

Universidad de Concepción

FACULTAD DE INGENIERÍA

AVANCE FINAL: PROYECTO SEMESTRAL

*Inteligencia Artificial*

Martín Molina  
Pamela Pinilla  
Carlos Venegas

Julio 2021

# Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Estudio Bibliográfico</b>	<b>3</b>
<b>3. Propuesta</b>	<b>5</b>
<b>4. Implementaciones</b>	<b>6</b>
4.1. Aplicación del método de regresión lineal . . . . .	6
4.2. Implementación ARIMA . . . . .	9
4.2.1. Pre-procesamiento de los datos . . . . .	9
4.2.2. Evaluar el modelo . . . . .	10
<b>5. Resultados y Análisis</b>	<b>11</b>
5.1. Pre-procesamiento de los datos . . . . .	11
5.1.1. Datos Mensuales . . . . .	12
5.1.2. Datos Semanales . . . . .	13
5.1.3. Datos Diarios . . . . .	14
5.2. Ajuste a mejor modelo . . . . .	15
5.2.1. Datos Mensuales . . . . .	15
5.2.2. Datos Semanales . . . . .	16
5.2.3. Datos Diarios . . . . .	17
5.3. Resultados de predicción . . . . .	18
5.3.1. Datos Mensuales . . . . .	19
5.3.2. Datos Semanales . . . . .	19
5.3.3. Datos Diarios . . . . .	20
<b>6. Dificultades</b>	<b>21</b>
<b>7. Conclusiones</b>	<b>22</b>

## 1. Introducción

El proyecto se centrará en la predicción del precio/dirección del Bitcoin, una criptomoneda descentralizada, cuyas transacciones no están ligadas a ningún país (no existe una autoridad de control como un banco que sea responsable de su emisión y registro de transacciones) y, a diferencia del mercado del stock, no se ve afectada por intervenciones de gobiernos o cualquier empresa.

Considerando lo anterior y el hecho de que su lanzamiento es relativamente nuevo (2009), se han desarrollado distintos métodos de inteligencia artificial para predecir su precio para así invertir correctamente en el futuro.

## 2. Estudio Bibliográfico

Dado la novedad del Bitcoin, no existen muchos modelos de predicción del precio actualmente, Shah y Zhang, 2014 utilizan regresión Bayesiana usando series de tiempo, en datos con intervalos de tiempo de 10 segundos a lo largo de un par de meses. Dado este vector de series de tiempo, generan 3 subsets de data en intervalos de 30, 60 y 120 minutos en el cual aplican GLM/Random Forest para generar 3 modelos lineales.

D. Kwon y Han, 2019 utilizan una red neuronal recursiva con el modelo LSTM (long short-term memory) para clasificar la dirección de series de tiempo del precio de distintas cripto-monedas(precio sube o precio baja), recolectando datos en intervalos de 10 minutos. Similar al artículo citado, hemos encontrado repositorios de proyectos similares utilizando redes neuronales recurrentes con el modelo LSTM:

- <https://github.com/sudharsan13296/Bitcoin-price-Prediction-using-LSTM>
- <https://github.com/rahulworld/bitcoin-prediction> (implementa también métodos de ML como SVM y LR)

Existen también otros métodos documentados para predecir el precio del Bitcoin con algoritmos de Machine Learning, Karasu y col., 2018, utiliza series de tiempo de los datos del precio de Bitcoin entre los años 2012-2018 y aplica 2 métodos de Machine Learning: Lineal Regression(LR) y Support Vector Machine (SVM).

Un artículo reciente también compara distintos métodos de predicción de dirección del precio, Ibrahim, Kashef y Corrigan, 2021 comparan estrategias del estado del arte en predecir la dirección del precio del Bitcoin, entre ellas prueban modelos como:

- Autoregressive Integrated Moving Average (ARIMA)
- Prophet (por Facebook)
- Random Forest
- Random Forest Lagged-Auto-Regression
- Multi-Layer Perceptron(MLP) Neural Networks

Pastor Cobo y Torre Barrio, 2019 en su trabajo de fin de grado, elaboran un algoritmo predictivo, que estudia si el precio del Bitcoin va a subir o bajar. Esto lo hace a través de dos métodos de redes neuronales: LSTM(Long Short-Term Memory) y GRU(Gated Recurrent Unit) y obtienen una respuesta binaria al problema de la predicción y además una no binaria (con tres niveles de respuesta).

Azari, 2019 realiza una predicción del Bitcoin utilizando ARIMA, donde se recopilieron datos del precio del Bitcoin desde el 2013 hasta el 2019. Para así predecir el precio de los primeros 7 días de enero del 2020.

Al igual que Azari, Ayaz y col., 2020 realizan una predicción del Bitcoin utilizando ARIMA, pero con un data set de 3 años y utilizando sub-períodos para disminuir el error de predicción.

### 3. Propuesta

A partir de lo investigado, hemos decidido enfocarnos en la resolución del problema de clasificación binaria del precio del Bitcoin, a través de distintos métodos de Machine Learning y comparar sus resultados. La predicción de la dirección del precio es un problema que sólo requiere datos históricos del valor diario del Bitcoin y por ende su modelo y complejidad para implementar no es muy grande. Se tomó esta decisión ya que los recursos computacionales que se poseen no son del nivel que se requiere para realizar una investigación más compleja.

La idea principal es a partir de nuestro estudio bibliográfico, reproducir algunos de estos métodos y comparar resultados, si nuestra implementación logra predecir la dirección del precio del Bitcoin y con qué precisión, a modo de aprendizaje, empezaremos implementando uno de los métodos más simples: Regresión Lineal (LR).

## 4. Implementaciones

### 4.1. Aplicación del método de regresión lineal

El Método de Regresión Lineal es un modelo matemático usado para aproximar la relación de dependencia entre una variable independiente,  $m$  variables dependientes y un término aleatorio epsilon.

Para comenzar, se implementa una Regresión lineal a un dataset de <https://www.kaggle.com/mczielinski/bitcoin-historical-data>, el cual consiste en una recopilación del precio del bitcoin en intervalos de 1 minuto, desde el 2012 hasta marzo del 2021.

El procesamiento del dataset se hizo a través de python( a través de jupyter notebook), donde a través del paquete pandas de python, se cargo el dataset y se paso a fechas de año-mes-dia:

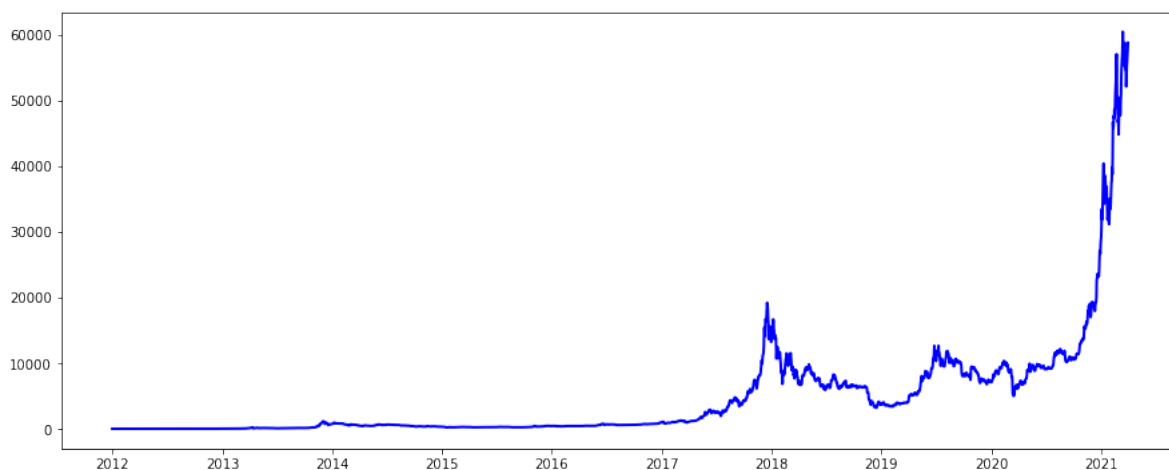


Figura 1: Gráfico del precio del bitcoin en dólares desde el año 2012 al 2021.

