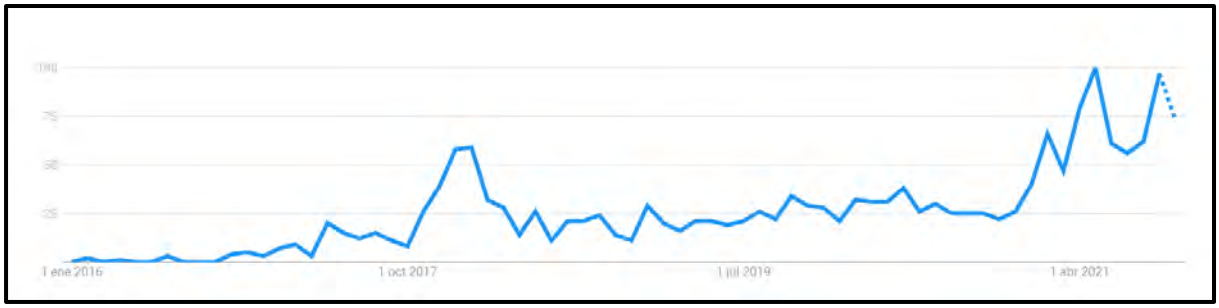


Figura 1: Interés por parte de los peruanos en criptomonedas 2016-01-01 a 16-10-2021
Fuente: (Google Trends 16-10-2021)

En el capítulo 1 se realizará un estudio de la tecnología detrás de las criptomonedas, los métodos utilizados para el análisis de texto. Asimismo, las herramientas analíticas requeridas para poder llevar a cabo modelos predictivos.

En el capítulo 2 se hace una revisión bibliográfica de trabajos de investigación cuyo tema central es similar al del presente trabajo de estudio. El motivo de ello es poder recolectar información ya sea de sugerencias hechas por los autores, así como las APIs utilizadas para poder obtener la información mediante *Web Scraping*, etc.

En el capítulo 3 se aborda a la cadena de bloques (o *blockchain*) Ethereum, ya que la criptomoneda en estudio (Ether) es la criptomoneda nativa de la plataforma, por lo tanto, cualquier actualización o mejora en la red conllevaría a un alza del precio de esta.

En el capítulo 4 se va a presenta el flujo que se va a seguir para poder aplicar metodologías tradicionales como modernas para pronosticar el rango del precio del Ether. Asimismo, se presentarán pruebas estadísticas para poder validar las condiciones requeridas.

En el capítulo 5 se detallarán las conclusiones y recomendaciones para futuros trabajos relacionados con el pronóstico de series de tiempo en general.

CAPÍTULO 1. MARCO TEÓRICO

En este capítulo se desarrollarán conceptos para poder entender cómo funciona la tecnología *blockchain* y sus aplicaciones. Se enfatizará en una de sus aplicaciones más destacadas las criptomonedas para que el lector pueda entender el desarrollo del presente estudio. Posterior a ello, se abordará el concepto de análisis de sentimiento donde se expondrá casos aplicados al mercado bursátil. Por último, se describirán las series de tiempo y sus componentes principales para poder realizar un modelo predictivo.

1.1. Blockchain

1.1.1. Sistemas distribuidos

Para poder entender cómo funciona la tecnología *blockchain* es necesario tener claro qué es un sistema distribuido pues la *blockchain* es uno, de manera más específica se trata de un sistema distribuido descentralizado.

Los sistemas distribuidos son un paradigma informático donde dos o más nodos trabajan en conjunto de manera coordinada con el fin de lograr un resultado en común.

Según Bashir (2018), los nodos pueden ser honestos, defectuosos o maliciosos. Asimismo, aquellos que presentan un comportamiento arbitrario son conocidos como nodos bizantinos[1].

De acuerdo con Drescher (2017), existen dos formas de organizar los nodos de manera distribuida y centralizada. La diferencia fundamental recae en que en un sistema centralizado solo el nodo central esta conectado a los demás nodos a diferencia del distribuido donde todos los nodos están enlazados entre sí, pero ningún nodo conecta con todos (Ver Figura 2).

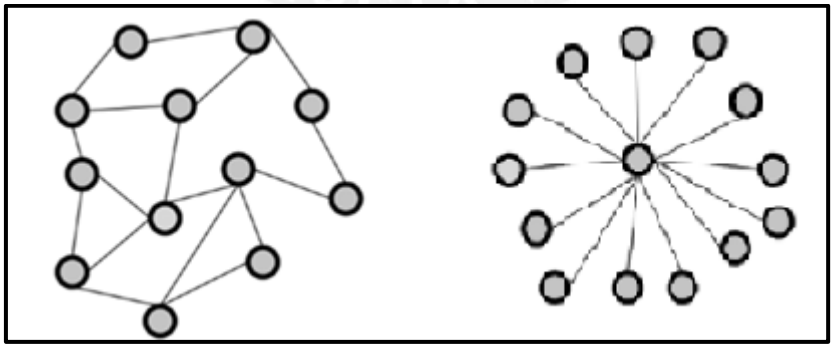


Figura 2: Sistema distribuido(izquierda) vs Sistema centralizado (derecha)
Fuente: (Drescher 2017:11)

[1] Nodo bizantino es un nodo malicioso que puede mentir o engañar intencionalmente a otros nodos de la red

El principal reto en el diseño de un sistema distribuido es la coordinación entre nodos y la tolerancia a las fallas pues incluso si alguno de los nodos se vuelve malicioso o alguno de los enlaces se rompe, el sistema distribuido debe tolerar esto y seguir estando operativo, sin problemas para lograr el resultado deseado.

Los sistemas distribuidos son tan difíciles de diseñar que han sido objeto de estudio durante muchos años, se han propuesto varios algoritmos y consensos para superar los problemas de coordinación entre nodos y la tolerancia a las fallas. Sin embargo, un teorema conocido como el teorema CAP ha sido demostrado y enuncia que ningún sistema distribuido puede tener coherencia, disponibilidad y tolerancia a la partición simultáneamente.

- **Coherencia** garantiza que todos los nodos en el sistema distribuido tengan una única y última copia de datos.
- **Disponibilidad** significa que un sistema está en funcionamiento ya que cumple las tres funciones: está accesible para su uso, está recibiendo solicitudes y responde con datos, sin ningún tipo de fallo, cuando es requerido.
- **Tolerancia a la partición** garantiza que, si un grupo de nodos falla, el sistema distribuido siga funcionando correctamente.

Para lograr la tolerancia a las fallas, se utiliza la replicación y la coherencia se consigue mediante algoritmos de consenso para garantizar que todos los nodos tienen la misma copia de los datos, esto también es conocido como *State machine replication*. Según Mougayar (2016), un *State* puede ser definido como información almacenada en un punto específico en el tiempo entonces una *State machine* es un dispositivo que recuerda el estado de algo en un instante de tiempo determinado. La *blockchain* es básicamente un método para poder replicar un *State machine*.

Tabla 1 Ventajas y desventajas de los sistemas distribuidos

Ventajas	Desventajas
Mayor potencia de cálculo	Gastos de coordinación
Reducción de costos	Gastos de comunicación
Mayor fiabilidad	Dependencia de las redes
Capacidad de crecimiento natural	Alta complejidad del programa

Fuente: (Drescher 2017:12-13)

1.1.2. Mecanismos de consenso

Un mecanismo de consenso es un conjunto de pasos que son tomados por todos los nodos o la mayoría de estos para alcanzar un estado o valor propuesto. Hay varios requisitos que deben cumplirse para proporcionar los resultados deseados de un mecanismo de consenso. A continuación, se exponen los requisitos en mención:

- **Acuerdo. -** todos los nodos honestos deciden el mismo valor.
- **Terminación. –** todos los nodos honestos terminan el proceso de consenso y llegan a una decisión
- **Validez. –** el valor acordado por todos los nodos honestos debe ser el mismo que el valor inicial propuesto por al menos un nodo honesto.
- **Tolerancia a las fallas. –** el algoritmo de consenso debe ser capaz de ser ejecutado en la presencia de nodos bizantinos.
- **Integridad. –** cada nodo toma una decisión única cada vez por cada ciclo de consenso.

Existe varios tipos de mecanismo de consenso, a continuación, se describirán los más comunes:

- **Basado en tolerancia a faltas bizantinas. –** se basa en un esquema simple de nodos que envían mensajes firmados, cuando se alcanza un cierto número de mensajes recibidos entonces se llega a un acuerdo.
- **Basado en líder. –** este tipo de mecanismo requiere que todos los nodos compitan por la lotería de la elección del líder y el nodo que la gana propone un valor final.

1.1.3. Dinero electrónico

Al igual que entender los conceptos de sistemas distribuidos es necesario para entender la *blockchain*, conocer sobre dinero electrónico también es esencial para apreciar la primera y más conocida aplicación de *blockchain*: el bitcoin, o de forma más general las criptomonedas.

Los problemas fundamentales que deben abordarse en los sistemas de *e-cash* son la contabilidad y el anonimato. En 1983, David Chaum abordó ambos problemas en su artículo *Blind Signatures for Untraceable Payments*, donde introdujo dos operaciones criptográficas las firmas ciegas y la compartición de secretos. Según Bashir (2018), las firmas ciegas permiten firmar un documento sin necesidad de verlo y la compartición de secretos permite detectar el uso del mismo token dos veces comúnmente llamado *double spending* (o doble gasto). Dwork y Naor (1992), introdujeron la idea de utilizar rompecabezas computacionales o *pricing functions*[2] para evitar el spam por correo electrónico. En 1997, Adam Back introduce el concepto de *hashcash* lo cual conllevo al uso de funciones *hash* de computación como *Proof of Work*. Posterior a ello, Dai (1998), introdujo *b-money,* el cual propuso la idea de crear dinero mediante la resolución de rompecabezas computacionales como el *hashcash*. Se basa en una red *peer-to-peer* donde cada nodo mantiene su lista de transacciones. Szabo (1998), propuso una idea similar a la de *b-money* denominada *Bit Gold*, la cual consistía en resolver rompecabezas computacionales para minar monedas digitales. Asimismo, Finney (2015), introdujo el concepto de monedas criptográficas combinando las ideas de rompecabezas computacionales y *b-money*, no obstante, aún requería de una autoridad central de confianza.

[2] Pricing functions hace referencia a las funciones duras que requieren ser calculadas para poder acceder a un recurso.

Los esquemas propuestos por los autores descritos anteriormente presentaban múltiples problemas, además de ser inviables. Estos problemas iban desde la ausencia de una solución clara entre los desacuerdos entre nodos, la necesidad de un tercero confiable y un sellado de tiempo confiable.

En 2008, se desarrolló un *whitepaper* que trataba sobre un sistema de efectivo electrónico usuario a usuario, el cual fue publicado de forma anónima por el seudónimo de Satoshi Nakamoto, que formó la base de la tecnología *blockchain.* En el aborda la problemática de contar con un tercero confiable, instituciones financieras, para el comercio electrónico ya que el costo de mediación de estas encarecía el costo de transacción. Para ello, Nakamoto (2008), propuso un modelo de *Proof of Work* en lugar del modelo basado en confianza, ya que el segundo contaba con algunas falencias. Por un lado, no era posible realizar transacciones completamente no reversibles dado que las instituciones financieras no pueden evitar medir disputas. Por otro lado, se debía aceptar un cierto porcentaje de fraude en las transacciones.

Por tal motivo se plantea la necesidad de un sistema de pagos electrónicos que brinde la misma seguridad que realizar la transacción en persona con dinero físico con bajos costos asociados y sin la necesidad de un tercero confiable.

Nakamoto (2008), define una moneda electrónica como una cadena de firmas digitales donde cada dueño transfiere la moneda al próximo al firmar un hash[3] de la transacción previa y la clave pública del próximo dueño y agregando estos al final de la moneda. Para que se logre esto sin un tercero confiable son requerimientos indispensables que las transacciones sean anunciadas públicamente y se necesita un sistema de participantes que estén de acuerdo con un único historial del orden en el que las monedas fueron recibidas con el fin de que un beneficiario pueda verificar las firmas para verificar la cadena de propiedad.

Sin embargo, no fue hasta 2009 donde se realizó la primera implementación de una criptomoneda denominada bitcoin. Por primera vez, se resolvió el problema de consenso distribuido en una red sin un tercero confiable. Utilizaba la criptografía asimétrica junto a *hashcash* como *Proof of Work (PoW)* para proporcionar un método seguro, controlado y descentralizado de minar monedas digitales. La clave para resolver los problemas antes mencionados fue el uso de una lista ordenada de bloques compuesta por transacciones y criptográficamente aseguradas por el mecanismo *PoW*.

[3] Hash es una función matemática que convierte una entrada de longitud arbitraria en una salida cifrada de longitud fija.

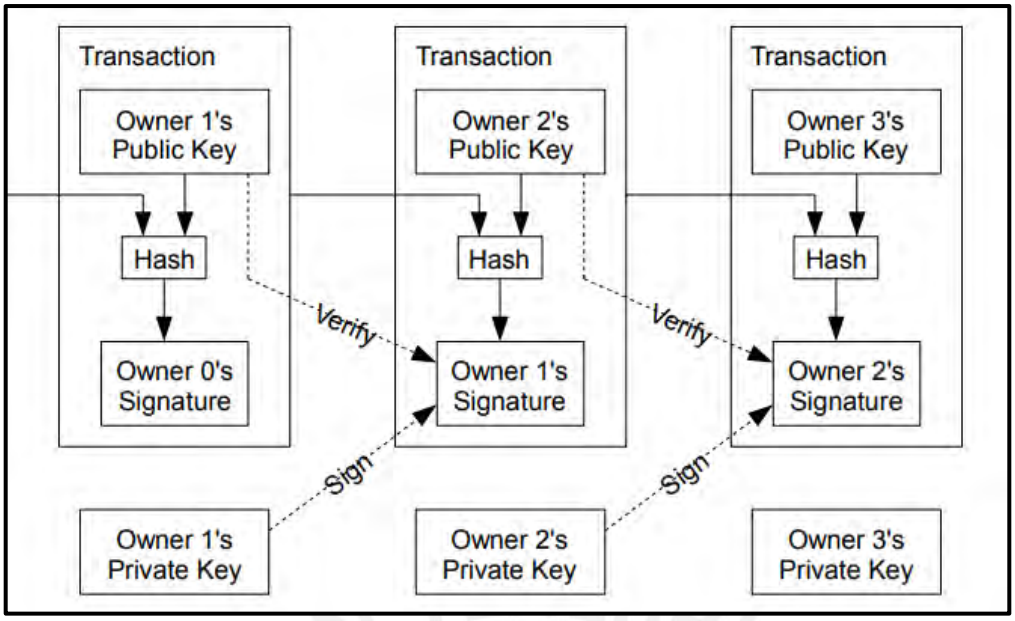


Figura 3: Cadena de firmas digitales
Fuente: (Nakamoto 2008:2)

1.1.4. Tecnología Blockchain

En retrospectiva, se observa cómo se combinaron ideas y conceptos descritos en los párrafos anteriores tales como sistemas distribuidos, mecanismos de consenso, rompecabezas computacionales y *hashcash* para poder crear el bitcoin y lo que se conoce ahora como *blockchain* (Ver Figura 4).

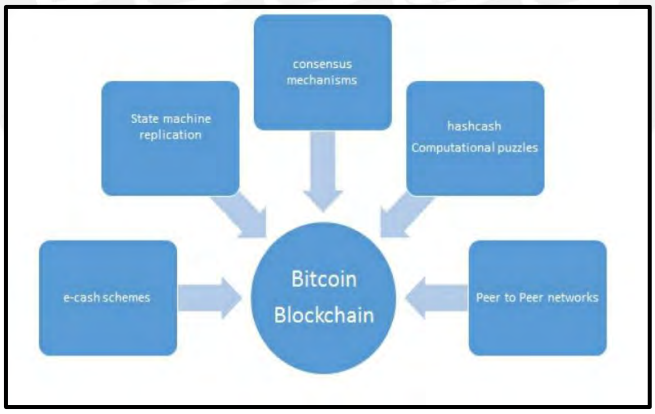


Figura 4: Invención del Bitcoin y Blockchain
Fuente: (Bashir 2018:12)

Bashir (2018), define la *blockchain* como un libro contable distribuido entre pares que es criptográficamente seguro, inmutable y que solo puede ser actualizado a través de un consenso entre pares.

Por su parte, Mougayar (2016) define la *blockchain* desde 3 perspectivas:

En términos técnicos, la *blockchain* es una base de datos que mantiene un libro mayor distribuido públicamente.

En términos de Negocios, la *blockchain* es una red de intercambio para transferir valor y activos entre pares sin la necesidad de intermediarios.

En términos Legales, la *blockchain* valida las transacciones, reemplazando a las entidades confiables.

Swan (2015), categoriza a las generaciones de la tecnología *blockchain* con base en 3 niveles:

- ***Blockchain* 1.0.** – Se introdujo con la creación del bitcoin y se utiliza para las criptomonedas ya que se enfoca en la descentralización del dinero y los pagos.
- ***Blockchain* 2.0.** – Se introducen los *smart contracts*[4] con el fin de implementar aplicaciones que ayuden a descentralizar los mercados financieros. Se enfoca en servicios financieros, pero no contempla las criptomonedas.
- ***Blockchain* 3.0**.- Es utilizada para implementar aplicaciones más allá de la industria de los servicios financieros y se utilizan en industrias como el gobierno, la salud, los medios de comunicación, las artes y la justicia.

Existen 3 tipos de *blockchains*, sin embargo, las más comunes son las públicas y las privadas. Laurence (2017), las define de la siguiente manera:

- **Redes públicas** son grandes y descentralizadas, cualquiera puede participar en ellas. Suelen ser más seguras que las privadas o autorizadas, pero son más lentas y caras. Están aseguradas con una criptomoneda y su capacidad es limitada.
- **Redes privadas** son privadas y solo está abierta a un grupo de individuos. Su principal atributo es su bajo coste y velocidad. La mayoría de estas no utiliza una criptomoneda y su capacidad puede ser ilimitada.
- **Redes autorizadas** son similares a las redes públicas pero la participación está controlada. Muchas de estas redes utilizan una criptomoneda, pero tienen un menor coste para las aplicaciones que se construyen sobre ella lo que genera una mayor escalabilidad y volumen de transacciones.

Tabla 2 Diferencias entre blockchains públicas y privadas

Blockchain pública	*Blockchain* privada
Los participantes no son necesariamente conocidos	Los participantes son conocidos
Los participantes no son necesariamente de confianza	Los participantes son de confianza
Cualquiera puede acceder a la información	Solo los participantes autorizados pueden acceder a la información
Cualquiera puede escribir información	Solo los participantes autorizados pueden escribir información

Fuente: (Morabito 2017:9)

[4] Smart contracts son programas autónomos automatizados que se encuentran alojados en la *blockchain*. No obstante, no todas las *blockchain* cuentan con esta función.

Se han descrito los tipos de redes dentro de la *blockchain,* las cuales sirven como métodos de autorización siendo las más comunes las redes públicas y privadas, sin embargo, hasta el momento no se han mencionado cómo se logra realizar la transacción. Para ello, es necesario hablar de los mecanismos o protocolos de consenso dentro de la *blockchain*. Previamente, se definió los mecanismos de consenso. Los cuales consistían en un conjunto de pasos tomados por los nodos para poder llegar a un acuerdo de un estado o valor. Bashir (2018), menciona que los nodos en una red *blockchain* pueden ejercer distintos roles, de acuerdo al tipo de *blockchain* que se esté utilizando, tales como proponer y validar transacciones hasta realizar minería para facilitar el consenso y asegurar la *blockchain*.

Según Mougayar (2016), Bitcoin inició el protocolo de consenso proof of work y una alternativa a proof of work para lograr el consenso es proof of stake. Bashir (2018), presenta y define a los algoritmos más importantes:

1. **Proof of Work (PoW)** se basa en la prueba de que se han gastado suficientes recursos computacionales antes de proponer un valor que sea aceptado por la red. Actualmente, es el único algoritmo que ha demostrado ser exitoso contra ataques sybil[5].
2. **Proof of Stake (PoS)** se basa en la idea de que los usuarios deben demostrar que poseen una determinada cantidad de tokens, demostrando así que tiene participación en la moneda. Fue introducida por primera vez por Peercoin y se va a utilizar en la red de Ethereum.
3. **Delegated Proof of Stake (DPoS)** se trata de una variación de PoS donde cada nodo o usuario que tiene participación en el sistema puede delegar la validación de una transacción a otro nodo a través de la votación. Es usado por Cardano.
4. **Proof of Elapsed Time (PoET)** fue introducido por Intel, utiliza Entorno de Ejecución confiable (TEE) para garantizar aleatoriedad y seguridad en el proceso de elección de líder mediante un tiempo de espera. Es usado por Hyperledger Sawtooh.
5. **Deposit-based consensus** los nodos o usuarios que desean participar en la red deben de depositar una fianza antes de proponer un bloque.
6. **Proof of Importance (PoI)** se basa en mantener o bloquear criptomonedas en el sistema y supervisar el movimiento de los estas por parte del usuario para asignar un nivel de confianza e importancia. Es usado por Nemcoin.
7. **Federated consensus or federated Byzantine consensus (FBA)** los nodos se agrupan de pares de confianza y propagan solo las transacciones que han sido validadas por la mayoría de los nodos de confianza.
8. **Reputation-based mechanism** un líder es elegido con base en la reputación que ha construido a lo largo del tiempo en la red. Esto se da mediante la votación de los nodos o usuarios.

[5] Ataque sybil es un ataque informático donde una persona trata de hacerse con el control de la red creando múltiples cuentas, nodos o computadoras, que son de su propiedad.

9. **Practical Byzantine Fault Tolerance (pBFT)** se basa en un esquema simple de nodos que envían mensajes firmados, cuando se alcanza un cierto número de mensajes recibidos entonces se llega a un acuerdo.

De acuerdo con Mougayar (2016), una de las desventajas de utilizar PoW es que no es ecoamigable ya que requiere una gran cantidad de gasto energético debido a la gran cantidad de poder computacional que se necesita para llevarse a cabo. Es por ello que PoS es una gran alternativa pues no requiere una gran capacidad de procesamiento como PoW, sino que se basa en el concepto de la minería virtual y en la votación basada en tokens.

Tabla 3 Principales características de los protocolos de consenso PoW, PoS y PoW/PoS

Esquema	Baja latencia	Bajo coste energético a largo plazo
Proof of Work (PoW)	No	No
Proof of Stake (PoS)	Sí	Sí
Hybrid Proof of Work and Proof of Stake (PoW/PoS)	Sí	Sí

Fuente: (Morabito 2017:12)

1.1.5. El mercado de las criptomonedas

En los años posteriores al lanzamiento del Bitcoin en 2009, fueron desarrolladas otras criptomonedas conocidas como *altcoins*[6]. Estas se diferencian del Bitcoin en aspectos como los mecanismos de minería, los métodos de distribución de monedas o la capacidad de crear aplicaciones descentralizadas. A menudo, estas *altcoins* se desarrollan con un propósito diferente o con el fin de mejorar las limitaciones del Bitcoin, como el suministro limitado de Bitcoin, el alto consumo de energía de la red o el mecanismo de consenso del usuario de *PoW*. Según Bashir (2018), en 2013 y 2014, el mercado de *altcoins* creció exponencialmente y se crearon diferentes tipos de proyectos de *altcoins* de los cuales algunos tuvieron éxito y otros perecieron. A continuación, se mostrará en la Tabla 4 las 4 principales criptomonedas según CoinMarketCap.

Se observa que tanto Bitcoin como Ethereum utilizan el algoritmo o protocolo de consenso PoW. El cual, como se ha mencionado antes, no es ecoamigable pues genera un excesivo gasto energético ya que requiere utilizar una gran cantidad de potencia computacional para poder resolver acertijos matemáticos complejos. Por ello, los posibles candidatos a estudiar serían Cardano y Binance Coin, ya que ambos utilizan un mecanismo PoS. Sin embargo, Cardano implementará los *smart contract,* el 12 de septiembre de 2021, con la actualización Alonzo[7]. La actualización permitirá a Cardano entrar a

[6] Altcoin es una criptomoneda alternativa a Bitcoin

[7] Farooque, M. (2021). Cardano Will Surge to New Highs After the Alonzo Release. *Nasdaq*.https://www.nasdaq.com/articles/cardano-will-surge-to-new-highs-after-the-alonzo-release-2021-08-25

competir en el ecosistema de las aplicaciones descentralizadas y DeFi. A diferencia de la red Binance Smart Chain (BSC) que ya se encuentra compitiendo pues, de acuerdo con Leandro França de Mello, es el ecosistema preferido del mercado NFT[8] pues muchas de las aplicaciones descentralizadas han migrado de la red de Ethereum (ERC20) a BSC por sus bajas comisiones de gas. Por tal motivo, el presente estudio analizará la criptomoneda Binance Coin (BNB) ya que es el token nativo de las redes Binance Chain (BC) y Binance Smart Chain. El token BNB se utiliza para pagar las comisiones de gas y otros servicios de la red BC o BSC.

Tabla 4 Capitalización del mercado (a octubre 2021) y características de las principales criptomonedas

Nombre	Precio ($)	Capitalización del mercado ($)	Protocolo de consenso	Ecosistema
Bitcoin	56,439.99	1,064,073,600,738	PoW	Bitcoin
Ether	3,580.64	423,376,639,619	PoW	Ethereum
Cardano	2.22	71,481,544,738	DPoS	Cardano
Binance Coin	419.65	70,678,980,822	PoS	Binance Chain y Binance Smart Chain

Fuente: (CoinMarketCap 2021)

1.2. Análisis del sentimiento en Twitter y mercados financieros

1.2.1. Análisis de Sentimiento

Es una técnica que utiliza el procesamiento de lenguaje natural, análisis de texto y herramientas computacionales para categorizar comentarios emitidos por usuarios en la Web 3.0. Estos comentarios pueden denotar sentimiento u opiniones positivas, negativas o neutras sobre diversos tópicos o temas. De acuerdo con Montesinos (2014) los desafíos a superar para poder implementar este tipo de análisis son:

- Determinar si el comentario del *tweet* es una opinión ya que puede tratarse de un comentario objetivo o de una respuesta de un usuario a otro.
- Determinar el tema del cual hacen referencia los *tweets* ya que si este no está asociada al tema de estudio entonces la información contenida en los *tweets* es irrelevante.
- Identificar las jergas, abreviaciones y modismos típicos utilizados en los *tweets*.
- Determinar la polaridad contenida en los *tweets* ya que un mensaje puede poseer sentimientos positivos y negativos en la misma oración. Por ejemplo: "*Me alegro que se haya terminado, pésimo el espectáculo*".

[8] Non-fungible tokens son tokens que pueden ser utilizados para representar la propiedad de objetos únicos tales como pinturas, música, objetos de colección e incluso inmuebles. https://ethereum.org/en/nft/

Para poder determinar la polaridad de un *tweet* existen diversos métodos, no obstante, los dos más usados son el aprendizaje computacional o aprendizaje de máquina y el uso de diccionarios léxicos. Sin embargo, el primero esta fuera del alcance de este trabajo ya que se hará uso de algoritmos que utilicen diccionarios léxicos ya que presentan un mayor porcentaje de efectividad respecto al primer método.

1.2.2. Diccionario léxico

El método del diccionario léxico tal como su nombre lo infiere se basa en un diccionario que contiene un conjunto de palabras que presentan un determinado peso y/o categoría emocional. La mayoría de estos diccionarios circulan por la web, gran parte de ellos esta en inglés, aunque una pequeña proporción de estos se encuentra en español, sin embargo, dado que son muy pocos aún se encuentran en etapa beta lo que significa que no son tan precisos. Montesinos (2014) describe algunos de los diccionarios más conocidos:

- *Dictionary of Affect in Language* contiene aproximadamente 8000 palabras en inglés cuyo rango de valores va desde 1 (negativo) a 3 (positivo).
- *Linguistic Inquiry and Word Count* es uno de los diccionarios más completos, posee una versión beta en español. Las palabras son etiquetadas en una categoría además se les atribuye un peso.
- *SentiWordNet* categoriza cada palabra en positivo, neutro o negativo, sus valores están entre 0 y 1.
- *General Inquirer* posee una lista de palabras y características asociadas, las cuales son: activa, pasiva, fuerte, débil, placentera o dolorosa, etc.
- *MPQA Subjectivity Lexicon* posee alrededor de 8000 palabras, con su respectiva polaridad y categoría gramatical.

1.2.3. Método de Corpus

El método corpus es un método alternativo al diccionario léxico. Este se compone de pequeños diccionarios de palabras, en general estos son adjetivos positivos o negativos cuya finalidad es identificar otros términos que indiquen similares sentimientos. Un claro ejemplo del funcionamiento de este método se puede encontrar en el estudio de Montesinos (2014) donde menciona que al conector "and", el cual es utilizado para unir adjetivos de similar polaridad, por ejemplo: "*This car is beautiful and spacious"*, si el término "*beautiful"* este contenido en el diccionario y esta representa un sentimiento positivo entonces la palabra "*spacious"* se agrega al diccionario con una polaridad positiva también. Posterior al análisis de los textos y haber podido identificar si los términos contenidos en los textos presentaban similar u opuesta polaridad son agrupados en palabras positivas y negativas. No obstante, según Montesinos (2014) este método presenta algunas desventajas frente al método de

diccionario léxico, pues es complicado juntar un texto que incluya todas las palabras en un idioma en específico.

1.2.4. Sentimiento y predictibilidad en los mercados financieros

Kraaijeveld y De Smedt (2020) define el sentimiento como cualquier percepción errónea que puede conllevar a valorar de forma incorrecta el valor fundamental de un activo. Por lo tanto, los sentimientos pueden hacer que los activos sean especulativos, sin embargo, según Baker y Wurgler (2007) lo que vuelve un activo más especulativo que otro es la dificultad y subjetividad al momento de determinar su verdadero valor. De acuerdo con Kraaijeveld y De Smedt (2020) basado en trabajos anteriores se sabe que los mercados financieros son afectados por las noticias y estas a su vez afectan el sentimiento.

Los inversionistas utilizan indicadores técnicos para medir el sentimiento tales como VIX, MACD, media móvil de 50 y 200 días.

1.2.5. Análisis del sentimiento en Twitter

El análisis de sentimiento es conocido también como minería de opiniones, de acuerdo con Liu (2015) es el campo de estudio que analiza opiniones, sentimientos, valoraciones, actitudes y emociones de las personas hacia entidades (productos, servicios, organizaciones, individuos, eventos, problemas y temas) expresado de forma escrita. Sin embargo, Liu (2015) menciona que el análisis de sentimiento se centra principalmente en las opiniones que expresan o implican sentimientos positivos o negativos. Cabe resaltar que, al hablar de sentimientos positivos y negativos, se debe considerar también las expresiones sin sentimientos, es decir, expresiones neutrales. En otras palabras, el análisis de sentimiento tiene como objetivo identificar las opiniones o sentimientos positivos, negativos o neutrales expresados o implícitos en el texto, así como también a los destinatarios de estas opiniones o sentimientos.

Un ámbito de aplicación popular del análisis de sentimiento es la predicción del mercado bursátil. Por ejemplo, Zhang et al. (2010) identificó los estados de ánimos positivos y negativos de los usuarios en la red social Twitter y los utilizo para predecir el movimiento de los índices bursátiles como el Down Jones, el S&P 500 y el NASDAQ. Ello revelo que cuándo los usuarios expresan mucha esperanza, miedo o preocupación, el índice Down Jones baja al siguiente día de forma contrario cuándo expresan menos esperanza, miedo o preocupación, el índice Down Jones sube. No obstante, realizar un análisis de sentimiento no es tan simple como parece, en realidad es un tema complicado pues un texto puede contener múltiples sentimientos a la vez. Por ejemplo:

"La actuación estuvo buena, pero la película pudo estar mejor"

La oración posee dos sentimientos a la vez positivos y negativos.

La red social Twitter cada vez está siendo más utilizada como una fuente de datos para el análisis de sentimiento ya que según Kraaijeveld y De Smedt (2020) las redes sociales se han convertido en la fuente primaria de información sobre las criptomonedas. Las principales fuentes son Twitter, foros relacionados a la criptomonedas y Noticias sobre criptomonedas. Sin embargo, en el caso de Twitter, los tweets tienen una longitud máxima de 280 caracteres lo cual hace que los datos sean extremadamente ruidosos. Por dicho motivo, es necesario hacer un preprocesamiento a los datos para posteriormente aplicar el algoritmo VADER para extraer el sentimiento de los tweets. De acuerdo con Kraaijeveld y De Smedt (2020) VADER es un *lexicon* y un modelo de análisis de sentimiento basado en reglas que esta específicamente entrenado y es adecuado para los sentimientos expresados en las redes sociales.
Por su parte, Beri (2017) dice lo siguiente sobre el modelo VADER:

> Es usado para el análisis de sentimiento de textos que es sensible tanto a la polaridad (positiva o negativa) como a la intensidad de la emoción. El análisis sentimental de VADER se basa en un diccionario que asigna características léxicas a intensidades de emoción conocidas como puntuaciones de sentimiento. La puntuación del sentimiento de un texto se obtiene sumando la intensidad de cada palabra del texto.

1.3. Series de tiempo

El análisis experimental de datos que se han observado en diferentes instantes de tiempo conduce a nuevos y únicos problemas de modelación e inferencia estadística. Sin embargo, la correlación causada por una muestra de puntos(datos) adyacentes a lo largo del tiempo impide que se puedan aplicar métodos estadísticos convencionales. El enfoque que se encarga de resolver las cuestiones matemáticas y estadísticas que plantean estas correlaciones temporales se le conoce como análisis de series de tiempo. De acuerdo con Shumway y Stoffer (2017) existen dos enfoques distintos, pero no necesariamente excluyentes para el estudio de las series de tiempo: *time domain approach* y *the frequency domain approach*. El primero se encarga de estudiar las relaciones retardadas mientras que el segundo estudia los ciclos.

1.3.1 Definición de series de tiempo

Las series de tiempo pueden ser definidas como todo lo que se observa secuencialmente durante un periodo de tiempo (Hyndman y Athanasopoulos, 2018). Por su parte, Shumway y Stoffer (2017) definen las series de tiempo como un conjunto de variables aleatorias indexadas según el orden en el cual han sido obtenidas a lo largo del tiempo, ello también es conocido como proceso estocástico. En términos formales, Auffarth (2021) define las series temporales como un proceso estocástico que se modela de la siguiente forma:

$$\{X_t\}_{t \in T}$$

, donde X(t) o Xt representa el valor de la variable aleatoria en el instante de tiempo t. Si T es un conjunto de números reales se trata de un proceso estocástico tiempo continuo mientras que si T es un conjunto de números enteros se conoce como proceso estocástico en tiempo discreto.

White Noise es un tipo de serie de tiempo que contiene variables aleatorias no correlacionadas (Wt) con media 0 y varianza finita. En ecuación adjunta, se observa la denotación de este proceso.

$$w_t \sim wn(0, \sigma_w^2)$$

Según Shumway y Stoffer (2017) existe otras variaciones del proceso *white noise* como *white independent noise*, el cual tiene como requisito que el ruido sea independiente e idénticamente distribuido; y el *Gaussian white noise* donde las variables aleatorias independientes siguen una distribución normal.

Los objetivos que se persiguen con el estudio o análisis de las series de tiempo son los siguientes:

- Estimar cómo continuará la secuencia de observaciones en el futuro (Hyndman y Athanasopoulos, 2018).
- Desarrollar modelos matemáticos que proporcionen información sobre los datos de muestra (Shumway y Stoffer, 2017).
- Controlar el proceso generador de la serie mediante el conocimiento de los parámetros que afectan el modelo o estableciendo políticas intervencionistas cuando el proceso se desvié de un objetivo preestablecido (Quesada, 2021).

Existen diferentes enfoques para el estudio o análisis de las series de tiempo:

- **Métodos tradicionales. –** Se basan en dividir la serie de tiempo en varios componentes mediante una descomposición aditiva o multiplicativa donde cada uno de los componentes representa una categoría de patrón subyacente. Asimismo, las técnicas de aislamiento exponencial también recaen dentro de este grupo.
- **Método de Box – Jenkins. –** Se basa en encontrar el mejor modelo dentro de la familia de los modelos de ARIMA para una serie de tiempo dada.
- **Análisis univariado y multivariado. –** En el primer caso, se centra en la comprensión de la variable para ello se calcula la media y varianza mientras que en el segundo se busca descubrir las relaciones entre las variables, ello se obtiene mediante el diagrama de dispersión.
- **Análisis en el dominio del tiempo y análisis en el dominio de las frecuencias. –** Explotan las características fundamentalmente de la función de correlación y densidad espectral (Quesada, 2021).

1.3.2. Patrones de Series de Tiempo

Tendencia

La tendencia puede ser definida como un aumento o disminución en los datos a largo plazo (Ver Figura 9), este no necesariamente es lineal, asimismo es utilizado para hacer referencia a cambios de direcciones (Hyndman y Athanasopoulos,2018). Por su parte, Auffarth (2021) define la tendencia como la dirección en la que se mueve una serie cuando no se tienen en cuenta las oscilaciones.

Estacionalidad

De acuerdo con Hyndman y Athanasopoulos (2018) una serie presenta patrones estaciones si es que esta afectada por factores estacionales tales como la época del año o día de la semana. La estacionalidad es siempre una frecuencia fija y conocida (Ver Figura 5).

Ciclo

Hyndman y Athanasopoulos (2018) un ciclo se produce cuando los datos presentan subidas y bajadas sin una frecuencia fija. Estas fluctuaciones están asociadas a lo que es comúnmente conocido como el ciclo del negocio. La duración de estas fluctuaciones suele ser de aproximadamente 2 años por lo menos (Ver Figura 5).

La mayoría de personas suele confundir el comportamiento cíclico de una serie de tiempo con el patrón estacional. Por dicho motivo, Hyndman y Athanasopoulos (2018) sugiere que se observen las fluctuaciones, si estas son invariables y están asociada a una frecuencia fija entonces se trata de una estacionalidad de lo contario se trataría de un patrón o comportamiento cíclico.

