Bitcoin Price Prediction Using Data Mining

What is Bitcoin?

Bitcoin is the first cryptocurrency. It is an open-source virtual or digital currency designed to work as a medium of exchange secured by cryptography. It maintains a public ledger to manage transactions. The cryptography concept is used to achieve authentication. Its most prominent feature is that it is entirely decentralized, i.e., not regulated or controlled by any central authority. Unlike regular currency, it is created, distributed, traded, and stored as a decentralized ledger system known as a blockchain. There is no physical existence of bitcoin. It is digital and is maintained by many computers or nodes that maintain its code and blockchain, which is a collection of blocks representing each transaction. The ledgers, also called miners, maintain bitcoin transactions are paid with new bitcoins for each block of transactions verified by them, which keeps the market going and new bitcoins being generated.

Why Bitcoin?

Since its launch in 2009, Bitcoin has gained immense popularity and is accepted as a legal currency in many countries, including the US. What attracts people towards bitcoin are the features that it provides, which cannot be given by regular currency. Firstly, all bitcoin transactions are transparent, so there is no scope for anyone to cheat. Secondly, Bitcoin is highly secured. In May 2020, it had 47,000 nodes, and the number is still growing. If someone wants to attack Bitcoin, they need to have at least 51% of the computation power that makes up Bitcoin, which is seemingly impossible. Also, in case of an attack, people could fork to a new blockchain making the attack fail. Also, bitcoins provide significantly fewer transaction rates compared to traditional internet banking options. All this makes Bitcoin a popular choice among people for transactions and investors.

Applications of bitcoin

The Bitcoin market finds many applications with increasing popularity.

- It allows for easy and cheap international payments. This is because Bitcoin is not related to any country of government regulation.
- More number of vendors and service providers are allowing bitcoin transactions. Thus, people can shop through bitcoins just with bitcoin payment apps on their phones.
- Unlike traditional internet banking options, it does not require one to give their personal information.
- No permission from any authority is required for the transactions.

- People don't have to pay extra transaction fees like that in standard internet banking.
- It is secured by powerful cryptographic algorithms.
- It has a huge potential for investments, as the market is rapidly growing and gaining popularity. This creates the need for systems that can predict bitcoin prices with the help of past data.

The need for Price Prediction

The Bitcoin market is massively growing. It was valued at 293.66 million US dollars in 2019. Due to its almost zero risk of inflation and all the factors discussed above, the market is trending, and people are investing in it. That's why there is a requirement of analysis of bitcoin price trends and value prediction so that investors can be benefited in making the right decisions about investing in bitcoins.

Proposed method

In the project, we propose to use data mining for Bitcoin price prediction. Starting with the required pre-steps like data cleaning, we suggest using several techniques then predicting the prices using them. We shall then do a comparative analysis of the results to understand which approach suits the task best. We expect that some method or the other would give us satisfactory results. If not, we shall explore if we can use some other way for the prediction.

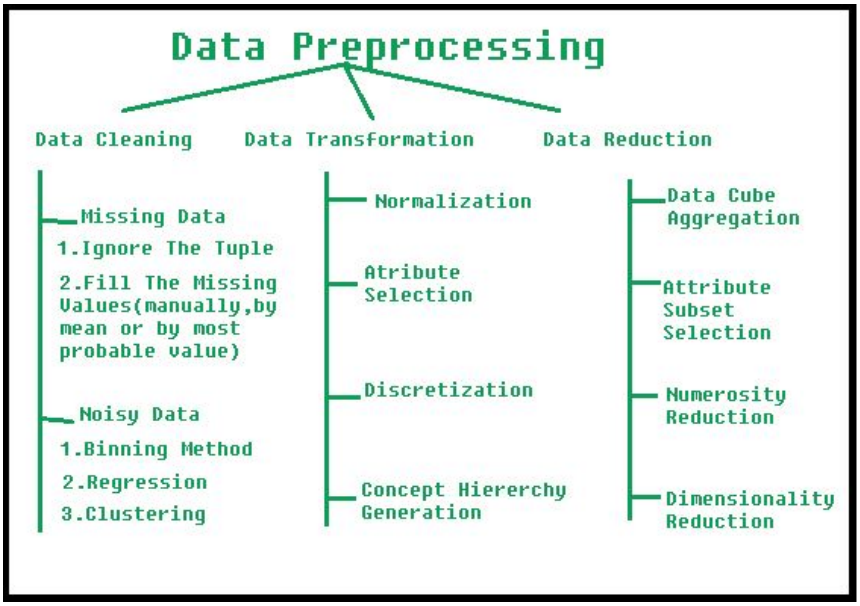
This would give us an understanding of the data mining process, along with the knowledge of the different techniques that can be used. In the end, we should be able to have a comparative understanding of the different algorithms and also appreciate the process of data mining with hands-on experience.