El primer paso a realizar fue establecer el forecast que se quería hacer, en este caso fue una predicción de 30 días y utilizando el paquete de python de sklearn, se procede a estandarizar el set de datos con `preprocessing.scale()`, lo cual centra el precio del bitcoin alrededor de su media y la escala por la varianza de la muestra.

Luego de estandarizar el dataset y establecer el tiempo de forecast, utilizando `train_test_split` de sklearn se separa el set de datos en subsets aleatorios de proporciones iguales para entrenar y validar el método de regresión lineal.

A través de sklearn, utilizando una `LinealRegression()`, se ajustan los subsets de entrenamiento y se predicen con el conjunto de datos de validación, lo cual entrega un nivel de confianza del 96 %.

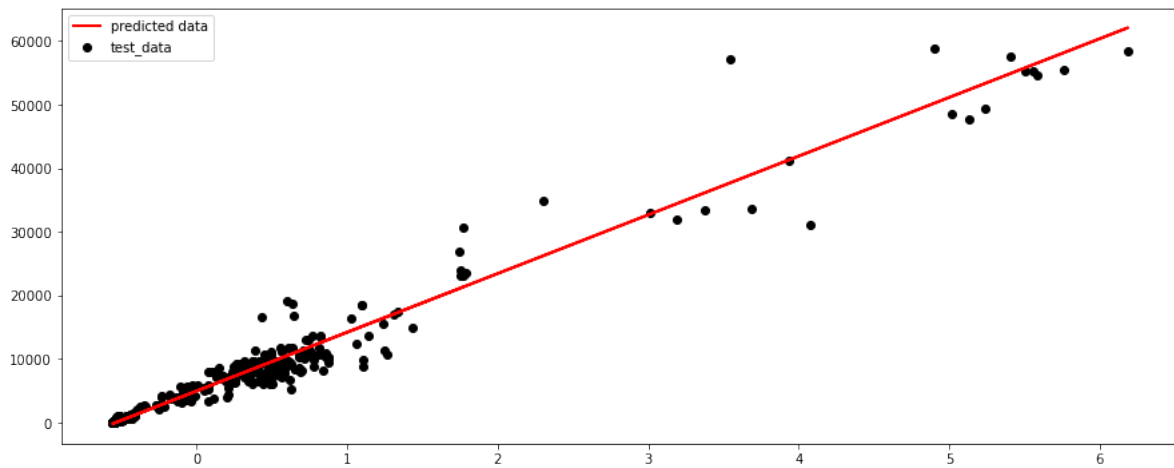


Figura 2: Gráfico comparativo entre el conjunto de datos de validación y la predicción lineal de los datos.

Resulta un modelo creado con un error de mínimos cuadrados de 1656, lo cual dice que este modelo no ajusta muy bien los datos.

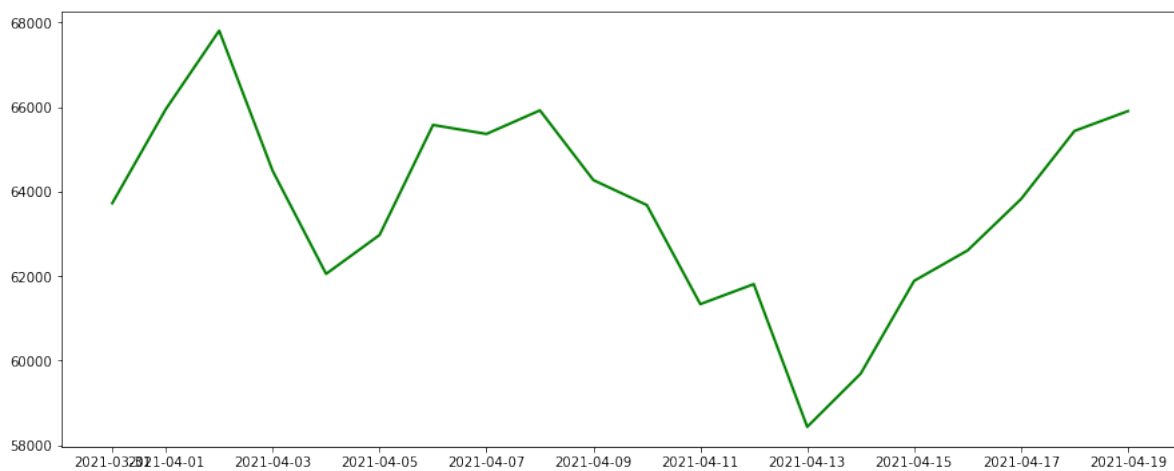


Figura 3: Predicción del precio del bitcoin de 30 días



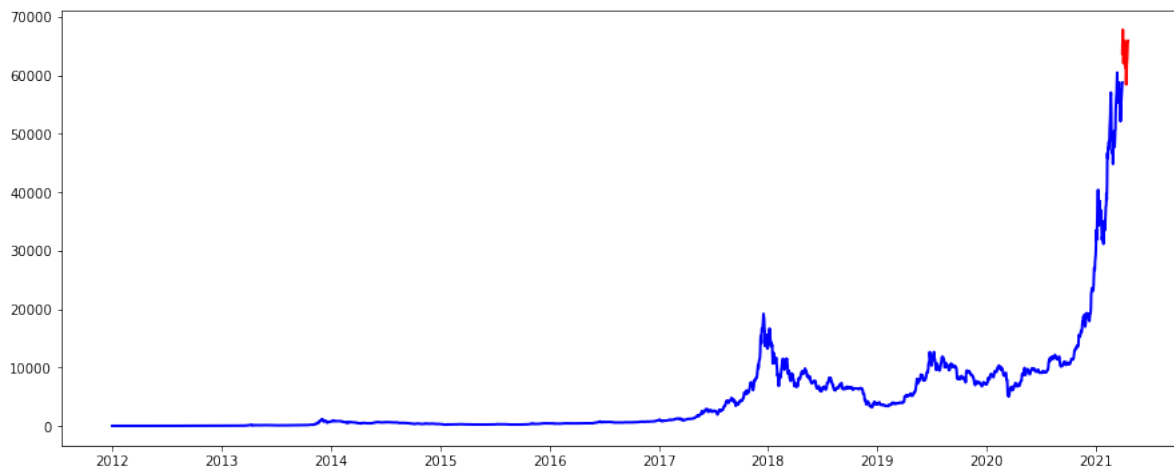


Figura 4: Gráfico comparativo del precio del bitcoin del conjunto de data (línea azul) con lo predicho por el modelo lineal (línea roja)

A partir de lo realizado se concluye que este método no es adecuado para la predicción con series de tiempo, ya que en una regresión lineal se asume que los errores en cada momento son independientes entre sí y están igualmente distribuidos, lo cual no es algo que ocurra en una serie de tiempo. Por eso será necesario aplicar a futuro métodos de redes neuronales (RNN) y/o métodos de auto regresión como ARIMA.