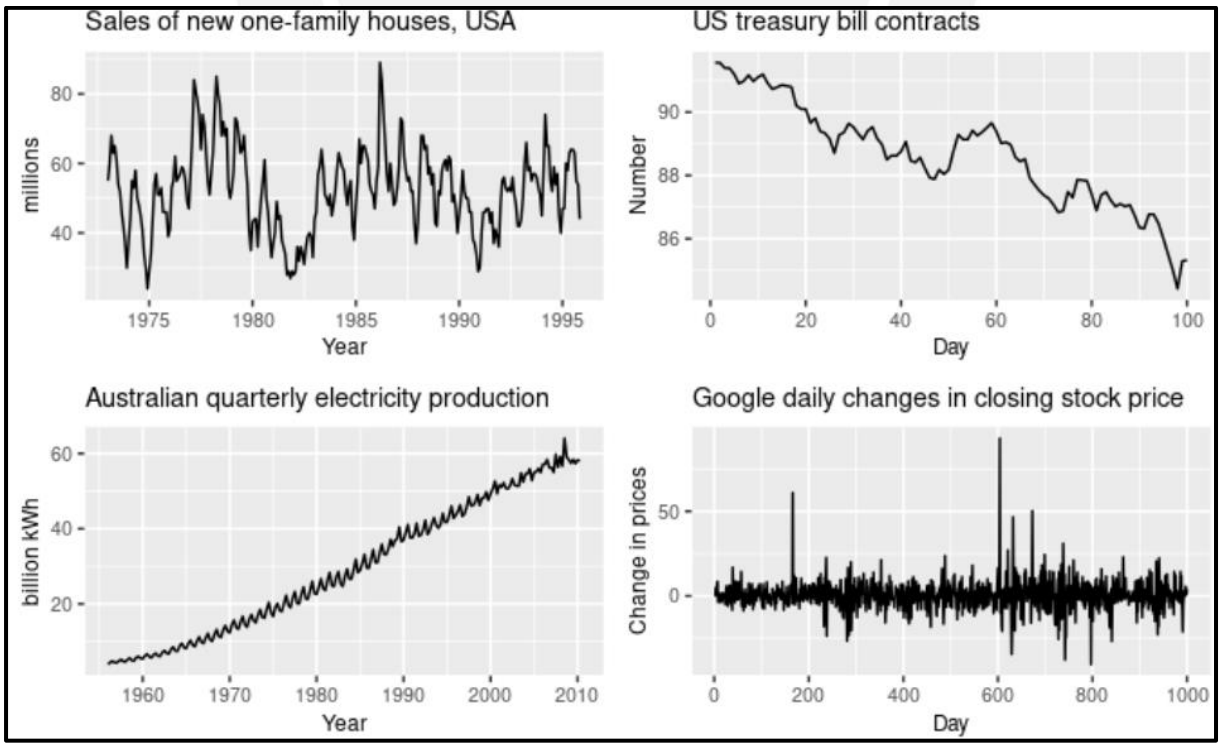


Figura 5: Ejemplos de series de tiempo que presentan patrones
Fuente: (Hyndman y Athanasopoulos 2018)

1.3.3. Autocorrelación

La autocorrelación mide la relación lineal entre los *lags* de una serie de tiempo. Hyndman y Athanasopoulos (2018) enuncia lo siguiente sobre la autocorrelación:

> Existen varios coeficientes de autocorrelación, correspondientes a cada panel en el gráfico de desfase, Por ejemplo, r_1 mide la relación entre y_t y y_{t-1} y r_2 mide la relación entre y_t y y_{t-2} y así sucesivamente.

De forma general, el coeficiente r_k puede ser definido de la siguiente forma:

$$r_k = \frac{\sum_{t=k+1}^{T}(y_t-\bar{y})(y_{t-k}-\bar{y})}{\sum_{t=1}^{T}(y_t-\bar{y})^2},$$

donde el parámetro T representa la longitud de la serie de tiempo.

Cuando los datos presentan tendencia, la autocorrelación de los pequeños *lags* tienden a ser grandes y positivos dado que las observaciones cercanas en el tiempo también lo son. Por lo tanto, la función de autocorrelación de las series de tiempo que presentan tendencia tiende a tener valores positivos que disminuyen lentamente a medida que aumentan los *lags*. Sin embargo, cuando los datos presentan estacionalidad, la autocorrelación será mayor para los *lags* estacionales que para los otros *lags*.

1.3.4. Modelos de regresión de series de tiempo

El método de predicción adecuado depende de la información disponible. Si no hay datos disponibles o si los datos disponibles no son relevantes para las previsiones entonces se debe utilizar métodos de predicción cualitativos. Estos métodos no son mera imaginación o adivinanza, ya que detrás de ellos hay todo un estudio estructurado bien desarrollado con el fin de obtener buenas predicciones sin utilizar data histórica. Sin embargo, en este trabajo no se hará uso de métodos de predicción cualitativos sino cuantitativos. Para ello se deben satisfacer 2 condiciones, según Hyndman, R.J., y Athanasopoulos, G. (2018):

- Data histórica numérica disponible
- Suponer que algunos patrones pasados se repetirán en el futuro

Existe una gran cantidad de métodos de predicción cuantitativos, cada uno de ellos ha sido desarrollada para un fin en concreto. En la Figura 6, se observa como el modelo ha aprendido el patrón estacional observado en la data histórica y lo ha reproducido para los próximos dos años. La región azul sombreada muestra los intervalos de predicción a un 80%, por ende, se espera que con un 80% de probabilidad cada valor futuro se encuentre en la región sombreada mientras que la región azul lineal muestra los intervalos de predicción a un 95%. Estos intervalos de predicción sirven para mostrar las incertidumbres de las predicciones.

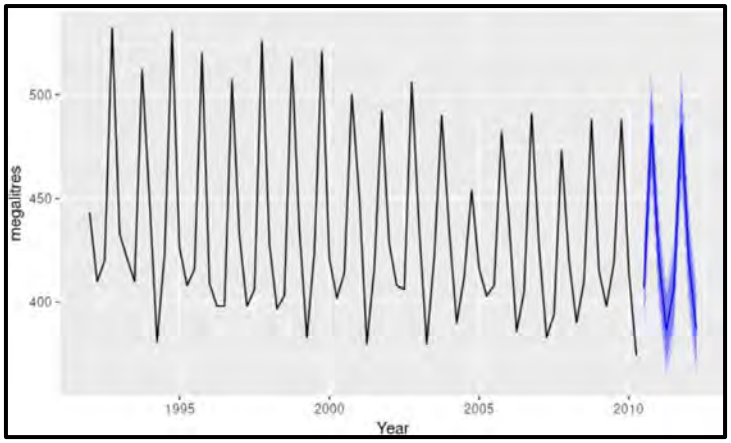


Figura 6: Producción trimestral de cerveza australiana: 1992Q1-2010Q2
Fuente: (Hyndman y Athanasopoulos 2018)

1.3.5. Pronóstico de series de tiempo con modelos clásicos

Los modelos clásicos de series de tiempo requieren que el proceso de generación de datos sea *stationarity*. Auffarth (2021) define *stationarity* (o *weak stationarity*) con base en 3 características:

- Variación finita
- Media constante
- Variación constante

La variación constante significa que la variación de la serie de tiempo en una ventana entre dos puntos es constante a lo largo del tiempo: $\gamma_x(s,t) = \gamma_x(s+h, t+h)$, aunque esto puede cambiar con el tamaño de la ventana.

Según Auffarth (2021), los modelos clásicos pueden ser agrupados en las siguientes familias de modelos: medias móviles (MA), autorregresivos (AR), ARMA y ARIMA.

La **media móvil simple**, se define usualmente como la media aritmética no ponderada sobre un período de k puntos, donde x_i representa la serie de tiempo observada.

$$\frac{x_1 + x_2 + \cdots + x_k}{k} = \frac{1}{k}\sum_{i=0}^{k} x_i$$

Una de sus principales características es que puede ser utilizada para suavizar la tendencia y el ciclo de una serie temporal, eliminando el ruido y las fluctuaciones periódicas que se producen a corto plazo. Sin embargo, la media móvil también puede ser utilizada como modelo para predecir el futuro. Auffarth (2021) define el modelo de media móvil de orden q de la siguiente forma:

$$\boldsymbol{MA(q)}: x_t = \mu + \epsilon_t + \sum_{i=0}^{q} \varphi_i \epsilon_{t-i},$$

donde μ representa la media de x_t (usualmente se asume que es 0), φ_i son parámetros y ϵ_t es *random noise*.

Por su parte, el **modelo autorregresivo** hace una regresión de la variable sobre sus propios valores retardados, dicho de otra forma, el valor actual es impulsado por sus vecinos inmediatos anteriores mediante una combinación lineal. El modelo autorregresivo de orden *p* puede ser denotado de la siguiente manera:

$$\boldsymbol{AR(p)}: x_t = c + \epsilon_t + \sum_{i=1}^{p} \emptyset_i x_{t-i},$$

donde $\emptyset_i$ es un parámetro del modelo, *c* es una constante y ϵ_t es *random noise*. En esta ecuación, *p* mide la autocorrelación entre valores sucesivos de la serie de tiempo.

El **modelo ARMA** está compuesto por dos tipos de valores retardados, uno para el componente autorregresivo y el otro para el componente de la media móvil. Por lo tanto, se define ARMA de la siguiente forma:

$$\boldsymbol{ARMA(p,q)}: x_t = c + \epsilon_t + \sum_{i=1}^{p} \emptyset_i x_{t-i} + \sum_{i=0}^{q} \varphi_i \epsilon_{t-i},$$

donde *p* y *q* representan el orden de la autoregresión y de la media móvil respectivamente.

El **modelo ARIMA (p, d, q)** incluye un paso de preprocesamiento de datos llamado integración. El cual tiene como objetivo volver estacionaria la serie de tiempo a través de una transformación llamada diferenciación. El parámetro *d* denota el número de veces que se han tomado diferencias entre los valores actuales y los anteriores.

A diferencia de los modelos presentados anteriormente que requieren que las series de tiempo presenten un proceso *stationary*, **el modelo SARIMA**, desarrollado como una extensión del modelo ARIMA, puede describir procesos que presentan un comportamiento no estacionario dentro como a través de las estaciones. Los modelos SARIMA suelen ser expresados como ARIMA (p, d, q) (P, D, Q) m, donde *m* define el número de períodos en una temporada mientras que *P*, *D* y *Q* parametrizan los componentes autorregresivos, de integración y de media móvil de la parte estacional de la serie de tiempo. Cabe resaltar que el parámetro *P* mide también el grado de autocorrelación que existe entre los componentes estacionales sucesivos de la serie de tiempo. Se procederá a descomponer el modelo SARIMA en sus componentes estacionales para poder ilustrar de mejor forma los parámetros que forman parte de estos.

$$\boldsymbol{SAR(P)}: x_t = c + \epsilon_t + \sum_{i=1}^{P} \emptyset_i x_{t-s \cdot P},$$

donde *s* representa la duración de la estacionalidad.

$$\boldsymbol{SMA(Q)}: x_t = \mu + \epsilon_t + \sum_{i=0}^{Q} \varphi_i \epsilon_{t-s \cdot Q}$$

Modelo Vectorial Autorregresivo (VAR), en este método, todas las variables se tratan como endógenas. Athanasopoulos et al. (2012) define al modelo VAR como una generalización del modelo autorregresivo univariante para predecir un vector de series de tiempo. Comprende una ecuación por cada variable en el sistema, el lado derecho de cada ecuación incluye una constante y retardos de todas las variables del sistema. En la Figura 5, se observa un VAR de dos variables con un retardo, donde:

$$y_{1,t} = c_1 + \emptyset_{11,1} y_{1,t-1} + \emptyset_{12,1} y_{2,t-1} + e_{1,t}$$
$$y_{2,t} = c_2 + \emptyset_{21,1} y_{1,t-1} + \emptyset_{22,1} y_{2,t-1} + e_{2,t}$$

Si la serie de tiempo es estacionaria, se pronosticará ajustando los datos a un modelo VAR directamente, no obstante, si la serie es no estacionaria se agruparán los datos para volverlos estacionarios y recién ajustarlo a un modelo VAR. En ambos casos, los modelos se estiman ecuación por ecuación utilizando el principio de mínimos cuadrados.

De acuerdo con Hyndman y Athanasopoulos (2018) una de las críticas a las que se enfrentan los modelo VAR es que son ateóricos, es decir, no se basa en ninguna teórica económica que imponga una estructura teórica de las ecuaciones. A pesar de esto, los modelos VAR son útiles en los siguientes contextos:

1. Predicción de un conjunto de variables relacionadas entre sí que no requieren una interpretación explicita.
2. Comprobar si una variable es útil para predecir otra.
3. Para el análisis de respuesta al impulso, donde se analiza la respuesta de una variable a un cambio repentino pero temporal de otra variable.
4. Descomponer el error de la varianza de la predicción

1.3.6. Pronóstico de series de tiempo mediante suavizado exponencial

El **método de suavizado exponencial simple** es una técnica utilizada para suavizar los datos de las series de tiempo utilizando una función de ventana exponencial. La cual puede ser utilizada para pronosticar series de tiempo que presentan estacionalidad y tendencia.

$$s_0 = x_0$$
$$\boldsymbol{SES}(\boldsymbol{\alpha}): s_t = \alpha x_t + (1-\alpha) x_{t-1},$$

donde x_t representa la serie de tiempo y α la tasa de aprendizaje del nivel cuyo rango de valor fluctúa entre 0 y 1.

El **método theta** es otro método de suavizado exponencial, es muy similar al método de suavizamiento exponencial simple a diferencia de que este incluye un término conocido como *drift*. Auffarth (2021) define este método de la siguiente forma:

$$T_t = c + b_0(t-1) + \epsilon_t$$

$$\boldsymbol{Theta}: \hat{X}_t = (1-\alpha)T_t + \alpha\hat{X}_{t-1},$$

donde T_t estima el componente de tendencia, c es el intercepto, b_t es un coeficiente multiplicado por el paso del tiempo y ϵ_t es el residual. Los parámetros c y b_t pueden ser determinados mediante el método de mínimos cuadrados ordinarios.

El **método de suavizado doble** logra resolver la limitación del método de suavizado exponencial simple, ya que logra identificar los niveles y la tendencia de la serie de tiempo. Sin embargo, aún es incapaz de poder identificar el patrón de estacionalidad de la serie de tiempo. Además de ello, el modelo trabaja bajo la asunción de que la tendencia es siempre la misma. Por dicho motivo, según Vandeput (2021) Gardner y McKenzie desarrollaron en 1985 el **método de suavizado doble con tendencia amortiguada** que incluye un factor de amortiguamiento que reducirá exponencialmente la tendencia a lo largo del tiempo. El modelo puede ser formulado de la siguiente forma:

$$a_t = \alpha d_t + (1-\alpha)(a_{t-1} + \phi b_{t-1})$$
$$b_t = \beta(a_t - a_{t-1}) + (1-\beta)\,\phi b_{t-1}$$
$$f_{t+1} = a_t + b_t\phi,$$

donde a_t estima el nivel de la serie de tiempo, b_t estima la tendencia mediante la tasa de aprendizaje de la tendencia (β) y ϕ representa la tasa de amortiguamiento. Si a la tasa de amortiguamiento se le asignara un valor de 0, entonces, se obtiene el método de suavizado doble.

Por último, el **método Holtz-Winters** también conocido como suavizado exponencial triple requiere remover la tendencia y estacionalidad de la serie de tiempo para ser aplicado. Este método captura tres componentes: una estimación de un nivel para cada instante de tiempo (L_t), un componente de tendencia (T), estacionalidad (S_t) con m ,el cual representa el número de estaciones en un año. Según Auffarth (2021) existen diversas variaciones de este método, estas pueden ser aditivas o multiplicativas, ya que tanto la tendencia como la estacionalidad pueden ser aditivas o multiplicativas.

El método Holtz-Winters con tendencia y estacionalidad aditiva puede ser definido de la siguiente forma:

$$\boldsymbol{HW}: x_{t+k} = L_t + kT_t + S_{t+k-m}$$

Con tendencia aditiva y estacionalidad multiplicativa se formula de la siguiente manera:

$$\boldsymbol{HW}: x_{t+k} = (L_t + kT_t) \cdot S_{t+k-m}$$

El nivel se define de la siguiente forma:

$$L_t = \alpha\frac{x_t}{S_{t-m}} + (1-\alpha)(L_{t-1} + T_{t-1})$$

La tendencia aditiva sigue la siguiente fórmula:

$$T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1}$$

Por su parte la estacionalidad multiplicativa es descrita de la siguiente manera:

$$S_t = \gamma\frac{x_t}{L_t} + (1-\gamma)S_{t-M}$$

***Random Walk with Drift*,** de acuerdo con Shumway y Stoffer (2017) este modelo es de gran ayuda si se busca analizar la tendencia de la serie de tiempo. La constante inicial X0 = 0, y la constante *drift* esta definida por la letra griega delta. Si el valor de la constante *drift* es 0, entonces se trataría de un modelo de *random walk* simple. El modelo random walk with Drift esta representado por la siguiente ecuación:

$$x_t = \delta + x_{t-1} + w_t$$

Por su parte Hyndman y Athanasopoulos (2018) comentan que el modelo *random walk* es comúnmente utilizado en datos no estacionarios, particularmente en datos económicos y financieros. Asimismo, se menciona que presenta dos características en común:

- Períodos largos de tendencia alcista o bajista
- Cambios de dirección repentinos e imprevisibles

1.3.7. Selección del modelo

Se conoce como *model selection* a la metodología para decidir entre modelos que compiten entre sí. En métodos tradicionales de regresión y teoría de selección del modelo, se suele utilizar un enfoque que consiste en considerar la complejidad del modelo para la selección del modelo, ya que modelos más complejos tienden a sobre ajustar los datos, por ende, poseen una mala capacidad de generalización. Bajo este contexto, una de las ideas principales en las que se basa esta teoría de selección de modelos es en la navaja de Ockahm, la cual establece que se debe preferir la solución que presente menor número de supuestos. De acuerdo con Auffarth (2021) los modelos de ARMA y otros se suelen estimar con la estimación de máxima verosimilitud (EMV). Uno de los criterios de selección de modelos más utilizados para evaluar el método EMV es el criterio de información de Akaike (AIC).

$$AIC = 2k - 2l,$$

donde l representa la log-verosimilitud del método de máxima verosimilitud y k, el número de parámetros del modelo.

Para un modelo ARIMA, el criterio AIC puede ser definido de la siguiente forma:

$$AIC = 2(p + q) - 2l$$

Existe otro criterio muy similar al AIC que realiza la misma función de comparar y seleccionar el mejor modelo, se le conoce como criterio de información bayesiano (BIC). La única diferencia es que este presenta un parámetro adicional N que representa el número de muestras del conjunto de datos.

$$BIC = k \cdot \ln(N - 2) \cdot \ln(l)$$

En ambos criterios se busca o prefiere aquel con el menor valor. No obstante, Arlot y Celisse (2010) argumentan que en la practica el método de *cross validation* es aplicable a una gran variedad de

problemas, por lo que sin conocer los datos es probable que produzca mejores resultados que los criterios de información penalizados.

La práctica más común para el particionamiento del conjunto de datos dentro de los métodos tradicionales de evaluación de pronósticos de series de tiempo es el *last block* evaluation. Este método, según Bergmeir y Benítez (2012), posee 4 posibilidades de entrenamiento y evaluación:

Fixed-origin evaluation este método de evaluación consiste en entrenar el modelo mediante los datos de entrenamiento y realizar un pronóstico por cada valor presente en el conjunto de prueba o evaluación. El origen del pronóstico se fija en el último valor del conjunto de datos de entrenamiento, por ende, por cada horizonte de tiempo solo se puede calcular un pronóstico. Cabe resaltar que suele ser utilizado en las competencias de pronosticó de series de tiempo. Sin embargo, una de las desventajas de este método es que las características estadísticas presentes en los datos de entrenamiento pueden influir en los resultados de evaluación.

Rolling-origin-recalibration evaluation consiste en transferir secuencialmente datos del conjunto de prueba al conjunto de entrenamiento. En consecuencia, el origen del pronóstico cambiará o se actualizará siguiendo la misma secuencia. Por lo tanto, por cada pronóstico realizado el modelo será reentrenado utilizando toda la información disponible en el conjunto de datos de entrenamiento.

Rolling-origin-update evaluation este tipo de evaluación es similar al *rolling-origin-recalibration evaluation* a diferencia de que este no transfiere valores del conjunto de prueba al conjunto de entrenamiento y no realiza un reentrenamiento del modelo. En su lugar, utiliza los valores pasados del conjunto de datos de prueba para actualizar la información de entrada del modelo.

Bergmeir y Benítez (2012) mencionan que a estos dos tipos de evaluación *rolling-origin evaluation* se les suele denominar *n-step-ahead evaluation* donde *n* representa el horizonte del pronóstico utilizado en la evaluación del modelo.

Rolling-window evaluation este método es similar al *rolling-origin-recalibration evaluation*, la única diferencia es que el tamaño del conjunto de datos de entrenamiento se mantiene constante, es decir, conforme se recolecten nuevos datos con el fin de que el tamaño siga siendo constante, los datos más antiguos del principio de la serie de tiempo son descartados. Este método solo puede ser aplicado si el modelo es reconstruido en cada ventana de tiempo y posee ventajas estadísticas teóricas, las cuales pueden ser identificadas en la práctica si los valores pasados tienden a perturbar la generación del modelo.

En algoritmos de machine learning tanto de regresión como de clasificación, este proceso de selección del modelo (o *model selection*) se suele llevar a cabo, a través de la evaluación de los distintos modelos mediante métodos estadísticos, con el fin de determinar si el modelo escogido es apropiado para el DGP (proceso de generación de datos) que este presenta. Para ello es necesario dividir el conjunto de datos disponibles en datos de entrenamiento y prueba. El primero será utilizado para la

construcción del modelo y el segundo para evaluar el desempeño del modelo en datos no conocidos. Sin embargo, ello genera dos problemas. Por un lado, dado que el conjunto de datos de prueba no es utilizado durante el entrenamiento del modelo, si el tamaño del conjunto de datos de entrenamiento es pequeño puede que el modelo no aprenda correctamente las características de la serie de tiempo y produzca malos resultados. Por otro lado, para seleccionar los datos de entrenamiento y prueba se realiza un muestreo estadístico lo cual resulta en que los datos que dispone el investigador son solo una realización posible de una muestra de proceso estocástico. De la misma forma, el error o residual hallado es solo una muestra de una variable aleatoria estocástica que posee sus posibles realizaciones y distribución de probabilidad. Estos dos problemas se relacionan directamente con *adequacy* y *diversity*. El término *adequacy* refiere a que por cada horizonte de tiempo se dispone de suficientes pronósticos mientras que *diversity* significa que el error medido no debe depender de eventos especiales dentro de la serie de tiempo. Por lo tanto, es requisito indispensable que los procedimientos de selección de modelo logren *adequacy* y *diversity*. Para abordar dichos problemas, en los métodos de aprendizaje de máquina supervisado es una práctica habitual utilizar *k-fold cross validation*.

Out of sampling validation consiste en separar un porcentaje del conjunto de datos para evaluar el modelo y el resto para entrenar el modelo. Este porcentaje varía, de acuerdo con el tamaño del conjunto de datos, sin embargo, normalmente se encuentra en el rango de 10% y 30%. Una desventaja que presenta este método es que no permite aprovechar los datos al máximo, pero elimina los problemas teóricos en las series de tiempo respecto a los efectos evolutivos temporales y dependencias dentro de los datos.

Cross validation este método es similar al out of sampling validation. Sin embargo, integra un paso adicional que consiste en cruzar los datos de entrenamiento y validación en rondas sucesivas de tal forma que cada observación tenga la oportunidad de ser validada (Refaeilzadeh et al., 2009). La forma básica de este tipo de validación cruzada es la k-fold cross validation, aunque existen otras variaciones de este método. Estas son casos especiales del k-fold cross validation o involucran rondas repetidas del k-fold cross validation.

K-fold cross validation consiste en dividir el conjunto de datos en k subconjuntos, $S_1 \ldots S_k$, cada uno de ellos recibe el nombre de *fold*, los cuales poseen similar o igual tamaño. El algoritmo de aprendizaje se aplica entonces k veces, para i=1…k, utilizando en cada iteración la unión de todos los subconjuntos diferentes de S_i como datos de entrenamiento y utilizando S_i como datos de prueba (Shultz et al., 2011). El k suele tomar valores como 3, 5 o 10, sin embargo, este depende intrínsecamente del tamaño del conjunto de datos. En la Figura 7, se observa un ejemplo del procedimiento a seguir con un k igual a 3.

No obstante, de acuerdo con Bergmeir y Benítez (2012) este enfoque tradicional del método k-fold cross validation no puede ser aplicado a series de tiempo, pues durante la autoregresión se utilizan los mismos valores tanto como features (*lags*) así como datos de referencia, por ende, los datos de entrenamiento y prueba no son independientes entre sí por más que se escojan al azar. Asimismo, las

series temporales pueden ser generadas por un proceso que evoluciona a lo largo del tiempo lo cual contradice los supuestos fundamentales en los que se basa el método de cross validation que los datos son independientes e idénticamente distribuidos.

Cross validation with omission of dependent data este método aborda los problemas teóricos que presenta el método cross validation respecto a los datos correlacionados a través de dos supuestos. Por un lado, el método asume que la serie de tiempo es estacionaria. Por otro lado, elimina de los datos de entrenamiento no solo los datos que serán usados en el conjunto de datos de prueba, sino que también aquellos puntos u observaciones que no son independientes de los datos de prueba. En la Figura 8, se observa el uso de esta técnica empleando un 5 *folds*. Asimismo, este empleará únicamente los últimos 4 puntos de entrenamiento para realizar el pronóstico del siguiente punto. Sin embargo, con el fin de lograr la independencia de las observaciones, por cada punto que pertenezca al conjunto de datos de prueba, las observaciones que se encuentre en un radio de 4 no podrán ser utilizadas para entrenar el modelo. Ello conduce a que en ciertas áreas casi todos los puntos tengan que ser excluidos del conjunto de datos de entrenamiento. Como es de esperarse, una de las ventajas de este método sería que para casos donde se utilice una gran cantidad de *lags* y *folds*, no se pueda hacer uso de este.

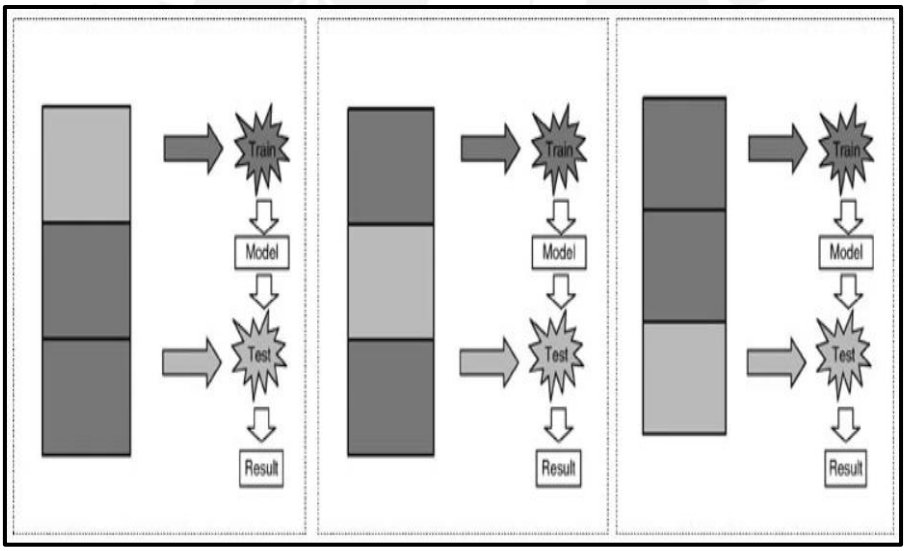


Figura 7: 3-fold cross validation
Fuente: (Refaeilzadeh et al. 2018)

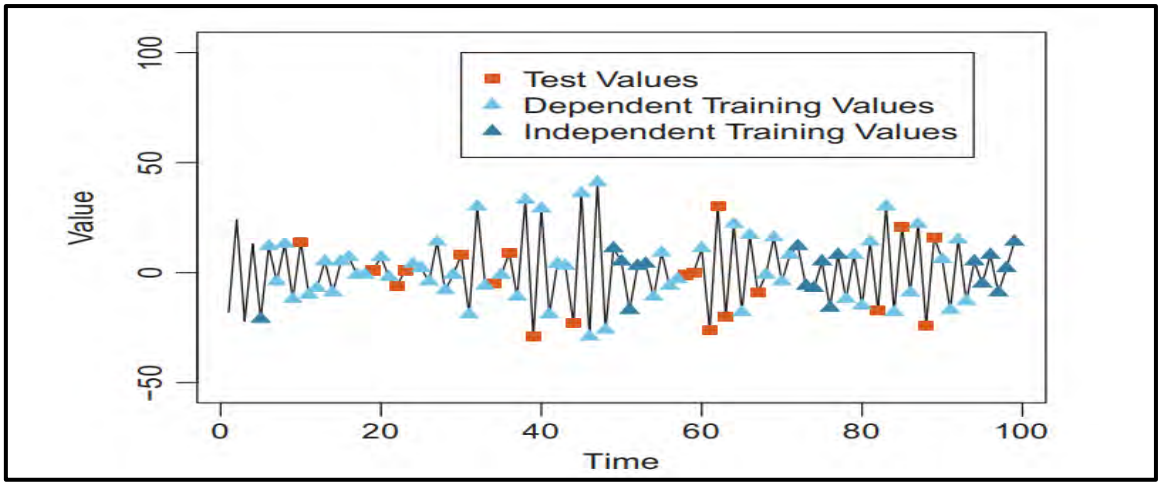


Figura 8: 5-fold cross validation with omission of dependent data
Fuente: (Bergmeir y Benítez 2012:199)

Cross validation with blocked subsets este método consiste en el uso de submuestras de la serie de tiempo sin interrumpir su proceso evolutivo. Racine (2000) propone un método relacionado a este tipo de procedimiento, el cual denomina *hv-block cross-validation*. Este método propuesto por Racine es una extensión del *h block cross-validation*, ya que emplea un bloque de tamaño v, en lugar de un solo valor en la validación o prueba. El autor resalta que el método *hv-block cross-validation* es asintóticamente consistente para procesos estacionarios lo cual significa que a medida que la cantidad de observaciones tiende a infinito, la probabilidad de seleccionar el modelo con la mejor capacidad de predicción converge a 1. No obstante, una de sus desventajas es que requiere que la cantidad de observaciones disponibles sea mayor a 500, lo cual en la práctica no siempre es posible.

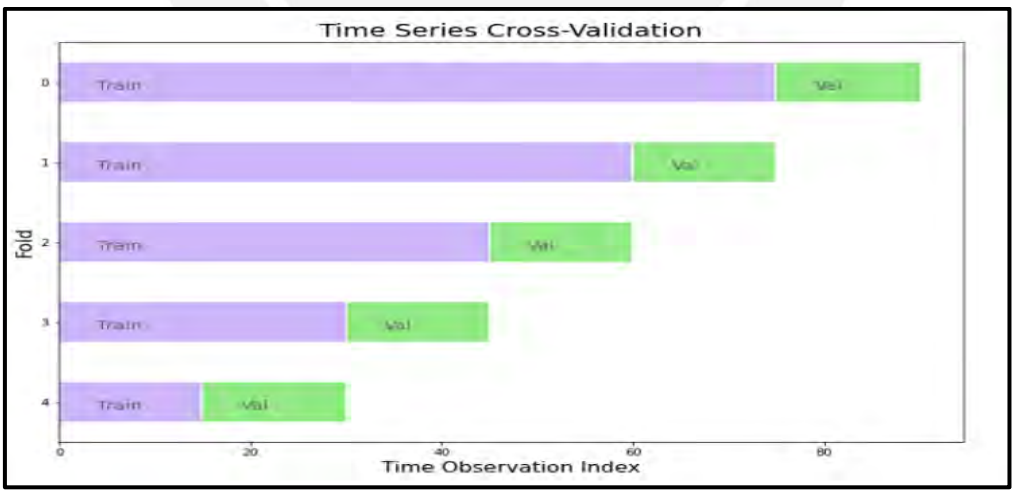


Figura 9: 5-fold blocked cross validation
Fuente: (Keith 2022)

1.3.8. Métricas de error

Se han abordado los modelos de regresión para series de tiempo, así como que metodología emplear para seleccionar entre modelos. Sin embargo, se requiere ser capaz de cuantificar el desempeño

de los modelos. Para ello es requisito indispensable utilizar métricas pues estas cumplen la función de evaluar los modelos ya que capturan el valor del rendimiento que esperamos o buscamos alcanzar. De forma general mientras más pequeño sea en magnitud el valor obtenido en la métrica de error, mejor será el pronóstico, por lo tanto, se deberá iterar alternando los valores asignados a los hiperparámetros del modelo con el fin de poder reducir dicho error. La principal característica diferencial de las métricas de error es que estas buscan expresar la distribución que siguen los residuales, de tal forma que un valor alto indica que el modelo no presenta un buen desempeño mientras que un valor bajo indica lo opuesto. Por ende, se debe buscar un valor bajo asociado a las métricas de error que se planeen utilizar. Auffarth (2021) define el error de pronóstico o residual como la diferencia entre los valores reales y los valores que predice el modelo empleado.

$$e_t = y_t - f(x_t),$$

donde $f(x_t)$ representa el valor pronosticado por el modelo para el instante de tiempo t y y_t denota el valor real para dicho instante de tiempo. Por su parte, comenta que las métricas de error más populares utilizadas en los modelos de series de tiempo son las siguientes:

Error cuadrático medio (MSE) se calcula el residual para cada instante de tiempo, posteriormente se eleva al cuadrado, para que los errores positivos y negativos no se anulen entre sí, ello contribuye a que los errores crezcan de forma cuadrática. Finalmente, se divide entre *N* que represente el número de puntos.

Error absoluto medio (MAE) es similar al MSE solo que en lugar de elevar al cuadrado se aplica el valor absoluto lo que resulta en que los errores contribuyan en proporción lineal. Cabe resaltar que la principal diferencia entre el MSE y MAE es el impacto que genera la presencia de valores atípicos ya que la función cuadrada obligar a dar un mayor a pesos a valores que son muy diferentes, es decir castiga mucho más a los valores extremos lo que resulta en que el MSE sea sensible a valores atípicos a diferencia del MAE que es más robusto en estos casos. Auffarth (2021) resalta que conocer la distribución que siguen los errores es fundamental pues permite seleccionar el error de medida adecuado.

Raíz del error cuadrático medio (RMSE) es equivalente a la desviación estándar y se calcula mediante la raíz cuadrado del MSE.

Desviación mediana absoluta (MdAE) es similar al MAE solo que en lugar de utilizar la media se emplea la mediana como operación de integración. Esto significa que el MdAE es más robusta que la MAE ya que la mediana no está afecta a valores atípicos.

Error absoluto medio porcentual (MAPE) se calcula escalando el MAE por el valor real que se busca predecir.

Error porcentual absoluto medio simétrico (SMAPE) es similar al MAPE solo que emplea la media de la predicción y el valor objetivo como escalamiento.

Error cuadrático medio normalizado (NMSE) se calcula dividiendo el MSE por la varianza. Es útil cuando se busca poder comparar modelos validados en diferentes *datasets* pues escala el desempeño del modelo con la desviación.

Tabla 5 Métricas de regresión más populares

Nombre de la métrica	Fórmula
Error cuadrático medio	$MSE = \frac{1}{N}\sum_{t=1}^{N} e_t^2$
Error absoluto medio	$MAE = \frac{1}{N}\sum_{t=1}^{N} \|e\|_t$
Raíz del error cuadrático medio	$RMSE = \sqrt{MSE}$
Desviación mediana absoluta	$MdAE = mediana(\|e\|_t)$
Error absoluto medio porcentual	$MAPE = \frac{1}{N}\sum_{t=1}^{N} \frac{\|e_t\|}{\|y_t\|}$
Error porcentual absoluto medio simétrico	$SMAPE = \frac{1}{N}\sum_{t=1}^{N} \frac{\|e_t\|}{(\|y_t\| + \|f(x_t)\|)/2}$
Error cuadrático medio normalizado	$NMSE = \frac{MSE}{\sigma^2}$

Fuente: (Auffarth 2021:109-111)

Hewamalage (2022) señala que la clave para seleccionar una medida de error en específico para evaluar los valores pronosticados es que sea simple matemáticamente y robusta para los datos proporcionados. De la misma forma la autora, proporciona un marco de trabajo para seleccionar correctamente la medida o métrica de error según los requerimientos del usuario o características propias de los datos, este puede ser visualizado en el Anexo 1.

1.4. Machine Learning

1.4.1. Definición de machine learning

Machine learning (o aprendizaje de máquina) es un subcampo de la inteligencia artificial que se encarga del desarrollo algoritmos basados en los datos, los cuales son capaces de realizar predicciones de algo en el mundo. De manera más formal, Tom M. Mitchell define los algoritmos de machine learning de la siguiente forma: Se le dice a un programa de ordenador que aprenda de la experiencia (E) con respecto ciertas clases de tareas (T) y se mide su desempeño mediante P, si su desempeño en las tareas en T mejora con la experiencia E. Los algoritmos de *machine learning* pueden ser divididos en 3 grupos: aprendizaje supervisado, aprendizaje no supervisado y aprendizaje por refuerzo. Dado que el objetivo del presente estudio es el de predecir el precio de la criptomoneda de Ethereum, este se

enfocará únicamente en algoritmos de aprendizaje supervisado. Sin embargo, se proporcionará una breve definición de cada uno de estos con el fin de que el lector tenga un panorama general de sus casos de uso.

Aprendizaje supervisado a un sistema inteligente se le presentan datos de entrada (*objects*) y datos de salida (*labels*) donde el objetivo es aprender una función que relacione las entradas con las salidas (Manokhin, 2022).

Aprendizaje no supervisado a diferencia del aprendizaje supervisado, este tipo de algoritmos no cuenta con *labels*, por ende, el objetivo de este algoritmo es el de aprender los patrones o estructura que siguen los datos. El resultado de este tipo de algoritmos es la agrupación de los datos de entrada mediante *clusters*.

Aprendizaje por refuerzo este tipo de algoritmo se ocupa del comportamiento de agentes inteligentes que maximizan su función de recompensa realizando acciones en un entorno (Manokhin, 2022).

1.4.2. Tipos de sistemas de Machine Learning

Existe una gran variedad de sistemas de Machine Learning, por lo cual según Géron (2019) resulta útil clasificarlos en categorías.

- *Supervised versus unsupervised learning*
- *Online versus batch learning*
- *Instance-base versus model-based learning*

Instance-based versus model-based learning

Los sistemas de *Machine Learning* también pueden ser clasificados de acuerdo a como generalizan la información pues gran parte de las tareas que este realiza consiste en hacer predicciones ya sea de clasificación o regresión. Esto quiere decir que el sistema tiene que ser capaz de realizar buenos pronósticos con información nueva. Esto se logra a partir de un previo entrenamiento con información o *data* histórica con el fin de entrenar el sistema para poder desempeñarse en escenarios que no conoce. De acuerdo con Géron (2019), existen dos enfoques principales:

Instance-based learning

Este sistema aprende mediante la memorización de los ejemplos obtenidos a través de los datos de entrenamiento, para posteriormente generalizarlo a nuevos casos utilizando una medida de similitud que le permite comparar estos nuevos ejemplos con los aprendidos previamente (Géron,2019). En la Figura 8, se observa con mayor detalle de qué manera el sistema utiliza la medida de similitud para poder generalizar nuevos escenarios. Como se logra visualizar, el nuevo caso será clasificado como triángulo ya que la gran mayoría de los casos más similares a este pertenecen a dicha clase.

