

BITMONEY:

Propuesta de Solucion



Autores

M.I.C. Carlos Abraham Carballo Monsivais

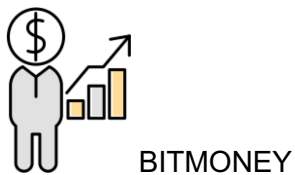
I.S.C. Leticia Edith Trujillo Ballesteros

C.L.M.A. Sacbe García García



Hackathon Blockchain 2020

Elaboración 2020, Primera edición



Propuesta de Solución

Contenido

4.	Propuesta de la Solución.....	4
4.1.	Resumen Solución Propuesta	4
4.1.	Datos de Criptomonedas	5
4.2.	Minería de Datos.....	5
4.3.	Extracción de Características	6
4.4.	Entropía de Shannon	7
4.5.	Agrupamiento de Características.....	7
4.6.	Transformación de Serie de Tiempo a Imagen.....	9
4.7.	Red Neuronal Convolucional - CNN	10

4. Propuesta de la Solución

4.1. Resumen Solución Propuesta

Se propone un procedimiento para clasificar en tres grupos (vender, comprar, riesgo:incertidumbre) la recomendación de inversión y estimar el nivel de confianza al inversionista mediante datos financieros de las criptomonedas en combinación con Redes Neuronales Convolucionales (CNN), donde se realiza una transformación de la serie de tiempo del precio en dolares de las criptomonedas en imágenes. La Figura 1 muestra los pasos que se siguen para realizar la clasificación y estimación del nivel de confianza para el inversionista.



Figura 1. Metodología de la Propuesta de Solución

Se propone un nuevo método de etiquetado para financieros de las criptomonedas, según el nivel de clasificación, basado en las técnicas estadísticas clásicas en el que primero se extraen las características de los datos sin procesar utilizando una transformación logaritma para estabilizar el comportamiento de la criptomoneda (LOG), y luego se aplica la entropía de Shannon para resaltar las



BITMONEY

características extraídas. Después, la agrupación de las características se realiza utilizando el algoritmo *K-Means* para hacer el etiquetado de los datos.

Este procedimiento es ideal cuando se tienen datos que no están etiquetados de datos financieros como las criptomonedas, dado que proporciona una forma automática de clasificarlos, proporcionando las etiquetas necesarias para supervisar el aprendizaje, para situaciones donde no se cuente con el conocimiento de un experto que haya etiquetado previamente los datos, en esta estrategia se utilizan las similitudes de los mismos datos, lo cual permite dar una alternativa al etiquetado.

A continuación, se realiza la conversión de los datos financieros de las criptomonedas sin procesar a imágenes, debido a que la extracción de características en un proceso sumamente importante y con la conversión a imágenes proporciona una buena manera de explorar las características en dos dimensiones. Además, permite identificar de una manera más visual el cambio entre los diferentes valores del precio sobre el tiempo dado que la información contenida en las imágenes representa el comportamiento del precio en dolares de las criptomonedas. En la fase de entrenamiento, las imágenes obtenidas junto con sus etiquetas se utilizan para entrenar una CNN para la tarea de clasificación multiclase, donde las clases representan valor del precio diario de cada criptomoneda en dolares. Se propone una arquitectura de CNN basada en AlexNet para hacer el clasificador del precio en dolares. Esta arquitectura fue seleccionada debido a que al tener imágenes pequeñas (32x32 pixeles) para entrenar a la CNN se requiere una arquitectura no tan compleja con muchas capas, además de que esta arquitectura ha demostrado ser buena en otros contextos. Finalmente, una vez entrenado el modelo, la clasificación puede ser implementada en tiempo real, tomando una pequeña muestra de las de los precios provenientes de fuentes de la internet de criptomonedas para convertirla en imágenes y luego ser clasificadas de acuerdo al nivel aproximado del precio. Esto último se conoce como fase de prueba.

4.1. Datos de Criptomonedas

Se determino solo tres criptomonedas: Bitcoin, Ethereum y Decred para fines comparativas y el desarrollo de la funcionalidad básica de BITMONEY. Los datos de estas criptomonedas fueron extraídos de las siguientes fuentes de internet: Yahoo, Kraken, Bitstamp. Los datos se reciben crudos, con algunas inconsistencias que se deben de trabajar con minería de datos.