## 4.2. Implementación ARIMA

Un modelo autoregresivo integrado de promedio móvil o ARIMA es un modelo estadístico que utiliza variaciones y regresiones de datos estadísticos con el fin de encontrar patrones para realizar una predicción futura. Este modelo es dinámico de series temporales.

Un modelo ARIMA se caracteriza por 3 términos  $p$ ,  $d$ ,  $q$ :

- $p$  es el orden del término AR (auto regresivo), que indican el número de retrasos en las observaciones del modelo.
- $d$  es el número de diferenciaciones necesarias para que la serie de tiempo sea estacionaria.
- $q$  es el orden del término MA(media móvil).

Para nuestra implementación utilizamos Python y librerías como `statmodels`, que contienen funciones como `SARIMAX` para ajustar modelos ARIMA, entre otras funciones para descomponer datos y analizarlos.

### 4.2.1. Pre-procesamiento de los datos

Azari, 2019 utilizan ARIMA en su dataset, analizan empleando ARIMA para predicciones a largo plazo (3 años) o más, y encuentran que los errores de predicción son muy altos. Esto se debe a la alta volatilidad del precio del Bitcoin, por lo que se sugiere entrenar ARIMA para sub intervalos de tiempo a lo largo de distintas épocas, donde cada una tenga una tendencia única.

Para eso se decide, utilizar el dataset de <https://www.kaggle.com/mczielinski/bitcoin-historical-data>, donde se divide el dataset en datos diarios, datos semanales y mensuales y crear 3 modelos ARIMA, donde se compara su precisión en el ajuste. Para los datos mensuales y semanales, se utiliza todo el largo del dataset para ajustar el modelo, mientras que para el dataset diario, se utiliza los datos desde el 2017, para evaluar el impacto en la predicción considerando que ese fue el periodo de tiempo donde el precio del Bitcoin empezó a tener una tendencia más volátil.

Para el tratamiento del dataset, es necesario conocer si es estacionario o no, esto quiere decir que carece de una tendencia marcada en el tiempo (cambios en la media y varianza). Utilizando ADF (Augmented Dickey-Fuller), un test estadístico cuya hipótesis nula es que los datos no son estacionarios.

El método que se utilizan para estandarizar los datos antes de ajustar el modelo es una transformación box-cox, que similar a la transformación logarítmica, logra eliminar parte de la varianza de los datos. Es importante recordar que antes de realizar la predicción, se aplica la inversa de la transformación para obtener los datos originales antes de la transformación.

ADF entrega un nivel de significancia conocido como  $p - value$  el cual, si es menor a 0.05 se puede rechazar la hipótesis nula con un nivel de confianza del 95 %. En el caso en que los datos no sean estacionarios se realiza una diferenciación que nos entrega un estimado del parámetro  $d$ , que dice cuántas veces es necesario diferenciar el dataset para que sea estacionario.

#### 4.2.2. Evaluar el modelo

Una vez determinado el parámetro  $d$ , es necesario determinar la parte ARMA del modelo, donde los parámetros  $p$  y  $q$  deben ser determinados. Para esto se eligen los parámetros  $p$  y  $q$  que se ajustan mejor al modelo en base al criterio **AIC** (Akaike Information Criterion), este valor que retorna la función SARIMAX, que explica que tan bien un modelo se ajusta a los datos tomando en cuenta la complejidad del modelo, es decir, mientras más bajo sea su AIC, mejor se ajusta y más simple es el modelo. Por lo que se eligen parámetros que minimicen el AIC. Esto se hace a través de una búsqueda iterativa del producto cartesiano entre un rango de parámetros  $p$  (0 a 3) y  $q$  (0 a 3).

## 5. Resultados y Análisis

A partir de lo explicado anteriormente, se expone y discuten los resultados de cada parte de la aplicación de ARIMA desde el pre-procesamiento de los datos hasta la predicción.

### 5.1. Pre-procesamiento de los datos

Primero, se separa el dataset como datos mensuales, datos semanales y datos diarios, donde los diarios empiezan desde el 2017-03-01.

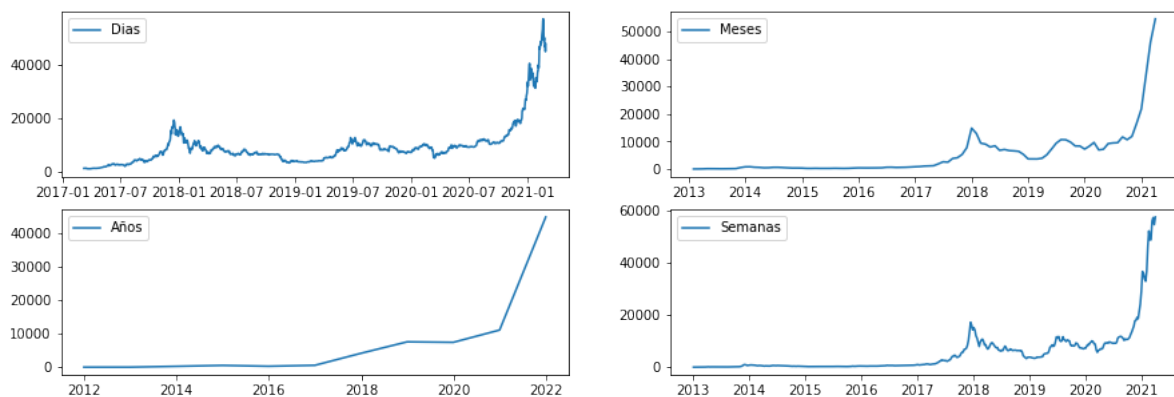


Figura 5: Gráfico comparativo del dataset separados en días, semanas, meses y años. Como se discutió anteriormente, se utilizan los datos diarios, semanales y mensuales para crear 3 modelos ARIMA.