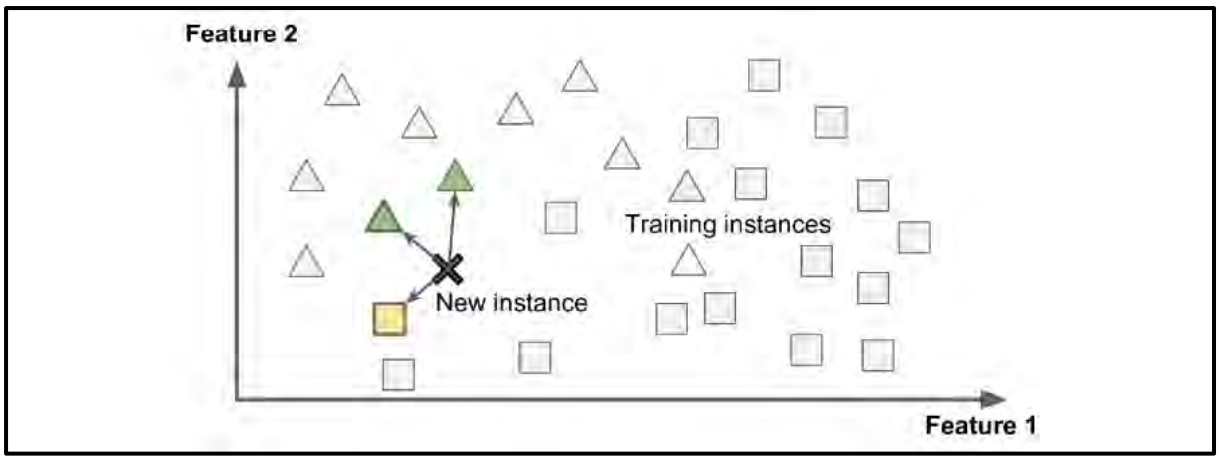


Figura 10: Instance-based learning
Fuente: (A. Géron 2019:18)

Model-based learning

Este sistema se basa en la construcción de un modelo mediante el uso de los datos de entrenamiento, para posteriormente hacer uso de este mismo con el fin de realizar predicciones (Géron, 2019).

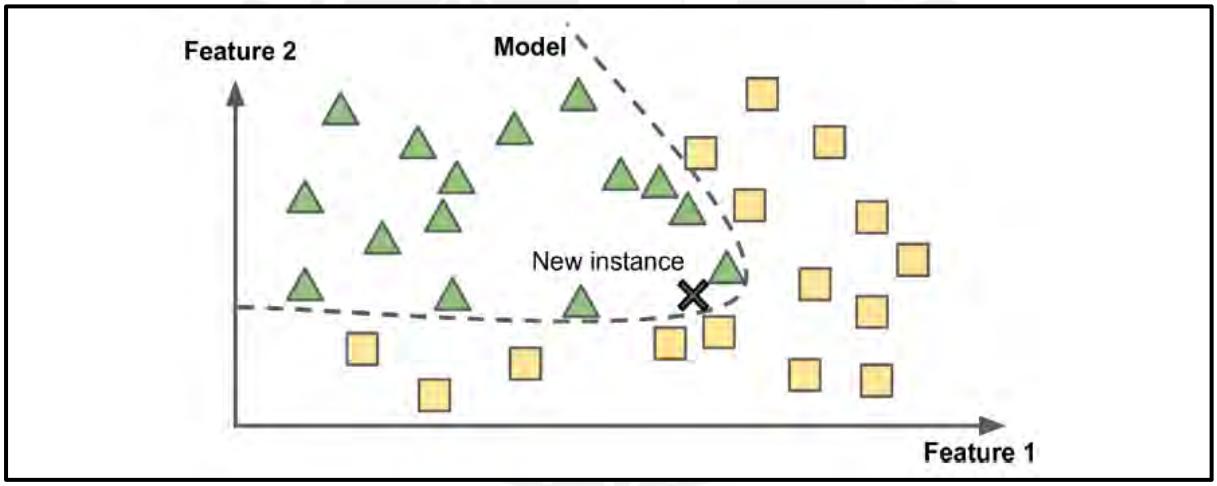


Figura 11: Model-based learning
Fuente: (A. Géron 2019:18)

1.4.3. Principales desafíos del machine learning

De acuerdo con lo mencionado en párrafos anteriores, para poder desarrollar un modelo de *machine learning.* Un requisito previo e indispensable es haber seleccionado un algoritmo de aprendizaje para posteriormente entrenarlo con algunos ejemplos comúnmente conocidos como datos de entrenamiento, ya que los resultados obtenidos del modelo pueden verse afectados por dos factores: seleccionar un mal algoritmo y utilizar malos datos. Por dicho motivo, Géron (2019), divide los problemas asociados a un mal modelo de *machine learning* con base en dichos factores:

Ejemplos asociados a malos datos

- **Cantidad insuficiente de datos de entrenamiento**

La gran mayoría de algoritmos de *machine learning* requieren de muchos datos para trabajar correctamente. Incluso para problemas sencillos se requieren miles de ejemplos y para problemas complejos como reconocimiento de voz o imágenes se necesitan millones de ejemplos. En 2001, Michele Banko y Eric Brill demostraron que los diferentes algoritmos de aprendizaje, incluso los más simples, lograban un desempeño similar en problemas complejos, una vez se les haya proporcionado una gran cantidad de datos (Ver Figura 9). Por ello, como regla general un algoritmo tonto con una gran cantidad de datos puede superar a uno inteligente con una cantidad modesta de estos (Pedro, 2012). Además, Pedro menciona que todos los algoritmos de aprendizaje funcionan esencialmente de la misma forma, agrupando los ejemplos cercanos en la misma clase, sin embargo, la principal diferencia en estos recae en el significado de "cercano" (Ver Figura 10).

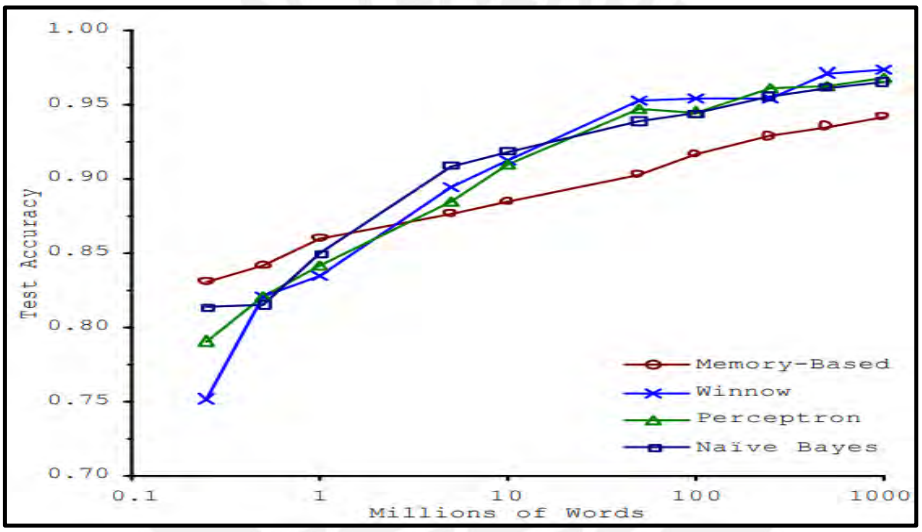


Figura 12: Importancia de los datos frente los algoritmos
Fuente: (Banko y Brill 2001:2)

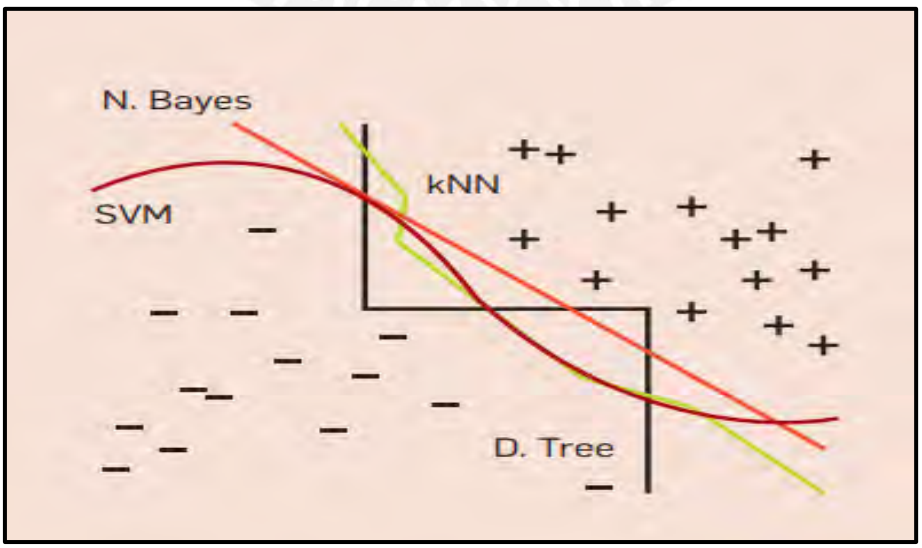


Figura 13: Fronteras delimitadas por algoritmos de aprendizaje diferentes pueden dar predicciones similares

Fuente: (Pedro 2012:8)

- **Datos de entrenamiento no representativos**

Para generalizar bien se requiere que los datos de entrenamiento sean representativos de los nuevos ejemplos que se busca predecir. Esto usualmente es complicado ya que existen 3 tipos de sesgo muestral:

1. *Sampling noise*. – Sucede cuando la muestra es muy pequeña.
2. *Sampling bias*. – Ocurre cuando el método o técnica de muestreo utilizado es defectuoso.
3. *Nonresponse bias*. – Se da cuando los que no respondieron la encuesta son sistemáticamente diferentes de los que sí lo hicieron lo cual produce que los resultados obtenidos difieran de manera significativa.

- **Datos de mala calidad**

Si los datos de entrenamiento presentan errores, datos atípicos y ruido debido a malas prácticas o mala calidad de las mediciones. Ello conllevará a que el modelo no pueda detectar los patrones correctamente lo cual afectará el desempeño de este al momento de predecir nuevos escenarios.

- **Features o características irrelevantes**

De acuerdo con Pedro (2012), el factor más importante que determina si un proyecto de *machine learning* tiene éxito o fracasa son los *features* utilizados. Por su parte, Géron (2019) menciona que a esto se le conoce como ingeniería de características (o *feature engineering*), este proceso involucra los siguientes pasos:

1. *Feature selection*. – Se basa en seleccionar las *features* más útiles entre todas las existentes para el entrenamiento del modelo.
2. *Feature extraction*. – Se basa en combinar *features* existentes para obtener una más útil.
3. Creación de nuevos *features* mediante la recolección de nuevos datos.

Ejemplos asociados a seleccionar un mal algoritmo

- **Sobreajuste**

Esto ocurre cuando un modelo tiene un buen desempeño en los datos de entrenamiento, pero no se desempeña bien en nuevos ejemplos. Géron (2019) propone algunas soluciones para este tipo de problema:

- Seleccionar un modelo más simple con menos parámetros, esto se logra reduciendo la cantidad de atributos o restringiendo el modelo.
- Recolectar más datos de entrenamiento.
- Reducir el ruido en los datos de entrenamiento.

- **Subajuste**

Esto ocurre cuando el modelo utilizado es muy simple para lograr aprender los patrones subyacentes de los datos. Géron (2019) propone algunas soluciones para este tipo de problema:

- Seleccionar un modelo más potente con mayores parámetros.
- Entrenar el modelo con mejores características o *features*.
- Reducir las restricciones del modelo

1.4.4. Ingeniería de características en series de tiempo

Ingeniería de características es el proceso que se encarga de extraer características de los datos utilizando sobre todo conocimientos del negocio con el fin de hacer que el proceso de aprendizaje sea más fluido y eficiente para el algoritmo. Crear buenas características es esencial para obtener o mejorar el buen desempeño de cualquier modelo de *machine learning*. De acuerdo con Joseph (2022), esta es una etapa clave en el proceso pues que tan bien el modelo de *machine learning* aprenda sobre el tiempo depende de lo bien que se creen o diseñen las características para capturarlo. Asimismo, según Joseph existen dos ideas principales para integrar el tiempo en los modelos de regresión:

Time delay embedding consiste en integrar el tiempo en términos de observaciones previas a través de *lags*.

- *Lags* o *backshift*. – son observaciones previas de la serie, las cuales son utilizadas para pronosticar el valor futuro (Ver Figura 14).

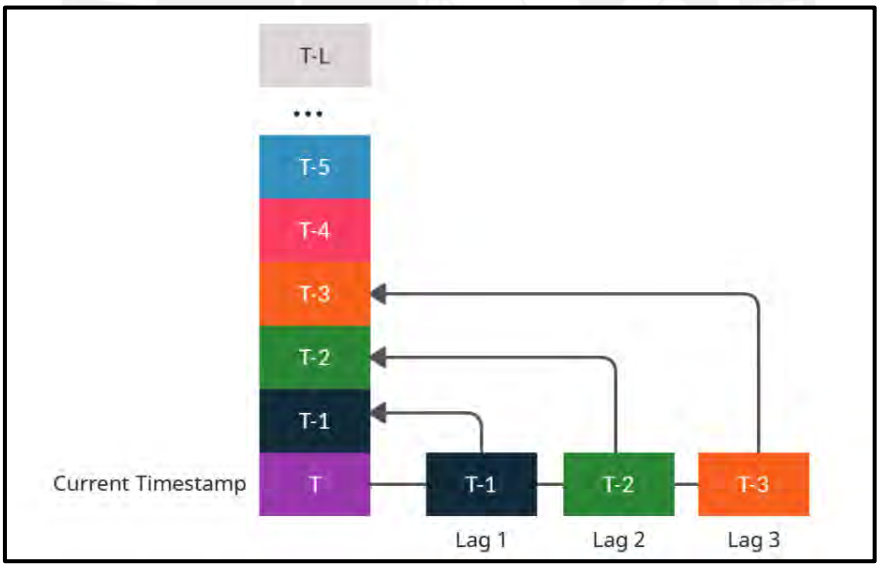


Figura 14: *Lag features*
Fuente: (Joseph 2022:125)

- *Rolling window*. – son agregaciones estadísticas de un grupo de observaciones previas de la serie (Ver Figura 15), estas agregaciones estadísticas pueden ser el promedio, desviación estándar, mínimo, máximo, etc.

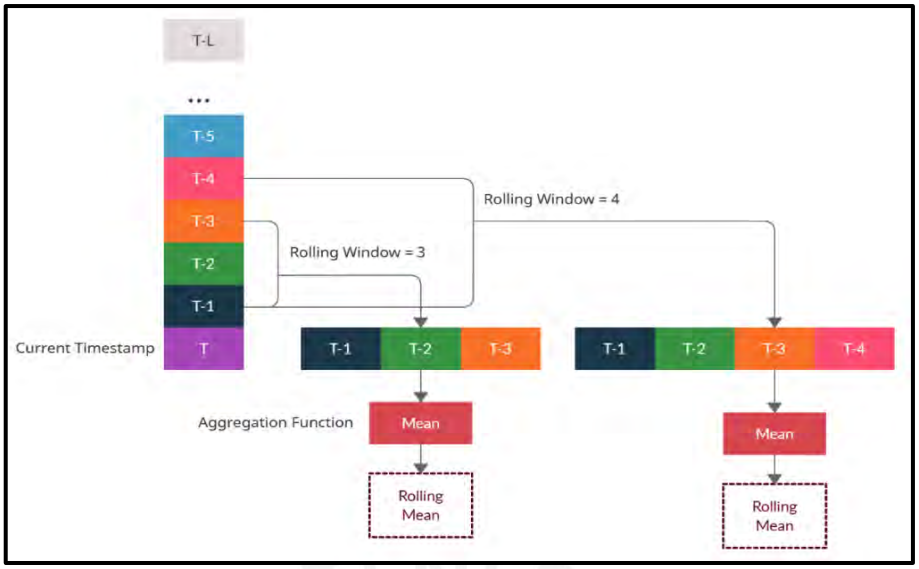


Figura 15: *Rolling window*
Fuente: (Joseph 2022:127)

- *Seasonal rolling window*. – es similar al rolling window tradicional, pero a diferente del primero, esta toma ventanas de tiempo estacionales omitiendo un número constante entre cada elemento de una ventana (Ver Figura 16).

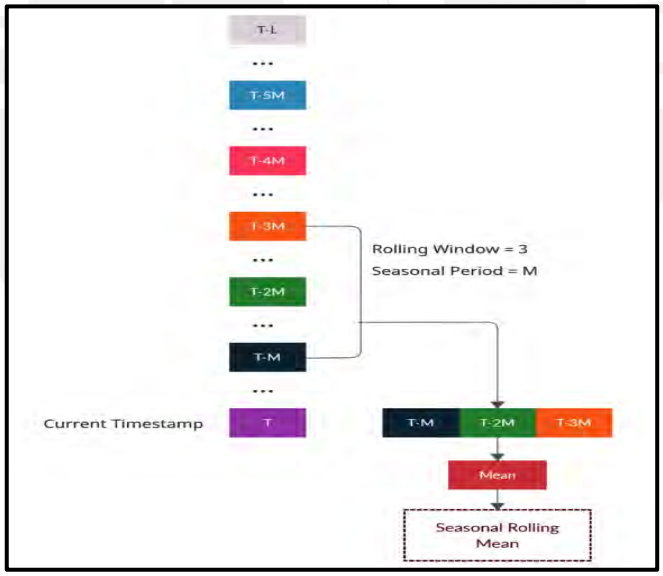


Figura 16: *Seasonal rolling window*
Fuente: (Joseph 2022:129)

Temporal embedding permite integrar el tiempo como *features* que el modelo de *machine learning* pueda aprovechar, pues en las series de tiempo hay dos aspectos importantes sobre el tiempo que son necesarios capturar: el paso del tiempo y la periodicidad de este mismo.

- Características de calendarios. - consiste en extraer *features* del tiempo, dado que las series de tiempo la mayoría de veces tienen indexado una marca de tiempo es posible extraer de esta información como el año, mes, semana, día, hora, minutos, etc. Este tipo de características ayuda a

capturar la periodicidad del tiempo y con ello al modelo de *machine learning* a capturar la estacionalidad presente en la serie de tiempo.

- *Time elapsed.* - ayuda a capturar el paso del tiempo, pues esta característica aumenta a medida que pasa el tiempo lo cual le permite al modelo de *machine learning* a tener una noción de este.
- Términos de Fourier. – refiere a la descomposición de una señal periódica en términos de funciones sinusoidales y cosinusoidales, se representa de la siguiente manera:

$$S_{N(x)} = \frac{a_0}{2} + \sum_{n=1}^{N} \left(a_n \cdot \cos\left(\frac{2\pi}{P} \cdot n \cdot x\right) + b_n \cdot \sin\left(\frac{2\pi}{P} \cdot n \cdot x\right)\right),$$

donde s_N es la aproximación del término N a la señal S. De forma teórica, cuando N tiene al infinito la aproximación resultante es igual a la señal original. P es la longitud máxima del ciclo.

1.4.5. Pronóstico de series de tiempo mediante machine learning

Se comenzará definiendo un modelo clásico y conocido por la gran mayoría, la **regresión lineal**, esta pertenece al grupo de familia de funciones que siguen la siguiente forma:

$$\hat{y} = \beta_0 + \sum_{i=1}^{k} X_i \beta_i,$$

donde k representa el número de *features* en el modelo y β son los parámetros del modelo. Existe un β_i por cada *feature* y un único β_0 que se conoce como intercepto. Los parámetros β pueden ser hallados a través de métodos de optimización como el de vector gradiente descendente pero el método más popular es el uso del método MCO (Mínimos Cuadrados Ordinarios). Para que los resultados obtenidos por el modelo de regresión lineal sean válidos deben cumplir los siguientes supuestos:

- La relación entre la variable independiente (*features*) y la variable dependiente (o *target variable*) debe ser lineal.
- Los errores o residuales siguen una distribución normal.
- La varianza de los errores es constante en todos los valores de la variable independiente, es decir, presenta homocedasticidad.
- Los errores no presentan autocorrelación
- Hay poca o no existe correlación entre las variables independientes (multicolinealidad).

La **regresión lineal regularizada**, la regularización es un tipo de restricción que se implementa durante el proceso de aprendizaje con el fin de reducir la complejidad del modelo y aumentar los grados de libertad de este. La forma en que se implementa en los modelos de regresión lineal es a través de la técnica *weight decay* donde se agrega un término λ que penaliza la magnitud de los coeficientes en la función de pérdida (o *loss function*).

La función de pérdida suma residual de cuadrados (o *Residual Sum of Squares*) que se utiliza para hallar los parámetros β pasa a ser la siguiente:

$$RSS = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 + \lambda W,$$

donde W es el *weight decay* y $\lambda > 0$ es el grado de regularización, es fácil notar que si λ es igual a 0, entonces se trata de una regresión lineal sin regularización. W es la norma de la matriz de pesos. En algebra lineal, la norma de una matriz indica el tamaño de sus elementos. Existen muchas normas para una matriz, pero las dos más comunes que se utilizan para regularización son $L1$ y $L2$. Cuando se utiliza la norma $L1$ para la regularizar la regresión lineal se denomina regresión Lasso mientras que si se utiliza la norma $L2$ se denomina regresión Ridge. La norma $L1$ se define como la suma de los valores absolutos de la matriz:

$$W = \sum_{i=1}^{k} |\beta_i|$$

Por su parte la norma $L2$ se define como la suma de los valores cuadrados de la matriz:

$$W = \sum_{i=1}^{k} \beta_i^2$$

Existe otra familia de funciones que son mucho más expresivas que una función lineal, se trata de los **árboles de decisión**. Estos dividen el espacio de *features* (o *feature space*) en diferentes subespacios y ajustan un modelo muy simple a cada uno de ellos (Ver Figura 17).

Figura 17: *Feature space* particionado por un árbol de decisión
Fuente: (Joseph 2022:183)

Para poder hallar el número de subespacios adecuado Joseph (2022), menciona que el árbol de decisión crea un conjunto de reglas de decisión (Ver Figura 18) e intenta hallar la mejor forma de dividir el *feature space* de tal forma que se maximice la homogeneidad de la variable objetivo dentro de la división o subespacio.

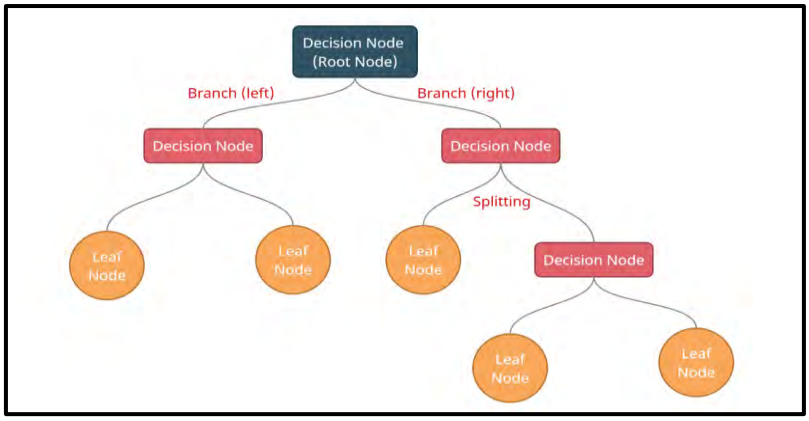


Figura 18: Partes de un árbol de decisión
Fuente: (Joseph 2022:184)

Formalmente se puede definir la función que ha sido generada por un árbol de decisión que posee M particiones de la siguiente forma:

$$\hat{y} = \sum_{m=1}^{M} c_m I(x \in P_m),$$

donde x es el *feature*, c_m es la constante de respuesta para la partición o división P_m y I es una función que resulta en 1, si $x \in P_m$, caso contrario el valor será 0.

El modelo de **bosques aleatorios** se basa en una modificación del *bagging*, el cual es una forma de *ensemble learning* que utiliza el muestreo repetido con remplazo de una población para extraer diferentes subconjuntos del conjunto de datos con el fin de entrenar *weak learners*, modelos cuyo desempeño es ligeramente superior al de adivinar al azar, en cada uno de los subconjuntos y combinarlos mediante votación o promediándolos. De acuerdo con Joseph (2022), el modelo de bosques aleatorios fue desarrollado por Leo Brieman en 2001 y la diferencia respecto al *bagging* recae en que modifica el procedimiento de construcción de los árboles con el fin de asegurar que todos los árboles no estén correlacionados entre sí.

Un parámetro principal en el algoritmo de bosques aleatorios es la elección del número de árboles,M, a construirse. Para cada árbol se repiten los siguientes pasos:

1. Extraer una muestra con remplazo del conjunto de datos de entrenamiento (o *training set*).
2. Seleccionar f *features* al azar del total disponible.
3. Escoger la mejor división empleando únicamente las *features* seleccionadas previamente y dividir el nodo en dos nodos hijos.
4. Repetir los pasos 2 y 3 hasta que se alcancen alguno de los criterios de finalización definidos.

La principal característica de este tipo de modelo es que gracias al muestreo aleatorio de *features* en cada división, ello incrementa la aleatoriedad y reduce la correlación entre los resultados de los diferentes árboles. Para realizar la predicción se utiliza cada uno de los M árboles para obtener un resultado. No obstante, en problemas de regresión el resultado final se obtiene mediante el promedio de

las predicciones de los M árboles. En el caso de problemas de clasificación, el resultado final se obtiene mediante un proceso de votación donde se escoge el que presenta mayor cantidad de votos. El modelo de bosques aleatorios puede ser representado de la siguiente forma:

$$\hat{y} = \frac{1}{M}\sum_{t=1}^{M} \tau_t(x),$$

donde $\tau(x)$ representa el resultado del t-ésimo árbol del bosque aleatorio.

De forma similar al *bagging*, existe otro método que utiliza el *boosting* para construir un modelo final robusto a través de *weak learners*. La principal diferencia radica en la forma en la que se utilizan los *weak learners*. En el caso del *boosting*, utiliza los *weak learners* de forma secuencial, es decir, entrenando a cada *weak learner* con diferentes versiones de datos modificados de forma repetitiva. El método de *gradiente boosting decision trees* puede ser formulado de la siguiente manera, considerando el caso de que se utilicen árboles de decisión como *weak learners:*

$$\tau(x) = \sum_{m=1}^{M} c_m I(x \in P_m)$$

$$\hat{y} = \sum_{k=1}^{M} \tau_k(x)$$

De acuerdo con Joseph (2022), existen muchas implementaciones de *regression gradient-boosted trees*, las más populares son las siguientes:

- XGBoost por T Chen
- LightGBM por Microsoft
- CatBoost por Yandex
- GradientBoostingRegressor y HistGradientBoostingRegressor en Scikit-Learn

1.5. *Conformal Prediction*

La predicción consiste en realizar pronósticos sobre el futuro, mientras que la predicción probabilística trata de cuantificar la incertidumbre de un pronóstico (Gneiting y Katzfuss, 2014). Los pronósticos probabilísticos son un componente importante en el proceso óptimo de la toma de decisiones y son utilizados para cuantificar la incertidumbre en distintos campos como el pronóstico del clima, resultados de elecciones, proyecciones demográficas y gestión del riesgo financiero. Un claro ejemplo de ello es el Banco de Inglaterra pues según Manokhin (2022), el Comité de Política Monetaria del Banco de Inglaterra ha hecho uso de modelos probabilísticos para los tipos de interés durante casi dos décadas (Ver Figura 19). Cabe resaltar de que luego de esto, los bancos centrales del mundo también adoptaron dicho enfoque. Manokhin (2022), menciona que no solo los bancos centrales del mundo están haciendo uso de modelos probabilísticos, sino que también grandes empresas tecnológicas como Amazon y Uber.

El objetivo general de la predicción probabilística es el de maximizar la precisión de una distribución predictiva sujeta a calibración de la predicción probabilística (Gneiting y Katzfuss, 2014).

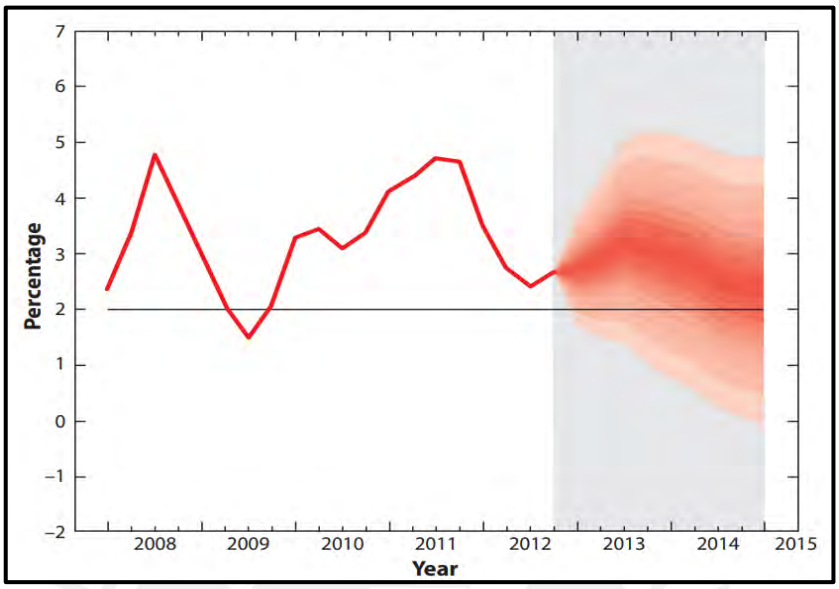


Figura 19: Pronóstico de febrero de 2013 del Banco de Inglaterra sobre la inflación en el Reino Unido
Fuente: (Gneiting y Katzfuss, 2014:127)

1.5.1. Definición de *conformal prediction*

Es un método que te permite convertir una predicción (o *point prediction*) en un conjunto de puntos (o *prediction set*) para cualquier modelo. De forma general, este método funciona de la siguiente manera. Primero, se requiere de un modelo de predicción previamente entrenado, el cual se denominará $\hat{f}$. Posteriormente, se realizará la calibración para ello es necesario un conjunto de datos nuevos que no hayan sido utilizados para el entrenamiento del modelo $\hat{f}$ con el fin de poder predecir un conjunto de puntos que posiblemente contenga en valor real. En términos formales conformal prediction puede ser expresado de la siguiente forma, utilizando $\hat{f}$ y los datos de calibración, se busca construir una función C que cumpla las siguientes condiciones:

$$1 - \alpha \leq \mathbb{P}\big(Y_{test} \in C(X_{test})\big) \leq 1 - \alpha + \frac{1}{n+1},$$

donde (X_{test}, Y_{test}) es un par de datos de prueba de la misma distribución o proceso de generación de datos, y α es el nivel de significancia que va desde 0 hasta 1. Es decir, la probabilidad de que el conjunto de datos predichos por la función C contengan el valor real es aproximadamente $1 - \alpha$. A esta propiedad se le conoce como cobertura marginal (o *marginal coverage*), dado que la probabilidad se promedia sobre la aleatoriedad en los puntos de calibración y prueba. Por su parte, Angelopoulos y Bates (2022) definen conformal prediction como un método para tomar cualquier noción heurística de incertidumbre de cualquier modelo y convertirla en rigurosa (Ver Figura 20). La principal característica de este método

es que puede ser utilizado independientemente del modelo $\hat{f}$ que se utilice o la distribución que siguen los datos.

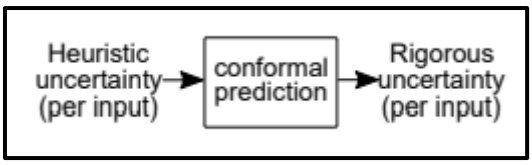


Figura 20: Conformal prediction
Fuente: (Angelopoulos y Bates, 2022:5)

1.5.2. *Conformal prediction* para problemas de clasificación

Angelopoulos y Bates (2022), esbozan los pasos a seguir para llevar a cabo *conformal prediction* para cualquier tipo de tarea ya sea de clasificación o regresión:

1. Identificar la noción heurística de incertidumbre del modelo entrenado $\hat{f}$.
2. Definir la función de puntaje (o *score function*) $s(x, y) \in \mathbb{R}$.
3. Calcular $\hat{q}$ mediante $\frac{\lceil (n+1)(1-\alpha) \rceil}{n}$, el cual define el cuantil de las puntuaciones de calibración.
4. Utilizar $\hat{q}$ para construir la función $C(X_{test}) = \{y: s(X_{test}, y) \leq \hat{q}\}$.

El método de clasificación simple define como *score function*:

$$s(x_i, y_i) = 1 - \hat{f}(X_i)_{Y_i},$$

donde $\hat{f}(X_i)$ representa el puntaje arrojado por la función de activación *softmax* de la verdadera clase. De ello, se puede intuir que cuando el puntaje asociado al *score function* es alto se debe a que la probabilidad estimada por la función de activación *softmax* es baja, es decir al modelo $\hat{f}(X_i)$ se le hace difícil reconocer a qué clase pertenece dicha observación. Luego, se calcula $\hat{q}$ para ello se calculan los puntajes de todos los datos de calibración empleando el *score function* definido. Posteriormente, se ordenan de menor a mayor (Ver Figura 21) donde como se explicó previamente un puntaje alto indica una alta incertidumbre y viceversa.

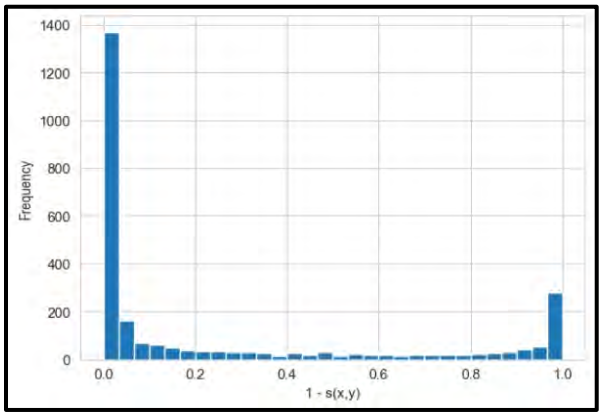


Figura 21: Distribución de los puntajes del *score function*
Fuente: (Molnar, 2022)

El corte o umbral $\hat{q}$ se debe escoger de tal forma que proporcione cobertura por lo menos al $1 - \alpha$ de las clases verdaderas. No obstante, para algunos puntos u observaciones existirá el caso en el cual más de una clase sea mayor a $1 - \hat{q}$ (Ver Figura 22).

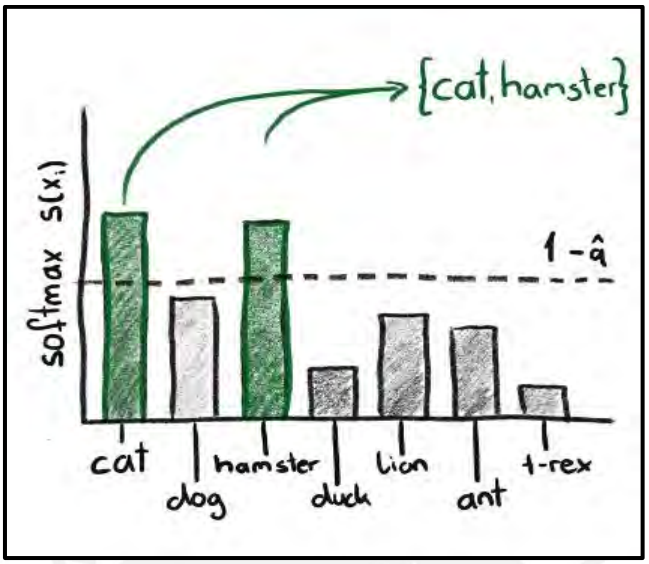


Figura 22: Prediction set para clasificación
Fuente: (Molnar, 2022)

Por último, se construye la función C que incluye todas las clases que presenten una probabilidad estimada suficientemente alta.

$$C(X_{test}) = \{y : \hat{f}(X_{test})_y \geq 1 - \hat{q}\}$$

El método de clasificación con ***Adaptive Prediction Sets*** resuelve el inconveniente que presentaba el método previo, el cuál tendía a no cubrir correctamente los subgrupos difíciles de identificar y sobre cubrir los fáciles. Si bien los métodos de *conformal prediction* garantizan la cobertura marginal, estos no garantizan la cobertura condicional (o *conditional coverage*). Los predictores adaptativos aproximan la cobertura condicional.

De manera similar que el método anterior, el primer paso a realizar es identificar la noción heurística de incertidumbre del modelo. Según Angelopoulos y Bates (2022), el modelo puede ser representado mediante el siguiente algoritmo:

$$\{\pi_1(x), \ldots, \pi_k(x)\}, k = sup\left\{k' : \sum_{j=1}^{k'} \hat{f}(X_{test})_{\pi_{j(x)}} < 1 - \alpha\right\} + 1,$$

donde $\pi(x)$ es la permutación de las K clases que ordena $\hat{f}(X_{test})$ de más probable a menos probable. Sin embargo, en la práctica este modelo fallaría en proveer cobertura marginal, dado que $\hat{f}(X_{test})$ no es perfecto solo proporciona una noción heurística de incertidumbre. Por lo tanto, se utilizará conformal prediction para convertir la noción heurística de incertidumbre en una rigurosa. Para ello es necesario definir el non-conformity score, el cual puede ser expresado de la siguiente forma:

$$s(x,y) = \sum_{j=1}^{k} \hat{f}(x)_{\pi_j(x)}, y = \pi_k(x)$$

En términos generales, se incluyen clases en el conjunto hasta llegar a la clase verdadera (Ver Figura 18). Posteriormente, se calcula $\hat{q}$ y se construye la función C.

$$C(x) = \{\pi_1(x), \dots, \pi_k(x)\}, k = sup\left\{k' : \sum_{j=1}^{k'} \hat{f}(x)_{\pi_{j(x)}} < \hat{q}\right\} + 1$$

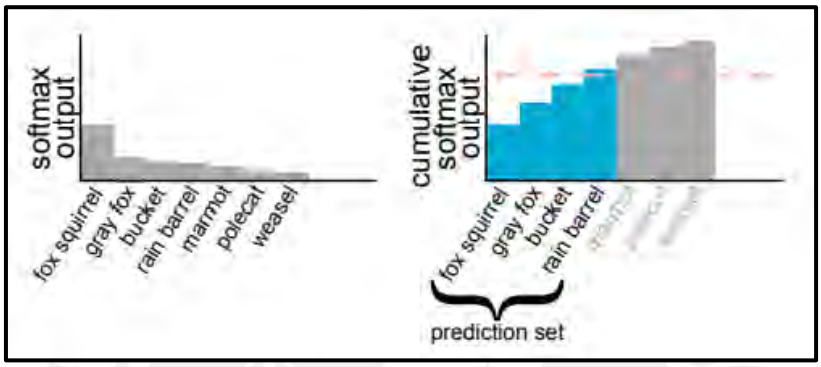


Figura 23: APS (*Adaptive Prediction Sets*)
Fuente: (Angelopoulos y Bates, 2022:7)

1.5.3. *Conformal prediction* para problemas de regresión

Conformal prediction también puede ser aplicado a problemas de regresión y se trabaja de forma similar que los problemas de clasificación lo único que varía es la forma en la que se define el *score function* y que los *prediction sets* no son un conjunto de clases, sino que en su lugar producen un intervalo que garantiza la cobertura del valor real en futuras observaciones. Existen dos formas de crear *prediction intervals*:

El primer método *Conformalized mean regression* se enfoca en transformar un punto (o *point prediction*) en un *prediction Interval* (Ver Figura 19).