4.2. Minería de Datos

La minería de datos fue utilizada para combinar los datos de las tres fuentes diferentes de cada criptomoneda: Yahoo, Kraken, Bitstamp para estrurar una nueva base de datos de cada criptomoneda combinando el promedio de las tres fuentes y imputando por la tecnica de spline, los datos perdidos. Después de este proceso de depuración se obtiene datos completo e integros de cada criptomoneda para fines de análisis o desarrollo de modelos en las fases siguientes.



BITMONEY

4.3. Extracción de Características

La extracción de características es uno de los procesos más importantes en el área del aprendizaje de máquina debido a que permite hacer la reducción de la dimensionalidad de todo el conjunto de datos. Esto permite que la información sea manejable de una manera más fácil, debido a que se disminuye la complejidad al momento de manejar todos los datos sin procesar. Por otra parte, también se acorta la complejidad computacional en el procesamiento, al evitar trabajar con una carga masiva de datos.

Para realizar la extracción de las características de los precio en dolares, fue necesario hacer uso de las Características estadísticas tradicionales (TSF), porque son excelentes herramientas que se utilizan en el dominio del tiempo para caracterizar los cambios del precio en dólares de las criptomonedas durante su evolución. Además, también permiten estimar los cambios de precio sobre el tiempo.

Cuando el mercado sufre cambios o es afectado por crisis como COVID-19, los precios se intensifican, y los valores de las TSF varían considerablemente, indicando el comportamiento de su volatilidad. Existen numerosas TSF que se utilizan frecuentemente para el análisis de datos financieros, algunas de las más importantes son RMS, curtosis, oblicuidad, pico a pico, factor de cresta, factor de forma, factor de impulso y factor de margen, las cuales se muestran en la Tabla 1, donde n indica el número de puntos discretos de la muestra, x_i es el i -ésimo punto en la señal de la muestra, \bar{x} es la media de los valores de la muestra, σ representa la desviación estándar y min y max son los valores mínimo y máximo de la muestra, respectivamente.

Nombre	Formula
RMS (Root Mean Square)	$\sqrt{\left(\frac{1}{n} \sum_{i=1}^n x_i^2\right)}$
Curtosis	$\frac{1}{n} \sum_{i=1}^n \frac{(x_i - \bar{x})^4}{\sigma^4}$
Oblicuidad	$\frac{1}{n} \sum_{i=1}^n \frac{(x_i - \bar{x})^3}{\sigma^3}$
Pico a pico	$x_{min} - x_{max}$
Factor de cresta	$\frac{max x_i }{RMS}$
Factor de forma	$\frac{RMS}{\frac{1}{n} \sum_{i=1}^n x_i }$
Factor de impulso	$\frac{max x_i }{\frac{1}{n} \sum_{i=1}^n x_i }$
Factor de margen	$\frac{max x_i }{\left(\frac{1}{n} \sum_{i=1}^n x_i ^{\frac{1}{2}}\right)^2}$

Tabla 1. Características estadísticas tradicionales (TSF)



La idea principal de estas características es identificar una tendencia monótona, es decir, cuando el precio de la criptomoneda tiene un comportamiento crecien d o decreciente, el valor de estas características aumenta o disminuye indicando un comportamiento volatil. Cuando hay cambios en el mercado que afectan el precio de la criptomonedas, estos son detectados con mayor oportunidad por TSF.

4.4. Entropía de Shannon

La entropía de Shannon es la parte central de la teoría de la información, y también se conoce como la medida de la incertidumbre. La entropía de Shannon $H(x)$ fue introducida con la teoría de la comunicación en 1948 por Shannon. Luego, en 2014, modificaron la fórmula original fue modificada, y se define como:

$$H(x) = -\frac{1}{n} \sum_{i=1}^n TSF(x_i) \log_2(TSF(x_i))$$

donde n es la longitud de la ventana deslizante, \log_2 es la función logarítmica de base 2 y TSF representa una de las características estadísticas tradicionales que fue seleccionada en la etapa anterior.

Una vez obtenida la TSF, se puede calcular la entropía de Shannon, lo que permite realzar las características obtenidas de las TSF. De esta manera, se puede elegir la TSF junto con la entropía de Shannon para observar una tendencia de aumento o disminución de los datos a lo largo del tiempo. Una vez que se ha identificado una de las TSF junto con la entropía de Shannon, procede pasar a la etapa de agrupación.