### 5.1.1. Datos Mensuales

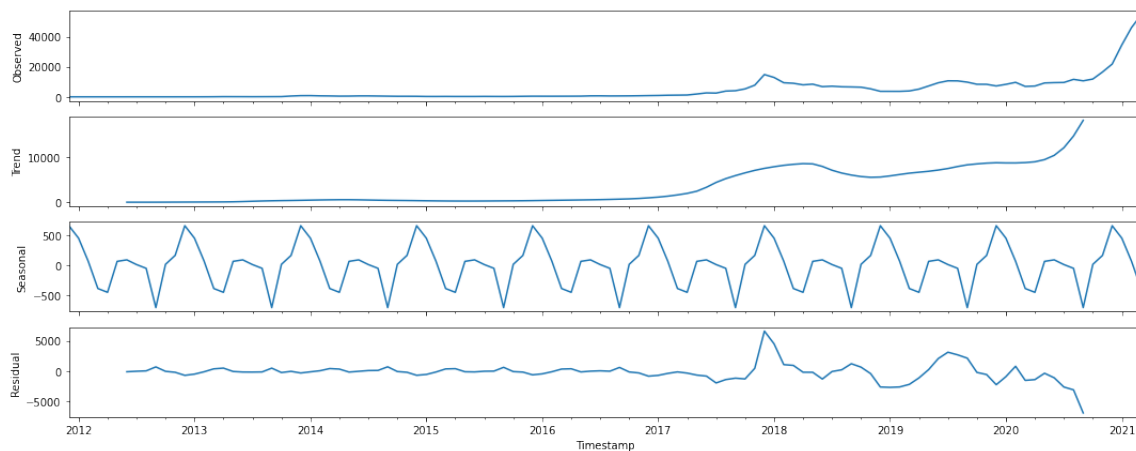


Figura 6: Descomposición estacional de la serie de tiempo mensual, la cual nos permite visualizar la tendencia, la componente estacionaria y los residuos.

Como se puede ver en la figura aún existe una tendencia marcada y pareciera haber una componente estacional anual en los datos mensuales, se procede a hacer un test ADF y resulta un p value de 0.998 por lo que aún no se puede rechazar la hipótesis nula. Para estandarizar el dataset se hace una transformación box-cox.

Al aplicar la transformación box-cox (desde la librería de paquetes statsmodels de python), se realiza nuevamente un test de estacionalidad con un p value de 0.92, por lo que se continúan diferenciando los datos.

Al diferenciar una vez, el test estadístico entrega un p value de 0, por lo que se puede rechazar la hipótesis nula de que el dato no es estacionario, por lo que el parámetro  $d$  es 1. También se prueba si al diferenciar datos actuales al dato del año anterior puede eliminar la componente estacional anual, pero esta entrega un p value de 0.44, por lo que no se consideró utilizar SARIMA (Seasonal ARIMA).

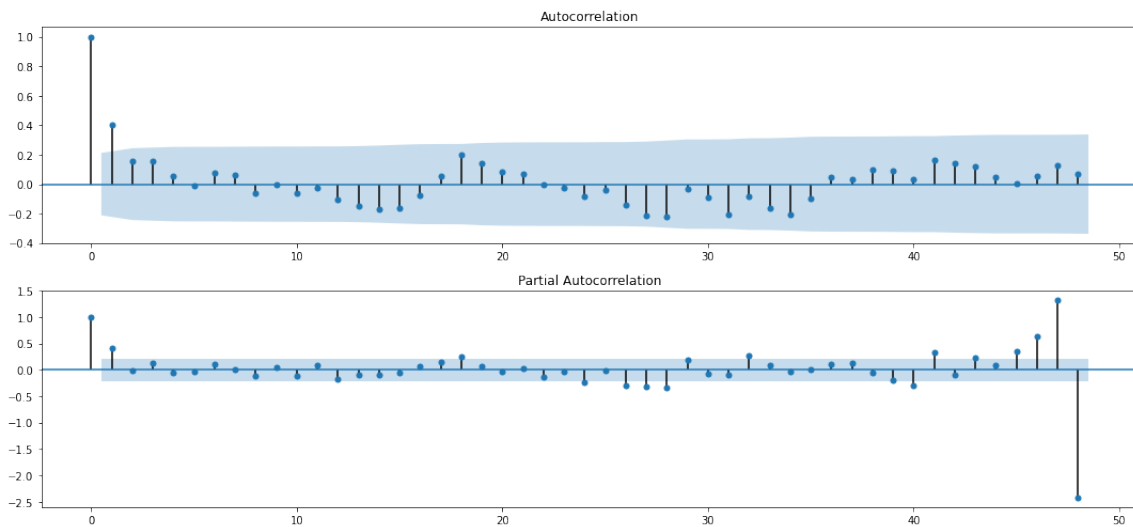


Figura 7: Graficos ACF/PACF (Auto-correlation function/partial auto-correlation function) post diferenciación, no existen muchos valores fuera del intervalo de confianza (zona azulada) por lo que no se puede extraer mucha información AR y MA de los residuos.

### 5.1.2. Datos Semanales

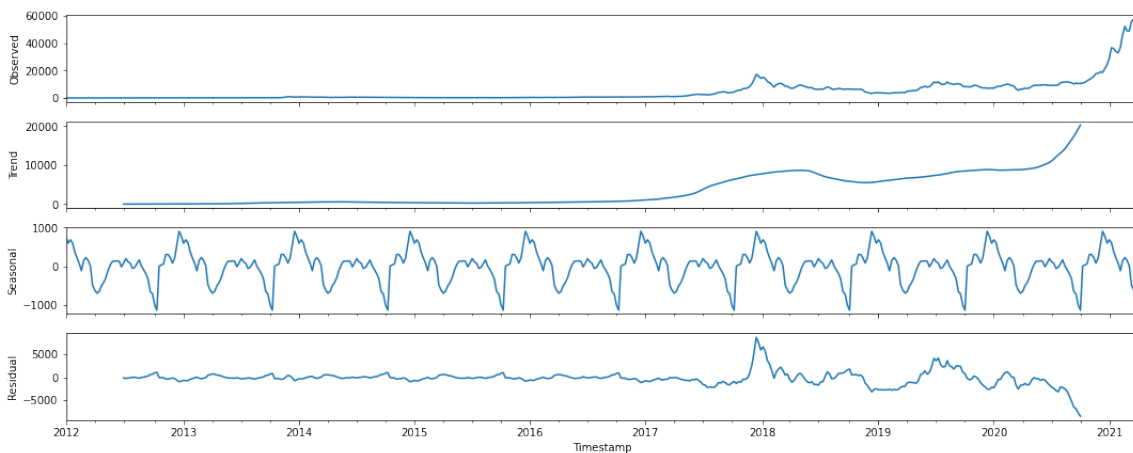


Figura 8: Descomposición estacional de la serie de tiempo semanal.

Existe una tendencia marcada como se ve en la la descomposición estacional, que empieza a incrementar desde el 2017, al realizar el test ADF entrega un p value de 0.999, así que se procede a estandarizar mejor los datos mediante transformación box-cox.

Luego de hacer la transformación, el test estadístico entrega un p value de 0.95, por lo que debe ser necesario diferenciar los datos para obtener estacionalidad.

La diferenciación de primer orden entrega un p value de 0 en el test ADF, por lo que se rechaza la hipótesis nula lo que implica estacionalidad.