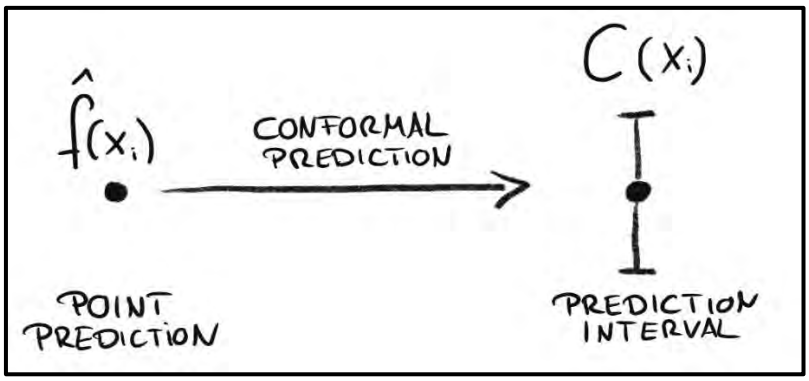


Figura 24: *Conformalized mean regression*
Fuente: (Molnar, 2022)

Como se mencionó en líneas previas la principal diferencia recae en la definición del *score function*. *Conformalized mean regression* define su *score function* de la siguiente forma:

$$s(x_i, y_i) = |y_i - \hat{f}(x_i)|,$$

donde $\hat{f}(x_i)$ es el valor pronosticado por el modelo y y_i el valor real. Conformalizar este *score* significa encontrar el punto de corte $\hat{q}$ de modo que el $1 - \alpha$ de los pronósticos tengan un puntaje por debajo y el α un puntaje superior. Este método trata ambas direcciones por igual, es decir, no se adapta a las distintas regiones del *feature space*.

El segundo método conocido como *Conformalized Quantile Regression* transforma una regresión cuantil en un *prediction Interval* (Ver Figura 20). A diferencia del método *Conformalized mean regression*, este método sí proporciona intervalos de predicción adaptativos.

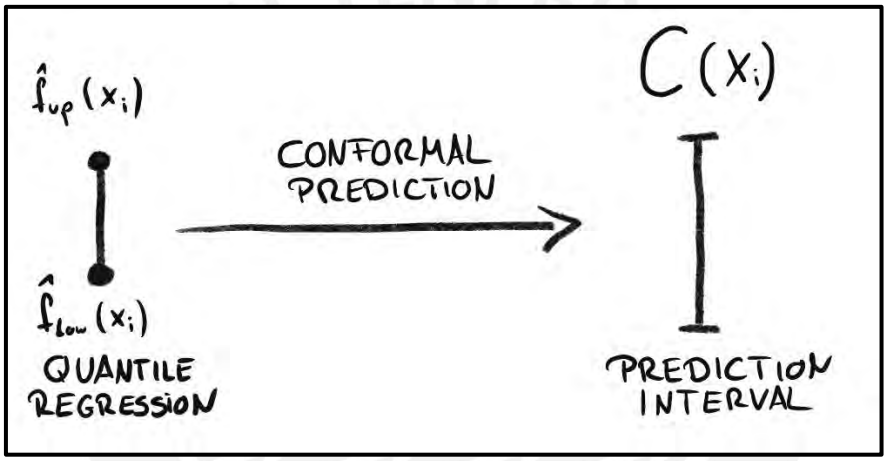


Figura 25: *Conformalized Quantile Regression*
Fuente: (Molnar, 2022)

El *score function* de este método puede ser expresado de la siguiente manera:

$$s(x_i, y_i) = max\left(\hat{q}_{low}(x_i) - y_i, y_i - \hat{q}_{up}(x_i)\right)$$

De esta expresión se puede observar que si el valor real y_i cae fuera del intervalo entonces el *score* $s(x_i, y_i)$ será positivo y si cae dentro será negativo. El punto de corte $\hat{q}$ puede ser interpretado como el término por el que debe ampliarse o acortarse el intervalo. Un $\hat{q}$ mayor a 0 indica que los intervalos originales eran muy cortos y un $\hat{q}$ menor a 0 indica que los intervalos eran muy amplios.

1.5.4. Evaluación de *conformal prediction*

En los párrafos anteriores, se ha descrito cómo formar conjuntos de predicción (o *prediction sets*) válidos en el caso de problemas de clasificación e intervalos de predicción (o *prediction interval*) válidos en el caso de problemas de regresión, pero aún no se sabe cómo evaluarlos. Por dicho motivo, esta sección abordará dicha problemática, la cual se divide en dos categorías: evaluación de la adaptabilidad y verificación de la cobertura.

Evaluación de la adaptabilidad, Angelopoulos y Bates (2022) proponen las siguientes métricas para evaluar un procedimiento conforme (o *conformal procedure*).

1. *Set size*. - es necesario graficar un histograma de los *sets sizes* que produce el *conformal procedure*. De esta forma, se podrá visualizar mediante el tamaño de las barras, si el procedimiento conforme es preciso o no, ya que un tamaño promedio alto de estas indica un problema con el *score* o modelo subyacente. Por otro lado, la distribución de las barras muestra si los conjuntos de predicción se adaptan adecuadamente a la dificultad de los ejemplos.

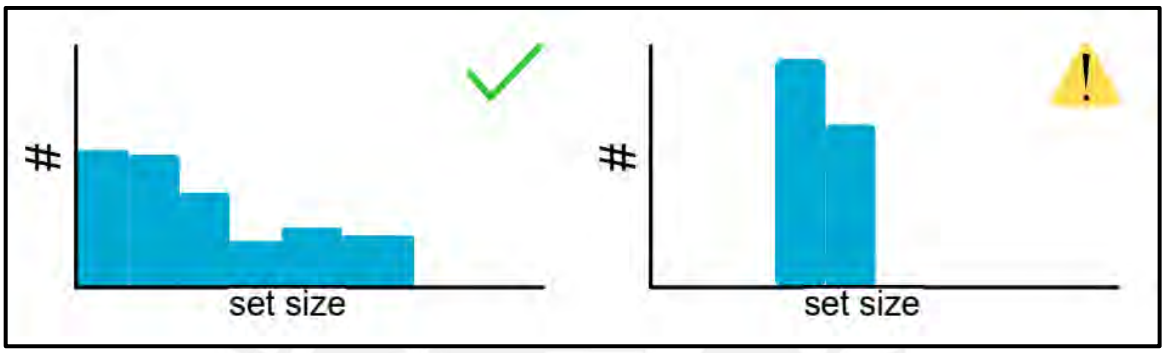


Figura 26: Evaluación del set size de un procedimiento conforme
Fuente: (Angelopoulos y Bates, 2022:12)

Una buena distribución de las barras usualmente es deseable, pero ello no necesariamente indica que los conjuntos se adapten adecuadamente a la dificultad de los ejemplos. Por dicho motivo, se requiere una métrica que permita verificar que se producen conjuntos grandes únicamente para los ejemplos difíciles de identificar.

2. *Conditional coverage*. – la adaptabilidad suele ser la consecuencia de la propiedad de la cobertura condicional:

$$\mathbb{P}[Y_{test} \in C(X_{test}) \mid X_{test}] \geq 1 - \alpha$$

Esa expresión indica que por cada observación de X_{test} se busca devolver un conjunto de predicción con una cobertura del $1 - \alpha$. Sin embargo, en el caso más general es imposible lograr una cobertura condicional, por lo cual se debe verificar que tanto el procedimiento conforme se aproxima a esta.

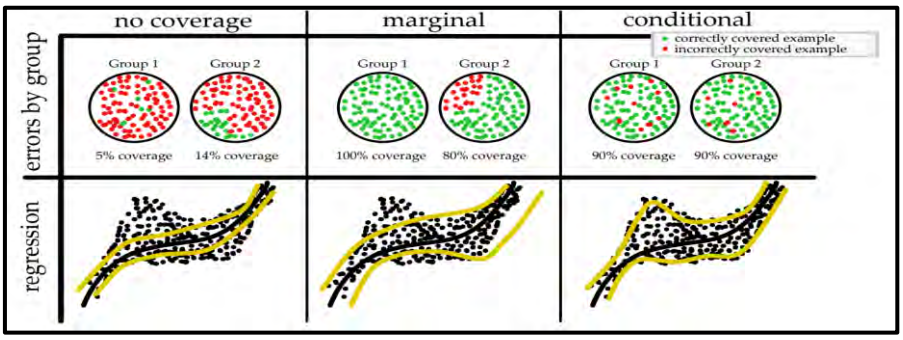


Figura 27: Diferencias entre cobertura marginal y condicional
Fuente: (Angelopoulos y Bates, 2022:13)

Una de las métricas para evaluar la cobertura condicional es *feature-stratified coverage.* En términos generales, es la cobertura observada entre todos los casos en los que pertenecen a la clase g. Si se logrará la cobertura condicional, esta sería de $1 - \alpha$ y valores muy por debajo de $1 - \alpha$ indicarían un mayor incumplimiento a la cobertura condicional. Considerar que esta métrica solo puede ser utilizada con una *feature* continua, pero es necesario agruparla o discretizarla con el fin de conseguir un número finito de categorías. Está métrica puede ser expresada de la siguiente forma:

$$FSC: \ min_{g \in \{1,\dots,G\}} \frac{1}{|\mathcal{L}_g|} \sum_{i \in \mathcal{L}_g} 1\left\{Y_i^{(val)} \in C\left(X_i^{(val)}\right)\right\},$$

donde la matriz $X_{i,1}^{(val)}$ representa los *features* y G las clases o categorías. El índice i indica la posición de la observación del conjunto de validación y 1 expresa la primera coordenada de cada *feature* lo cual indica el grupo o clase a la cual pertenece. $\mathcal{L}_g$ expresa la cantidad de observaciones que $X_{i,1}^{(val)} = g$ para todo g que se encuentre en $1 \leq g \leq G$.

Otra métrica utilizada para evaluar la cobertura condicional es *size- stratified coverage*. De forma general, esta métrica es la cobertura observada para todas las observaciones cuyo *set size* cae dentro de un mismo intervalo (o *bin*) g. En términos formales, esta métrica se define de la siguiente manera:

$$SSC: \ min_{g \in \{1,\dots,G\}} \frac{1}{|\mathcal{L}_g|} \sum_{i \in \mathcal{L}_g} 1\left\{Y_i^{(val)} \in C\left(X_i^{(val)}\right)\right\},$$

donde G representa la cantidad de *bins* en los cuales ha sido discretizado $C(x)$ y $\mathcal{L}_g$ indica la cantidad de observaciones que caen dentro del *bin* $B_1 \dots B_G$. Se observa que está métrica es similar a la expresada anteriormente (*feature-stratified coverage)*, la diferencia recae en como se define $\mathcal{L}_g$, en este caso el usuario no tiene que definir un conjunto importante de *features* discretas a priori, lo cual quiere decir que puede aplicarse a cualquier ejemplo.

Verificación de la cobertura, para poder llevar a cabo este procedimiento es necesario realizar R ensayos sobre la función $\mathrm{C}(x)$ (o *conformal procedure*) con nuevos conjuntos de datos de calibración y validación, y calcular la cobertura empírica para cada uno.

$$\mathrm{C}_i = \frac{1}{n_{val}} \sum_{i=1}^{n_{val}} 1\left\{Y_{i,j}^{(val)} \in C_j\left(X_{i,j}^{(val)}\right)\right\}, \forall\, j = 1 \dots R,$$

donde n_{val} es el tamaño del conjunto de datos de validación y $\left(X_{i,j}^{(val)}, Y_{i,j}^{(val)}\right)$ es la observación i del conjunto de datos de validación del ensayo j. C_j es calibrado utilizando el conjunto de datos de calibración del ensayo j. Según Angelopoulos y Bates (2020), el histograma de los C_j debe centrarse en aproximadamente $1 - \alpha$ (Ver Figura 23).

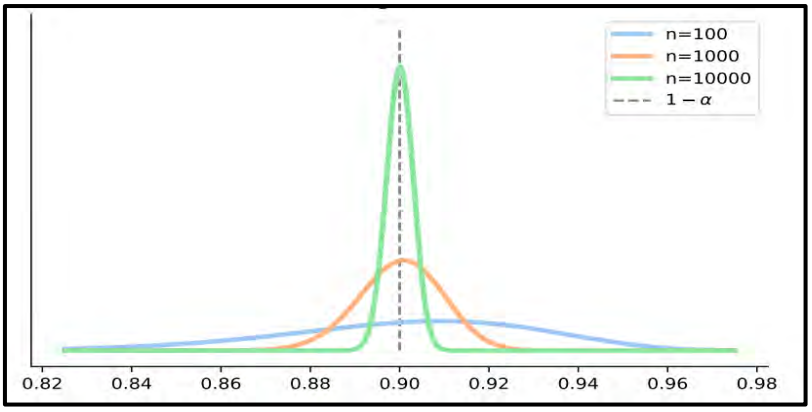


Figura 28: Distribución de la cobertura según el tamaño de los datos de calibración
Fuente: (Angelopoulos y Bates, 2022:15)

No obstante, en la práctica solo se cuenta con $n + n_{val}$ puntos en total para evaluar el algoritmo conformal, por ende, no se puede extraer nuevos datos para cada una de las rondas R. Por lo tanto, se calculará los valores de cobertura dividiendo aleatoriamente las $n + n_{val}$ observaciones, R veces en datos de calibración y validación.

CAPÍTULO 2. CONTENIDO DE LA INVESTIGACIÓN

2.1. Casos de estudio

En este apartado, se procederá a explicar la aplicación del análisis de sentimiento en la red social Twitter para la predicción del valor de las criptomonedas, donde se va a dar una explicación breve de los problemas que presentaban, la secuencia que siguieron para construir y evaluar el modelo y un análisis de los resultados obtenidos.

2.1.1. Predicción del precio del Bitcoin mediante análisis de sentimiento en Twitter

El estudio de investigación realizado por O. Sattarov, H. S. Jeon, R. Oh y D. Lee (2020), publicado por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) consiste en el estudio de la variación del precio del Bitcoin del 12 de marzo al 12 de mayo del 2018 (Ver Figura 29). La razón principal por la cual se escogió este período de estudio es debido a que durante este período de tiempo el precio del Bitcoin ha aumentado y disminuido lo cual permite evaluar la efectividad del modelo propuesto. Para poder obtener la curva de precios se utilizó el módulo Quandl de Python con el cual se extrajo el precio del Bitcoin de 4 plataformas de intercambio: Coinbase, Bitstamp, itBit y Kraken. Algunos de estos presentaban valores perdidos. Por dicho motivo, para obtener una curva suavizada se sumó el precio de las 4 plataformas y se dividió entre 4.

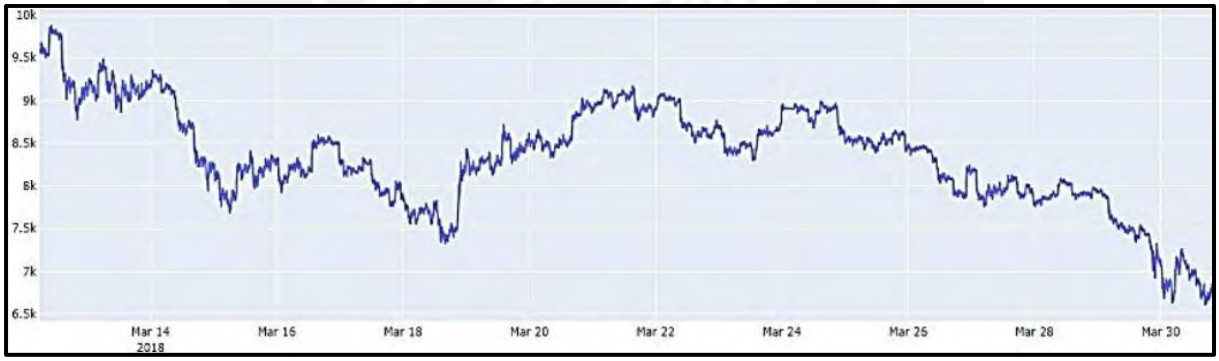


Figura 29: Variación del precio del Bitcoin del 12 de marzo al 12 de mayo del 2018
Fuente: (O. Sattarov, H. S. Jeon, R. Oh y D. Lee 2020:3)

Además de ello, mediante el uso del APIs y un poco de *web scraping* se extrajo 92550 tweets durante el mismo intervalo de tiempo de estudio, los cuales servirán para alimentar el modelo. Los tweets están conformados por una combinación de expresiones, emoticones, símbolos, URLs y menciones de usuarios. La data(tweets) extraída de Twitter suele contener mucho ruido es por ello que es necesario realizar un procesamiento de datos para poder tener una data estructurada. Por dicho motivo, se realizó un preprocesamiento de la data con el fin de estandarizar la información y reducir su escala. Para llevar a cabo dicho preprocesamiento se seguirán los siguientes pasos:

- Convertir el tweet a minúsculas.

- Remplazar dos o más puntos por un espacio en blanco.
- Eliminar los espacios y las comillas simples o dobles de los extremos de los tweets.
- Remplazar dos o más espacios en blanco con solo uno.
- Remover los lugares mencionados por los usuarios en los tweets.
- Convertir los hashtags en palabras.
- Clasificar los emoticones y eliminar los retweets.

Tabla 6 Ejemplo de la aplicación de las técnicas de preprocesamiento

	Técnica de preprocesamiento	Resultado
0	Tweet original	RT@bitcoin https://twitter.com/FT/status/10226050 Bitcoin ETF rejected but buuuuuy!!! Ask yourself why you aren't buying lol, tomorrow it will reach 8000 #BUY #NOW #BITCOIN
1	Remover "RT"	@bitcoin https://twitter.com/FT/status/10226050 Bitcoin ETF rejected but buuuuuy!!! Ask yourself why you aren't buying lol, tomorrow it will reach 8000 #BUY #NOW #BITCOIN
2	Eliminar URLs, espacios en blanco excesivos y menciones	Bitcoin ETF rejected but buuuuuy!!! Ask yourself why you aren't buying lol, tomorrow it will reach 8000 #BUY #NOW #BITCOIN
3	Reducir caracteres repetitivos >3 a 3	Bitcoin ETF rejected but buuuy!!! Ask yourself why you aren't buying lol, tomorrow it will reach 8000 #BUY #NOW #BITCOIN
4	Convertir a minúsculas	bitcoin etf rejected but buuuy!!! ask yourself why you aren't buying lol, tomorrow it will reach 8000 #buy #now #bitcoin
5	Remover hashtags	bitcoin etf rejected but buuuy!!! ask yourself why you aren't buying lol, tomorrow it will reach 8000 buy now

Fuente: (O. Sattarov, H. S. Jeon, R. Oh y D. Lee 2020:2)

Luego de haber realizado el preprocesamiento de datos con el fin de homogenizar la data. Se procede a aplicar el análisis de sentimiento para ello se utilizó el algoritmo VADER, el cual se ajusta de mejor manera al sentimiento de las redes sociales. Para poder evaluar el modelo propuesto se utilizó el algoritmo de Random Forest. Los autores escogieron este algoritmo debido a que, en comparación con otros, el algoritmo de Random Forest es bastante eficaz trabajando con datos de entrada que no están relacionados. Los datos de entrada que se utilizarán para alimentar el modelo de Random Forest son el score del análisis de sentimiento y el precio histórico del Bitcoin. Se experimentó con 10 árboles utilizando características de presencia y frecuencia, obteniendo que los rasgos de presencia se comportan mejor que los de frecuencia, aunque esta mejora no fue significativa. De ello, el modelo arrojo los siguientes resultados (Ver Figura 30 y Tabla 7).

Tabla 7 Estadísticas de los errores de predicción

Definición	Valor
Número de tweets	92550
Cantidad de valores predichos	80491
Valor máximo de error (%)	43.83
Valor mínimo de error (%)	21.84
Valor promedio de error (%)	37.52

Fuente: (O. Sattarov, H. S. Jeon, R. Oh y D. Lee 2020:3)

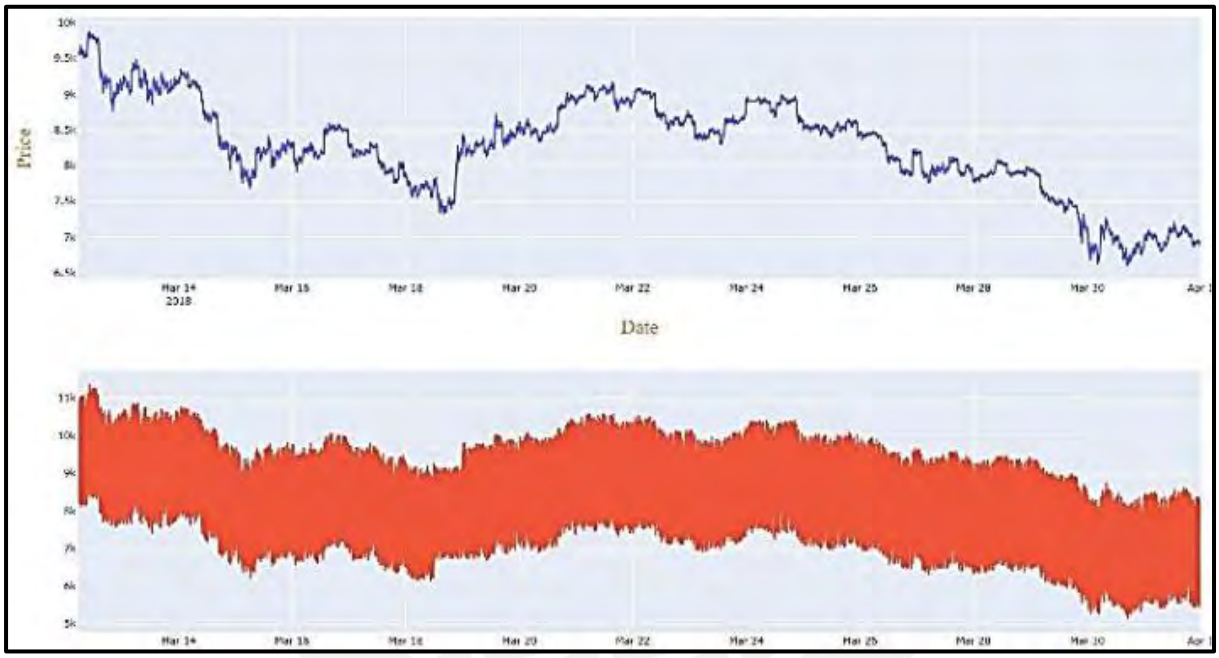


Figura 30: Precio real del Bitcoin(primero) vs Predicción del modelo (segundo)
Fuente: (O. Sattarov, H. S. Jeon, R. Oh y D. Lee 2020:3)

La Tabla 7 muestra que durante el proceso de predicción se perdieron aproximadamente más de 10 000 valores lo cual afecta el rendimiento del modelo.

Los autores concluyen que existe una fuerte correlación entre los cambios de tendencia del precio del Bitcoin y los sentimientos de Twitter. Asimismo, mencionan que desarrollar un léxico especial de sentimientos de Bitcoin mejoraría la relación entre el análisis de sentimientos y el precio de Bitcoin considerando características como los hashtags, emoticones y menciones de usuarios.

2.1.2. Predicción de la rentabilidad de altcoins a través de las redes sociales

La investigación realizada por Steinert, L. y Herff, C. (2018), expone como la actividad y el sentimiento de Twitter pueden ser utilizados para predecir los rendimientos de algunas altcoins ya que a diferencia de Google Trends, Twitter no solo proporciona información sobre la actividad del usuario sino también sobre su estado de ánimo asociado. Para llevar a cabo este estudio los autores recolectaron

los precios de 181 altcoins durante un periodo de 71 días y la actividad en la red social referida a estas, utilizando Twitter Search API. Se realizó la captura de información en dos períodos, por ende, los conjuntos de datos obtenidos no son continuos. El primer período comienza el 21 de marzo y termina el 5 de mayo del 2017. En la Figura 31, se observa el precio y la actividad en la red social, la cual esta representada por el número de tweets de la criptomoneda PinkCoin. En primera instancia, se puede sospechar que existe cierta relación. Por su parte, el segundo período cubre 26 días, empieza el 9 de mayo y finaliza el 4 de junio del 2017. En resumen, el conjunto de datos incluye el precio de las 181 altcoins y un total de 426,520 tweets referidos a estas.

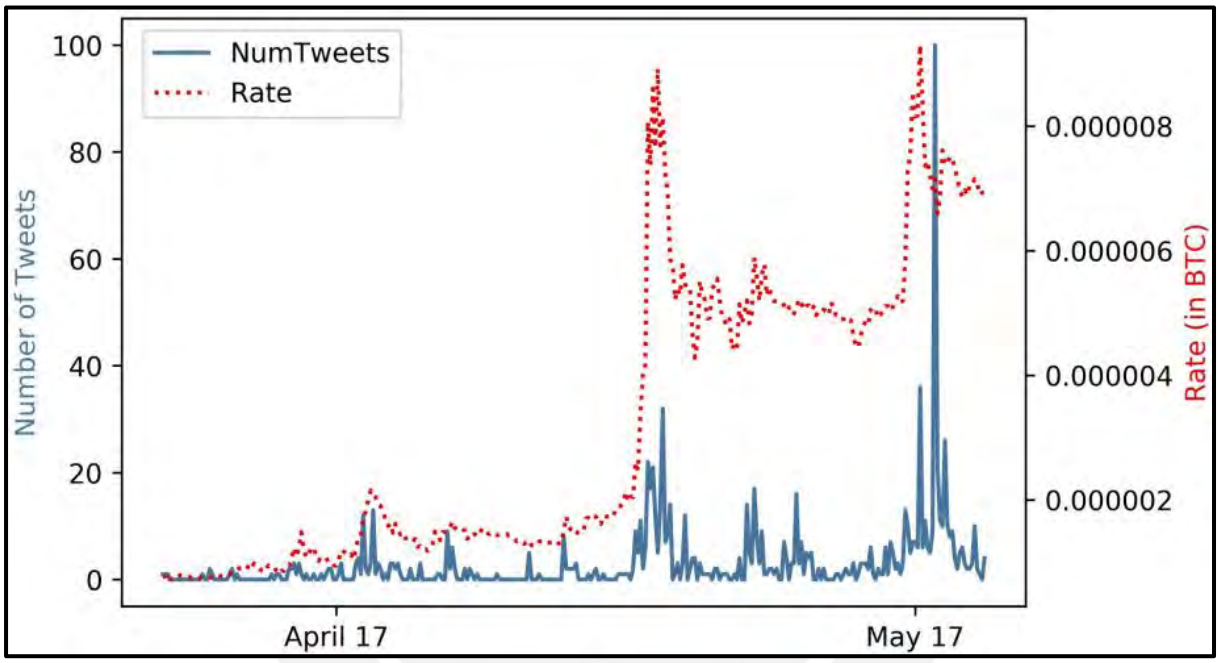


Figura 31: Número de tweets y precio del Pinkcoin durante un período de 45 días
Fuente: (Steinert, L. y Herff, C. 2018:2)

Posterior a ello, se utilizó el algoritmo de VADER para obtener el sentimiento de los tweets extraídos. Cuando se utiliza el algoritmo de VADER para el análisis de sentimientos se calculan cuatros puntuaciones diferentes: positiva, neutral, negativa y compuesta. Las tres primeras son porciones de texto que corresponden con su respectivo grupo mientras que la última es la suma de estas porciones o segmentos de texto cuya suma total debe dar 1. Para demostrar cómo funciona el modelo VADER se utilizarán dos tweets como ejemplo (Ver Figura 32 y 33).

Figura 32: DigiByte tweet
Fuente: (Steinert, L. y Herff, C. 2018:5)

Figura 33: Bitcoin tweet
Fuente: (Steinert, L. y Herff, C. 2018:5)

En la Figura 32, se observa que el tweet relacionado con la altcoin DigiByte contiene expresiones positivas como "WOW" y emoticones de felicidad. En consecuencia, el tweet posee una puntuación positiva alta y una puntuación compuesta muy positiva.

En la Figura 33, sucede todo lo contrario pues se puede visualizar sentimientos negativos acerca del Bitcoin como "*What the hell are you doing???*". Esto conlleva a que el tweet obtenga una puntuación negativa alta y una puntuación compuesta muy negativa. Esto puede ser representado de mejor manera en la Figura 34.

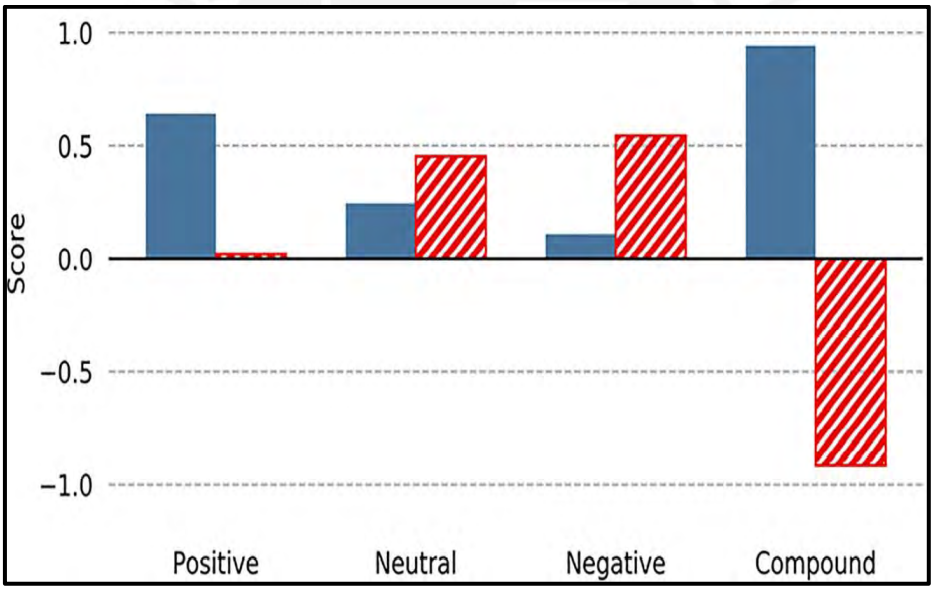


Figura 34: Puntuaciones de VADER para DGB (azul) y BTC (rojo)
Fuente: (Steinert, L. y Herff, C. 2018:5)

Tabla 8 Estadísticas generales del conjunto de datos de entrenamiento

Variable	Media	Desv. Estándar	Mínimo	Máximo
Número de tweets	3.635	12.243	0	100
Positivos	0.032	0.068	0	0.700
Neutrales	0.354	0.443	0	1.000
Negativos	0.008	0.032	0	0.727
Compuestos	0.058	0.161	-0.922	0.956
Volumen	637.150	3736.074	0	97574.006

Precio	0.006	0.057	0	0.912

Fuente: (Steinert, L. y Herff, C. 2018:6)

Predicción

Para crear modelos de predicción, los autores aplicaron análisis de regresión lineal por mínimos cuadrados ordinarios al conjunto de entrenamiento. Las variables de entrada que serán utilizadas para el modelo son el número de tweets y el puntaje obtenido mediante el algoritmo de VADER. Esto resultó en un total de 333 muestras y 5 atributos. De acuerdo con Steinert, L. y Herff, se emplearán los retornos en lugar del precio de las altcoins como variable dependiente o de respuesta ya que de esta manera se reduce el riesgo de obtener datos que presentan autocorrelación.

$$return_t = price_t - price_0$$

No obstante, los autores solo analizaron las altcoins que presentaban un mayor grado de correlación entre las variables de entrada (números de tweets y sentimiento) y la variable de salida (retorno). Para ello, se utilizó el coeficiente de determinación (R2), el cual indica que porcentaje de la variación del retorno esta explicado por los números de tweets y sentimiento.

Resultados

Los resultados obtenidos se basan en el conjunto de datos de entrenamiento de 5 altcoins (Ver Figura 35), estas fueron escogidas con base en su actividad en la red social Twitter, volumen de comercio y precio. Solo se analizaron 5 altcoins pues los rendimientos de las demás altcoins no respondían a los cambios en la cantidad de tweets en la red social Twitter y del sentimiento encontrado en estos. No obstante, al aplicar estos modelos al conjunto de datos de prueba se obtuvo coeficientes de determinación menores a los hallados con el conjunto de datos de entrenamiento (Ver Figura 36), se halló significancia estadística para 16 de las 40 predicciones.

Se concluye del análisis que existe conexión entre la actividad en la red social Twitter y el sentimiento encontrado en esta con el retorno de las altcoins analizadas. Asimismo, los autores recomiendan considerar los seguidores que poseen los usuarios como variable de entrada ya que ello indica el número de personas que pueden ser influenciadas por los tweets. El considerar esta variable de entrada en el modelo y emplear redes neuronales mejorará la precisión de la predicción.

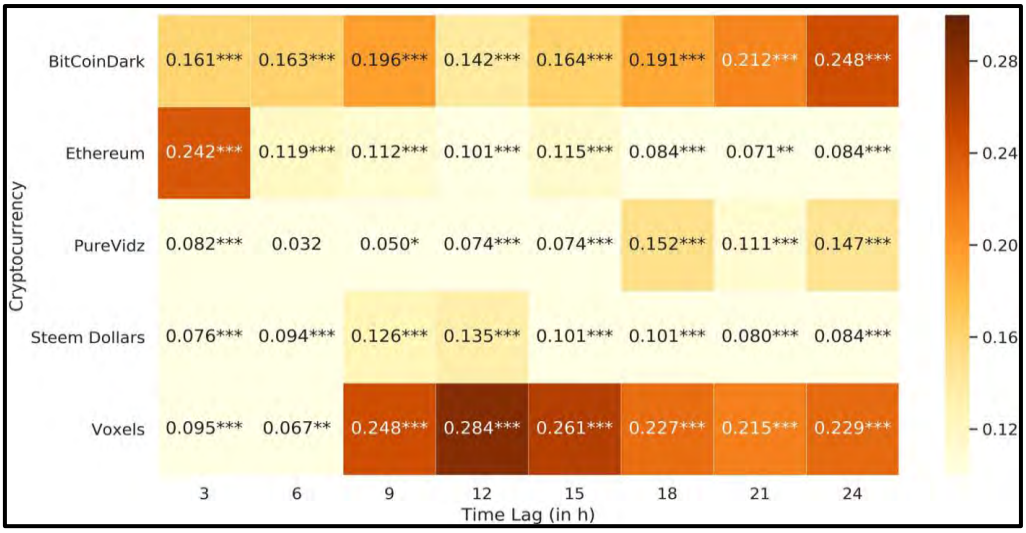


Figura 35: Coeficiente de determinación de los modelos de regresión lineal (training set)
Fuente: (Steinert, L. y Herff, C. 2018:7)

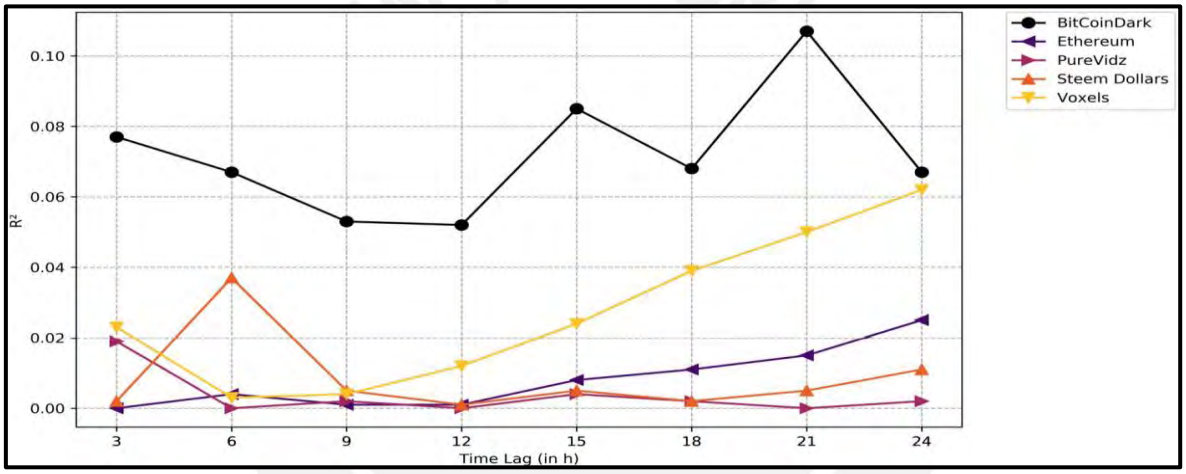


Figura 36: Coeficiente de determinación de los modelos de regresión lineal (test set)
Fuente: (Steinert, L. y Herff, C. 2018:8)

2.1.3. Pronóstico de los movimientos del BTC utilizando análisis de sentimiento de Twitter

El estudio de investigación realizado por Ibrahim, A. (2021), publicado por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) desarrolla un modelo de predicción de ensamble compuesto (CEPM). El marco de trabajo del CEPM consta de 5 etapas:

1. **Preprocesamiento de texto**

   En esta etapa, el autor utilizo el algoritmo de Porter ya que usualmente proporciona mejores resultados pues posee una menor tasa de error de *stemming* respecto a otros modelos. Posterior a ello, procede a realizar un *word stop removal* que consiste en eliminar conectores como "*and*", "*are*" y "*because*" contenidos en los tweets. Se realiza este paso ya que, si bien estos conectadores no contribuyen de manera significativa la precisión del modelo, realizando la eliminación de estos conectores se reduce la cantidad de información alojada en el conjunto de datos lo cual acelera el tiempo de respuesta en el proceso de minado de datos.

2. Puntuación del sentimiento

En esta etapa, el autor emplea el algoritmo VADER para asignar a cada tweet una puntuación de sentimiento basado en el grado de positividad, neutralidad o negatividad de las palabras contenidas en el tweet. La puntuación final del sentimiento toma en cuenta el número de seguidores en Twitter, retweets y me gusta asociados a cada tweet.

3. Clasificaciones XGBoost individuales

En esta etapa, se realizaron varias variantes de los clasificadores XGBoost.

4. Agregación de conjuntos compuestos

El modelado de conjuntos está diseñado para maximizar el desempeño del modelo. En este estudio, se empleó un método de validación cruzada de 10 veces. El conjunto de datos fue dividido en 10. De estos, 9 de ellos fueron utilizados como un conjunto de datos de entrenamiento y el restante como conjunto de datos de prueba.

5. Validación del modelo

De acuerdo con Ibrahim, A. (2021) la matriz de confusión es la medida más utilizada para determinar la calidad de los métodos usados en la predicción del valor de las criptomonedas. Los indicadores más populares para evaluar los resultados obtenidos de la matriz confusión son: *accuracy, precision, recall* y *F-score*.

Accuracy: Se calcula determinando el porcentaje de observaciones que fueron etiquetadas correctamente mientras más cercano este a 100% mejora será el desempeño del modelo al momento de predecir. No obstante, no es el indicador más confiable ya que proporciona resultados engañosos si es que las clases no están correctamente balanceadas.

Precision: Mide el ratio de entradas positivas correctas. Si el ratio es alto indica una capacidad de predicción robusta por parte del modelo.

Recall: Mide el ratio de predicciones positivas correctas sobre el total de predicciones positivas que el modelo ha realizado.

F-score (F1): Se calcula mediante el promedio ponderado de *Precisión y Recall*. Este indicador resulta útil cuándo se evalúa casos con distribuciones de clases desiguales.

En la Tabla 9, se observan los resultados obtenidos del modelo CEPM respecto al resto de modelos.