4.5. Agrupamiento de Características

El siguiente paso es agrupar los datos obtenidos de la entropía de Shannon junto con una de las TSF, con el fin de obtener las etiquetas necesarias para realizar el entrenamiento de la CNN. Para este paso se implementa el algoritmo *K-Means* para hacer la agrupación de los datos con características similares, de tal manera que se puedan etiquetar los diferentes grupos vender, comprar y riesgo. Este algoritmo se utiliza cuando no se tienen datos etiquetados, debido a que tiene como objetivo poder encontrar grupos de datos que son similares o tienen las mismas características. El algoritmo de K-Means ha sido utilizado en varios trabajos en la literatura para hacer la detección de anomalías o fraudes, debido a que es uno de los enfoques que se usan cuando se tienen datos de entrenamiento no etiquetados. La detección de anomalías se puede utilizar en desfalcos de dinero, en la mejora de la calidad de los datos, en el aumento de la producción, en la prevención de ataques a una red, en el uso de drogas en hospitales, series de tiempo, detección de intrusos, entre otros. Por esta razón se ha utilizado el algoritmo de K-Means, debido a que se pretende detectar comportamientos anormales en los precios de las criptomonedas y también el conjunto de datos de Bitcoin, Ethereum y Decred no contiene etiquetas. En esta fase se tiene la posibilidad de elegir el número de clases que se quiere etiquetar en el conjunto de datos dado. Se toma por defecto $k = 3$ grupos.



BITMONEY

El algoritmo *K-Means* fue propuesto en (Lloyd, 1982) y es uno de los más importantes algoritmos de clasificación no supervisado que permite agrupar datos en un número específico de grupos que tienen características similares. Estos grupos se llaman “*clusters*” y el número de *clusters* se define por K. Este algoritmo consiste en minimizar la suma de las distancias euclidianas de cada uno de los puntos con respecto al centroide del *cluster*. La fórmula de la distancia euclidiana se muestra en la Ecuación 1.

$$D(q, m) = \sqrt{\sum_{i=1}^n (q_i - m_i)^2}$$

donde $D(q, m)$ representa la distancia entre los puntos, q representa el primer punto y m es el segundo punto. En la Figura 2 se muestra un ejemplo del funcionamiento del algoritmo *K-Means*, donde se realiza la clasificación de tres diferentes grupos de datos. En la primera imagen (a) se muestran los datos sin procesar, en la segunda imagen (b) se muestra la primera iteración del algoritmo, donde se seleccionan los K centroides aleatoriamente y se empiezan a calcular las distancias euclidianas de cada uno de los puntos con respecto a cada uno de los centroides. Se puede observar que los datos corresponden a un *cluster* de acuerdo al centroide, donde se puede diferenciar por el color. En la tercera imagen (c) Se recalculan los centroides de los grupos obtenidos en el paso anterior, en donde se obtienen nuevamente las distancias euclidianas de cada uno de los puntos con respecto a los centroides y estos puntos se actualizan de acuerdo al *cluster* que pertenecen. Finalmente, en la cuarta imagen (d) se muestra la iteración final del algoritmo, donde los *clusters* convergen de manera exitosa al separar correctamente los datos.