Identifying the dataset

The first step for the project would be to identify and get a suitable dataset. The characteristics that we should note while looking for a dataset are accuracy and precision of the data and its reliability, consistency, timeliness, completeness, and comprehensiveness.

Data pre-processing

To be able to use the data for predicting results, it is crucial to convert the raw data into a usable and efficient form. After identifying the dataset, this is the next step to be performed.



This would involve the following steps:

1. Data Cleaning: We have to deal with missing and irrelevant data using methods like filling the missing values with mean, ignoring some tuples, binning, regression, or clustering.
2. Data Transformation: Then, to make the data into an appropriate form that is suitable for processing, we need to perform the transformation. It can be done using Normalization, attribute selection, discretization, and concept hierarchy generation techniques.
3. Data Reduction: In case the data amount is enormous, it would require a lot of storage and analysis costs. To reduce this, we might have to use data reduction. This can be done using data cube aggregation, attribute subset selection, numerosity reduction, or dimensionality reduction.

Proposed Algorithms:

A. Linear regression model :

Linear regression is a linear approach for modeling the relationship between a dependent variable and independent variables, represented by the main equation:

$y = b_0 + \bar{b}_1 \cdot \bar{x}_1 + \bar{b}_2 \cdot \bar{x}_2 + \bar{b}_3 \cdot \bar{x}_3 + ...$

where y is dependent variable and $\bar{x}_1, \bar{x}_2, \bar{x}_3, ...$ are independent variables, while b_0 is the intercept and $\bar{b}_1, \bar{b}_2, \bar{b}_3, \ldots$ are the vectors of coefficients. This algorithm aims to find the curve that best fits the data, which best describes the relationship between the dependent and independent variables. The algorithm finds the best fitting curve plotting

all the possible trend curves through our data and for each of them calculates and stores the amount $(y-\overline{y})^2$, and then choose the one that minimizes the sum of the squared differences $\sum_i (y_i-\overline{y}i)^2$, namely the curve that minimizes the distance between the real points and those crossed by the curve of best fit.

B. K-Nearest Neighbour :

K-nearest neighbour algorithm creates k groups/clusters of data points available to us based on the similarity between them and classifies unknown data points using distance functions.

Between two dependent and independent variables, we can compute the distance between them using some distance function D(x,y), where x,y are composed of several features, such that x={x1,…,xN}, y={y1,…,yN}.

Following is the pseudo-code for implementing the KNN algorithm:

- Load the data.
- Initialize k.
- Iterate over training data points.
    1. Using distance function, calculate the distance between each row of training data and test data. There are various methods for calculating distance. The most common is the Euclidean distance.
    2. Sort obtained distances in ascending order.
    3. From the top k rows, find the most frequent class among these.
    4. The most frequent class will be the predicted class. Return predicted class.

C. Naive Bayes :

The algorithm is mainly based on the Bayes Theorem of probability. The Bayes theorem is based on the conditional probability that the probability of occurrence of an event A given that an event B has already occurred is:

P(A|B)= $\frac{P(B|A)P(A)}{P(B)}$ P(y|X)= $\frac{P(X|y)P(y)}{P(X)}$

Here we assume that the features are independent, and all the features have an equal effect on the outcome. We are going to classify the features given their properties.

The variable y is the class variable, which represents the class given the conditions. Variable X represents the features. The variable X is described as a set of n rows from x1,x2,.....,xn. We can obtain the values for each by looking at the dataset and substitute them into the equation. For all entries in the dataset, the denominator does not change. It remains static. Therefore, the denominator can be removed, and proportionality can be introduced.

D. Support Vector Machines:

It uses a supervised learning-based algorithm. Support Vector machines use margins, which is the distance between the closest data points and the hyperplane. All the data-points inside the margins or on the edge of margin are called Support Vectors. SVM starts with data in a relatively low dimension, moves the data into a higher dimension, and then finds the support vector classifier that separates the higher dimensional data. To decide the higher dimension for data, SVM uses Kernel functions. The trick is that Kernel functions only calculate the relationship between every pair of data points using dot products and pretending that they are in higher dimensions; they don't actually change the dimension. This trick is called the Kernel trick. It reduces the amount of computation by avoiding the actual transformation. The kernel simply computes the dot product of two vectors x and y in some (very high dimensional) features.

$$\text{maximize}_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(X_i^T \cdot X_j)$$

$$0 <= \alpha_i <= C \text{ for all } i=1,2,3,..,n \text{ and } \sum_{i=1}^{n} \alpha_i y_i = 0$$

There are three types of Kernels: i)Linear ii)Polynomial and iii)RBF

E. Recurrent Neural Networks (RNN):

A Recurrent Neural Network is a type of neural network where every node has a connection between them, and these nodes are connected along a temporal sequence. Temporal here refers to time-dependent data. RNN is preferred over traditional neural networks because they can only consider the current input state. They can't handle or implement sequential inputs as they don't have a memory to store previous states. On the other hand, RNN is recurring in nature, and they have loops for considering previous states.