Tabla 9 Resultados de Indicadores Precision, Recall, F-score y Accuracy

Modelo	Precision	Recall	F-score	Accuracy
Regresión Logística	0.6743	0.4532	0.54207141	0.61
SVM	0.64843	0.5543	0.59768152	0.65
Naive Bayes	0.665732	0.65421	0.65992071	0.66
XGBoost	0.78953	0.809532	0.7994059	0.72
CEPM	0.8926	0.883474	0.88801355	0.88

Fuente: (Ibrahim, A. 2021:4)

Tabla 10 Porcentaje de mejora en los indicadores mediante el uso del CEPM

Precision	Recall	F-score	Accuracy
21%	16%	18%	22%

Fuente: (Ibrahim, A. 2021:4)

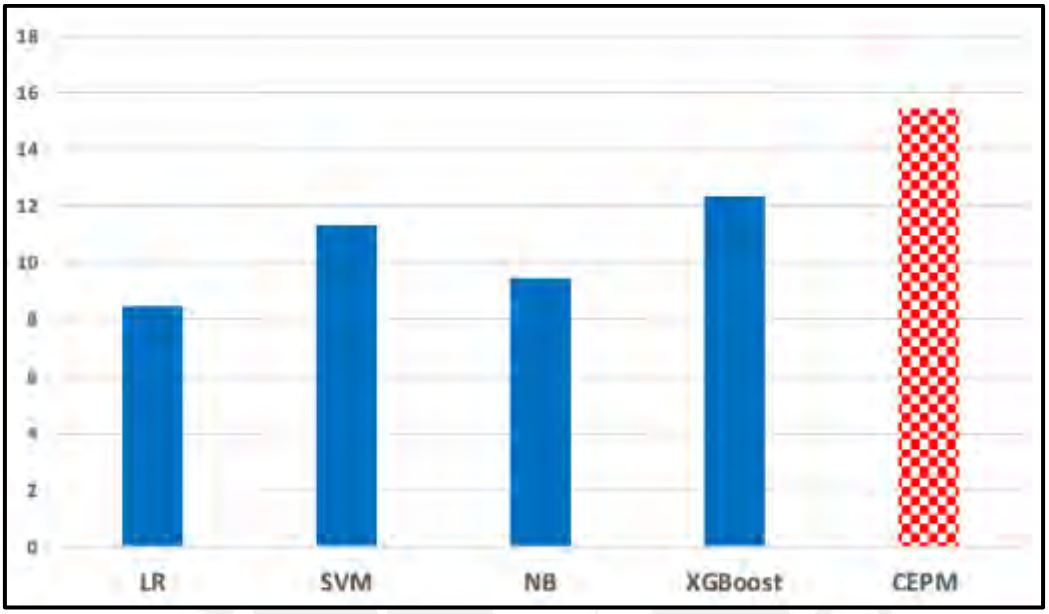


Figura 37: Tiempo de ejecución de los modelos en segundos
Fuente: (Ibrahim, A. 2021:4)

Con base en los resultados obtenidos, se concluye que el CEPM presenta un mayor desempeño respecto a los otros modelos utilizando el conjunto de datos de Twitter obtenido durante la era del COVID-19. Sin embargo, el modelo propuesto (CEPM) también puede servir para predecir el mercado del BTC incluso después de la pandemia del COVID-19.

2.1.4. El poder predictivo del sentimiento de Twitter para predecir el precio de las criptomonedas

El estudio de investigación realizado por Kraaijeveld, O., & De Smedt, J. (2020), publicado por la Revista de Mercados Financiero Internacionales, Instituciones y Dinero se enfoca en la predicción del precio de retorno de las 9 criptomonedas con la mayor capitalización del mercado en mayo de 2018. El estudio fue dividido en 5 etapas (Ver Figura 38):

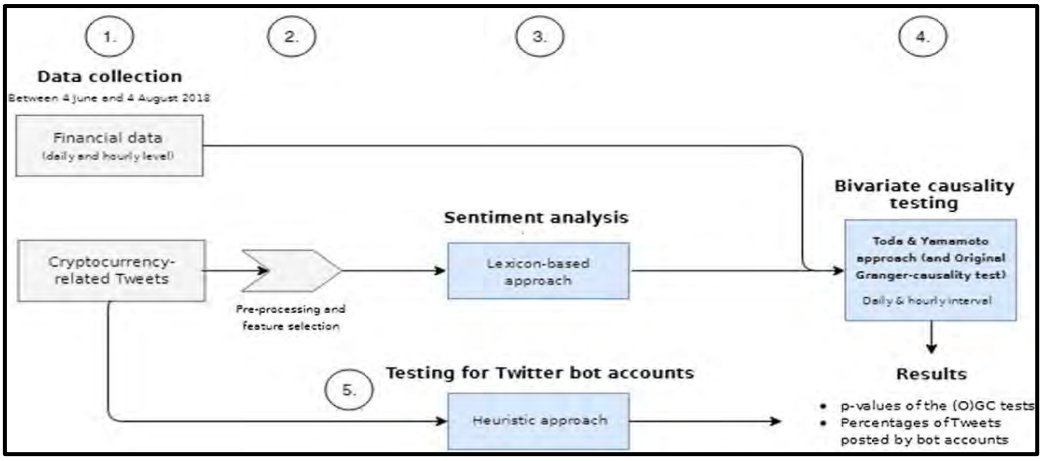


Figura 38: Flujo de las fases de la metodología aplicada
Fuente: (Kraaijeveld, O., & De Smedt, J. 2020:7)

1. Recolección de datos

El conjunto de datos se dividió en dos secciones. La primera sección se enfoca en los tweets extraídos de Twitter y la segunda sección en la información financiera extraída de CoinMarketCap. El período en el cual se recolecto la información de Twitter y CoinMarketCap data del 4 de junio al 4 de agosto de 2018. Se obtuvo un total de 24,035,075 tweets de los 9 *datasets*, es decir, un conjunto de datos por criptomoneda. La distribución de los tweets por criptomoneda se puede observar en la Tabla 11. Por otro lado, de CoinMarketCap se extrajo las siguientes variables financieras: precio del token/USD, precio del token/BTC, volumen diario de operaciones, capitalización de mercado en dólares americanos y el suministro.

Tabla 11 Distribución de tweets por criptomoneda

Criptomoneda	Cantidad de Tweets recolectados
BTC	9,768,425
ETH	6,286,602
XRP	1,635,570
BCH	816,634
EOS	619,889
LTC	1,212,446
ADA	489,321
XLM	1,310,418
TRX	1,895,760
Total	**24,035,075**

Fuente: (Kraaijeveld, O., & De Smedt, J. 2020:8)

2. Preprocesamiento de datos y selección de features

Los datos de Twitter son conocidos por ser una base de datos no relacional, además de presentar un alto nivel de ruido. Por tal motivo, es requisito indispensable realizar una limpieza de datos con el fin de disminuir el ruido presente en ellos y poder realizar el análisis de sentimiento. Para ello se aplicaron las siguientes técnicas de preprocesamiento:

- Remover RT si se encuentra presente en los tweets
- Remover los URLs, espacios en blanco y menciones presente en los tweets
- Eliminar el prefijo del hashtag de la palabra si este se encuentra en el diccionario de inglés NLTK Reuters. Si la palabra no se encuentra en el diccionario se elimina todo el hashtag.
- Las contracciones de las palabras se expanden (Por ejemplo: we're a we are)
- Remover los dos símbolos de dólar y las palabras que contienen caracteres numéricos
- Transformar las palabras contenidas en el tweet a minúsculas
- Remover *stop words*
- Tweets que contengan letras que se repiten más de 3 veces se reduce a 3.
- Remover los signos de puntuación

Por último, con el fin de disminuir las diferencias horarias de los tweets extraídos de Twitter y las variables financiera de CoinMarketCap se utilizará marcas de tiempo de UTC-1. Un claro ejemplo de las técnicas previamente descritas se observa en la Tabla 12. Por su parte, en la Tabla 13 se visualiza un resumen estadístico de los tweets posterior al preprocesamiento.

3. Métodos

Comprobar la presencia de bots en Twitter

Si bien parte del estudio consiste en validar la existencia de bots en Twitter, debido al alcance y limitaciones del estudio no se analizará el efecto de estos en el sentimiento de Twitter o el precio de las criptomonedas. Para probar la existencia de tweets extraídos de Twitter asociados a cuentas bots se aplicaron seis heurísticas. No obstante, para garantizar la identificación de los bots se considera que un tweet es publicado por un bot de criptomonedas si cumple 2 o más de los siguientes criterios:

1. El tweet contiene las palabras "*give away*" o "*giving away*"
2. El tweet contiene las palabras "*register*", "*pump*" o "*join*".
3. El tweet contiene más de 14 hashtags
4. El tweet contiene más de 14 simbolos
5. La fuente de la plataforma contiene la palabra "*bot*"
6. El usuario sigue a menos de 1000 cuentas y el ratio entre el número de cuentas seguidas y las cuentas que siguen a ese usuario es superior a diez

Análisis de sentimiento

Se obtiene la puntuación de la polaridad del sentimiento de los tweets mediante el uso del algoritmo VADER. Este es complementado agregando los *tokens* contenidos en el 2016 Loughran & McDonald *financial corpus* y un léxico compilado de criptomonedas con 63 palabras y abreviaturas con el fin de aumentar los *tokens* que no están presentes en el léxico de VADER. El algoritmo de VADER calcula una puntuación compuesta ponderada normalizada cuyo rango de valores va desde -1 hasta +1. Esta puntuación indica si un tweet es positivo (>=0.5), neutral (>-0.5 y <0.5) o negativo (<=-0.5). Las puntuaciones obtenidas son convertidas a una serie de tiempo y agregadas a intervalo de tiempo de diarios o de horas lo cual se logra tomando la puntuación promedio por intervalo de tiempo deseado.

Test de Wiener-Granger

Se utiliza el *augmented Todo & Yamamoto (T&Y) Granger-causality test* debido a que según los estudios de Engle y Granger (1987) cuando las series de tiempo no son estacionarias y cointegradas, la prueba original de Wiener-Granger no es válida. Asimismo, la prueba de Dickey-Fuller aumentada es empleada con el fin de probar la estacionalidad y determinar el orden máximo de integración (Dmax). Finalmente, se utiliza la prueba de LM de Breusch-Godfrey para evaluar la autocorrelación de los residuos de los modelos autorregresivos vectoriales (VAR). El procedimiento de T&Y se basa en los siguientes modelos:

$$x_t = \mu + \sum_{t=1}^{l'+D_{max}} \alpha_t Y_{t-1} + \sum_{t=1}^{l'+D_{max}} \beta_t X_{t-1} + \mu_{1t}$$

$$Y_t = \mu + \sum_{t=1}^{l'+D_{max}} \gamma_t X_{t-1} + \sum_{t=1}^{l'+D_{max}} \delta_t Y_{t-1} + \mu_{2t}$$

, donde l′ representa *lags orders* y μ representa *error terms*. Este enfoque se asegura de que no exista autocorrelación entre los *errors terms* no estén al aumentar el número de *lags* hasta llegar al número seleccionado de *lags*.

Indicadores y Variables

Los autores utilizan como variables predictoras a *daily and hourly sentiment* en Twitter ST, *bullishness* (B) y *message volumen* Vmes con el fin de validar si influyen de manera significativa, es decir presentan un poder predictivo para cada una de las variables de respuesta o dependientes *price returns* Pr y *daily trading volumen* Vd/trad. Se utiliza el *return price* en lugar del precio de la criptodivisa con el fin de evitar relaciones espurias. La variable *bullishness* se calcula de la siguiente manera, donde c denota la criptomoneda dado que en el caso de estudio se están analizando 9 criptomonedas, número de tweets con recomendación de comprar (Mbuy) y con recomendación de

vender (Msell) en el intervalo de tiempo t. La cantidad total de tweets está representada por M. Los indicadores y variables presentadas se definen de la siguiente forma:

$$B_t^c = \ln\left(1 + \frac{M_c^{buy}}{M_c^{sell}}\right)$$

$$V_{mes}^c = \ln\left(1 + \sum_t^c M\right)$$

$$V_{trad}^c = \ln(1 + V_{trad}^c)$$

$$P_{RC}^t = \ln\left(1 + \frac{P_t^c}{P_{t-1}^c}\right)$$

Conclusión

Se concluye con base en los resultados obtenidos que para el intervalo de tiempo diario, el sentimiento de Twitter puede ser utilizado para predecir el *price return* de Bitcoin, Bitcoin Cash y Litecoin. En el caso de las criptomonedas tales como EOS y TRON, el *bullishness ratio* puede ser utilizado para predecir el *price return*. Por su parte, la variable predictora *message volume* es un predictor del *price return* de Litecoin y XRP. Asimismo, se logro identificar que las predictores más fuertes a nivel diario son el sentimiento de Twitter y *message volume*. Respecto al intervalo de tiempo por hora, no se observó ningún poder de predicción en ninguna de las variables de estudio.

Por otra parte, se encontró que aproximadamente el 1-14% de los tweets obtenidos correspondían a bots, identificar el porcentaje de tweets asociados a bots es importante ya que estos pueden ser utilizados para difundir información falsa con el fin de dirigir el sentimiento de los inversores lo cual afectaría los resultados de este estudio. Finalmente, Los autores recomiendan ampliar el periodo de observación, experimentar con varios niveles de granularidad e investigar los efectos de la influencia social del usuario así como utilizar un modelo de aprendizaje de máquina supervisado o *hybrid approach*. Si bien este trabajo no estudia los efectos de los tweets que corresponden a bots en los precios o volúmenes de comercio de las criptomonedas se podría investigar como identificar estos bots con mayor precisión y el efecto que generan en el sentimiento de Twitter o el precio de las criptomonedas.

Tabla 12 Ejemplo de aplicación de las técnicas de preprocesamiento

	Técnica	Resultado
0	Tweet original	RT @bitcoin https://twitter.com/FT/status/1022605086172872704 Bitcoin ETF rejected but buuuuuy!!! Ask yourself why you aren't buying lol, tomorrow it will reach 8000 #BUY #NOW #BITCOIN #BTC $BTC $ETH
1	Remover "RT"	@bitcoin https://twitter.com/FT/status/1022605086172872704 Bitcoin ETF rejected but buuuuuy!!! Ask yourself why you aren't buying lol, tomorrow it will reach 8000 #BUY #NOW #BITCOIN #BTC $BTC $ETH
2	Remover los URLs, espacios en blanco y menciones	Bitcoin ETF rejected but buuuuuy!!! Ask yourself why you aren't buying lol, tomorrow it will reach 8000 #BUY #NOW #BITCOIN #BTC $BTC $ETH
3	Reducir palabras que se repitan >3 a 3	Bitcoin ETF rejected but buuuy!!! Ask yourself why you aren't buying lol, tomorrow it will reach 8000 #BUY #NOW #BITCOIN #BTC $BTC $ETH
4	Transformar a minúsculas	bitcoin etf rejected but buuuy!!! ask yourself why you aren't buying lol, tomorrow it will reach 8000 #buy #now #bitcoin #btc $btc $eth
5	Eliminar el tweet si el número de palabras es <4	bitcoin etf rejected but buuuy!!! ask yourself why you aren't buying lol, tomorrow it will reach 8000 #buy #now #bitcoin #btc $btc $eth
6	Remover hashtags si no se encuentran en el diccionario Reuters	bitcoin etf rejected but buuuy!!! ask yourself why you aren't buying lol, tomorrow it will reach 8000 buy now $btc $eth
7	Expandir contracciones	bitcoin etf rejected but buuuy!!! ask yourself why you are not buying lol, tomorrow it will reach 8000 buy now $btc $eth
8	Transformar jergas o acrónimos	bitcoin etf rejected but buuuy!!! ask yourself why you are not buying laughing out loud, tomorrow it will reach 8000 buy now $btc $eth
9	Eliminar símbolos de dólar	bitcoin etf rejected but buuuy!!! ask yourself why you are not buying laughing out loud, tomorrow it will reach 8000 buy now
10	Eliminar palabras que contengan caracteres numéricos	bitcoin etf rejected but buuuy!!! ask yourself why you are not buying laughing out loud, tomorrow it will reach buy now
11	Aplicar lematización WordNet	bitcoin etf rejected but buuuy!!! ask yourself why you are not buying laughing out loud, tomorrow it will reach buy now
12	Eliminar *stop words*	bitcoin etf rejected but buuuy ask not buying laughing out loud tomorrow reach buy

Fuente: (Kraaijeveld, O., & De Smedt, J. 2020:9)

Tabla 13 Resumen estadístico de los tweets posterior al preprocesamiento

Criptomoneda	Cantidad total de tweets	Cantidad de usuarios únicos	Cantidad de tweets únicos	Volumen promedio diario	Desv. Estánd. del volumen diario	Polaridad en promedio
Bitcoin (BTC)	9,568,223	978,066	3,332,389	156,856.1	18,166.7	0.315
Ether (ETH)	6,129,414	707,180	1,550,239	100,482.2	12,778.0	0.481
XRP (XRP)	1,534,870	300,320	622,703	23,613.4	8,023.5	0.2882
Bitcoin Cash (BCH)	733,504	123,818	366,982	11,461.0	3,260.7	0.23
EOS (EOS)	516,431	99,226	189,517	7,945.1	2,726.9	0.35
Litecoin (LTC)	1,128,391	197,770	442,052	17,631.1	6,169.3	0.328
Cardano (ADA)	418,380	77,290	210,839	6,436.6	2,094.1	0.297
Stellar (XLM)	1,082,282	428,779	368,036	16,153.5	9,835.5	0.314
TRON (TX)	1,800,544	546,635	309,023	28,133.5	23,437.1	0.367
Total	22,912,039					0.33

Fuente: (Kraaijeveld, O., & De Smedt, J. 2020:9)

2.1.5. Predicción del precio de criptomonedas a través de series de tiempo y sentimiento social.

La investigación realizada por Pang, Y., Sundararaj, G., & Ren, J. (2019) tenía como fin cerrar las brechas existentes entre el análisis y aplicación de los datos se sentimiento u opinión en línea a las estrategias de trading para ello diseñaron un algoritmo que permite evaluar estrategias de trading considerando dicho parámetro.

Datos de sentimiento social

Los autores resaltan el rol que cumplen las plataformas de redes sociales como nuevo medio de comunicación, interacción y compromiso. Asimismo, señalan que las emociones humanas pueden ser categorizadas en alegría, amor, odio, miedo, optimismo, miedo, duda, etc. Los datos de sentimiento de Bitcoin fueron obtenidos en colaboración con Thomson Reuters, el cual utilizo dos fuentes de información redes sociales y medios de comunicación. Los autores enfatizan en los estilos de comunicación que se utilizan en ambos medios. Pues en las redes sociales se utiliza un lenguaje informal, con faltas ortográficas y suele denotar un nivel significativo de ironía o sarcasmo a diferencia del lenguaje utilizado en los medios de comunicación. Thomson Reuters posee un *pipeline* que les facilita la extracción, análisis de léxico, filtrado de correlación y formación del índice. El índice en mención se le conoce como TRMI (Thomson Reuters Marketpsych Index), el cual se forma a partir de redes sociales y medios de comunicación; y se alimenta a través de tres fuentes: redes sociales, medios de comunicación y una combinación de contenidos asociado a medios sociales y noticias.

Cada TRMI está compuesto de una combinación de varias variables. En primero lugar, se determinan los valores absolutos de todas las variables que contribuyen al TRMI durante las últimas 24

horas. Posteriormente, estos valores absolutos son sumados para todos los componentes, a esta suma se le conoce como *Buzz* y se formula de la siguiente manera:

$$\boldsymbol{Buzz(a)} = \sum_{c\in C(a),v\in V}\left|Var_{c,v}\right|,$$

donde *V* representa el conjunto de todas las variables subyacentes a cualquier TRMI de la clase de activos, *a* denota un activo.

El TRMI de un activo puede ser calculado de la siguiente forma:

$$\boldsymbol{TRMI_t(a)} = \frac{\sum_{c\in C(A),v\in V(t)}\left(I(t,v)*PsychVar_v(C)\right)}{Buzz(Asset)},$$

se define la función *I (t, v)* para determinar si una Var es aditiva o sustractiva a una TRMI

$$\boldsymbol{I(t,v)} = \{+1, if\ addictive\ ;\ -1, if\ subtractive\}$$

En la Tabla 14, se observan algunos índices del TRMI.

Tabla 14 Índices de sentimientos del TRMI

Index	Description: *Score of references in news and social media to…*	Range
sentiment	overall positive references, net of negative references	-1 to 1
optimism	optimism, net of references to pessimism	-1 to 1
fear	fear and anxiety	0 to 1
joy	happiness and affection	0 to 1
trust	trustworthiness, net of references connoting corruption	-1 to 1
violence	violence and war	0 to 1
conflict	disagreement and swearing net of agreement and conciliation	-1 to 1

Fuente: (Pang, Y., Sundararaj, G., & Ren, J. 2019:37)

El rango de estudio del presente estudio va desde octubre de 2017 hasta marzo de 2018 con un intervalo de tiempo por hora. Los autores utilizan el precio del Bitcoin expresado en dólares, además de ello trabajan con variables como el precio de cierre, el precio de apertura, precio máximo y mínimo por cada hora. En la figura 39, se observa el comportamiento del precio del Bitcoin en el rango de estudio.

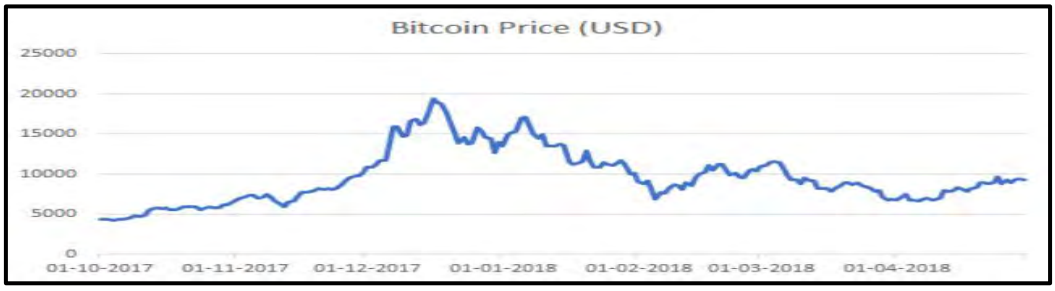


Figura 39: Valor del precio del Bitcoin durante 6 meses
Fuente: (Pang, Y., Sundararaj, G., & Ren, J. 2019:37)

En el EDA (Análisis exploratorio de datos), los autores tienen como objetivo principal hallar las correlaciones existentes entre el movimiento del precio del Bitcoin y los datos de las redes sociales. Para realizar ello se cuenta con un total de 43 índices de sentimiento del TRMI para realizar el estudio en mención. Del total de 43 índices de sentimiento, se utilizarán 4 índices de sentimiento a base de ejemplo para graficar el análisis llevado a cabo por estos.

En la Figura 40, se compara el precio del Bitcoin vs. *Joy*, *Joy* es definido como el estado de ánimo en el que se puede experimentar felicidad y afecto. El rango de valores del índice TRMI va desde 0 hasta 1. Del gráfico se pueden sacar dos conclusiones a priori, la correlación entre ambas variables de estudio es positiva, es decir, cuando el precio o valor del Bitcoin incrementa, el estado de *Joy* experimenta un ligero aumento. A su vez, se observa que la serie de tiempo no es estacionaria, por ende, no posee media ni desviación estándar constante a lo largo del tiempo.

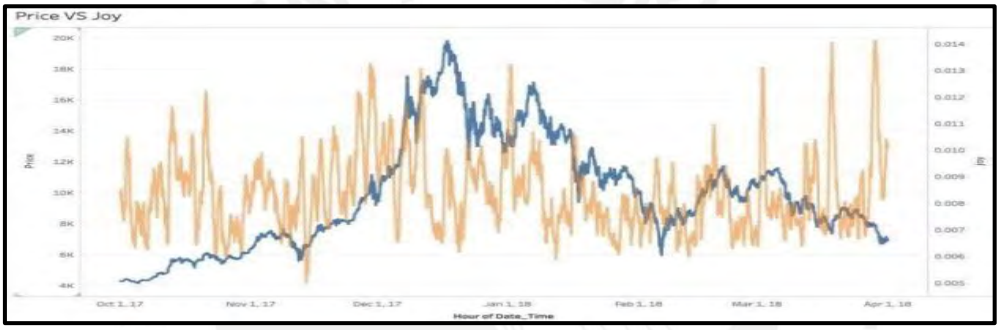


Figura 40: Price (línea azul) vs. Joy (línea anaranjada)
Fuente: (Pang, Y., Sundararaj, G., & Ren, J. 2019:38)

Por otro lado, se compara el precio del bitcoin con el índice de *optimism*, el cual representa la mentalidad positiva que tiene la gente respecto el mercado. En primera instancia, se observa (Ver Figura 41) una correlación negativa ya que un aumento exponencial del precio va seguido de una caída constante de optimismo pues sugiere a los inversores que se trata de especulación, esto puede ser observado a finales del 2017, sin embargo, se detecta otro comportamiento a finales del período de medición que indica que cuando los precios se han estabilizado, el sentimiento de optimismo aumenta. Los rangos asociados a este índice TRMI se encuentran en la escala de -1 a 1.

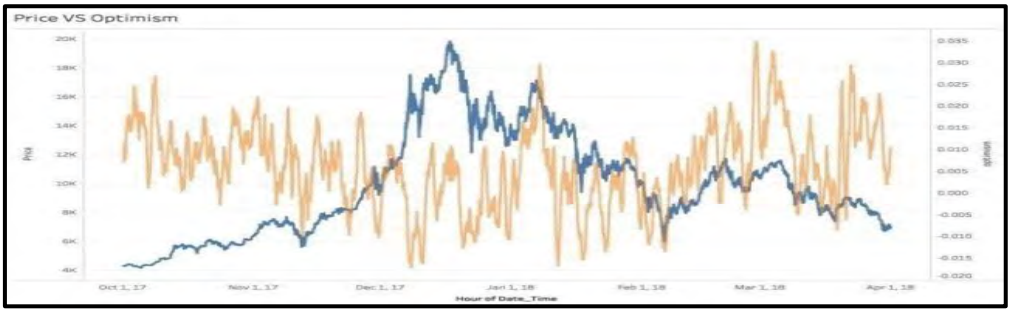


Figura 41: Price (línea azul) vs. Optimism (línea anaranjada)
Fuente: (Pang, Y., Sundararaj, G., & Ren, J. 2019:38)

Asimismo, se compara el índice de sentimiento *Gloom*, este es un indicador de depresión del mercado. Los valores asociados a este índice de TRMI se encuentran en la escala de 0 a 1. Existe un fuerte aumento del sentimiento de depresión cuando el precio del Bitcoin se encuentra en una caída constante como se observa en la Figura 42. Por último, se analiza el índice de sentimiento *Fear*, este sentimiento puede ser indicador de la respuesta reactiva de la gente sobre el mercado. El rango de valores asociado a este índice TRMI va desde 0 hasta 1. Esto se ejemplifica de la siguiente manera, cuando el precio del Bitcoin se encuentra en niveles muy bajos o se produce un descenso a dichos niveles (Ver Figura 43), existe un miedo exacerbado pues las personas temen perder su capital o inversión.

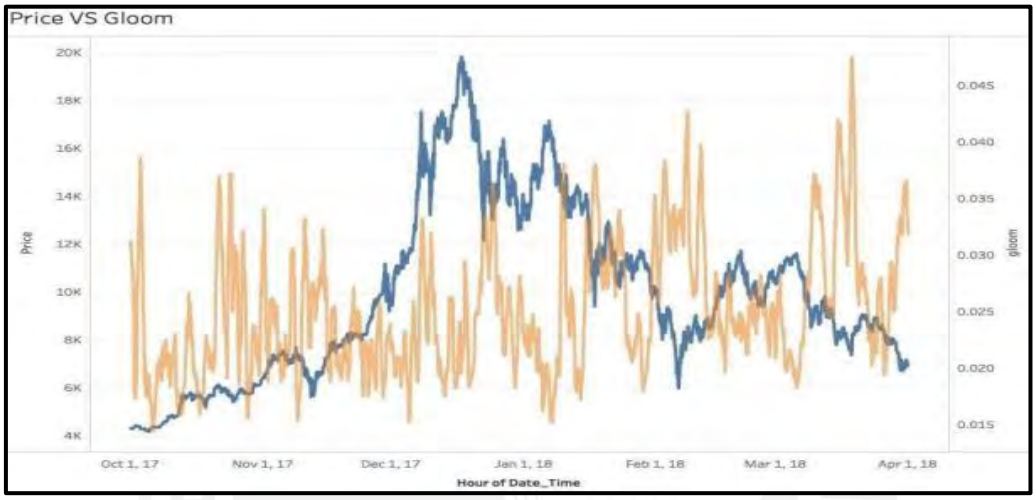


Figura 42: Price (línea azul) vs. Gloom (línea anaranjada)
Fuente: (Pang, Y., Sundararaj, G., & Ren, J. 2019:38)

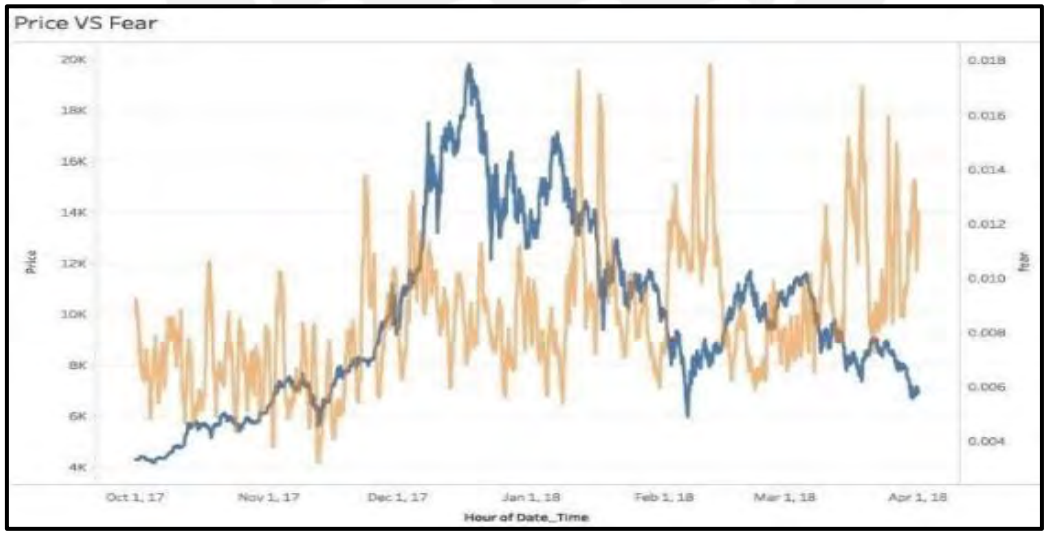


Figura 43: Price (línea azul) vs. Fear (línea anaranjada)
Fuente: (Pang, Y., Sundararaj, G., & Ren, J. 2019:39)

Procesamiento de datos y modelado

Los autores utilizaron el método *stepwise* para hallar las variables más significativas que contribuyen al precio del Bitcoin, el criterio utilizado fue el criterio de información Akaike. El método

de *stepwise* consiste en añadir y eliminar variables predictoras del modelo predictivo con el fin de encontrar el subconjunto de variables que minimicen el error de la predicción. Del total de 43 índices de sentimiento del TRMI se escogieron las 14 variables más significativas que el método StepAIC identificó.

Tabla 15 Selección de índices significativos del TRMI mediante stepwise

Variables	Tolerance	VIF
1 buzz	0.376	2.66
2 joy	0.652	1.53
3 conflict	0.389	2.57
4 fear	0.547	1.83
5 gloom	0.467	2.14
6 timeUrgency	0.506	1.98
7 uncertainty	0.361	2.77
8 emotionVsFact	0.609	1.64
9 marketRisk	0.337	2.97
10 adoption	0.387	2.58
11 anonymity	0.821	1.22
12 fOMO	0.680	1.47
13 fork	0.523	1.91
14 hodl	0.850	1.18

Fuente: (Pang, Y., Sundararaj, G., & Ren, J. 2019:39)

Los modelos estadísticos pueden ser utilizados para analizar el patrón de los datos ya sea la tendencia, estacionalidad y validar si el mismo patrón se repite. Los modelos de series de tiempo son un caso especial ya que el orden de los registros es importante. Estos casos especiales a menudo son resueltos empleando métodos de modelado de series temporales (ARIMA) y redes neuronales recurrentes (RNN).

Para aplicar modelos de ARIMA, la serie de tiempo debe ser estacionaria Si los gráficos de autocorrelación muestran que la serie no satisface dicha condición, entonces se la serie se puede volver estacionaria diferenciados los valores. Este proceso se repite sucesivamente hasta que la serie logre la estacionariedad. Los modelos de ARIMA poseen 3 parámetros: autoregresión, media móvil e integración. Estos parámetros se hallan a base de prueba y error con el fin de encontrar los valores más óptimos para el modelo. La forma de evaluar los modelos es encontrar la variable que posea alta significancia, además de pocos errores estándares y el valor AIC sea mínimo. El modelo es entrenado con datos de entrenamiento para posteriormente ser puesto a prueba con datos que no han sido vistos

por este para poder medir el rendimiento del modelo y su precisión al momento de pronosticar con base a datos nuevos no conocidos. El modelo puede seguir la tendencia a la baja con un MAE de 1000.

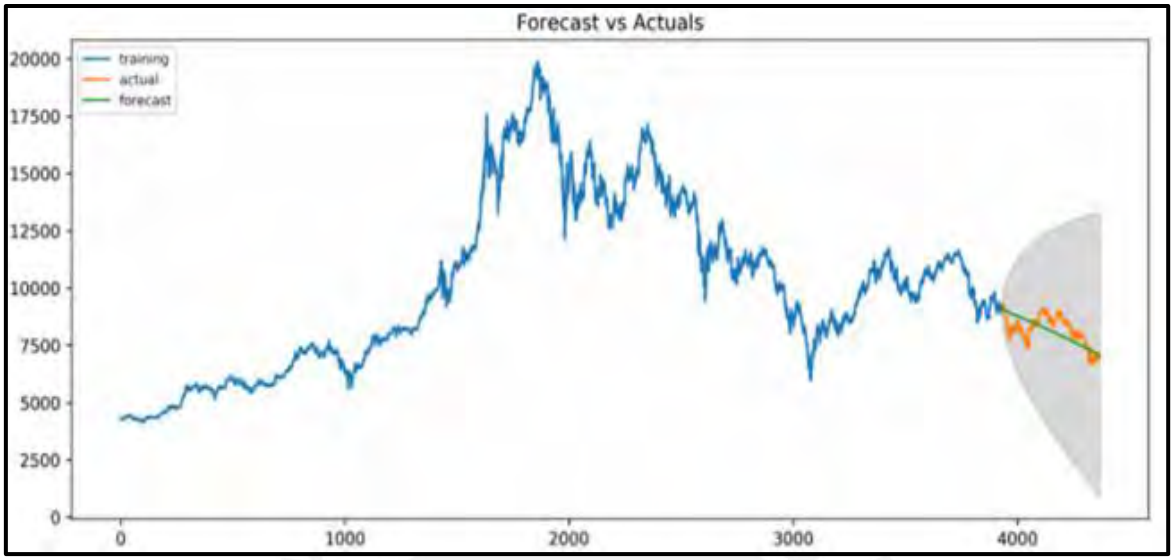


Figura 44: ARIMA – Comparativo del Pronóstico vs. Precio real
Fuente: (Pang, Y., Sundararaj, G., & Ren, J. 2019:39)

El modelo ARIMAX es una extensión del ARIMA, la única diferencia es que el modelo presenta una variable exógena adicional a la indexada a la serie de tiempo. Ello con el fin de poder incluir los 14 índices de sentimiento que contribuyen de forma significativo al precio del Bitcoin. Al añadir los datos de sentimiento se observa una mejora en la predicción del modelo pues se obtiene un MAE que se encuentra en el rango de 550.

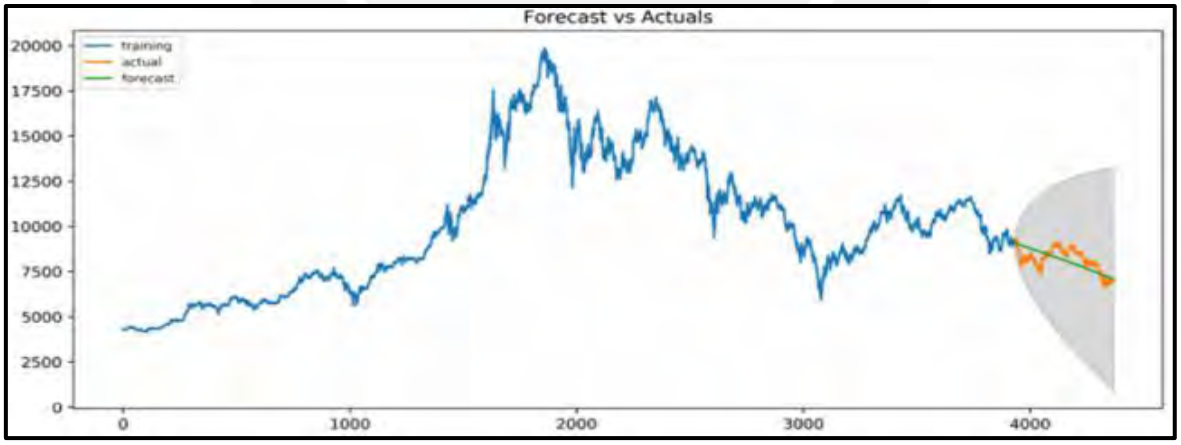


Figura 45: ARIMAX - Comparativo del Pronóstico vs. Precio real
Fuente: (Pang, Y., Sundararaj, G., & Ren, J. 2019:40)

Por su parte, las redes neuronales recurrentes (RNN) son un tipo de red neuronal artificial diseñada para reconocer patrones en datos que presentan secuencias como lo son los textos, escrituras a mano y series de tiempo. Este tipo de algoritmos toman en cuenta el tiempo y la secuencia ya que posee una dimensión temporal. La RNN no solo se alimenta de los datos actuales *t* sino que a su vez

también requiere los que percibieron en el tiempo *t-1*. Las RNN están compuestos por neuronas, capas ocultas y funciones de activación que sirven para calcular el valor esperado empleando *backpropagation*. La operación de cálculo puede ser descrita de la siguiente forma:

$$h_t = \varphi(Wx_t + Ux_{t-1}),$$

donde h_t representa el estado oculto en el paso de tiempo *t*. Es una función de la entrada x_t en el mismo paso de tiempo solo que esta se encuentra afectada por una matriz de pesos *W*, además se añade el estado oculto en el paso de tiempo *t-1* multiplicado por su propia matriz U. Las matrices de pesos son filtros que determinan la importancia que se le debe asignar a cada entrada actual como al estado oculto, estos pesos pueden ser definidos de forma inicial ya que mediante el *backpropagation* estos pesos se irán ajustado hasta encontrar el error mínimo, el cual podrá ser un mínimo local o mínimo global. La función de activación φ permite condensar valores muy grandes o muy pequeños en un espacio logístico, así como hacer que el vector gradiente funcione para el *backpropagation*. En el presente estudio los autores emplearon el modelo LSTM para describir la serie de tiempo y comparar el performance del modelo ARIMA.

Long Short-Term Memory (LSTM) es un modelo robusto de RRN que maneja el problema del gradiente decreciente, por lo tanto, se desempeña bien en datos temporales. El conjunto de datos es divido en datos de entrenamiento (*training set*) y datos de prueba (*test set*). La red LSTM se construye añadiendo capas de neuronas junto con funciones de activación y optimizadores. Se añade una capa de eliminación para eliminar el sobreajuste (*overfitting*). El modelo se entrena en lotes para un número definido de épocas y tamaño del lote. La cantidad de datos de entrenamiento puede ser expresada como una multiplicación del número de épocas y el tamaño del lote. La medida de error MAE ayuda en el cálculo del vector gradiente para los datos de entrenamiento.