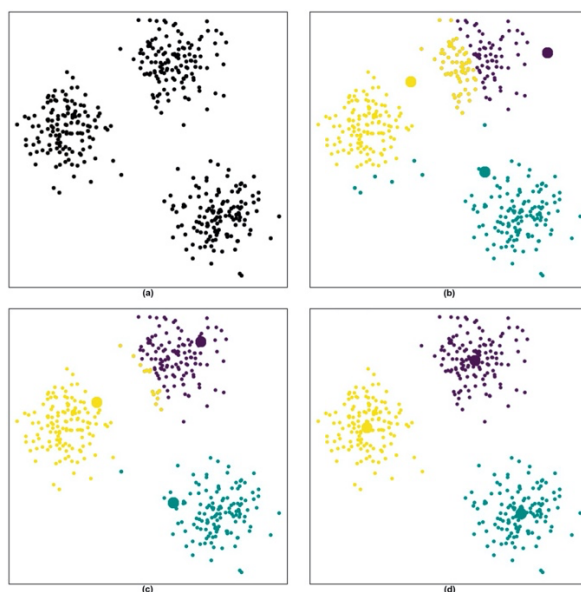


Figura 2. Algoritmo *K-Means* para separar tres grupos de datos en donde en la imagen (a) se muestran los datos sin ser procesados por el algoritmo. En la imagen (b) se observa la primera iteración del algoritmo *K-Means* donde los centroides se seleccionan de manera aleatoria. Por otra parte, en la imagen (c) se muestra la segunda iteración y se actualizan los centroides, y finalmente, en la imagen (d) se observa la última iteración donde los *clusters* están formados.

4.6. Transformación de Serie de Tiempo a Imagen

Los métodos tradicionales para las series de tiempo se basan en el análisis estadístico, los sistemas expertos de lógica difusa o los sistemas genéticos, entre otras aplicaciones. La extracción de características de series de tiempo sin procesar es una de las principales funciones de estos métodos, dado que una buena extracción de características tiene un gran impacto en los resultados. A diferencia de los métodos tradicionales, se realiza un método de preprocesamiento de datos que convierte las series de tiempo financieras en el dominio de tiempo en imágenes, con el fin de aprovechar las potentes herramientas de clasificación disponibles para el procesamiento de imágenes utilizando CNN, debido a que permiten extraer muchas más características en las imágenes. Además, convertir las series de tiempo sin procesar en imágenes proporciona una buena manera de explorar las características bidimensionales. Para una serie de tiempo sin procesar R con N puntos de muestra, la Figura 3 muestra el método de conversión a imágenes, donde cada punto de la serie de tiempo es un píxel de una imagen cuadrada en escala de grises con un tamaño de $M \times M$.

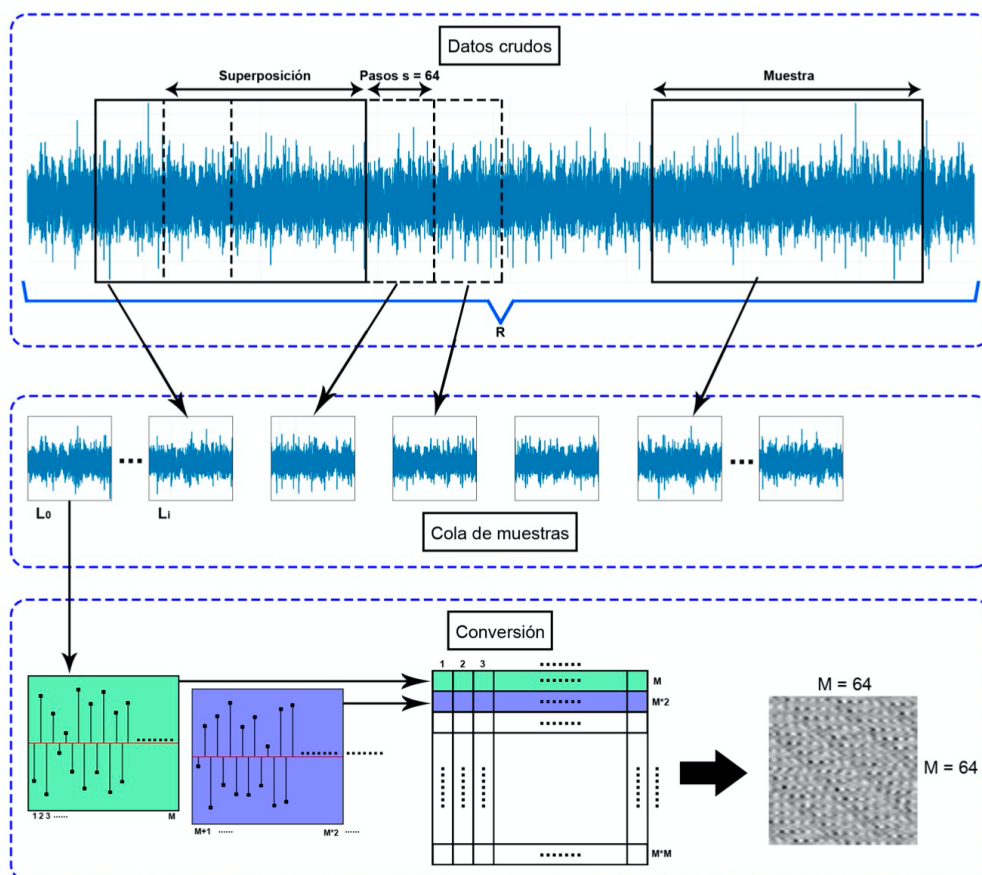
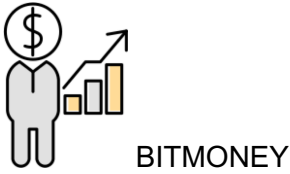


