

# **PROYECTO\_AUTOMATIZACION\_COMPRA\_VENTA\_DE\_BTC**

## **Breve reseña del BTC**

Bitcoin nace en noviembre de 2008, cuando una persona (o grupo de personas) bajo el pseudónimo de Satoshi Nakamoto, envió un mensaje a un correo sobre criptografía, describiendo un proyecto para crear una moneda digital que sirviera para contabilizar y transferir valor. Según afirmó el propio Nakamoto cuando surgió esta criptomoneda. Bitcoin nace con ambiciones elevadas: proporcionar a los ciudadanos un medio de pago que posibilite la ejecución de transferencias de valor rápidas, a bajo costo, y que, además, no pueda ser controlado ni manipulado por gobiernos, bancos centrales o entidades financieras. En enero de 2009 entró en funcionamiento la primera red basada en el protocolo Bitcoin, lo cual supuso el origen de las criptomonedas. A partir de 2011, algunas organizaciones empezaron a aceptar donaciones en bitcoins y los comerciantes que operaban en Internet empezaron a aceptar estas divisas como medio de pago. Desde este momento, su crecimiento fue imparable. Sin embargo, a lo largo de estos años, han ido surgiendo más criptomonedas nuevas, que compiten con Bitcoin y hacen que su valor fluctúa de manera constante.

## **Objetivo del proyecto**

Este proyecto tiene como objetivo desarrollar un sistema automatizado que tome decisiones de compra y venta de Bitcoin (BTC) basadas en datos históricos y patrones de comportamiento. A continuación, se presentan los aspectos clave del proyecto, que surgen de las siguientes preguntas:

- ¿Es posible crear un modelo que siempre genere ganancias al operar con BTC?
- ¿Qué tipo de modelo de aprendizaje automático sería más adecuado para este propósito?
- ¿Los datos históricos pueden utilizarse para entrenar al modelo y establecer patrones de referencia?

El objetivo principal es crear un modelo de aprendizaje automático que pueda operar de manera autónoma y generar ganancias al comprar y vender BTC, utilizando los datos históricos para identificar patrones significativos en los precios del BTC, con el objetivo de maximizar las ganancias.

## **Organización del Proyecto**

El proyecto se presenta en un repositorio de git con una estructura cookiecutter, link del repositorio:

[https://github.com/silviabolana2006/Automatizacion\\_de\\_compra\\_venta\\_de\\_btc.git](https://github.com/silviabolana2006/Automatizacion_de_compra_venta_de_btc.git)

Proceso de ETL

Para comenzar este proyecto se utilizó un csv que se compone de 6 columnas, 2860 filas de la página Kaggle, el nombre original del csv es “Precios de Bitcoin (2014-2024)” del colaborador Alexander Kapturov ( Propietario ) la licencia es CCO de dominio público, la última fecha de actualización del dataset fue hace en mayo 2024

	Date	Open	High	Low	Close
lunes, 21 de marzo de 2016	413417999	413417999	410381012	413307007	
martes, 22 de marzo de 2016	413131989	418375000	412531006	418088989	
miércoles, 23 de marzo de 2016	418161011	419268005	417364014	418040985	
jueves, 24 de marzo de 2016	418424011	418679993	415485992	416394012	
viernes, 25 de marzo de 2016	416507996	418079987	415558014	417177002	

foto1

En principio el dataset tenía 3375 filas y columnas,se realizó el proceso de ETL en PowerBi, en principio, donde fueron eliminados los primeros 2 años , una columna duplicada, y los datos nulos. Se unificó el tipo de datos con un decimal, tipo float y la fecha en formato fecha . Como figura en la foto 1. El segundo proceso de ETL fue en Python donde, se cambiaron los nombres al español de los atributos, se agregaron 2 columnas nuevas quedando 8 columnas Y 2942 filas. Se modificaron los decimales ya que era confuso para el modelo, se limitó al valor entero en usd. En la foto 2 se observa el data frame que trabajamos desde la librería panda, se guardó en una carpeta con el nombre mi\_dataframe\_BTC.