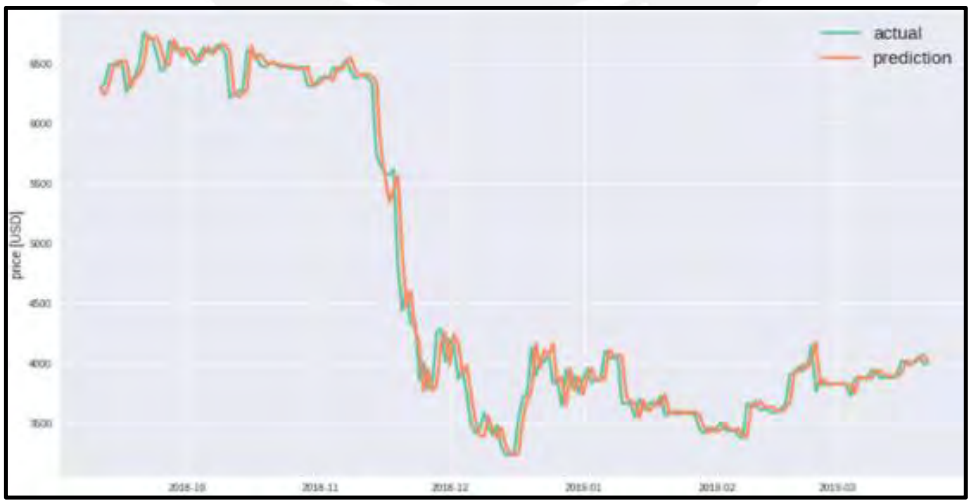


Figura 46: LSTM - Comparativo del Pronóstico vs. Precio real
Fuente: (Pang, Y., Sundararaj, G., & Ren, J. 2019:40)

En la Figura 46, se observa que los valores pronosticados son casi similares a los valores reales de la criptomoneda, sin embargo, se identifica que existe un desfase entre estas.

Los autores concluyen que los comportamientos del mercado pueden predecirse con cierto grado de precisión utilizando modelos de *machine learning*. Asimismo, resaltan el hecho de que los datos del sentimiento social están disponibles en abundancia y pueden ser aprovechados para estudiar el mercado y el comportamiento de las personas. Sin embargo, advierte que existen ciertos desafíos que superar ya que la recolección y almacenamiento de la información involucra cuestiones de privacidad y leyes de protección de datos. No obstante, los autores confían en que cuando se demuestre que este tipo de análisis es eficaz, la tecnología será acogida y adaptada a un ritmo mucho más rápido.

CAPÍTULO 3. DIAGNÓSTICO DE ETHEREUM

Ethereum es una *blockchain* descentralizada que establece una red *peer-to-peer* (P2P), la cual permite crear, ejecutar y verificar de forma segura el código de las aplicaciones comúnmente conocidas como *smart contracts*. Estos *smart contracts* son escritos en el lenguaje de programación Solidity[9] y estos a su vez utilizan la máquina virtual de Ethereum (EVM[10]) para poder ser ejecutados y desplegados a la *blockchain*. La máquina virtual de ETH puede ser definida matemáticamente como una función de transición de estado, ya que Ethereum a diferencia de las otras *blockchain* que son definidas como un libro contable público, esta se define a sí misma como una máquina de estado distribuida. La máquina virtual de Ethereum alberga un conjunto de reglas específicas para cambiar el estado de esta. De acuerdo con el capítulo 1.1.5. el protocolo de consenso de la red de Ethereum era PoW, el cual tenía como característica principal que no era ecoamigable pues demandaba una gran capacidad de poder computacional lo cual se traducía en altos costes energéticos. Según Digiconomist (2021) el consumo de energía por año de ETH y BTC en conjunto equipara al de un país como Italia y es casi igual al del Reino Unido (Ver Figura 47). No obstante, la actualización de Ethereum ETH 1.0 a ETH 2.0 busca abordar y solucionar dicha problemática ya que se plantea cambiar el protocolo de consenso de Prueba de trabajo (PoW) a Prueba de consenso (PoS) con lo cual se espera reducir en un 99.95% aproximadamente el costo energético, de acuerdo con estimaciones realizadas por Carl Beekhuizen (2021).

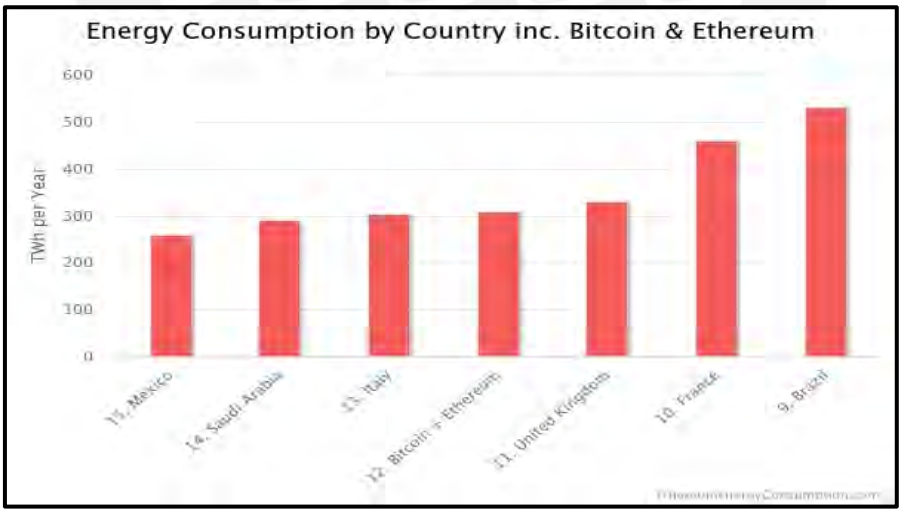


Figura 47: Consumo energético por países incluido BTC y ETH
Fuente: (Digiconomist 2021)

[9] Solidity es un lenguaje de programación de alto nivel orientado a objetos (POO) creado para desarrollar e implementar *smart contracts*. https://docs.soliditylang.org/en/v0.8.13/
[10] EVM es la máquina virtual global que permite a cada participante de la red de Ethereum ejecutar y almacenar acuerdos. https://ethereum.org/en/developers/docs/evm/

La criptomoneda de la red de Ethereum se denomina Ether (ETH), esta se utiliza para el pago de las comisiones de gas por cada transacción (Ver Tabla 16), por lo tanto, el Ether sirve como un mecanismo de control precio ya que el costo asociado al gas guarda relación con el poder computacional requerido para ejecutar la transacción y la demanda de la red para ejecutar dicho cálculo en ese instante de tiempo.

Tabla 16 Características de las principales Blockchain

	Ethereum	Avalanche	Solana	Polkadot
Transacciones por segundo	30 aprox.	10.29 aprox.	1954 aprox.	166.6 aprox.
Smart contracts	EVM	C-chain EVM	Solana BPF	Parachains
Nivel de descentralización	Alto	Medio	Bajo	Alto
Interoperabilidad	No	No	No	Sí
Comisión de Gas	3-15 USDT	0 aprox.	0 aprox.	0 aprox.
Criptomoneda	ETH	AVAX	SOL	DOT

Fuente: LeewayHertz

Las comisiones de gas ayudan a mantener la red segura pues de esta forma se asegura que personas maliciosas eviten enviar spam a la red ya que ello les generaría un costo asociado elevado por cada ejecución computacional. Otro uso del Ether es mediante *staking* en plataformas financieras descentralizadas (DeFi) donde el usuario puede prestar, solicitar y ganar intereses sobre sus ETH. En la Figura 48, se observa la evolución del precio del ETH y Gwei, este último representa los gastos en promedio de ETH asociados a las comisiones de gas por transacción, por lo tanto, se puede observar que los meses en los cuales hubo mayor actividad en la red fueron agosto de 2020 y marzo de 2021 (de Best, 2022).

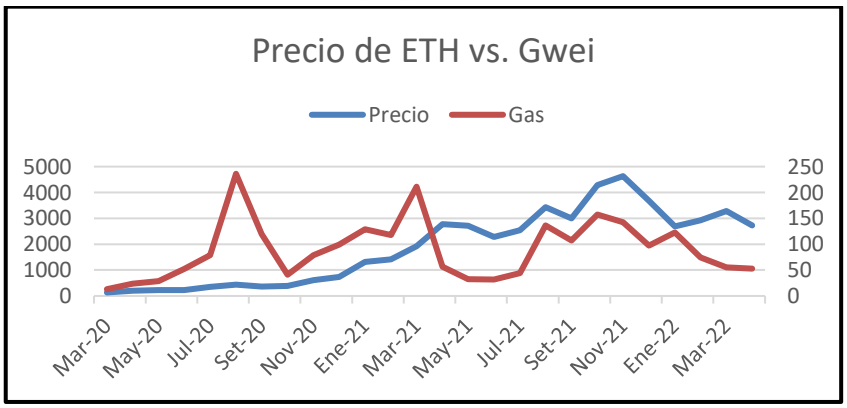


Figura 48: Precio del ETH vs. Gwei desde marzo-2020 hasta abril-2022
Fuente: Yahoo Finance y Statista

CAPÍTULO 4. CONSTRUCCIÓN DEL MODELO

El desarrollo del modelo constará de 5 etapas: recolección de información, preprocesamiento, aplicación de modelos, validación del modelo y análisis de resultados.

En la etapa de recolección de información, se extraerá la información financiera de la criptomoneda Ether (ETH) de Yahoo Finance mediante *Web Scrapping* para ello se utilizará la API de Yahoo Finance. Los datos financieros que ofrece esta plataforma son fecha, precio máximo, precio mínimo, precio al cierre, precio al cierre ajustado y volumen. Para el presente estudio se trabajará únicamente con el precio de cierre ajustado por día. Respecto a las opiniones o indicadores de sentimiento, serán recolectadas de la plataforma financiera y sitio web de noticias Investing. De igual forma se aplicará la técnica *Web Scrapping* para extraer los titulares de las noticias para ello se hará uso de la librería *Beautiful Soup* de Python que nos permite extraer información en formato HTML o XML, la información recopilada de Investing también se hará en días. El período de recolección de información constará desde el 12 de marzo de 2020 hasta 10 de abril del 2022.

En la etapa de preprocesamiento de datos, se aplicarán metodologías y se hará uso de librerías de Python que permitan limpiar la información recopilada y conservar su resumen estadístico, ello con el fin de disminuir el ruido presente en la información y no afectar la distribución de la fuente de origen de los datos. Respecto a los datos de Investing, se procederá a filtrar la información únicamente publicada en idioma inglés ya que los diccionarios léxicos a utilizar trabajan en dicho idioma, como se mencionó en líneas anteriores, los diccionarios léxicos en español todavía se encuentran en una etapa beta lo que significa que no son tan precisos a diferencia de los diccionarios léxicos de inglés. Posteriormente, se realizará la transformación de todas las palabras contenidas en los titulares de las noticias a minúsculo con el fin de homogenizar la data. Adicionalmente, se removerán los símbolos monetarios, caracteres numéricos, se eliminará los espacios en blanco, *stop words* y se reducirá la cantidad de letras que se repiten en las palabras. Finalmente, se utilizará el algoritmo de VADER para polarizar los titulares extraídos de Investing. Respecto a la data financiera de la criptomoneda en estudio, solo se realizarán transformaciones o diferenciaciones a los datos si es que no existe estacionariedad en la serie temporal todo ello con el fin de volverla estacionaria ya que el modelo a emplearse requiere que se cumpla dicho requisito.

En la etapa Aplicación de Modelos, luego de haber logrado que la serie de tiempo sea estacionaria y encontrado los parámetros del modelo ARIMA que minimizan el AIC se extraerán características de las fechas con el fin de capturar el paso del tiempo y la periodicidad de este mismo, además de ello se integrará una variable exógena, sentimiento del mercado, la cual ha sido hallada a través del análisis de los titulares de noticias sobre criptomonedas de la plataforma financiera Investing. Luego de haber seleccionado las características que contribuyen al performance del modelo se hará uso de modelos

lineares sin regularización y con regularización, así como modelos del grupo de la familia de funciones de árboles de decisión, bosques aleatorios, etc.

En la etapa validación del modelo, se utilizarán medidas de error como MAE O RSME para poder evaluar el desempeño de los pronósticos realizadas por el modelo desarrollado. Por último, en la etapa de análisis de resultados se presentará los resultados obtenidos de las pruebas y se contrastará el valor esperado versus el valor obtenido; y se dará recomendaciones para futuras investigaciones relacionadas con el tema de estudio. En las Figuras 49 y 50 se puede observar el diagrama de flujo que seguirá la metodología aplicada para el presente trabajo de estudio.

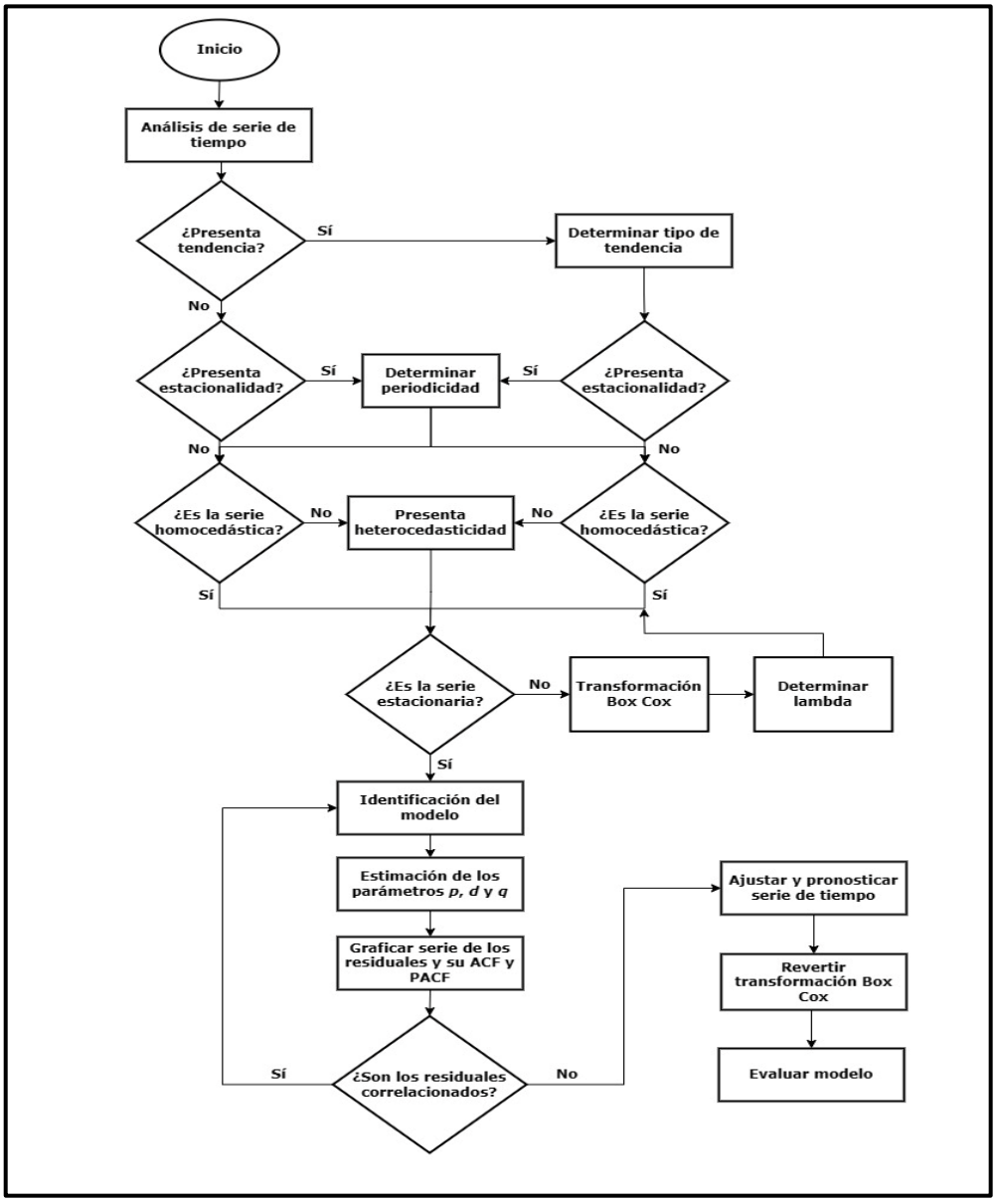


Figura 49: Flujo del método ARIMA

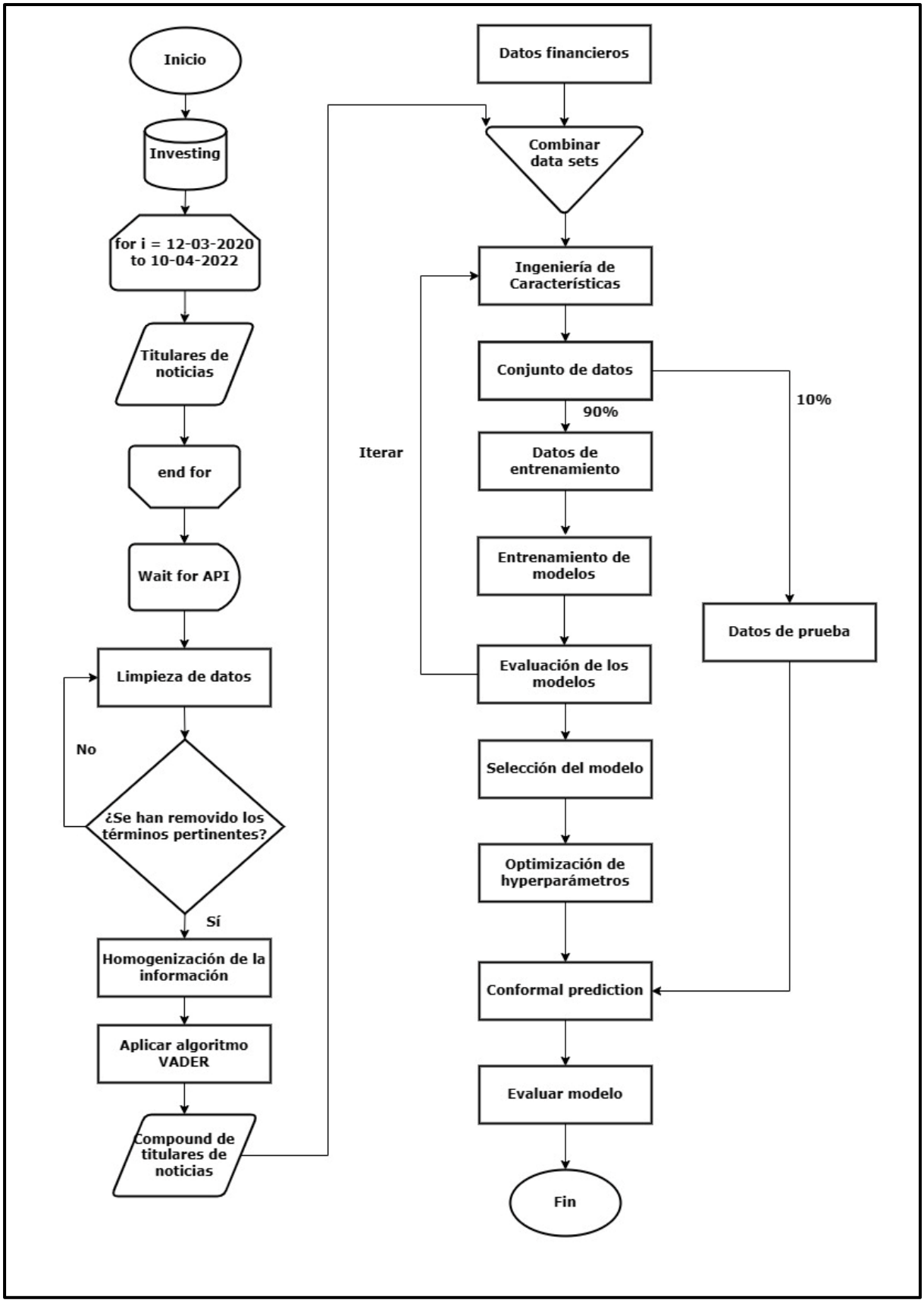


Figura 50: Flujo de la metodología del trabajo

4.1. Recolección de información

Datos Financieros

El intervalo de tiempo de estudio del precio de la criptomoneda Ether (ETH) va desde el 12 de marzo de 2020 hasta el 10 de abril de 2022. La información o data histórica será extraída de la plataforma Yahoo Finance mediante su API, dentro de esta se encontrarán variables tales como fecha, precio de apertura, precio máximo, precio mínimo, precio de cierre, precio de cierre ajustado y volumen todo ello será extraído en una frecuencia diaria. Sin embargo, para el presente estudió se trabajará únicamente con la fecha y el precio de cierre ajustado por día. En la figura 51, se puede observar el dataframe obtenido de Yahoo Finance, en el se encuentran las variables descritas anteriormente. Asimismo, en la Figura 52 se observa el script empleado para realizar la extracción de los datos financieros de la plataforma Yahoo Finance.

Date	High	Low	Open	Close	Volume	Adj Close
2020-03-12	195.147934	111.210709	194.738922	112.347122	22134741655	112.347122
2020-03-13	137.429535	95.184303	112.689995	133.201813	27864623061	133.201813
2020-03-14	134.484375	122.414474	133.582474	123.306023	12740784545	123.306023
2020-03-15	132.242142	121.853653	123.246063	125.214302	12719251813	125.214302
2020-03-16	124.996117	105.171440	124.996117	110.605873	15984904590	110.605873

Figura 51: Datos del precio del ETH del 12/03/2020 al 10/04/2022
Fuente: Yahoo Finance

```
import pandas as pd
import yfinance as yf
import datetime as dt
```

✓ 0.0s Python

```
ETH = yf.download("ETH-USD", start="2020-03-12", end="2022-04-11", interval="1d")
```

✓ 0.5s Python

Figura 52: Código para la extracción de la información financiera

Datos Informativos

Los datos sobre el sentimiento del mercado serán extraídos de la plataforma financiera y sitio web de noticias Investing con una frecuencia diaria en el rango de estudio que va desde el 12 de marzo del 2020 hasta el 10 de abril del 2022, todo esto se logrará mediante el uso de técnicas de Web Scrapping

para ello se hará uso de la librería *Beautiful Soup*[11], la cual nos permitirá extraer datos de archivos HTML y XML. De la plataforma Investing se podrá extraer variables tales como los titulares de las noticias, el contenido de la noticia, la fecha y el autor asociado a esta. Asimismo, se filtrarán las noticias según su idioma, pues se estudiarán únicamente las noticias que contemplen el idioma inglés, es decir, hayan sido redactadas en inglés. Esto debido a que la mayoría de diccionarios léxicos están en inglés y el método de VADER utiliza este tipo de diccionarios. En el presente estudio solo se hará uso de la fecha en la cual se publicó la noticia y el titular asociada a esta, ello con el fin de reducir la carga operativa y optimizar los recursos del computador. En la Tabla 17, se muestran los 10 registros más recientes de las noticias extraídas de Investing. En la Figura 53 se muestra el código para la extracción de los titulares de noticias aplicando la técnica de *Web Scrapping*, este script fue desarrollado en Python y a través de este se obtuvieron los resultados visualizados en la Tabla 17.

Tabla 17 Noticias del período 12/03/2020 al 10/04/2022

Statement	Date
Number of UK crypto firms operating under FCA temporary registration status drops	10/04/2022
NFT Investments PLC mulls £96M acquisition of Pluto Digital	10/04/2022
Starbucks joins NFT party, UK government seeks stablecoin regulations and Crypto Twitter rallies behind cancer fighter, Hodler's Digest: Apr. 3-9	10/04/2022
ECB executive board member talks about current state of digital euro CBDC research	10/04/2022
Texas-based Bitcoin mining operator files for $60M IPO	10/04/2022
Finance Redefined: Axie Infinity creator raises $150M, DApp daily users surge to 2.4M and more	10/04/2022
60 Minutes feature on El Salvador's Bitcoin Beach will air Sunday	10/04/2022
Altcoin Roundup: Interoperability push puts attention back on Polkadot	10/04/2022
Kyber Network (KNC) soars after integrating with Uniswap v3 and Avalanche Rush Phase 2	10/04/2022
Museums in the metaverse: How Web3 technology can help historical sites	10/04/2022

[11] https://beautiful-soup-4.readthedocs.io/en/latest/

```
[ ] from bs4 import BeautifulSoup
    import urllib.request,sys,time
    import requests
    import pandas as pd
    #Ajustar ancho de la columna del DataFrame a 900
    pd.set_option('max_colwidth', 900)
```

```
#Crear Lista en blanco donde se almacenarán los datos a extraer
frame=[]
#Loop de páginas que contienen las noticias publicadas en el rango de estudio
for page in range(272,2564):
#La consola nos indicará de qué página se está extrayendo la información
    print('processing page :', page)
#el dominio del sitio web más la pág. que contiene la información a extraerse
    url = 'https://www.investing.com/news/cryptocurrency-news/'+str(page)
    print(url)#Imprimira el enlace de donde se alojan las noticias extraídas

#Se utilizará para detectar en qué página ocurrió un problema
    try:
        #Se solicita a la API obtener información de la página
        page=requests.get(url)
#Se ejecutará si se encuentra un error al momento de que la API solicite info
    except Exception as e:
        error_type, error_obj, error_info = sys.exc_info()
         #Imprimir el URL donde ocurrió el problema
        print ('ERROR FOR LINK:',url)
         #Imprimir información del problema encontrado
        print (error_type, 'Line:', error_info.tb_lineno)
        continue
    #Tiempo de descanso que se le brinda a la consola entre solicitud
    time.sleep(9)
    #Esta transformando el contenido de la página a HTML
    soup=BeautifulSoup(page.text,'html.parser')
    #Especificar tag y clase del objeto donde se encuentra información
    links = soup.find('div',
                      attrs={'class':'largeTitle'}).find_all('article',

                      attrs={'class':'js-article-item articleItem '})
                                                                      r
    for j in links:
#Se selecciona del objeto anterior el Titulo asociado a la noticia
        Statement = j.find("a",attrs={'class':'title'}).text.strip()
#Se selecciona del objeto anterior la Fecha de publicación de la noticia
        Date = j.find('span',attrs={'class':'date'}).text[-12:]
#Se añade ambas informaciones a la lista que fue creada al inicio
        frame.append((Statement,Date))
#Transforma la variable frame del tipo de dato "lista" a "DataFrame"
data=pd.DataFrame(frame, columns=['Statement','Date'])
```

Figura 53: Código para la extracción de los titulares de noticias

4.2. Preprocesamiento de datos

Datos de Twitter

Con base a la información expuesta en la Tabla 17, se logra identificar que los titulares de las noticias extraídas no se encuentran listos para ser procesados en el modelo pues contienen símbolos de puntuación, símbolos de dólar ($), símbolos de libra esterlina (£), caracteres numéricos y *stop words*. Por lo tanto, es necesario limpiar la información y homogenizarla, esto último se logrará transformando todo el texto a minúsculo para posteriormente ser utilizada en el algoritmo VADER. En primer lugar, se procederá a realizar a remoción de los componentes señalados anteriormente (Ver Tabla 18).

Tabla 18 Limpieza y homogenización de los titulares de las noticias

Statement	Date
number uk crypto firms operating fca temporary registration status drops	10/04/2022
nft investments plc mulls m acquisition pluto digital	10/04/2022
starbucks joins nft party uk government seeks stablecoin regulations crypto twitter rallies cancer fighter hodler's digest apr	10/04/2022
ecb executive board member talks current state digital euro cbdc research	10/04/2022
texasbased bitcoin mining operator files m ipo	10/04/2022
finance redefined axie infinity creator raises m dapp daily users surge m	10/04/2022
minutes feature el salvador's bitcoin beach air sunday	10/04/2022
altcoin roundup interoperability push puts attention polkadot	10/04/2022
kyber network knc soars integrating uniswap v avalanche rush phase	10/04/2022
museums metaverse web technology help historical sites	10/04/2022

Se observa que los titulares de noticias ya no presentan símbolos de dólar, símbolos de libras esterlinas, símbolos de puntuación, caracteres numéricos ni *stop words*, por lo tanto, el paso siguiente a efectuarse se centrará en hacer un análisis exploratorio de los términos o palabras que contienen los titulares de noticias para ello se armará un corpus con el fin de contabilizar la frecuencia con la que aparecen los términos en todos los titulares contenidos en el dataset. Sin embargo, dado que se busca obtener los términos más significativos del conglomerado de noticias, se restringirá la cantidad de términos a visualizarse pues se mostrarán únicamente aquellas palabras que como mínimo presenten una frecuencia de 230. Para presenta la información se emplearán técnicas de visualización como gráficos de barra y nube de palabras con el fin de poder exponer la información encontrada de forma gráfica y natural. Asimismo, se presentará un agrupamiento jerárquico con la final de hallar los términos similares. En la Figura 54, se puede visualizar que el término que se presenta con mayor frecuencia y ocupa el primer lugar es bitcoin pues ha sido mencionado aproximadamente 2500 veces. El término que ocupa el segundo lugar es crypto, el cual posee 2384 menciones. En el tercer y cuarto puesto, se encuentran las palabras *price* y Ethereum, las cuales han sido mencionadas 1015 y 990 veces respectivamente. Cabe resaltar que la palabra bitcoin también se encuentra en el octavo puesto con su

forma abreviada BTC, por lo tanto, la cifra asociada a la frecuencia ascendería en realidad a 3214. No obstante, se observa que ADA, la criptomoneda de la blockchain de Cardano también se encuentra contenido en las noticias, lo mismo sucede con Ripple (XRP). Esto también puede ser observado de mejor manera en la nube de palabras (Ver Figura 55) donde los términos con mayor frecuencia poseen un mayor tamaño. Además de ello, presentan un color verde más oscuro mientras que aquellos términos que presentan una menor frecuencia de ocurrencia poseen una coloración tenue y un menor tamaño.

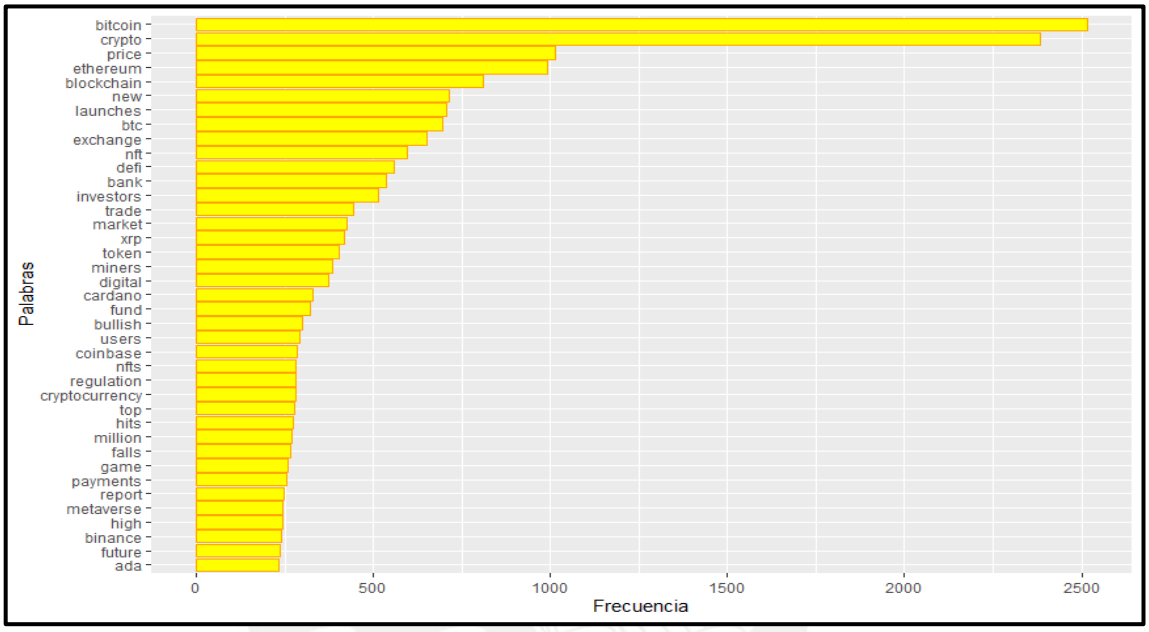


Figura 54 Términos con mayor frecuencia en el corpus de los titulares de noticias

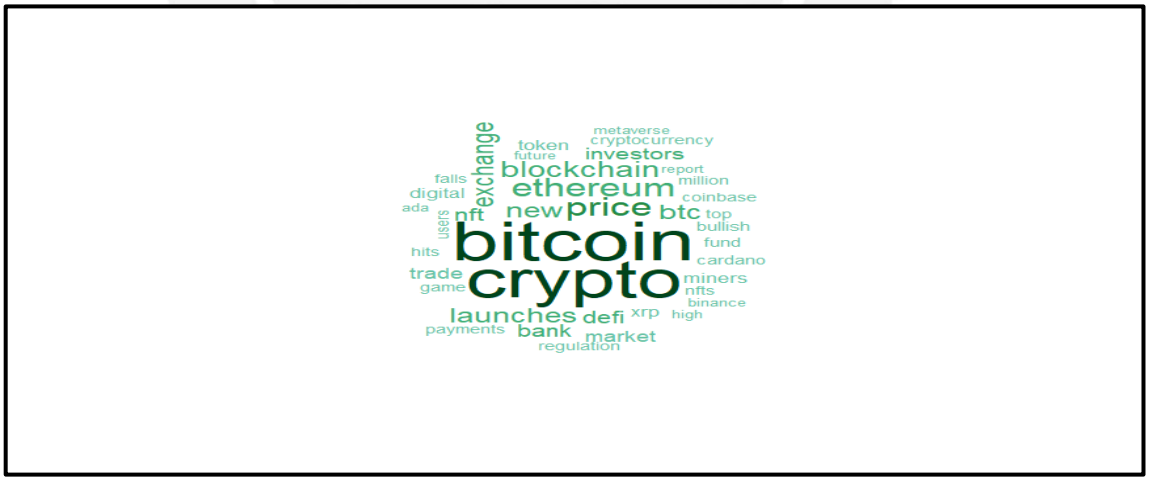


Figura 55: Nube de los términos más representativos del corpus

Por último, en la Figura 56 se presenta el dendograma que muestra el grado de proximidad o similitud entre los términos. Se observa que se utilizó una partición de 3 conglomerados. En el primero, se encuentra crypto, en el segundo bitcoin y el tercero conglomerado contiene los términos Ethereum, BTC y price se puede intuir que el nivel de similitud se debe a que hacen referencia al precio de las criptomonedas de ambas cadenas de bloque (*blockchain*). De la misma forma, las palabras *launches* y

new se encuentran agrupadas pues puede deberse a que habrá lanzamiento de nuevos tokens o criptomonedas mediante un ICO (*Initial coin offering*).

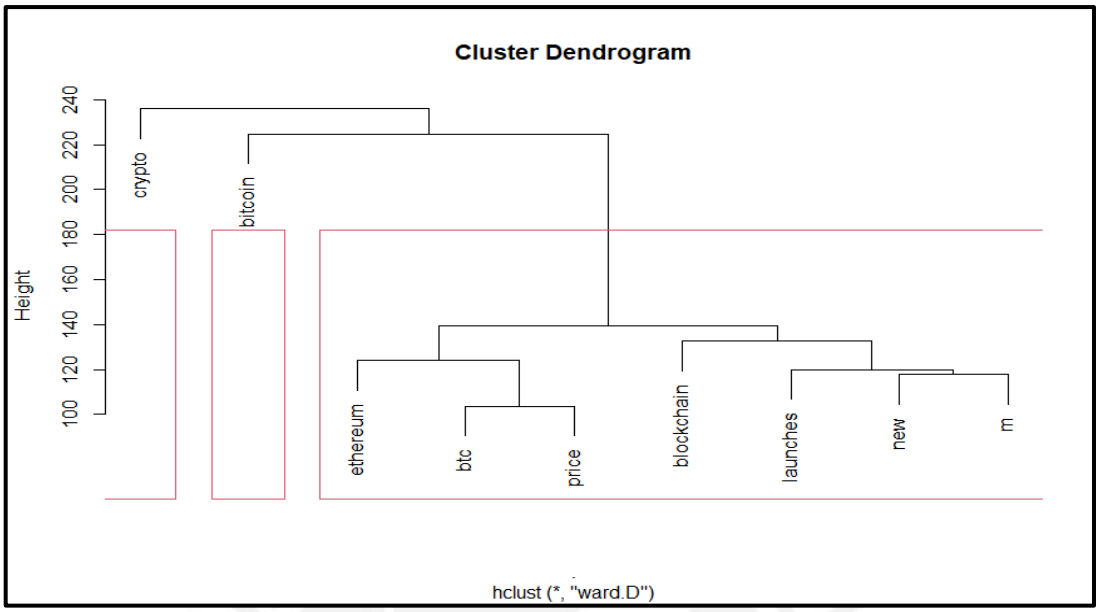


Figura 56: Dendograma de los términos más representativos del corpus

Finalmente, luego de haber preprocesado y homogenizado la información de los titulares de las noticias, se procederá a aplicar el algoritmo de VADER con el fin de poder obtener el sentimiento o polaridad asociado a cada uno de estos. En la Tabla 19, se observa el *compound* asociado a cada titular de noticia, este valor se halla mediante la suma de los sentimiento positivos, negativos y neutrales, luego de ello se normaliza entre -1 que representa el sentimiento más negativo y +1 que indica que el titular de la noticia es extremadamente positivo. El titular asociado al primer registro "*bitcoin plummets cryptocurrencies suffer market turmoil*" traduciéndolo al español sería lo siguiente "el bitcoin se desploma las criptodivisas sufren las turbulencias del mercado". Se observa que los términos *plummets*, *suffer*, market y *turmoil* denotan un sentimiento o polaridad negativa, por lo tanto, el resultado que se ve reflejado en el Compound (-0.7184) sería coherente. Si se analiza el segundo registro "jpmorgan chase settles suit credit card crypto purchases" de forma traducida indica lo siguiente "jpmorgan chase resuelve la demanda de compras de criptomonedas con tarjeta de crédito". Se sabe que JP Morgan Chase es una compañía financiera transnacional reconocida globalmente, por ende, el hecho que resuelva la demanda de compras de criptomonedas con tarjetas de crédito conllevaría a ser una noticia alentadora y positiva para las personas que acostumbran utilizar este medio de pago para la compra de criptomonedas. De la misma forma, se visualizar que el *compound* asociado a esta noticia es mayor a 0 (0.3818) lo cual corrobora lo mencionado anteriormente. Cabe resaltar que si se observa la columna *Date*, las fechas asociadas a este atributo son las mismas, esto se debe a qué cómo se mencionó al inicio cuando se realizó el *Web Scrapping* en el intervalo de tiempo de estudio puede que para un día determinado se hayan publicado más de una noticia, por ende, es necesario agrupar los términos

mediante la fecha indexada con el fin de obtener el promedio de los *compounds* para cada día en específico. Si no se realiza este procedimiento cuando realizamos la unión de los datos financieros y datos informativos mediante las fechas indexadas, el sistema nos arrojará un error pues el tamaño de los datos informativos es mayor al de los datos financieros.