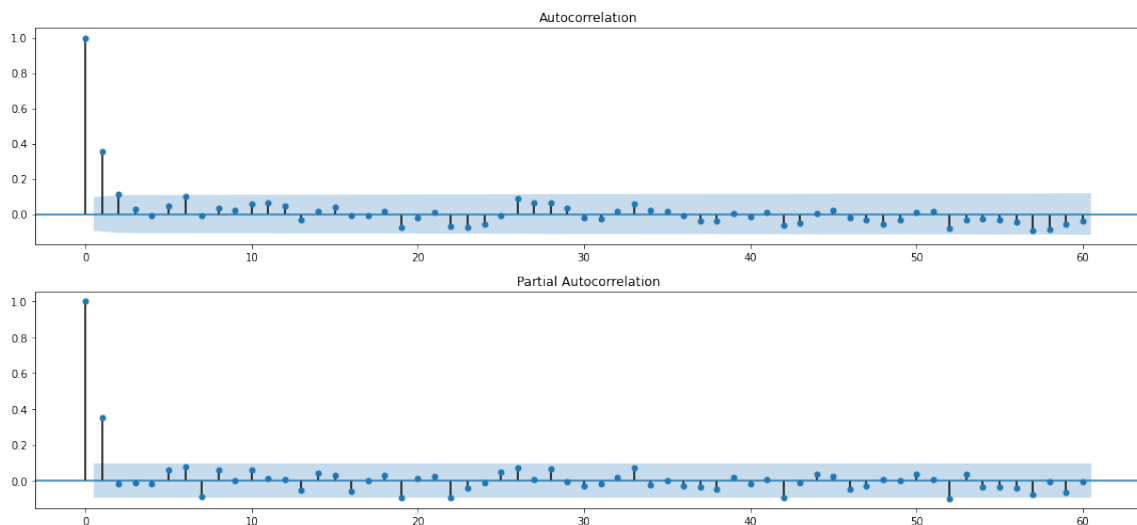


Figura 9: Plot ACF/PACF de residuos de datos semanales.

La figura no muestra lags significativos desde los primeros dos lags, y parece mantenerse en el intervalo de confianza, por lo que no se puede extraer información del residuo de esta forma, por lo que se utiliza una búsqueda de mejor modelo a través de AIC.

### 5.1.3. Datos Diarios

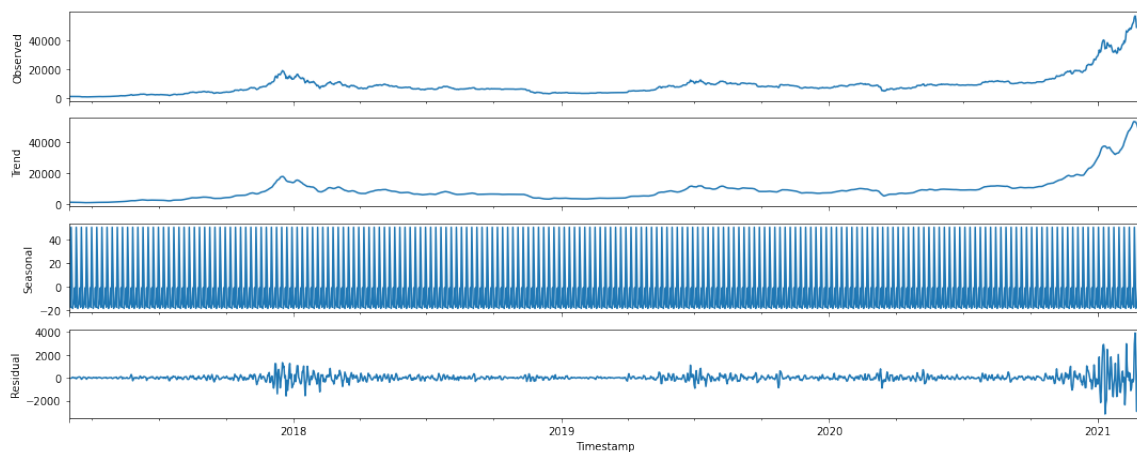


Figura 10: Descomposición estacional de los datos diarios.

La figura muestra una tendencia relevante y la parte estacional pareciera tener una componente semanal (cada 7 días), aunque esta vez solo se considera ajustar un modelo ARIMA para comparar resultados con otros estudios. El test estadístico antes de la transformación box-cox entrega un valor p de 1.0, y luego de aplicar la transformación box-cox 0.9, se procede a diferenciar una vez el dataset que da un valor p de 0, por lo que se rechaza la hipótesis nula y el parámetro d es 1.

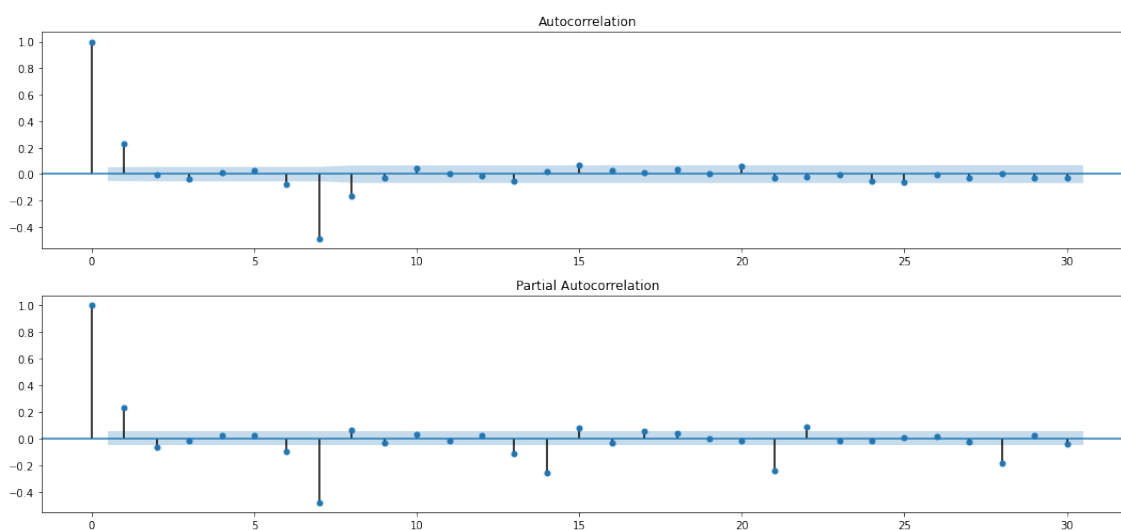


Figura 11: Plot ACF/PACF de residuos de datos diarios.

Los gráficos de auto-correlación muestra lags significantes cada 7 días, por lo que quizás a futuro sería interesante ajustar un modelo considerando la parte cíclica (SARIMA). Ahora centrará la investigación en el ajuste de un modelo ARIMA en base a los parámetros p y q que minimicen el AIC.

## 5.2. Ajuste a mejor modelo

Realizando una búsqueda que minimiza el valor AIC del modelo, se buscan los mejores parámetros p,q desde un rango de 0 a 3.

### 5.2.1. Datos Mensuales

Para los datos mensuales, el modelo ARIMA que minimiza el valor AIC tiene como parámetros (1,1,0).