1	Fecha	Precio_Apertura	Precio_Máximo	Precio_Mínimo	Precio_Cierre	Volumen	Variabilidad	Precio_Promedio
2	31/12/2015	426.0	433.0	419.0	431.0	45996600	5.0	1
3	1/01/2016	431.0	436.0	428.0	434.0	36278900	3.0	1
4	2/01/2016	435.0	436.0	432.0	433.0	30096600	-2.0	0
5	3/01/2016	434.0	434.0	425.0	430.0	39633800	-4.0	0
6	4/01/2016	430.0	435.0	429.0	433.0	38477500	3.0	1
7	5/01/2016	433.0	434.0	430.0	432.0	34522600	-1.0	0
8	6/01/2016	432.0	432.0	426.0	429.0	34042500	-3.0	0

foto2

## Estructura del dataframe

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2943 entries, 0 to 2942
Data columns (total 8 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Fecha                 2943 non-null   object
1   Precio_Apertura       2943 non-null   float64
2   Precio_Máximo         2943 non-null   float64
3   Precio_Mínimo         2943 non-null   float64
4   Precio_Cierre         2943 non-null   float64
5   Volumen               2943 non-null   int64
6   Variabilidad          2943 non-null   float64
7   Precio_Promedio       2943 non-null   int64
dtypes: float64(5), int64(2), object(1)
memory usage: 184.1+ KB
None
```

\*Fecha (object): Este campo contiene fechas en formato de texto. Se convirtió en un formato de fecha y hora (datetime) para facilitar el análisis temporal.

\*Precio Apertura, Precio Máximo, Precio Mínimo, Precio Cierre (float64): Estos campos contienen precios en formato de punto flotante. Representan los precios de apertura, máximo, mínimo y cierre de cada día. Son fundamentales para el análisis financiero y la generación de indicadores técnicos.

\*Volumen (int64): Este campo contiene el volumen de transacciones. Es un entero que indica la cantidad de activos negociados. Es útil para analizar la liquidez y el interés del mercado.

\*Variabilidad (float64): Este campo mide la variabilidad de los precios, se utiliza para evaluar la volatilidad del mercado.

\*Precio Promedio (int64): Este campo representa un precio promedio calculado, útil para tener una referencia general del precio durante el período analizado.

## Desarrollo del Modelo

1. Entrenamiento y prueba El modelo desarrollado incluye una arquitectura cookie cutter, se utilizó un algoritmo de regresión lineal donde se dividió el conjunto de entrenamiento y prueba, para X se tomó la variable precio cierre, para Y la variable precio apertura. El 20% de los datos se utilizó como conjunto de prueba (test\_size=0.2) y el 80% de los datos para entrenamiento;

2. Rendimiento del modelo Se Realizaron predicciones en los datos de prueba y el rendimiento del modelo. El modelo arrojó las siguientes predicciones:

\*Error Cuadrático Medio del modelo,(MSE): es de 6560.85, mide la diferencia entre las predicciones del modelo y los valores reales. Cuanto menor sea el MSE, mejor será el rendimiento del modelo En este caso como el valor del precio de BTC está en miles, parece alto pero es aceptable por el tipo de dato, indica que el modelo de predicción está funcionando.

\*Coeficiente de Determinación ( $R^2$ ): es 0.87 el  $R^2$  indica la proporción de la varianza en la variable dependiente que se puede predecir a partir de las variables independientes. Un valor de 0.87 es un ajuste bueno

\*Error absoluto medio: es de 11.46 es el promedio de errores absolutos de todos los puntos de datos en el conjunto de datos dado.

\*Error absoluto mediano: es de 2.94 la mediana de todos los errores en el conjunto de datos dado. La principal ventaja de esta métrica es que es robusta a los valores atípicos.

3. Curvas de Aprendizaje: Se utiliza para evaluar el rendimiento del modelo a medida que aumenta el tamaño del conjunto de entrenamiento, podemos trazar la puntuación de entrenamiento y prueba en función del tamaño del conjunto de datos. La Curva de Aprendizaje muestra cómo cambia el rendimiento del modelo a medida que aumenta el tamaño del conjunto de entrenamiento.

\* Entrenamiento: Esta línea muestra cómo se comporta el modelo en el conjunto de entrenamiento a medida que aumenta el tamaño del conjunto. Inicialmente, la puntuación es alta, pero luego disminuye. Esto puede indicar que el modelo está sobre ajustando los datos de entrenamiento.

\* Prueba: La línea naranja comienza baja y aumenta, alcanzando una meseta. Esto indica que el modelo mejora su rendimiento a medida que se entrena con más datos, pero eventualmente alcanza un punto donde las mejoras adicionales son menores