Figura 3. Método para convertir las señales sin procesar en imágenes. Primero, se toma una submuestra de la señal de tamaño $M \times M$ donde M representa la altura y el ancho total de la imagen cuadrada. Esta submuestra es entonces mapeada en una matriz y cada punto es normalizado en un rango de 0 a 255 para representar la intensidad de cada valor de píxel.



Primero una submuestra L de tamaño $M2$ es tomada de la serie de tiempo sin procesar R , por lo tanto, la i -ésima submuestra está dada por $L_i = \{R(i \cdot s + 1), R(i \cdot s + 2), \dots, R(i \cdot s + M2)\}$, donde $s \in \mathbb{Z}^+$ es el paso entre las submuestras, y el índice $i = \{0, 1, \dots, \lfloor N/s \rfloor\}$, y $\lfloor \cdot \rfloor$ denota la función piso. Tenga en cuenta que se llega a un importante solapamiento entre las muestras para obtener más imágenes, lo que es ventajoso para el clasificador de la

$$P_i(x, y) = \text{round} \left(\frac{L_i((x-1) * M + y) - \min(L_i)}{\max(L_i) - \min(L_i)} * 255 \right)$$

CNN, es decir, $s \ll M$. Entonces, cada punto de la submuestra L llena una matriz de $M \times M$ de izquierda a derecha y de arriba a abajo. Cada punto es normalizado de 0 a 255, y representa el valor de intensidad de cada píxel, con coordenadas x, y , de la imagen $P(x, y)$. Para la i -ésima imagen, este proceso se define de la siguiente manera:

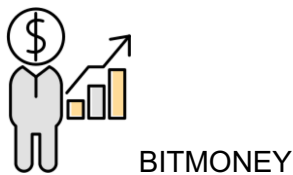
El tamaño y el número de imágenes pueden variar según la cantidad de datos de criptomonedas disponibles. Además, la complejidad de los cálculos también será proporcional al tamaño de las imágenes. De ahora en adelante, en caso de que la complejidad sea un problema, el tamaño de las imágenes debe ser reducido. Para nuestra propuesta se ha elegido un tamaño de 32×32 píxeles, con un paso de $s = 64$.

4.7. Red Neuronal Convolutacional - CNN

Las CNN son redes neuronales profundas que se centran principalmente en el procesamiento de imágenes y son excelentes para el reconocimiento de patrones. Además, son uno de los mejores métodos de clasificación. Las CNN obtienen automáticamente las características de las imágenes mediante filtros convolucionales, lo que las convierte en una herramienta con gran capacidad para aprender las características de forma robusta y sensible.

En las CNN existen tres tipos principales de capas: a) la capa convolutacional, b) la capa de submuestreo y c) la capa totalmente conectada (FC). La capa convolutacional sirve para adquirir mapas de características que se obtienen a través de un conjunto de filtros, mientras que la capa de submuestreo sirve para reducir el número de características de las entradas y la complejidad de los cálculos. Finalmente, la capa FC, que es una capa de una red neuronal normal donde cada píxel es considerado como una neurona, funciona para calcular las puntuaciones de cada una de las clases. Las capas FC tienen una función de pérdida como un clasificador SVM o *Softma*.

El modelo de la CNN que se propone para la tarea de clasificación se basa en la arquitectura del estado del arte llamada AlexNet, y se muestra en la Figura 4, donde se observa que la entrada de la CNN son imágenes de 64×64 de tamaño, posteriormente se aplica la primera convolución que se denota por C1, donde se aplican 96 filtros y se obtienen mapas de características de 32×32 de tamaño. Luego, se aplica la primera capa de submuestreo que se denota por S1, en donde se aplica un filtro a los 96 mapas de características y se reducen de tamaño a 16×16 . Así sucesivamente se aplican más capas de convolución y de submuestreo, y al final de la última capa de submuestreo se obtienen mapas de características de 4×4 de tamaño.