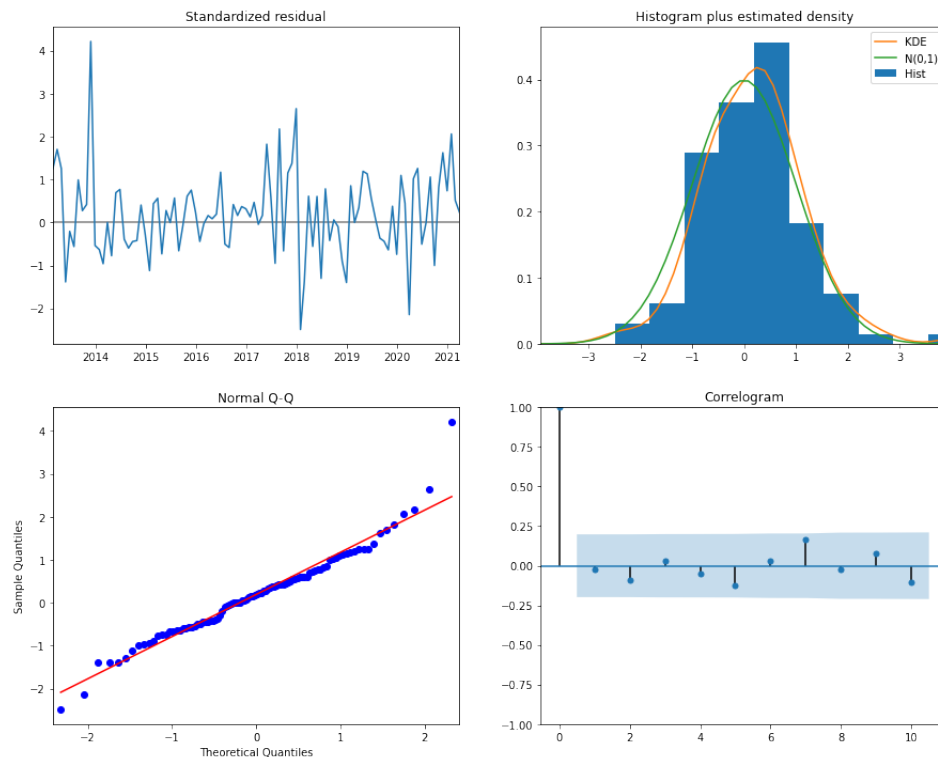


Figura 12: Diagnostico del mejor ajuste en base a AIC, la figura de arriba a la izquierda son los valores de residuos estandarizados, como se puede observar, los residuos no muestran un comportamiento cíclico por lo que la diferenciación del modelo pareciera estar correcto. El histograma de arriba a la derecha muestra la distribución de los residuos, donde la curva verde es una distribución normal estándar (0,1) y KDE la estimación de la densidad de los residuos. Se puede observar que los residuos se encuentran relativamente cercanos a estar normalmente distribuidos lo que es un buen indicio de un buen ajuste. La figura de abajo a la izquierda visualiza la distribución ordenada de los residuos (puntos azules) sobre la recta roja que son muestras tomadas de una distribución normal estándar, igualmente indica que los residuos están normalmente distribuidos. Finalmente la figura de abajo a la derecha es el correlograma (ACF) de los residuos, donde no parecen haber lags significantes fuera del intervalo de confianza.

A partir de lo analizado el ajuste a los datos parece estar en buenas condiciones para realizar una predicción.

### 5.2.2. Datos Semanales

El mejor ajuste en base a minimizar el valor AIC fue un modelo ARIMA(1,1,0).

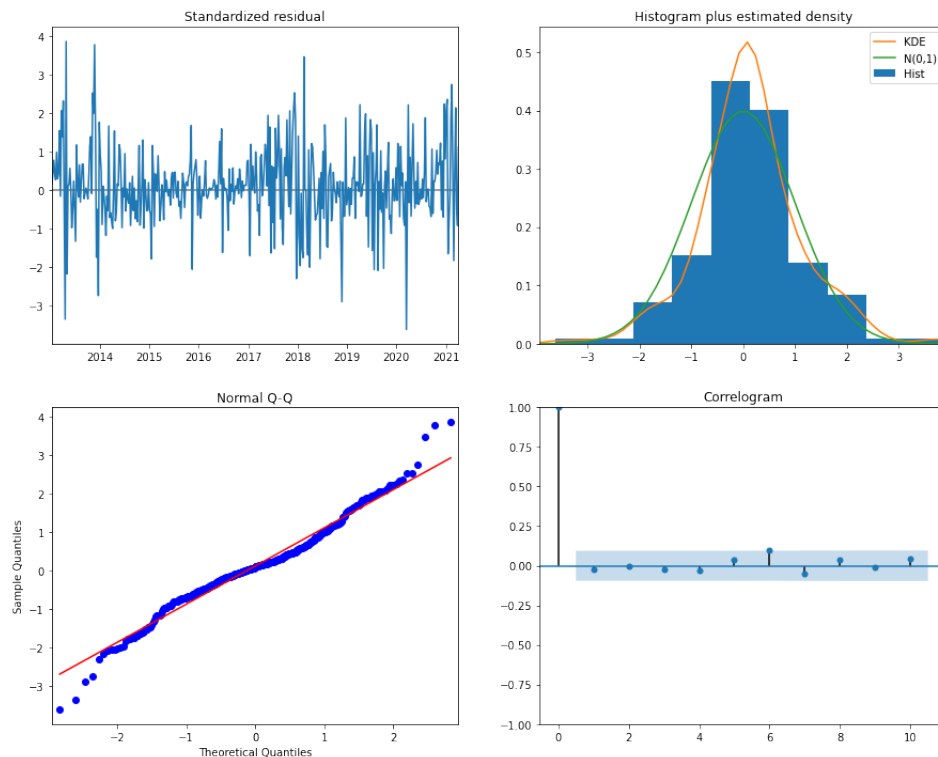


Figura 13: Diagnostico del ajuste a los datos semanales, el gráfico del residuo estandarizado no muestra comportamientos cíclicos. El histograma con la densidad estimada muestra que los residuos están igual distribuidos hacia el centro, si bien no tan distribuidos como la curva normal. Mientras que el gráfico Normal Q-Q indica que los residuos se ajustan correctamente a la normal lo que es un buen indicio. El correlograma no muestra lags significantes fuera del intervalo de confianza, fuera de eso no podemos extraer ninguna información acerca de este gráfico.

### 5.2.3. Datos Diarios

El mejor modelo de predicción en base a AIC para los datos diarios es  $ARIMA(0,1,1)$ .