Tabla 19 Compound de titulares de noticias

Date	Statement	Compound	Positivo	Negativo	Neutral
12/03/2020	bitcoin plummets cryptocurrencies suffer market turmoil	-0.7184	0	0.6	0.4
12/03/2020	jpmorgan chase settles suit credit card crypto purchases	0.3818	0.271	0	0.729
12/03/2020	libra members hedge bets joining rival stablecoin project	0	0	0	1
12/03/2020	carney sees big challenges boe eyes digital banknotes	0.0772	0.157	0	0.843
12/03/2020	bitcoin price slips k lows trump europe ban	-0.6597	0	0.474	0.526

En la Tabla 20, se muestra los resultados obtenidos luego de haber promediado todos los *compound* de acuerdo con su fecha indexada. Es fácil notar que para el período del 12 de marzo hasta el 21 de marzo del 2020, el cual se compone en total de 10 días. En 6 de estos se registraron sentimientos negativos mientras que solo en 4 el promedio de los tweets asociados a cada día tuvo una polaridad positiva.

Tabla 20 Compound promedio de los titulares de las noticias

Date	Average Compound
12/03/2020	-0.007206
13/03/2020	-0.036471
14/03/2020	-0.243395
15/03/2020	-0.141200
16/03/2020	-0.101484
17/03/2020	0.170627
18/03/2020	0.047238
19/03/2020	-0.013690
20/03/2020	0.044710
21/03/2020	0.6124

Datos Financieros

El conjunto de datos será divido en datos de entrenamiento y datos de prueba cuya proporción será de 90% y 10% respectivamente. A diferencia de la metodología tradicional empleada en *machine learning* donde las observaciones son mezcladas de forma aleatoria para posteriormente ser divididas

en datos de entrenamiento y prueba con el fin de poder validar el modelo. Este enfoque no es propicio para las series de tiempo ya que generaría que se entrene en algunos casos con datos recientes y se pronostique el pasado lo cual no sería muy coherente. Por dicha razón, en las series de tiempo se toman las observaciones más antiguas como datos de entrenamiento y las observaciones más recientes como datos de validación para validar el modelo, a ello se le conoce como validación fuera de tiempo (o *out-of-sample validation*) para lograr esto se emplearán las funciones *head* y *tail* de la librería *pandas*[12]. Posterior a la importación del dataset y la división de estos se procederá con el análisis de la serie de tiempo. Auffarth (2021) define el análisis de las series de tiempo (TSA) como el proceso de extracción de información mediante un resumen estadístico de la serie de tiempo, pero sobre todo y más importante el análisis de la tendencia y estacionalidad de la serie. Pues como se explicó en el capítulo 1.3, para poder pronosticar series de tiempo se requiere que la media y varianza sean constantes, es decir, las propiedades estadísticas de la distribución de los datos permanezcan constantes a lo largo del tiempo, por lo cual es requisito indispensable asegurar la estacionariedad de la serie ya sea mediante diferenciaciones de primer o segundo orden, o mediante transformaciones simples o complejas como la transformación Box-Cox. En la figura 57, se observa la evolución del precio del ETH en el rango de estudio.

Figura 57: Evolución del precio del ETH del 12/03/2020 al 10/04/2022

En primera instancia, se logra identificar de la serie de tiempo que tanto la media como la varianza no son constantes a lo largo del tiempo, por ende, la serie temporal en estudio no sigue un proceso estacionario.

[12] https://pandas.pydata.org/docs/

Si bien se ha determinado que la serie no es estacionaria de forma visual, existen pruebas estadísticas para determinar si un proceso es estacionario o no. La prueba más utilizada es la prueba de Dickey-Fuller aumentada (ADF), sin embargo, la prueba de Kwiatkowski-Phillips-Schmidt-Shin (KPSS) también es utilizada para comprobar si un proceso es estacionario. En la prueba ADF, la hipótesis nula señala que una raíz unitaria está presente en una muestra de series de tiempo para cierto intervalo de confianza y la hipótesis alternativa es que no existe presencia de una raíz unitaria, por ende, el proceso es estacionario. Por su parte la hipótesis nula de la prueba KPPS, indica que el proceso es estacionario y la hipótesis alternativa que el proceso no es estacionario. En ambos casos, se trabajará con un intervalo de confianza del 95%, por ende, el nivel de significancia o significación será de 5%. Para el caso de la prueba de Dickey-Fuller aumentada se obtiene un *p-value* de 0.347 (Ver Figura 41), por ende, no se rechaza la hipótesis nula lo cual quiere decir que la serie de tiempo no es estacionaria. De la misma forma, se observa en la Figura 42 que la prueba KPSS refuerza el resultado hallado ya que el *p-value* (0) que arroja esta prueba es menor a 0.05, por lo tanto, se rechaza la hipótesis nula, ello significa que el proceso no es estacionario.

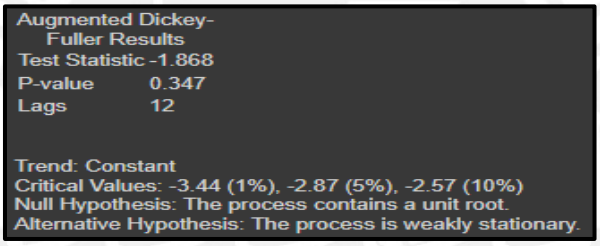


Figura 58: Resultado de prueba Dickey-Fuller Aumentada (ADF)

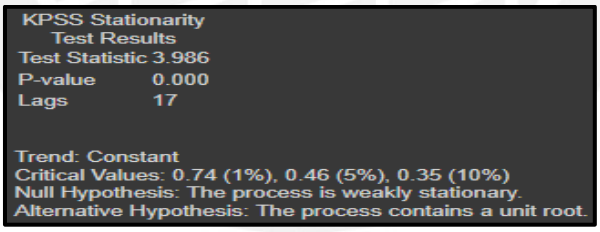


Figura 59: Resultado de prueba Kwiatkowski-Phillips-Schmidt-Shin (KPSS)

De acuerdo con los resultados obtenidos de las pruebas estadísticas realizadas, se observa que la premisa basada en el comportamiento de la serie de tiempo de forma visual era correcta pues las pruebas validan que efectivamente la serie no es estacionaria.

Existen diversas técnicas para remover la raíz unitaria y volver la serie estacionaria, una de ellas es mediante diferenciaciones, esto es restar las observaciones posteriores entre sí:

$$z_t = y_t - y_{t-1}$$

Otra forma consiste en dividir los valores posteriores entre sí:

$$z_t = \frac{y_t}{y_{t-1}}$$

No obstante, estos métodos no siempre resolverán el problema de la estacionariedad, ya que, si la serie de tiempo presenta una tendencia no lineal o por partes, estos métodos no eliminarán ese tipo de no estacionariedad. Además de ello, estos métodos presentan ciertas desventajas que se detallarán a continuación. Por ejemplo, se pierde la escala de la serie de tiempo al momento de modelarla. Otra desventaja, se debe a que cuando se utiliza la diferenciación para entrenar al modelo y posteriormente hacer pronósticos se requiere invertir la transformación después de obtener los valores de salida del modelo, ello agrega una complejidad adicional que gestionar.

Por lo tanto, se van a emplear otras técnicas para detectar si la serie presenta tendencia y de ser así proceder a removerla. Una de las formas de detectar si la serie de tiempo presenta tendencia es mediante el Tau de Kendall, el cual calcula la correlación de rango entre dos variables. Este tipo de prueba es no paramétrica, por ende, no realiza suposiciones sobre los datos. Si se utiliza la serie temporal como una de las variables a estudiar y la otra variable se define como la representación ordinal del tiempo entonces el Tau de Kendall resultante indicaría la tendencia de la serie temporal. En la Figura 61, se observan los resultados obtenidos. La dirección nos indica si se trata de una tendencia creciente o decreciente para este caso en particular se trata de una tendencia creciente. El valor asignado al *slope* representa al coeficiente de Tau. El rango de valores que puede tomar dicho coeficiente va desde -1 hasta 1, donde 0 indica que no existe correlación dado que para este ejemplo es 0.78179, es decir, es cercano a 1 lo cual indicaría que la tendencia presente en la serie de tiempo es fuerte. Además de ello, se observa que el *p-value* es menor que el nivel de significancia establecido (0.05), por ende, se concluye que la serie es estadísticamente significativa. El variable *deterministic* indica si la serie de tiempo presenta una tendencia determinística o estocástica, dado que el valor resultante es *False*, ello indicaría que se trata de una tendencia estocástica.

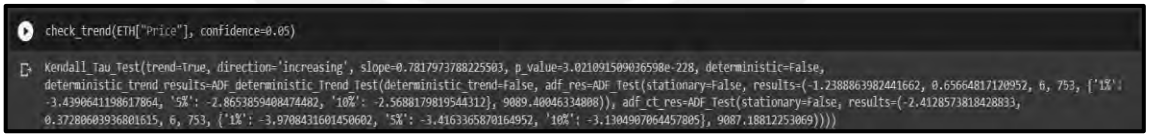


Figura 60: Resultado de prueba Tau de Kendall

Dado que se ha determinado que la serie de tiempo presenta una tendencia creciente estocástica, es necesario determinar si presenta estacionalidad para poder corregirla. Existen dos métodos que permiten identificar si la serie de tiempo en estudio presenta estacionalidad: gráfico de autocorrelación o las series de Fourier. Sin embargo, en este estudio se empleará únicamente el primero. Un gráfico de autocorrelación permite visualizar la correlación que existe entre la serie de tiempo y sus *lags*. Por lo tanto, si existiera estacionalidad se esperaría que la correlación aumente gradualmente hasta que llegue a un pico para posteriormente comenzar a decrecer a medida que se aleja en el tiempo. Se observa que la serie temporal no presenta estacionalidad (Ver Figura 61).

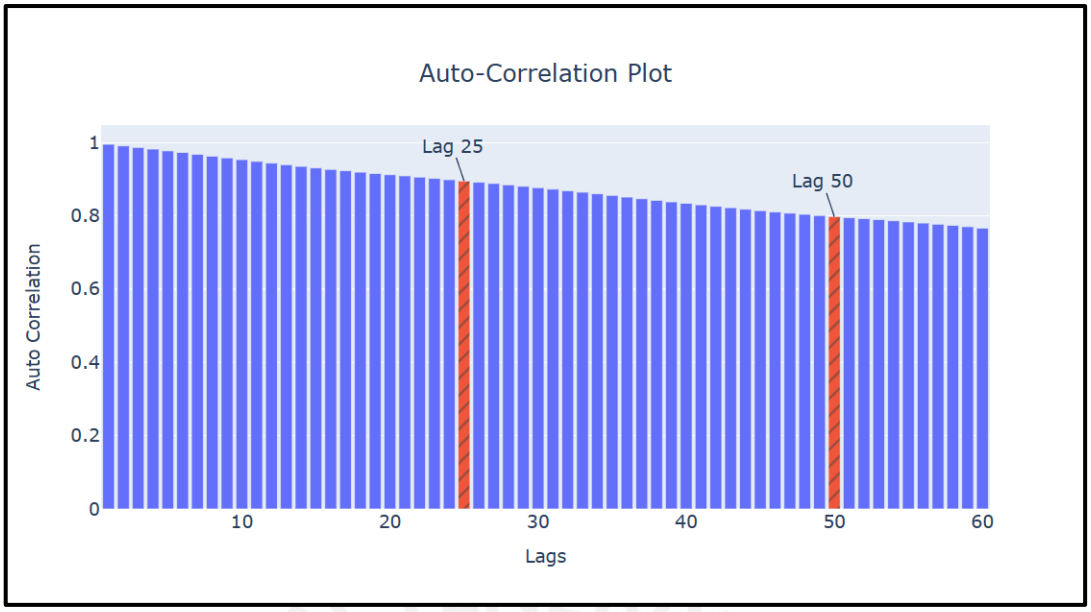


Figura 61: Autocorrelación del precio del ETH

Finalmente, se analizará si la serie presenta heterocedasticidad lo cual significa que la variabilidad o dispersión de la serie varía a lo largo del tiempo. Si bien en la Figura 57, se puede determinar si la serie presenta heterocedasticidad mediante una inspección visual es recomendable validar dicha hipótesis o premisa a través de una prueba estadística para ello se empleará el *White test*. Esta prueba utiliza una función de regresión para verificar si la varianza es constante. Cabe resaltar que este tipo de prueba únicamente considera la tendencia en la regresión, si existiese estacionalidad no funcionará correctamente. No obstante, dado que se ha comprobado que la serie temporal en estudio no presenta estacionalidad se procederá a emplearla.

```
check_heteroscedastisticity(ETH["Price"], confidence=0.05)
White_Test(heteroscedastic=True, lm_statistic=171.1337834385401, lm_p_value=6.898760891286815e-38)
```

Figura 62: White test

En la Figura 62, se observa que la serie presenta heterocedasticidad dado que el valor asociado al parámetro *heterocedastic* es igual a *True*, el parámetro *lm_statistic* indica el valor asociado al estadístico Multiplicador de Lagrande y el parámetro restante indica el *p-value* asociado al estadístico. Dado que se ha detectado que la serie presenta heterocedasticidad el paso siguiente debería ser removerla para ello se cuentan con diversas metodologías como aplicar una transformación logarítmica o una transformación Box-Cox. La transformación logarítmica tiene dos propiedades principales: estabilizar la variación y reducir la simetría. Sin embargo, Joseph (2022) menciona que se ha demostrado que la transformación logarítmica no siempre estabiliza la varianza. Además de ello, dicho

método genera otro problema, ya que ahora la optimización de la función de pérdida ocurre en la escala logarítmica lo cual podría afectar el aprendizaje del modelo pues este podría pensar que la pérdida es pequeña pero cuando se invierta la transformación este pequeño número puede transformarse en uno muy grande. Por otro lado, la transformación Box-Cox es una transformación logarítmica generalizada, por ende, la transformación logarítmica es solo un caso especial para cuando λ es igual a 0. Este método también presenta las mismas desventajas que la transformación logarítmica pero el grado que esos efectos están presente varía, además de ello mediante el uso del λ se puede controlar dichos efectos adversos. La transformación Box-Cox sigue la siguiente forma:

$$y(\lambda) = \frac{y^{\lambda} - 1}{\lambda}, \lambda \neq 0$$

$$y(\lambda) = \log(y), \lambda = 0$$

Existe una gran variedad de métodos automatizados que permiten hallar el λ óptimo para cualquier distribución de datos. Para este caso en específico, se utilizará el método de Guerrero que fue desarrollado en 1993.

Figura 63: Transformación Box-Cox con lambda -0.03759

De acuerdo con los resultados obtenidos de las pruebas estadísticas realizadas, se observa que la serie de tiempo presenta una tendencia estocástica. Por lo tanto, se procederá a realizar las diferenciaciones siguiendo el flujo del proceso presentado en la Figura 49. No obstante, primero se debe hallar el orden de la diferenciación con el fin de determinar si se realizará una diferenciación de primer o segundo orden. La diferenciación de primer orden representa las tendencias lineales mientras que la

diferenciación de segundo orden representa las tendencias cuadráticas. Para identificar si la serie temporal en estudio presenta una tendencia lineal o cuadrática es necesario descomponerla. Recordemos que una serie de tiempo puede ser descompuesta de forma aditiva o multiplicativa, por lo tanto, se procederá a descomponer la serie temporal y se graficarán sus componentes de tendencia, estacionalidad y error (Ver Figura 64).

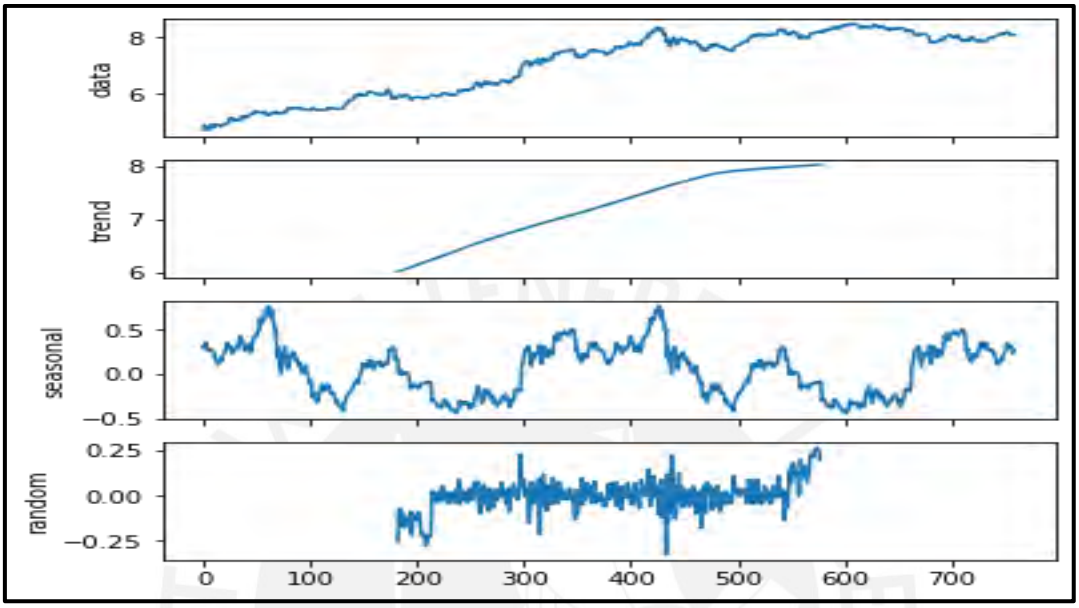


Figura 64: Serie de tiempo descompuesta en tendencia, estacionalidad y error

De la figura 64, se observa que la tendencia sigue un comportamiento lineal, por ende, se deberá utilizar una diferenciación de primer orden. Cabe resaltar que para determinar el grado de la diferenciación también se puede utilizar la prueba de Dickey-Fuller aumentada o la prueba KPP para ello se hará uso de la librería *pmdarima*. En la Figura 65, se observa que se obtiene 1 utilizando la prueba ADF lo cual corrobora el resultado hallado previamente de forma gráfica, por lo tanto, la diferenciación a realizarse deberá ser de primer orden.

```
from pmdarima.arima.utils import ndiffs
# Prueba ADF:
ndiffs(df_ETH, alpha=0.05, test='adf')
```

```
1
```

Figura 65: Número de diferenciaciones según prueba de ADF

Se procede a graficar nuevamente la evolución del precio del ETH luego de haber aplicado la diferenciación de primer orden. En la Figura 66, se observa que la serie de tiempo en estudio es una serie temporal de alta frecuencia. Las características principales de una serie de tiempo de alta

frecuencia son que la media es constante a lo largo del tiempo, no existe estacionalidad y una varianza que no es constante.

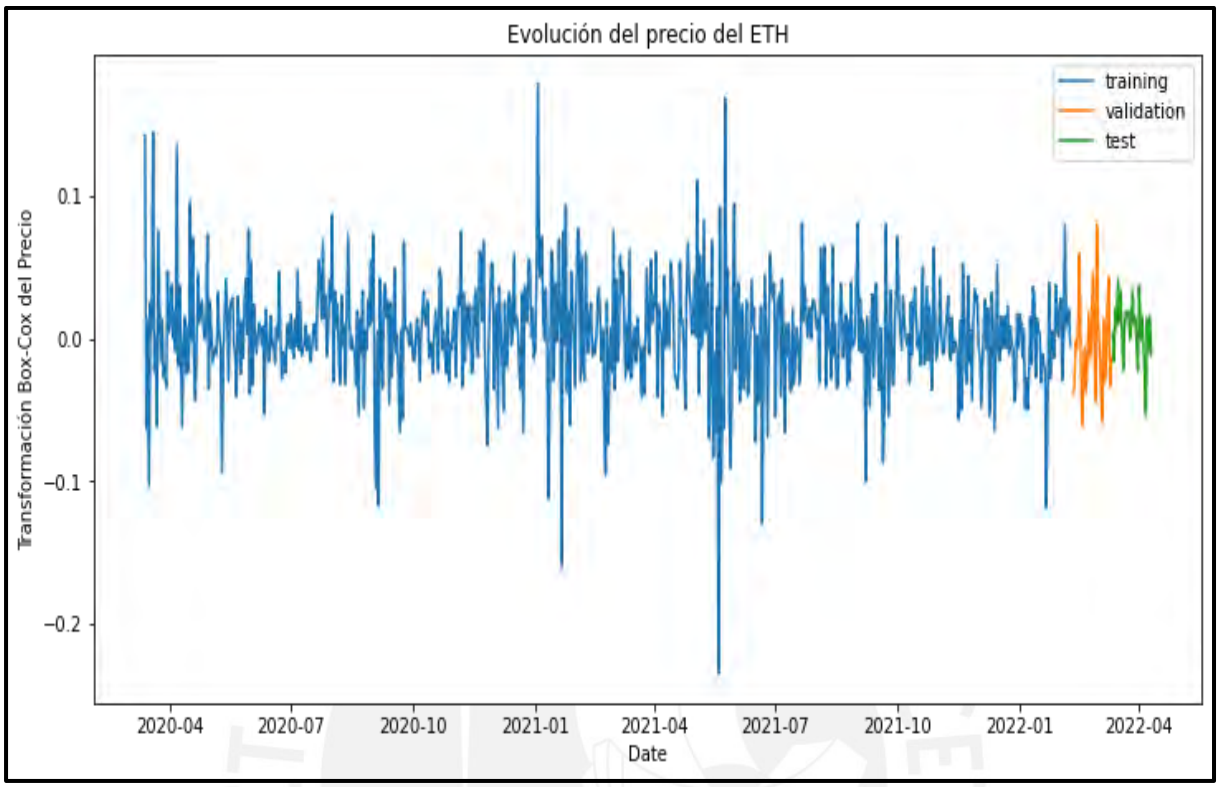


Figura 66: Evolución del precio del ETH con diferenciación de primer orden

Dado que se ha logrado volver a la media constante mediante la diferenciación es requisito indispensable comprobar si la serie de tiempo se ha vuelto estacionaria para ello se utilizará la prueba ADF, cabe remarcar que se obtendría el mismo resultado si se utiliza la prueba KPP como se demostró anteriormente. En la Figura 67, se observa que la prueba arroja un *p-value* igual a 0, ya que este es menor a 0.05 entonces se rechaza la hipótesis nula con lo cual se asegura con un nivel de confianza del 95% que el proceso es estacionario.

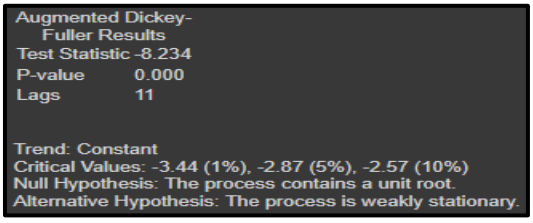


Figura 67: Resultado de prueba ADF con diferenciación de primer orden

Si bien los resultados expuestos indican que el proceso es estable, es decir, la serie presenta estacionariedad débil, existe un inconveniente con los métodos estadísticos tradicionales y es que requieren necesariamente que la serie sea estacionaria. Como se ha visto a lo largo de la etapa del análisis de la serie de tiempo y el preprocesamiento de los datos se han tenido que realizar pasos previos para solucionar la no estacionariedad, por ejemplo, realizar una transformación Cox-Box con el fin de

estabilizar la varianza y posteriormente realizar una diferenciación para poder remover la tendencia. Se ha realizado en ese orden ya que si primero se realizaba la diferenciación existía la posibilidad de que haya valores iguales a cero o menores a uno y cuando se realice la transformación Box-Cox sea necesario agregar una constante *M* para poder trabajarlos. No obstante, al momento de añadir dicha constante crea cierta perturbación de la distribución de los datos lo cual genera efectos adversos en el modelo. Por dicho motivo, se presentará también el pronóstico del precio del ETH mediante el uso de modelos de *machine learning*, los cuales no requieren que la serie sea necesariamente estacionaria para poder realizar pronósticos pues esta carencia puede solucionarse utilizando características adecuadas en el modelo.

Para ello se va a hacer uso del *time delay embedding* y *temporal embedding* con el fin de integrar el tiempo en el modelo. El *time delay embedding* permitirá obtener como regresores 30 *lags* y el *temporal embedding* ayudará al modelo a identificar el paso del tiempo y la periodicidad de este mismo para esto se piensa trabajar con la semana del año, día de la semana, día, mes y año (Ver Figura 68). Esto se ha realizado ya que según Auffarth (2021) las variables de fecha y hora contienen información que puede mejorar significativamente el performance del modelo gracias a la extracción de estas características. Además de ello, se agregará como variable exógena el sentimiento del mercado capturado a través de los titulares de noticias extraídos de la plataforma financiera Investing.

```python
eth_df["Date"] = pd.to_datetime(eth_df.index)
eth_df["Weekofyear"] = eth_df.Date.dt.isocalendar().week.astype("int64")
eth_df["Weekday"] = eth_df.Date.dt.isocalendar().day.astype("int64")
eth_df["Day"] = eth_df.index.day
eth_df["Month"] = eth_df.index.month
n_lags = 31
for day in range(1, n_lags):
    eth_df[f"Lag_{day}"] = eth_df["Price"].shift(day)
```

Python

Figura 68: Código para la extracción de las *features* del tiempo

4.3. Aplicación y validación del modelo

Dado que se ha asegurado la estacionariedad de la serie de tiempo, el paso a seguir debería ser hallar los parámetros del modelo ARIMA. Recordar que el modelo ARIMA presenta 3 parámetros *p*, *d* y *q*. Para poder hallar los parámetros *p* y *q* se hará uso de los gráficos de la función de autocorrelación (ACF) y la función de autocorrelación parcial (PACF), estos también son conocidos como correlogramas. Se comenzará hallando el parámetro *p* para ello se requiere graficar la función de autocorrelación parcial. Posteriormente, se graficará la función de autocorrelación con el fin de hallar el parámetro *q*. Respecto al parámetro *d*, este ya ha sido hallado pues el parámetro *d* representa número de diferenciaciones requeridas para volver la serie estacionaria, en nuestro caso es 1. Con base a la información visualizada en la Figura 70, se observa que la autocorrelación parcial para *lags* mayores a

1 presenta *p-values* cercanos a 0 en el eje de ordenadas lo cual indica que las observaciones no están correlacionadas entre sí. Por su parte, la autocorrelación para todos los *lags* presenta *p-values* cercanos a 1 en el eje de ordenadas lo cual indica que existe una fuerte correlación positiva. Además de ello, se puede visualizar que la autocorrelación es significativa ya que sus valores superan el umbral de color azul, el cual representa el intervalo de confianza. Se escogerá un valor de 1 para el parámetro *p* ya que se observa en la autocorrelación parcial que para este *lag* presenta un correlación alta y significativa. En el caso del parámetro *q* también se escogerá el *lag* 1 ya que si bien para todos los *lags* de 0 a 20 presenta una alta correlación se logra observar que a medida que aumenta el *lag* disminuye la significancia pues se vuelve más cercana al umbral.

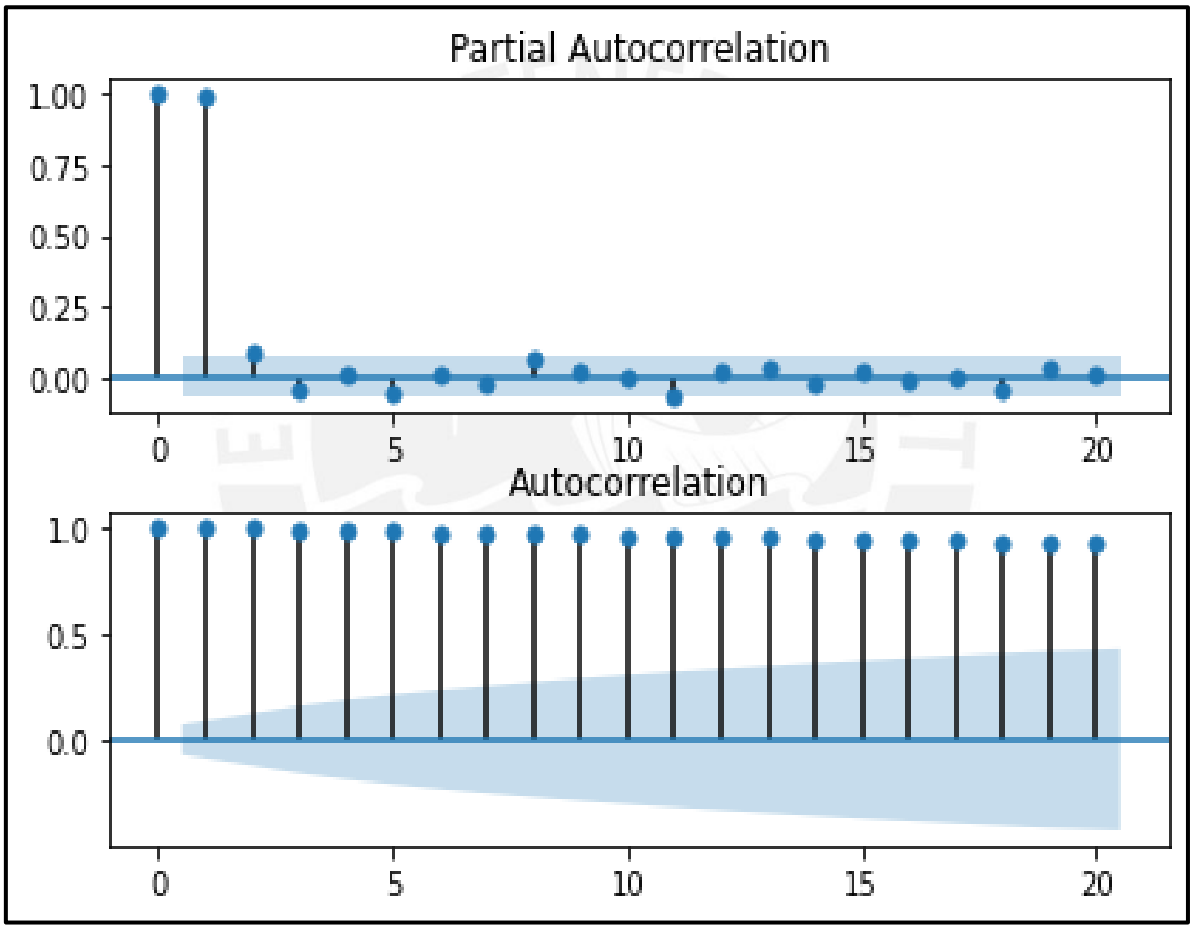


Figura 69: Autocorrelación y Autocorrelación parcial

De acuerdo con el flujo del proceso presentado en la Figura 49, el paso siguiente sería graficar la serie de los residuales junto a la función de autocorrelación y autocorrelación parcial para ello se empleará la librería *tsdiag* de R. En la Figura 70, se observa en la función de autocorrelación que los errores no están correlacionados ya que para todos los *lags* sus *p-values* se encuentran cercanos a 0, ninguno sobrepasa la región delimitada en color azul. Por su parte, el gráfico Ljung-Box nos indica que en teoría hay ruido blanco ya que si bien todos los *lags* se encuentran por encima del *p-value* (0.05) que esta denotado por la línea de color azul, en el lag 10 se observa que se encuentra por debajo de este

valor. No obstante, se utilizará la prueba estadística Ljung-Box para asegurar que efectivamente existe ruido blanco. En la Figura 71 se observa que el *p-value* (0.8528) obtenido es mayor a 0.05, por ende, existe ruido blanco y nuestro modelo se ajusta bien, ya que la media del error es igual a 0 (Ver Figura 72), la varianza es constante y los errores no están correlacionados.

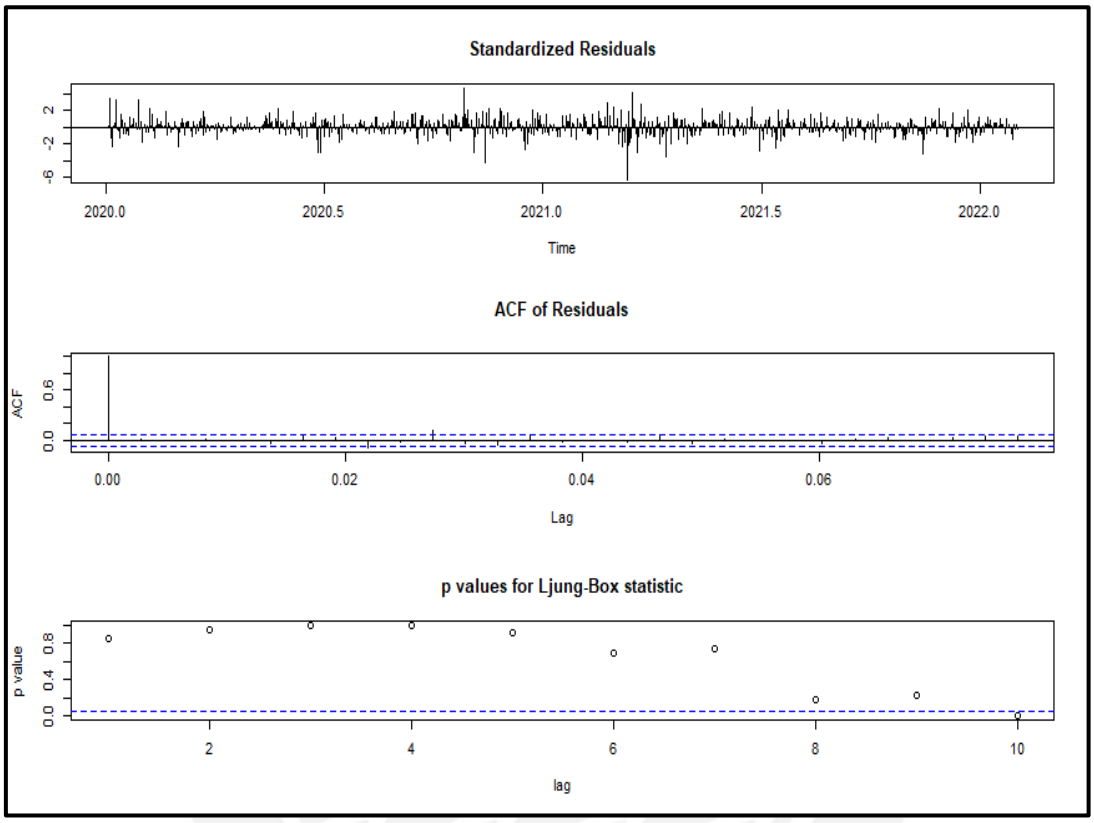


Figura 70: Gráfico de residuales ACF y Ljung-Box

```
        Box-Ljung test

data:  residuals(modelo1)
X-squared = 0.034446, df = 1, p-value = 0.8528
```

Figura 71: Prueba Ljung-Box en residuales

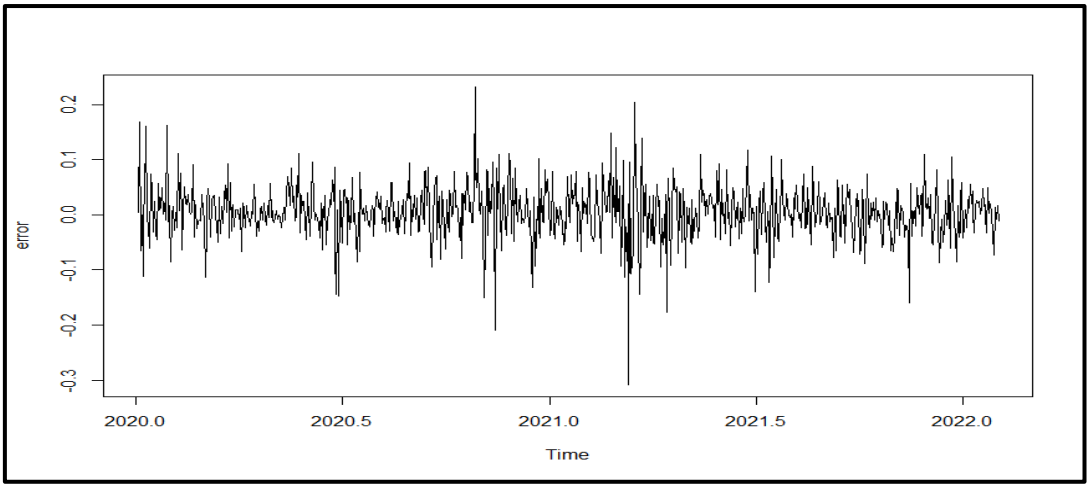


Figura 72: Grafico de serie de residuales

Con base a la información presentada anteriormente, se procede a entrenar el modelo ARIMA (1,1,1) con los datos de entrenamiento (*training set*) para ello se importará la librería *statsmodels* con el fin de poder utilizar la función ARIMA. Los resultados obtenidos se observan en la Figura 73, con base a lo expuesto en el acápite 1.3.5. los modelos de ARIMA se suelen estimar con la estimación de máxima verosimilitud (EMV) y los criterios de selección de modelos más utilizados para evaluar el método EMV son el criterio de información de Akaike (AIC) y el criterio de información Bayesiano (BIC). En la Figura adjunta, se observa que el modelo presenta un AIC de -2137.666 y BIC -2119.560. Sin embargo, no sé conoce con exactitud si con estos parámetros se obtiene el menor AIC y BIC, por lo cual, lo recomendable sería iterar los valores de los parámetros hasta poder alcanzar el mínimo AIC y BIC. No obstante, la librería *pmdarima* contiene una función denominada *stepwise* que nos ayudará a encontrar los valores óptimos de los parámetros *p*, *d* y *q*. El método stepwise es una regresión estadística paso a paso que permite construir un modelo de forma iterativa mediante la selección automática de variables independientes hasta que los resultados sean óptimos (Ver Figura 74).

```
# Construir modelo
import statsmodels.api as sm
from statsmodels.tsa.arima.model import ARIMA
model = ARIMA(train, order=(1, 1, 1))
fitted = model.fit()
print(fitted.summary())
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                 Precio   No. Observations:                  700
Model:                 ARIMA(1, 1, 1)   Log Likelihood                1312.006
Date:                Sun, 12 Mar 2023   AIC                          -2618.012
Time:                        03:40:03   BIC                          -2604.363
Sample:                    05-11-2020   HQIC                         -2612.736
                         - 04-10-2022
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.8423      0.127     -6.640      0.000      -1.091      -0.594
ma.L1          0.7988      0.144      5.550      0.000       0.517       1.081
sigma2         0.0014   4.41e-05     31.085      0.000       0.001       0.001
==============================================================================
```

Figura 73: Modelo ARIMA $\boldsymbol{y_t = -0.0014 - 0.8423y_{t-1} + 0.7988\epsilon_{t-1} + \epsilon_t}$

```
parameters_stepwise = pm.auto_arima(train, start_p=0, start_q=0,
                      test='adf',       # usar prueba ADF para encontrar 'd'
                      max_p=3, max_q=3, # maximo p y q
                      d=None,           #Dejar que el modelo determine 'd'
                      m=0, seasonal=False, #m:#observaciones estacionales
                      D=None,
                      max_Q=3,
                      max_P=3,
                      trace=True,
                      error_action='ignore',
                      suppress_warnings=True,
                      stepwise=True)

print(parameters_stepwise.summary())
```

```
Performing stepwise search to minimize aic
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=-2620.602, Time=0.52 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=-2620.078, Time=0.28 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=-2619.958, Time=1.25 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=-2617.475, Time=0.27 sec
 ARIMA(1,1,1)(0,0,0)[0] intercept   : AIC=-2621.436, Time=3.88 sec
 ARIMA(2,1,1)(0,0,0)[0] intercept   : AIC=-2617.395, Time=0.83 sec
 ARIMA(1,1,2)(0,0,0)[0] intercept   : AIC=-2619.468, Time=2.51 sec
 ARIMA(0,1,2)(0,0,0)[0] intercept   : AIC=-2619.351, Time=0.70 sec
 ARIMA(2,1,0)(0,0,0)[0] intercept   : AIC=-2619.394, Time=0.25 sec
 ARIMA(2,1,2)(0,0,0)[0] intercept   : AIC=-2617.487, Time=2.82 sec
 ARIMA(1,1,1)(0,0,0)[0]             : AIC=-2618.012, Time=0.68 sec

Best model:  ARIMA(1,1,1)(0,0,0)[0] intercept
Total fit time: 14.112 seconds
```

Figura 74: Código del método *stepwise* para hallar parámetros óptimos

Luego de haber hallado los mejores parámetros para el modelo ARIMA, se procede a encontrar el mejor modelo de *machine learning* que aprenda los patrones subyacentes de los datos. Esto se llevará a cabo mediante un comparativo de los modelos presentados en el 1.4.5. Es decir, se empleará un modelo de regresión lineal, regresión ridge, regresión lasso, *Random Forest*, XGBoost y LightGBM. Para poder determinar que modelo se desempeña mejor en los datos de entrenamiento se hará uso de la técnica *grid search*, la cual examina todas las combinaciones posibles de los parámetros declarados por el usuario para encontrar los valores óptimos de los parámetros que generen el mejor modelo. Luego de entrenar los modelos haciendo uso de la técnica *grid search* se obtiene que el modelo que se desempeña mejor es la Regresión Lasso (Ver Figura 76). Por lo tanto, se procederá a realizar el *hyper tuning* con el fin de ajustar los parámetros del modelo para optimizar su rendimiento en el conjunto de datos de prueba. Ello se logrará mediante el uso de la librería Optuna[13], si bien existen diferentes técnicas como *random search*, *grid search*, etc. Optuna hace uso de métodos de búsqueda bayesiana, este tipo de método utiliza una estrategia de selección inteligente de puntos de evaluación que minimiza la cantidad de evaluaciones necesarias para encontrar la solución óptima lo cual hace que este tipo de búsqueda sea más eficiente en recursos tanto computacionales como en el tiempo. El método realizará 10 iteraciones intentando encontrar aquellos hiperparámetros que minimicen el MSE.