Finalmente, se aplican las capas FC de submuestreo con *maxpooling* (Maxpool) y tres capas totalmente conectadas (FC), donde la última es la capa de salida. Se puede observar que los tamaños de los filtros de cada capa se modificaron con respecto a la propuesta original de AlexNet, para poder tratar imágenes más pequeñas. Las características de las capas de convolución están denotadas de la manera $(5 \times 5 \times 96)$, donde las primeras dos cifras (5×5) representan el tamaño del filtro que se aplica y la tercer cifra (96) indica la cantidad de filtros que se obtendrán. Mientras que las características de las capas de submuestreo (Maxpool) se muestran como (2×2) , indicando el tamaño del filtro que se aplica. Finalmente, las características de las FC se muestran como n que indica el número de neuronas y x el número de clases. Una vez que las señales han sido convertidas en imágenes, se puede realizar el entrenamiento de la CNN para clasificar el los tres grupos (comprar, vender, riesgo).

Las principales modificaciones de la arquitectura de AlexNet se hicieron para tratar imágenes más pequeñas, lo que resultó en una reducción del tamaño de los filtros utilizados para cada capa. Por lo tanto, en la primera capa convolucional el tamaño del filtro se redujo de 11×11 a 5×5 ; también el tamaño del filtro de la segunda capa convolucional se redujo de 5×5 a 3×3 ; mientras que las capas de *maxpooling* se redujeron de 3×3 a 2×2 . Además, se añadió una nueva capa de *maxpooling* entre la tercera y cuarta capa convolucional. El uso de imágenes más pequeñas es conveniente porque permite obtener más imágenes de los conjuntos de datos disponibles, lo que es favorable para el algoritmo de entrenamiento. También, las imágenes más pequeñas reducen significativamente el costo computacional de los algoritmos de entrenamiento. Además, en las dos primeras capas FC, se varió el número de neuronas, buscando la mejor configuración para nuestro caso particular, utilizando valores de 512 a 3584 neuronas, en la primera capa FC, y de 0 a 1024 en la segunda. Finalmente, la última capa sólo fue cambiada de acuerdo al número de clases etiquetadas, siete en nuestro caso. En la Figura 4, se presenta un ejemplo con imágenes de 64×64 . Esta arquitectura se tuvo que adaptarse a imágenes de 32×32 , por la cantidad de datos que tenemos en las criptomonedas.

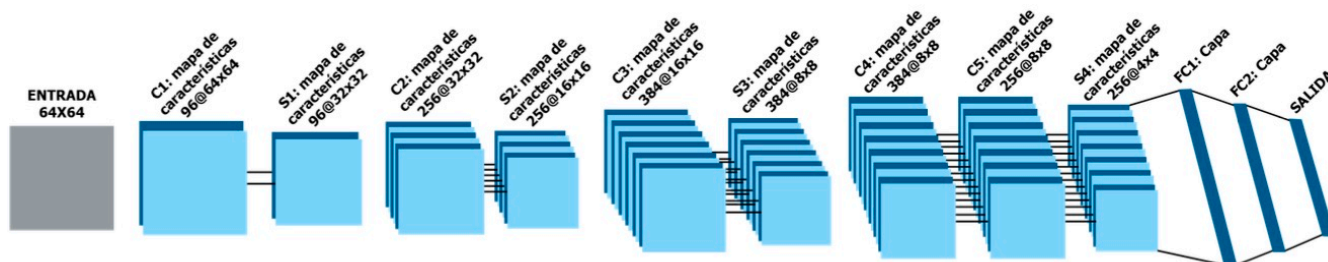


Figura 4. Propuesta de arquitectura basada en AlexNet para la estimación del desgaste de los rodamientos. Se cambió el tamaño de los filtros en la primera y segunda capa convolucional, además de añadirse una capa de submuestreo de *maxpooling* entre la tercera y cuarta capa convolucional. Finalmente las capas de submuestreo fueron redimensionadas y las capas totalmente conectadas cambiaron el número de neuronas con respecto a la propuesta original de AlexNet.