\* Tamaño del conjunto de entrenamiento (eje horizontal): Representa el número de ejemplos de entrenamiento utilizados para entrenar un modelo. Comienza en 0 y aumenta hasta 2000.

\* Puntuación (eje vertical): La puntuación generalmente se refiere al rendimiento del modelo. En este caso, parece estar en una escala negativa, lo que sugiere que se trata de una métrica de error. Cuanto más cerca de 0, mejor.

4. Gráfico de Regresión Lineal: El gráfico de regresión ayuda a visualizar cómo se ajusta el modelo a los datos, muestra la relación entre dos variables, una variable independiente (X) y una variable dependiente (Y).

La línea de regresión permite identificar tendencias en los datos, puede ser positiva, negativa o incluso cero.

En el modelo se analizaron las variables volumen con precio de cierre del modelo de regresión lineal, obteniendo como resultado una relación positiva entre el volumen y el precio de cierre. A medida que el volumen aumenta, también lo hace el precio de cierre, hay una relación directa entre las dos variables. Para determinar si había más variables que arrojen relación positiva en las predicciones del modelo de regresión lineal con los valores reales, se comparan la variable Volumen frente al Precio de Apertura, arrojando relación negativa, a medida que aumenta la variable independiente, la variable dependiente disminuye, esto significa que, en promedio, cuando el volumen aumenta, el precio de apertura tiende a disminuir.

5. Histograma de Residuos: Los residuos son las diferencias entre los valores reales y las predicciones. Un gráfico de residuos permite verificar si el modelo está capturando correctamente la variabilidad en los datos o si hay patrones no capturados.

El gráfico mostró la barra azul prominente en el valor 0 del eje horizontal, con una frecuencia de más de 500, sugiere que los residuos del conjunto de datos están fuertemente centrados alrededor de cero, indica un buen ajuste del modelo,

## **Análisis de Resultados**

\*En el gráfico de Variabilidad Mensual Promedio en 2016 y de 2020, se encontraron patrones de ganancia considerable que pueden determinar en un futuro cuándo podría ser un buen momento para comprar BTC:

\*Meses con baja variabilidad (negativa): Los meses en los que la variabilidad baja o es negativa podrían indicar una menor volatilidad y, potencialmente, precios más bajos. En el gráfico, esto parece ocurrir en los meses 7 y 8 de 2016, 3 y 9 de 2020, meses recomendables para la compra.

\*Meses con alta variabilidad (positiva): Aunque la alta variabilidad puede indicar volatilidad, también puede señalar oportunidades de compra antes de un aumento significativo en el precio. Los meses 10 y 12 en 2016 Y 2020 , muestran alta variabilidad positiva, lo que podría ser indicativo de un aumento en el interés y el precio de BTC.

## **Análisis de tendencia**

Se observa un patrón de aumento en la variabilidad positiva hacia el final del año, momento de vender los BTC, generando el patrón de venta y en los meses de baja variabilidad antes de un posible aumento en el precio, es momento de comprar BTC, generando el patrón de compra.

\*En caso de los halving que son cada 4 años, contando desde el 2012, se observa después de octubre una subida del precio consecutiva para finalmente en diciembre dispararse el precio de BTC.

\*Otro hallazgo es que en el año siguiente al halving en el primer semestre hay un mes donde el precio tiene una caída importante y después se mantiene; Se podría utilizar esa caída para comprar BTC,

## **Recomendaciones Y conclusión**

El modelo desarrollado y los análisis realizados proporcionan una base sólida para tomar decisiones informadas sobre la compra de BTC, aprovechando los patrones de variabilidad mensual y los efectos de los eventos de halving, este modelo podría utilizarse para compra y venta de BTC sin inconveniente, ya que puede predecir tendencias futuras, es un modelo que demostró un 87% de efectividad, es un modelo confiable y seguro, cumpliendo con el objetivo principal.

\*Se recomienda la compra de BTC: en meses de baja variabilidad, a precios potencialmente bajos y utilizar las caídas de precio del primer semestre del año siguiente al halving.

\*Se recomienda la venta de BTC: en meses de variabilidad alta, a precios récord en subida.