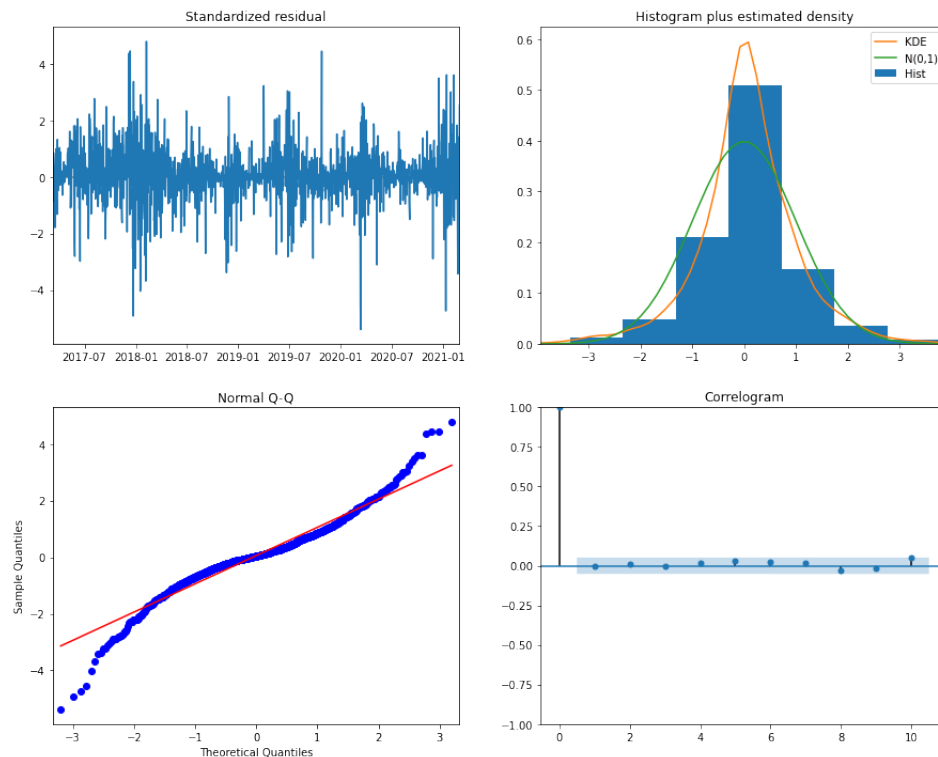


Figura 14: Diagnostico del mejor modelo evaluado, el gráfico de los residuos no presentan un comportamiento cíclico, lo que indica una correcta integración (parte I del modelo). En el histograma con la densidad estimada se puede observar que los datos están distribuidos hacia el centro como la distribución normal, con una varianza menor a la de la curva. El gráfico normal Q-Q nos muestra que los residuos están bien alineados con la muestra normal, exceptuando los valores iniciales y finales mostrando un comportamiento algo parecido al fenómeno de Runge, esto puede ser por la poca cantidad de datos que hay al principio y al final del dataset, aunque en general el modelo parece ser un buen ajuste a los datos. El correlograma no entrega mucha información ya que no se ven muchos lags fuera del intervalo de confianza.

### 5.3. Resultados de predicción

Para la predicción de los datos, se hizo 'one-step ahead static forecast', esto es, una predicción que utiliza el valor actual del dataset, sea  $x(T)$ , para predecir  $x(T+1)$  y así sucesivamente, prefiriendo utilizar los datos actuales del dataset (y no los predichos) para predecir el siguiente paso.

Sumando eso también se aumenta la muestra a pronosticar, donde en ese caso, dependerá de los datos predichos para predecir el siguiente paso; esto es para ver el desempeño del método ARIMA para predicciones a largo plazo.

Es importante que al momento de mostrar los resultados de la predicción/forecast, se

realice una transformada inversa box-cox para visualizarlos normalmente, debido al tratamiento hecho en el pre-procesamiento de los datos.

### 5.3.1. Datos Mensuales

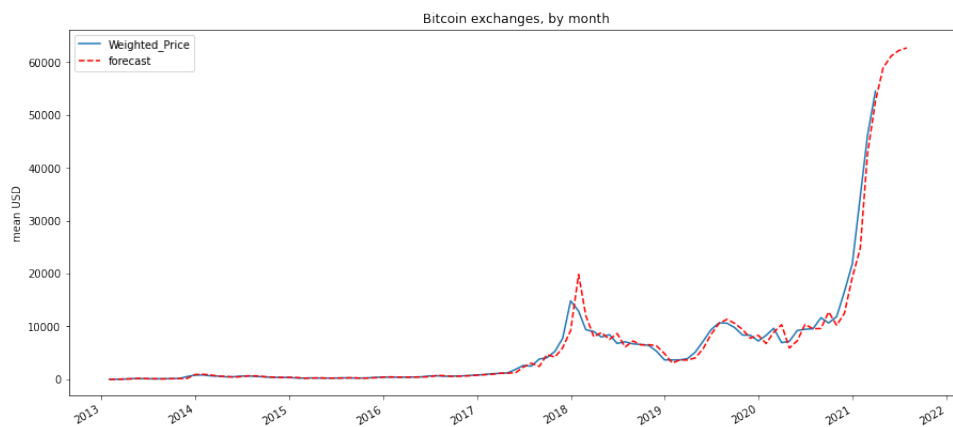


Figura 15: Gráfico de la predicción de datos mensuales, se puede observar un error más acentuado en los peaks del dataset, los errores MSE (Mean Square Error) y MAPE(Mean Absolute Percentage Error) son 1867 y 16.1 % respectivamente.

Un MAPE de 16.1% indica que quizás el ajuste del modelo ARIMA no sea muy confiable para predecir mensualmente los precios del Bitcoin.

### 5.3.2. Datos Semanales

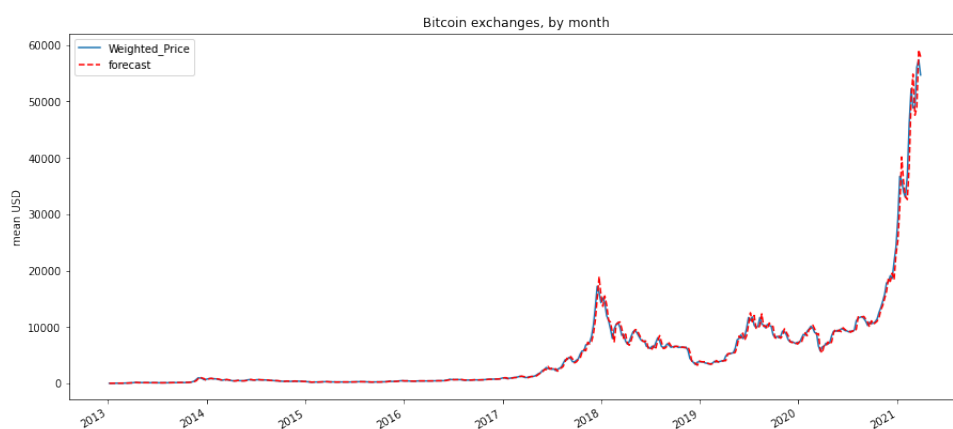


Figura 16: Gráfico de la predicción de datos semanales, en este caso la curva del forecast parece ajustarse mejor en los peaks que el modelo mensual, los errores MSE (Mean Square Error) y MAPE(Mean Absolute Percentage Error) son 737.32 y 6.56 % respectivamente.

Un MAPE de 6.56 % muestra una mejora significativa en comparación con la predicción mes a mes del precio, aún así, un error mayor del 5 % no es ideal si se quiere utilizar este modelo con confianza para estimar el futuro precio del bitcoin.

### 5.3.3. Datos Diarios

En los datos diarios cabe recalcar que se utiliza un sub intervalo de la muestra completa, tomando en cuenta solo la época donde el precio del bitcoin empezó a incrementar volátilmente. Esto puede influir en la precisión de la predicción, además de los intervalos más cortos de predicción (1 día). Esto es en vista y consideración de la conclusión de Azari, 2019, que dice que el modelo ARIMA tiene un mejor desempeño en la predicción a corto plazo, donde recomienda métodos como RNN (redes neuronales) para predicciones a largo plazo.