[13] https://optuna.org

```
mlr = {'normalizer':['scale','minmax',None]}
xgboost = {'max_depth':[3,4,5],
    'n_estimators':[50,75,100,150],
    'learning_rate':[0.01,0.1],
    'gamma':[0,3,5]}
ridge = {'normalizer':['scale','minmax',None],
    'alpha':np.linspace(0,2,100)}
lasso = {'normalizer':['scale','minmax',None],
    'alpha':np.linspace(0,2,100)}
lightgbm = {'n_estimators':[50,75,100,150],
    'boosting_type':['gbdt','dart','goss'],
    'max_depth':[3,4,5],
    'learning_rate':[0.01,0.1],
    'reg_alpha':np.linspace(0,1,5),
    'reg_lambda':np.linspace(0,1,5),
    'num_leaves':np.arange(5,50,5),}
rf = {'max_depth':[3,4,5],
    'n_estimators':[50,75,100,150],
    'max_samples':[0.7,0.8,0.9]}
```

Figura 75: *Grid* de parámetros para los modelos de *machine learning*

```
ms = f.export('model_summaries',to_excel=True,determine_best_by='TestSetRMSE', excel_name='eth_ml_metrics.xlsx')
ms[['ModelNickname','InSampleRMSE','TestSetRMSE']]
```

✓ 0.2s

	ModelNickname	InSampleRMSE	TestSetRMSE
0	lasso	114.595589	127.644091
1	ridge	111.343099	130.315062
2	mlr	111.343099	130.315062
3	lightgbm	84.239067	153.471613
4	rf	128.941627	160.966094
5	xgboost	13.710053	172.670971

Figura 76: Selección de modelo de *machine learning*

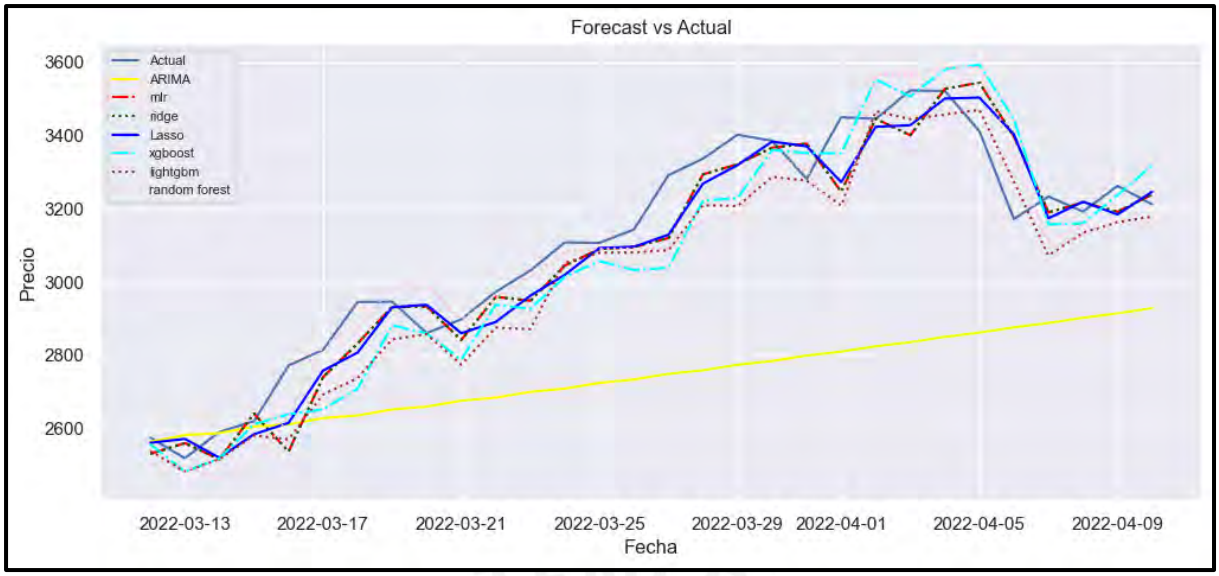


Figura 77: Desempeño del pronóstico de los modelos para 30 días del precio para ETH

Para poder hacer un uso más eficiente de los datos de entrenamiento se dividirá en 4 subconjuntos donde cada subconjunto contendrá datos de entrenamiento y datos de validación, esto siguiendo el método de *cross validation with blocked subsets* presentado en el capítulo 1.3.7. En el anexo 3, se puede observar el *feature importance* de cada subconjunto respectivamente. El gran beneficio de este método es que hace uso de submuestras de la serie de tiempo sin interrumpir su proceso evolutivo. La desventaja como se mencionó anteriormente era que se requería que la serie posea como mínimo 500 observaciones, ya que esta restricción no presenta ningún problema para el presente estudio se procederá a utilizarlo. Este método proporciona 4 métricas de error de MSE correspondientes a cada subdivisión, las cuales se promediarán para poder obtener un capturar un error que contenga todo el proceso evolutivo del modelo a lo largo de la serie de tiempo. Los códigos desarrollados en Python con su respectiva sintaxis con la que se obtuvo los resultados pueden ser visualizados en las Figuras 77,78 y 79.

Siguiendo el flujo definido en la Figura 50, el paso siguiente a realizar será hallar los intervalos de predicción empleando predicción conformal para ello se hará uso de la librería MAPIE[14].

[14] https://mapie.readthedocs.io/en/latest/index.html

```python
from sklearn.model_selection import KFold, TimeSeriesSplit
from sklearn.linear_model import Lasso
from sklearn.metrics import mean_squared_error
import optuna

def objective(trial: optuna.trial.Trial) -> float:
    """
    Dado un conjunto de hyperparámetros, se entrenará el modelo y calculará
    el error promedio de validación hallado en cada TimeSeriesSplit
    """
    #Hyperparámetros
    hyperparams = {
        "alpha": trial.suggest_float(name="alpha", low=0, high=3.0,step=0.001,log=False),
    }

    tss = TimeSeriesSplit(n_splits=4)
    scores = []
    for train_index, val_index in tss.split(X_train):

        # Dividir en datos de entrenamiento y validación
        X_train_, X_val_ = X_train.iloc[train_index, :], X_train.iloc[val_index,:]
        y_train_, y_val_ = y_train.iloc[train_index], y_train.iloc[val_index]

        # Entrenar el modelo
        ml_lasso = Lasso(**hyperparams)
        ml_lasso.fit(X_train_, y_train_)

        # Evaluar el modelo
        y_pred = ml_lasso.predict(X_val_)
        mse = mean_squared_error(y_val_, y_pred)

        scores.append(mse)

    # Retornar el promedio del mse
    return np.array(scores).mean()
```

Figura 78: Código del proceso de optimización de hiperparámetros

```
study = optuna.create_study(direction="minimize")
study.optimize(objective, n_trials=10)
best_params = study.best_trial.params
print(f'{best_params=}')
```

✓ 0.0s

```
best_params={'alpha': 2.466}
```

Figura 79: Hiperparámetros óptimos mediante Optuna

4.4. Análisis de resultados

Previamente se halló que el mejor modelo de ARIMA mediante stepwise se obtiene con los parámetros (1,1,1), por ende, los valores hallados anteriormente asociados al AIC y BIC son los óptimos. En resumen, el modelo se encuentra ajustado correctamente, la ecuación que lo representa esta

denotada por $\boldsymbol{y_t = -0.0014 - 0.8423y_{t-1} + 0.7988\epsilon_{t-1} + \epsilon_t}$. El paso siguiente debería ser pronosticar y analizar los resultados obtenidos del modelo ARIMA (1,1,1) con los datos de prueba (*test set*). En la Figura 80, se visualiza que los valores medios proyectados por el modelo (color verde) se encuentran por debajo del valor real de la criptomoneda, sin embargo, gran parte de los valores reales de la criptomoneda logran ser capturados por el intervalo de confianza de 95% (umbral gris). De la Figura 80, se visualiza que el modelo puede seguir la tendencia a la baja con un MAE de 0.0890 y un RMSE de 0.1005. Por lo tanto, se espera que el modelo de *machine learning* pueda mejorar su precisión incorporando variables exógenas. No obstante, si se visualiza la escala estos valores mencionados han sido hallados mediante la transformación Box-Cox, por ende, es necesario revertir dicha transformación para poder evaluar los resultados en términos reales.

Figura 80: Pronóstico del modelo ARIMA (1,1,1) sin reversión

Para poder llevar a cabo la reversión se importará la librería scipy para hacer uso de la función inv_boxcox. En la Figura 81, se observan los valores reales de la criptomoneda, si bien el intervalo de confianza sigue capturando los valores reales del precio de la criptomoneda en estudio, las métricas de error han aumentado pues el RMSE posee un valor de 413.48 y el MAE de 361.22.

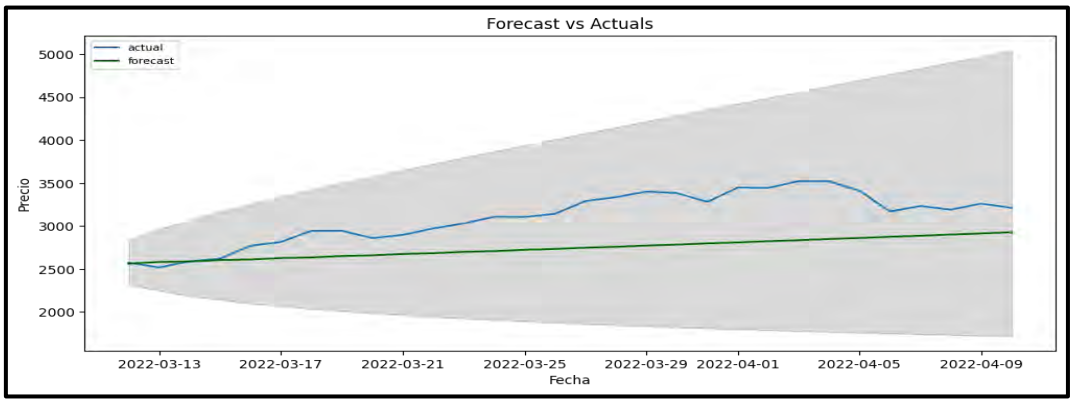


Figura 81: Pronóstico del modelo ARIMA (1,1,1) con reversión

Por otro lado, luego de haber comparado los distintos modelos de *machine learning* se llegó a la conclusión que aquel que tenía mejor desempeño era la regresión de Lasso. Por consiguiente, se realizó el proceso de *hyper tuning* con el fin de hallar los parámetros que optimizaban el desempeño del modelo, ello se daba cuando alpha tomaba el valor de 2.466 con ello se obtuvo una mejora en las métricas de error, pues los valores del RMSE y MAE son 126.03 y 100.59 respectivamente. Si bien la mejora no es significativa pues se redujo de 127.64 a 126.03 en el RMSE, ello indica que los parámetros encontrados mediante el *grid search* estuvieron cerca del valor óptimo, es decir, se encontraron con un mínimo local mas no con el mínimo global.

Por último, como se mencionó en el acápite 1.5. se requería hallar el modelo base óptimo para posteriormente poder aplicar *conformal prediction* con un $\alpha = 0.05$. En la Figura 82, se observa los intervalos del precio del ETH; así como el precio real y el pronosticado. Asimismo, se adjunta la cobertura (o *coverage*) que indica que el modelo presenta un 96.7% de probabilidad de contener al valor real y una anchura (o *width*) de 504.779. Ambos son indicadores de la precisión y confianza del modelo como se puede observar la anchura o tamaño del intervalo de predicción es demasiado ancho lo cual puede indicar que el modelo tiene menos confianza en su predicción se sugiere que la anchura sea más estrecha pero esta debe ser balanceada con la cobertura pues una anchura demasiado estrecha tenderá a tener menos probabilidad de contener al valor real.

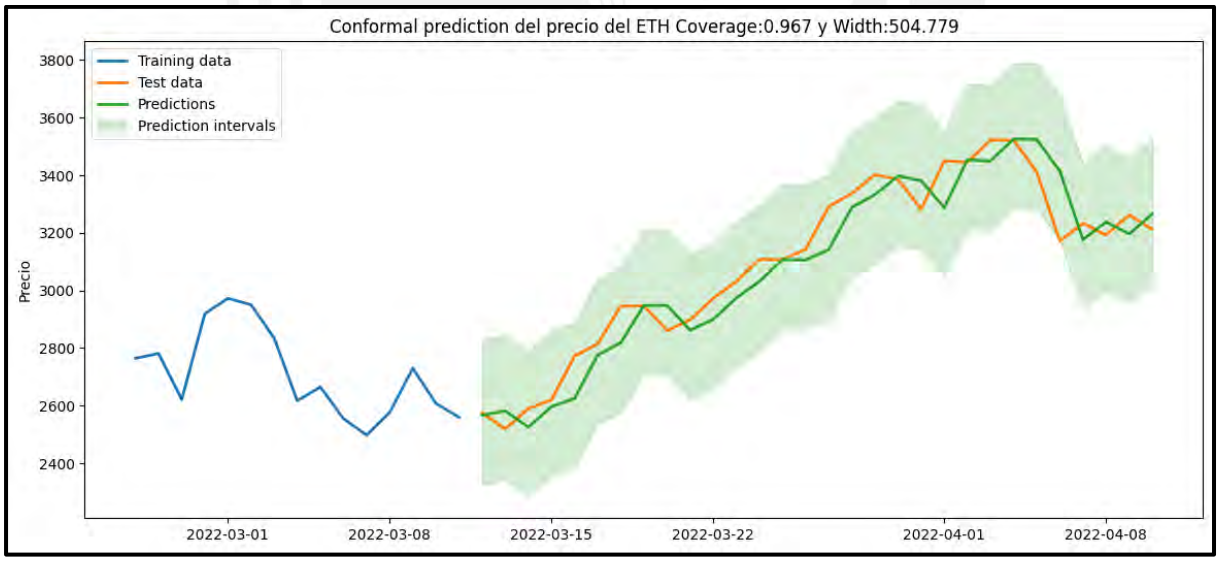


Figura 82: Intervalo del precio del ETH

Tabla 21 Resumen de métricas de error de los modelos

Modelo	RMSE(Training)	RMSE(Test)	MAE(Training)	MAE(test)
ARIMA(1,1,1)	113.04	413.48	67.62	361.22
Lasso	115.18	126.03	72.75	100.59
Ridge	111.34	130.31	69.05	101.43
Regresión Múltiple	111.34	130.31	69.05	101.43
LightGBM	84.23	153.47	55.82	123.76
Random Forest	128.94	160.96	91.33	136.73
XGBoost	13.71	172.67	9.60	144.68

CAPÍTULO 5. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- Se concluye que la predicción conformal resulta útil en industrias donde sea crucial contar con predicciones fiables tales como finanzas, salud, consumo masivo, transporte y farmacéutica. Ya que a través de rangos estadísticamente válidos de la oferta y la demanda se lograría reducir la demanda insatisfecha, los cuales estarían asegurados por la cobertura garantizada que proporciona el *framework*. Asimismo, el uso de variables exógenas como el sentimiento del mercado dependerá en gran medida del tipo de industria y datos de la serie de tiempo.
- Se concluye que a nivel país no existe aún inversión o desarrollo de aplicaciones en la tecnología *blockchain* por parte de organizaciones gubernamentales lo cual en cierta medida es desventajoso, ya que actualmente existe una diversidad de aplicaciones que podrían ser aplicadas en los sectores de salud, educación, etc. Un claro ejemplo de esto serían los *real world assets*.
- Se concluye que el modelo Lasso que integra como features el día del año, mes, año, día de la semana, semana del año, lags y el sentimiento de los titulares de noticias proporciona el menor RMSE (126.03) Y MAE (100.59) respecto a los demás modelos incluyendo el ARIMA (1,1,1) cuyas métricas de error RMSE y MAE asociadas son 413.48 y 361.22 respectivamente.
- La fuente de información de donde se han extraído los titulares de noticias relacionadas a criptomonedas corresponde a la plataforma financiera y sitio web de noticias Investing, el cual es uno de los tres sitios web financieros más visitados a nivel mundial. Este hecho reduce el riesgo de contar con *fake news*, ya que este tipo de noticias tiene como fin causa zozobra y desinformación lo cual impactaría en los resultados obtenidos del análisis de sentimiento y estos a su vez perjudicarían el performance del modelo ya que los datos de salida del algoritmo VADER (compound de titulares de noticias) son los datos de entrada del modelo Lasso.

5.2. Recomendaciones

- Se recomienda crear un diccionario léxico específico para las criptomonedas que contenga los términos relacionados a este tipo de tecnología *blockchain*, esto podría mejorar la relación entre el índice de sentimiento del mercado frente a la variación del precio de las criptomonedas.
- Se recomienda considerar para futuros proyectos el análisis de las declaraciones o anuncios de fuentes gubernamentales como el Sistema de Reserva Federal (FED) o el Comité Federal de Mercado Abierto (FOMC) y asignar un mayor peso a este tipo de noticias, ya que las declaraciones provenientes de estos organismos impactan en cierta medida sobre el mercado de valores y de las criptomonedas.

- Se recomienda explorar la correlación y grado de significancia del volumen de transacciones en la red de Ethereum, ya que en cada transacción se está haciendo una quema de ETH. En ese sentido, a mayor quema menor circulación de ETH, por ende, el precio tendería a aumentar por la oferta y la demanda. Sin embargo, se requeriría construir un modelo que pronostique el volumen de transacciones en la red de Ethereum y que este a su vez sirva como insumo para el modelo del pronóstico del precio.
- Se recomienda emplear para futuros estudios de Altcoins, el análisis en tiempo real de los videos subidos en la red social TikTok, extraer los hashtags de cada video para poder determinar que Altcoin está recibiendo mayor FOMO por parte de los usuarios.
- Se recomienda explorar los nuevos modelos globales para el pronosticó de series de tiempo.

BIBLIOGRAFIA

Angelopoulos, A. N., & Bates, S. (2022). A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification. ArXiv:2107.07511 [Cs, Math, Stat]. https://arxiv.org/abs/2107.07511

Auffarth, B. (2021). *Machine Learning for Time-Series with Python: Forecast, predict, and detect anomalies with state-of-the-art machine learning methods*. Packt.

Avan-Nomayo, O. (2021, 18 de febrero). Binance Coin alcanza un nuevo máximo histórico en medio de una actividad explosiva en BSC. *Cointelegraph*. https://es.cointelegraph.com/news/binance-coin-sets-new-all-time-high-amid-skyrocketing-activity-on-bsc

Arlot, S., & Celisse, A. (2010). A survey of cross-validation procedures for model selection. Statistics Surveys, 4(0), 40–79. https://doi.org/10.1214/09-ss054

Beekhuizen, C. (2021, 18 mayo). *Ethereum's energy usage will soon decrease by ~99.95%*. Ethereum Foundation Blog. https://blog.ethereum.org/2021/05/18/country-power-no-more/

Bloomberg News (2021, 24 de septiembre). China Widens Ban on Crypto Transactions; Bitcoin Tumbles. https://www.bloomberg.com/news/articles/2021-09-24/china-deems-all-crypto-related-transactions-illegal-in-crackdown

Beri, Aditya (2020, 27 de mayo). Sentimental Analysis using VADER: interpretation and classification of emotions. Towards Data Science. https://towardsdatascience.com/sentimental-analysis-using-vader-a3415fef7664

Bergmeir, C., & Benítez, J. M. (2012). On the use of cross-validation for time series predictor evaluation. Information Sciences, 191, 192–213. https://doi.org/10.1016/j.ins.2011.12.028

Bird, S., Edward, L. & Ewan, K. (2009), *Natural Language Processing with Python*. O'Reilly Media Inc.

Baker, M., & Wurgler, J. (2007). Investor Sentiment in the Stock Market. *Journal of Economic Perspectives*, 21 (2): 129-152. https://doi.org/10.1257/jep.21.2.129

Banko, M., & Brill, E. (2001). Scaling to very very large corpora for natural language disambiguation (pp. 26–33). Association for Computational Linguistics (ACL). https://doi.org/10.3115/1073012.1073017

Bashir, I. (2017). *Mastering Blockchain: Distributed ledgers, decentralization and smart contracts explained*. Packt. https://www.packtpub.com/product/mastering-blockchain/9781787125445

Bheemaiah, K. (2017). Debt-based Economy: The Intricate Dance of Money and Debt. In *The Blockchain Alternative*. Apress. https://doi.org/10.1007/978-1-4842-2674-2_1

Chaum D. (1983) Blind Signatures for Untraceable Payments. In: Chaum D., Rivest R.L., Sherman A.T. (eds) Advances in Cryptology. Springer, Boston, MA. https://doi.org/10.1007/978-1-4757-0602-4_18

de Best, R. (2022, 6 mayo). *Average daily gas price of Ethereum from August 2015 to May 5, 2022*. Statista. https://www.statista.com/statistics/1221821/gas-price-ethereum/

Drescher, D. (2017). *Blockchain Basics: A Non-Technical Introduction in 25 Steps*. Apress Media LLC. https://doi.org/10.1007/978-1-4842-2604-9

Domingos, P. (2012, October). A few useful things to know about machine learning. *Communications of the ACM*. https://doi.org/10.1145/2347736.2347755

Dwork C., Naor M. (1993) Pricing via Processing or Combatting Junk Mail. In: Brickell E.F. (eds) Advances in Cryptology — CRYPTO' 92. CRYPTO 1992. Lecture Notes in Computer Science, vol 740. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-48071-4_10

Ethereum Energy Consumption Index. (2021, 30 diciembre). Digiconomist. https://digiconomist.net/ethereum-energy-consumption

Engle, R. F., & Granger, C. W. J. (1987). Co-Integration and Error Correction: Representation, Estimation, and Testing. *Econometrica*, *55*(2), 251–276. https://doi.org/10.2307/1913236

França. L. (2021,31 de agosto). ¿Es Binance Smart Chain el mayor rival de Ethereum?. C*ointelegraph*.https://es.cointelegraph.com/news/in-a-year-ethereums-biggest-rival-meet-binance-smart-chains-blockchain-ecosystem

Finneseth, J. (2021, 10 de febrero). 3 razones por las que Binance Coin (BNB) alcanzó un nuevo máximo histórico de USD 148. *Cointelegraph*. https://es.cointelegraph.com/news/3-reasons-why-binance-coin-bnb-hit-a-new-all-time-high-at-148

Gestión (2021, 01 de enero). Blackrock abre la puerta al bitcoin. https://gestion.pe/economia/mercados/blackrock-abre-la-puerta-al-bitcoin-noticia/?ref=gesr

Géron, A. (2019). *Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Inc. https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/

Gneiting, T., & Katzfuss, M. (2014). Probabilistic Forecasting. Annual Review of Statistics and Its Application, 1, 125–151. https://doi.org/10.1146/annurev-statistics-062713-085831

George Athanasopoulos, D. S. Poskitt & Farshid Vahid (2012) Two Canonical VARMA Forms: Scalar Component Models Vis-à-Vis the Echelon Form, Econometric Reviews, 31:1, 60-83, DOI: 10.1080/07474938.2011.607088

Guerrero, Victor M. (1993), Time-series analysis supported by power transformations. Journal of Forecasting, Volume 12, Issue 1, 37-48. https://onlinelibrary.wiley.com/doi/10.1002/for.3980120104.

Hewamalage, H. (2022, 21 marzo). *Forecast Evaluation for Data Scientists: Common Pitfalls and Best Practices*. arXiv.Org. https://arxiv.org/abs/2203.10716

Hyndman, R.J., & Athanasopoulos, G. (2018). *Forecasting: principles and practice*, 2nd edition, OTexts: Melbourne, Australia. https://otexts.com/fpp2. Accesed on 16 October 2021.

Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.

Ibrahim, A. (2021). Forecasting the early market movement in bitcoin using twitter's sentiment analysis: An ensemble-based prediction model. In *2021 IEEE International IOT, Electronics and Mechatronics Conference, IEMTRONICS 2021 - Proceedings*. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/IEMTRONICS52119.2021.9422647

Joseph, M. (2022). *Modern Time Series Forecasting with Python*. Packt. https://www.packtpub.com/product/modern-time-series-forecasting-with-python/

Keith, M. (2022, 16 junio). Model Validation Techniques for Time Series. Towards Data Science.https://towardsdatascience.com/model-validation-techniques-for-time-series-3518269bd5b3

Kraaijeveld, O., & De Smedt, J. (2020). The predictive power of public Twitter sentiment for forecasting cryptocurrency prices. *Journal of International Financial Markets, Institutions and Money*, *65*. https://doi.org/10.1016/j.intfin.2020.101188

LeewayHertz (2022, 13 abril). *Comparison of blockchain protocols*. LeewayHertz - Software Development Company. https://www.leewayhertz.com/comparison-of-blockchain-protocols/

Liu, B. (2015). *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*. Cambridge University Press. https://www.cambridge.org/pe/academic/subjects/computer-science/knowledge-management-databases-and-data-mining/sentiment-analysis-mining-opinions-sentiments-and-emotions?format=AR

Manokhin, V. (2022). Machine Learning for Probabilistic Prediction. *Royal Holloway University of London*. https://pure.royalholloway.ac.uk/en/publications/machine-learning-for-probabilistic-prediction

Molnar, C. (2022, 20 de diciembre). Week #1: Getting Started With Conformal Prediction For Classification. *Mindful Modeler*. https://mindfulmodeler.substack.com/p/week-1-getting-started-with-conformal

McShane, A. (2021, 30 de septiembre). *Federal Reserve Chair Jerome Powell: U.S. Has No Plans to Ban Bitcoin and Crypto*. Nasdaq. https://www.nasdaq.com/articles/federal-reserve-chair-jerome-powell%3A-u.s.-has-no-plans-to-ban-bitcoin-and-crypto-2021-09

Molano, N. (2019). Claves para entender la tecnología 'blockchain'. *BBVA*. https://www.bbva.com/es/claves-para-entender-la-tecnologia-blockchain/

Morabito, V. (2017). *Business Innovation Through blockchain: The B3 perspective*. Springer International Publishing. https://doi.org/10.1007/978-3-319-48478-5

Mougayar, W. (2016). *The Business Blockchain*. John Wiley & Sons Ltd. https://www.wiley.com/en-us/The+Business+Blockchain%3A+Promise%2C+Practice%2C+and+Application+of+the+Next+Internet+Technology-p-9781119300335

Montesinos García, L. (2014). Análisis de sentimientos y predicción de eventos en twitter. Disponible en http://repositorio.uchile.cl/handle/2250/130479

Nakamoto, S. (2013). Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf

Pang, Y., Sundararaj, G., & Ren, J. (2019). Cryptocurrency Price Prediction using Time Series and Social Sentiment Data. *Proceedings of the 6th IEEE/ACM International Conference on Big Data Computing, Applications and Technologies - BDCAT '19*. https://doi.org/10.1145/3365109.3368785

Pozzi, F. A., Fersini, E., Messina, E., & Liu, B. (2016). *Sentiment Analysis in Social Networks*. *Sentiment Analysis in Social Networks*. Elsevier Inc. https://doi.org/10.1016/B978-0-12-804412-4.00001-2

Quesada, M. (2021). *Análisis de Series Temporales. Modelos Heterocedásticos*. [Trabajo fin de Master, Universidad de Granada]. Repositorio Institucional (Digibug). https://masteres.ugr.es/moea/pages/tfm1011/analisisdeseriesmodelosheterocedasticos

Quirós, F. (2021, 11 de febrero). ¿Por qué aumentó el valor de BNB?. *Cointelegraph*. https://es.cointelegraph.com/news/why-did-the-value-of-bnb-increase

Reuters (2021, 22 de julio). JPMorgan to give all wealth clients access to crypto funds. https://www.reuters.com/business/finance/jpmorgan-give-all-wealth-clients-access-crypto-funds-business-insider-2021-07-22/

Refaeilzadeh, P., Tang, L., Liu, H. (2009). Cross-Validation. In: LIU, L., ÖZSU, M.T. (eds) Encyclopedia of Database Systems. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-39940-9_565

Racine, J. (2000). Consistent cross-validatory model-selection for dependent data: hv-block cross-validation. *Journal of Econometrics*, *99*(1), 39–61. https://doi.org/10.1016/s0304-4076(00)00030-0

Sattarov, O., Jeon, H. S., Oh, R., & Lee, J. D. (2020). Forecasting bitcoin price fluctuation by twitter sentiment analysis. In *2020 International Conference on Information Science and Communications Technologies, ICISCT 2020*. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ICISCT50599.2020.9351527

Steinert, L., & Herff, C. (2018). Predicting altcoin returns using social media. *PLoS ONE*, *13*(12). https://doi.org/10.1371/journal.pone.0208119

Shumway, Robert H., Stoffer, David S. (2017). *Time Series Analysis and Its Applications: With R Examples*. Springer. https://doi.org/10.1007/978-3-319-52452-8

Swan, M. (2015). *Blockchain for a New Economy.* O'Reilly Media, Inc. https://www.oreilly.com/library/view/blockchain/9781491920480/

Shultz, T. R., Fahlman, S. E., Craw, S., Andritsos, P., Tsaparas, P., Silva, R., Drummond, C., Ling, C. X., Sheng, V. S., Drummond, C., Lanzi, P. L., Gama, J., Wiegand, R. P., Sen, P., Namata, G., Bilgic, M., Getoor, L., He, J., Jain, S., & Stephan, F. (2011). Cross-Validation. Encyclopedia of Machine Learning, 249–249. https://doi.org/10.1007/978-0-387-30164-8_190

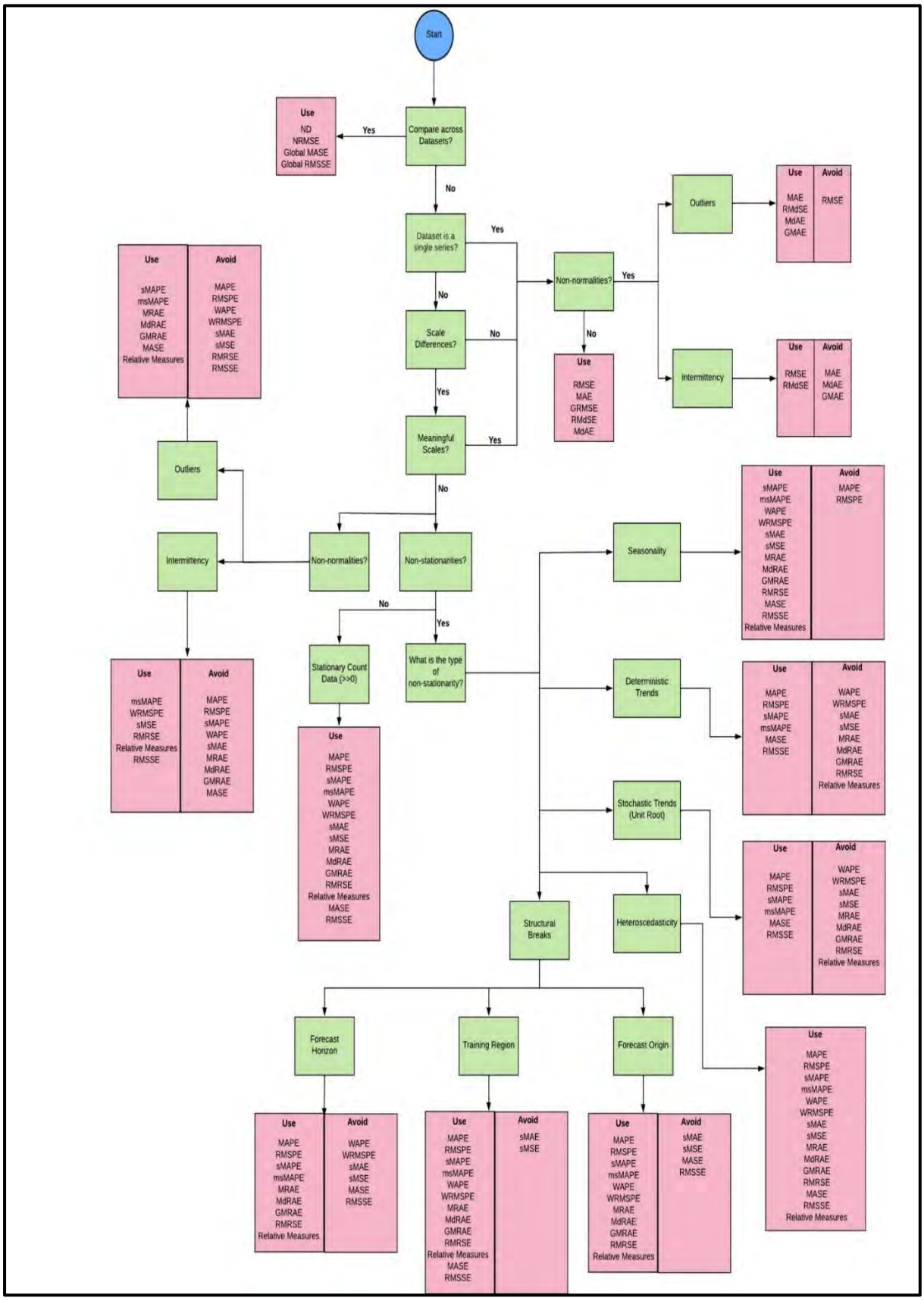
Team Ripple (2017). The Internet of Value: What It Means and How It Benefits Everyone. Ripple.https://ripple.com/insights/the-internet-of-value-what-it-means-and-how-it-benefits-everyone/

Tapscott, D. & Tapscott, A. (2016). *Blockchain Revolution: How the Technology Behind Bitcoin is Changing Money, Business, and the World.* Penguin Random House LLC. https://www.penguinrandomhouse.com/books/531126/blockchain-revolution-by-don-tapscott-and-alex-tapscott/

Vandeput, N. (2021). 7 Double Smoothing with Damped Trend. In *Data Science for Supply Chain Forecasting* (pp. 60-65). Berlin, Boston: De Gruyter. https://doi.org/10.1515/9783110671124-007

Zhang, X., Fuehres, H., & Gloor, P. A. (2011). Predicting Stock Market Indicators Through Twitter “I hope it is not as bad as I fear.” *Procedia - Social and Behavioral Sciences*, *26*, 55–62. https://doi.org/10.1016/j.sbspro.2011.10.562

Anexo 1. Diagrama de flujo para la selección de la métrica de error



Anexo 2. Código del preprocesamiento de los titulares de noticias.

```
#Instalar librerías gensim y Vader
!pip install gensim, vaderSentiment
#Cargar librerías
import re
from gensim.parsing.preprocessing import remove_stopwords
import json
from textblob import TextBlob
from datetime import date
import pandas as pd
import string
from datetime import timedelta
import datetime
plt.style.use("fivethirtyeight")
pd.set_option('max_colwidth', 900)
#Definir funciones necesarias para la limpiezas de los datos

def CleanTxt(text):
  text = re.sub(r"$","", text) #Remover $
  text = re.sub(r"£","", text) #Remover $
  text = re.sub(r"$[A-Za-z0-9_]+","", text) #Remover @menciones
  text = re.sub(r"£[A-Za-z0-9_]+","", text)
  text = re.sub(r'\d', "", text) #Remover números
  text = re.sub(r"RT[\s]+","", text) #Remover RTweets
  #text = re.sub(r'\d.', "", text) #Remover números que están acompañados de letras
  text = re.sub(r"#","", text) #Remover #
  text = re.sub(r"https?:\/\/\S+","", text) #Remover URL
  text = re.sub(r"RT","", text) #Remover RTweets
  return text

def Remove_SW(text):
  text = remove_stopwords(text)
  return text

def Transformar_Minuscula(text):
  text = text.casefold()
  return text

def remove_punctuation(text):
  table = str.maketrans("","",string.punctuation)
  return text.translate(table)

df_News["Statement"] = df_News["Statement"].apply(CleanTxt)
df_News["Statement"] = df_News["Statement"].apply(remove_punctuation)
df_News["Statement"] = df_News["Statement"].apply(Transformar_Minuscula)
df_News["Statement"] = df_News["Statement"].apply(Remove_SW)
```

```
#Importar módulo de la librería vaderSentiment para realizar el análisis de sentimientos
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
analyzer = SentimentIntensityAnalyzer()

scores = [] #Crear una lista en blanco donde se almacenarán las variables de salida.
# Declare variables for scores
compound_list = [] #Crear una lista en blanco donde se almacenarán los scores del compound
positive_list = [] #Crear una lista en blanco donde se almacenarán los valence score positivos
negative_list = [] #Crear una lista en blanco donde se almacenarán los valence score negativos
neutral_list = [] #Crear una lista en blanco donde se almacenarán los valence score neutrales

for i in range(df_News["Statement"].shape[0]):
    compound = analyzer.polarity_scores(df_News["Statement"][i])["compound"]
    pos = analyzer.polarity_scores(df_News["Statement"][i])["pos"]
    neu = analyzer.polarity_scores(df_News["Statement"][i])["neu"]
    neg = analyzer.polarity_scores(df_News["Statement"][i])["neg"]

    scores.append({"Compound": compound, # Unir los score positivos, negativos y neutrales.
                   "Positivo": pos,
                   "Negativo": neg,
                   "Neutral": neu
                })
# Convertir el diccionario en un dataframe
sentiments_score = pd.DataFrame.from_dict(scores)
# Unir el dataframe que contiene los titulares de noticias con el dataframe que contiene los scores
hallados de las variables mediante el index.
df_News = df_News.join(sentiments_score)
```

Anexo 3. Importancia de las características en el modelo Lasso.