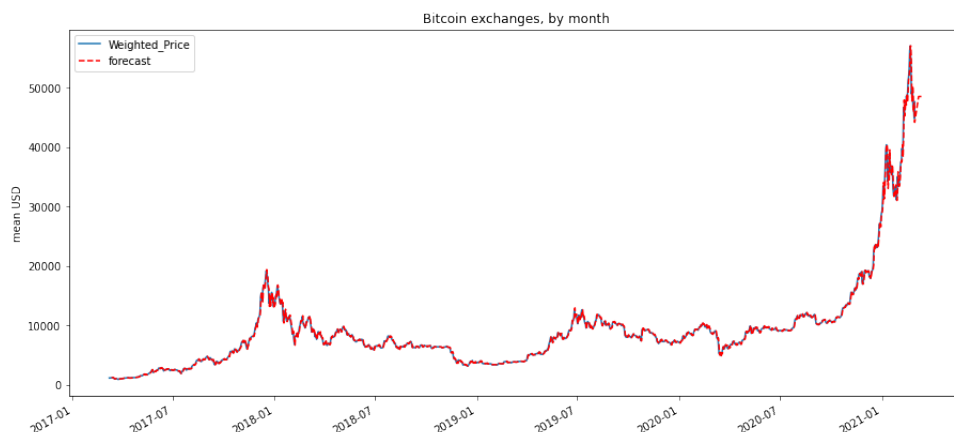


Figura 17: Gráfico de la predicción de datos diarios donde la curva azul representa el precio ponderado y la roja rayada la predicción, en este caso la curva del forecast parece ajustarse mejor en los peaks que en los modelos anteriores, los errores MSE (Mean Square Error) y MAPE(Mean Absolute Percentage Error) son 549.64 y 2.19 % respectivamente.

Con un MAPE de 2.19 % se puede confirmar lo concluido por Azari, 2019, donde las predicciones a corto plazo en ARIMA resultan bastante más precisas, esto puede mejorarse aún más incorporando otros métodos al pre-procesar los datos, y sub dividir los intervalos aún más para entrenar un modelo que incluya una tendencia única.

## 6. Dificultades

A lo largo de la realización del proyecto, se presentaron diferentes dificultades. Una de ellas fue el pre-procesamiento de los datos al momento de elegir una transformación buena para la serie de tiempo. Para estimar los valores  $p$ ,  $d$  y  $q$  de ARIMA, a parte de las mencionadas anteriormente, en la realización del código se realiza la “búsqueda por grilla”, buscando el modelo que minimizara el mean square error, que es un proceso que busca exhaustivamente a través de un subconjunto especificado manualmente del espacio de hiperparámetros del algoritmo de destino. Este, al momento de programarlo es muy costoso, ya que se utiliza un triple for en la búsqueda.

Por otra parte la volatilidad de los datos fueron un problema en diversas circunstancias, ya que se debía realizar el entrenamiento en distintos sub-intervalos de tiempo y probar así cuál era el más preciso. En relación a esto, solo considerar el tiempo como variable no permite obtener una buena predicción. Aunque sea una moneda descentralizada, se deben considerar otros factores, como la prohibición de la minería en China, los twitts relacionados al bitcoin y otros factores sociales y económicos que pueden afectar directamente al bitcoin.

## 7. Conclusiones

Primeramente, el método de regresión lineal se veía interesante para resolver el problema de predicción, pero no se considera un factor sumamente importante: una regresión lineal asume que los errores en cada momento son independientes entre sí y están igualmente distribuidos y esto no cumple en una serie de tiempo. Por esto no se considera para continuar con el estudio y se prefiere utilizar ARIMA.

Con relación a ARIMA, las predicciones que se realizan son de un solo paso, esto quiere decir que se toma un dato de la muestra y se predice el siguiente, por lo que en cada dato de la muestra el modelo ajustaba el siguiente paso. Esto fue una complicación al momento de predecir valores a futuro, ya que estos no existen y no se pudo realizar una predicción certera.

Aunque los resultados son mejores, esto no quiere decir que se haya predicho una tendencia correcta. Como se dijo en el apartado de dificultades, la volatilidad del precio del bitcoin es un gran problema al momento de realizar el pre-procesamiento de los datos, por lo que se divide el dataset en sub-intervalos mensuales, semanales y diarios. Esta volatilidad dificulta la predicción, ya que no tiene un comportamiento el cual el modelo pueda aprender. Por esto, en un futuro se podría tomar en consideración otros métodos de pre-procesamiento, incluir modelos de ARIMA en subintervalos del dataset donde su tendencia sea única e incluir y comparar distintos datasets. En relación a esto, se considera también poder integrar o predecir de alguna manera otros factores al cálculo de la predicción; como la cantidad de twitts que hablen del bitcoin o también factores socio-económicos internacionales como la prohibición de la minería en China. También se puede considerar comparar un forecast dinámico, para evaluar resultados diferentes.

## Referencias

- Ayaz, Zeba y col. (abr. de 2020). “Bitcoin Price Prediction using ARIMA Model”. En: DOI: 10.36227/techrxiv.12098067.v1. URL: [https://www.techrxiv.org/articles/preprint/Bitcoin\\_Price\\_Prediction\\_using\\_ARIMA\\_Model/12098067](https://www.techrxiv.org/articles/preprint/Bitcoin_Price_Prediction_using_ARIMA_Model/12098067).
- Azari, Amin (2019). *Bitcoin Price Prediction: An ARIMA Approach*. arXiv: 1904.05315 [cs.SI].
- D. Kwon J. Kim, J. Heo C. Kim e Y. Han (2019). “Time Series Classification of Cryptocurrency Price Trend Based on a Recurrent LSTM Neural Network”. En: *JIPS* 15.3, págs. 694-706. DOI: 10.3745/JIPS.03.0120.
- Ibrahim, Ahmed, Rasha Kashef y Liam Corrigan (2021). “Predicting market movement direction for bitcoin: A comparison of time series modeling methods”. En: *Computers Electrical Engineering* 89, pág. 106905. ISSN: 0045-7906. DOI: <https://doi.org/10.1016/j.compeleceng.2020.106905>. URL: <https://www.sciencedirect.com/science/article/pii/S0045790620307576>.
- Karasu, Seçkin y col. (2018). “Prediction of Bitcoin prices with machine learning methods using time series data”. En: *2018 26th Signal Processing and Communications Applications Conference (SIU)*, págs. 1-4. DOI: 10.1109/SIU.2018.8404760.
- Pastor Cobo, Manuel y Pablo de Torre Barrio (2019). “Elaboración de un algoritmo predictivo sobre la evolución del precio de las criptomonedas”. El software generado a lo largo del trabajo está disponible en un repositorio público de GitHub: <https://github.com/PabloDeTorre/NeuralBTCPredictor> Trabajo de Fin de Grado, Universidad Complutense, Facultad de Informática, Departamento de Sistemas Informáticos y Computación, Curso 2018/2019. URL: <https://eprints.ucm.es/id/eprint/56582/>.
- Shah, Devavrat y Kang Zhang (2014). “Bayesian regression and Bitcoin”. En: *2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, págs. 409-414. DOI: 10.1109/ALLERTON.2014.7028484.