RNN first converts independent activations into dependent and assigns the same weights and bias to all the layers, which provides previous output as input to the next layer and hence provides memory to previous outputs. These three layers combined together to form a single recurrent unit.

- Formula for calculating current state:

  $h_t = f(h_{t-1}, x_t)$

  where,

  h_t - > current state

  h_{t-1} - > previous state

  x_t - > input state

- Formula for applying activation function:

$h_t = \tanh (W_{hh}h_{t-1} + W_{xh}x_t)$
where,
w_{hh} - > weight at recurrent neuron
w_{xh} - > weight at input neuron

- Formula for calculating output:
$Y_t = W_{hy}h_t$
where,
y_t - > output
W_{hy} - > weight at output layer

F. Convolutional Neural Networks:

Generally, Convolutional Neural Networks(CNN) are used for image classification, but on the basis of recent research papers, it is shown that CNN also helps in dealing with sequential data. So, for this project, we will develop a CNN-based prediction model.CNN consists of a sequence of layers which are Input Layer, Convolution Layer, Activation Function Layer(e.g. RELU), Pool Layer, and Fully connected Layer.

Input Layer: holds the raw input.

Convolution Layer: calculates dot product to compute output volume.

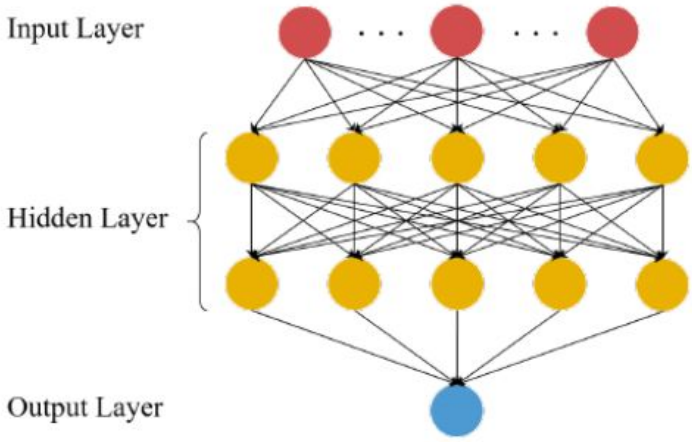
Activation Function Layer: applies an activation function to the output of the convolution layer.

Pool Layer: reduces the size of the volume to make computation fast, reduces memory, and removes overfitting.

Fully connected Layer: connects every neuron of adjacent layers. It returns a 1D array of class scores which is computed from the inputs of the previous layer.

G. Deep Neural Networks:

Deep Neural Networks usually consists of an input layer, several hidden layers, and an output layer, as shown in Figure below:



In our case, the number of nodes in the input layer will be equal to the number of features, which means each node will correspond to each feature. Also, if we analyze m days at a time, then each input node will take a column of size m as input. Also, our deep neural network will be fully connected, which means that every node in each layer will be connected to each node in the next layer. That is, each node in the hidden layer takes input vectors x_1, x_2, ….., x_n from the previous layer, where n is the number of nodes in the previous layer and the size of each vector is m. Its output h is then defined as

$h = \sigma_h(\omega_1 \odot x_1 + \ldots. + \omega_n \odot x_n + b)$

where weight vectors ω_1, …. , ω_n and the bias b are the hyperparameters, $\odot$ is the element-wise multiplication and σ_h is a nonlinear function like the logistic sigmoid.

In the output layer, the set of input vectors will be converted to a single vector and then converted to a single value using dot product between the converted single vector and a weight vector and this single value y will be used as the predicted bitcoin price.

H. Auto-Regressive Integrated Moving Average(ARIMA):

ARIMA model is used for time series analysis and predictions. This model is used on time series data which will be transformed into a stationary time series, and the predictions will be linear regression upon features including time differences and moving averages. In ARIMA, the data we use is the difference which means that price features are transformed into the difference between prices.

p: number of autoregressive terms.
d: number of non-seasonal differences needed for stationary
q: number of lagged forecast errors in the prediction equation

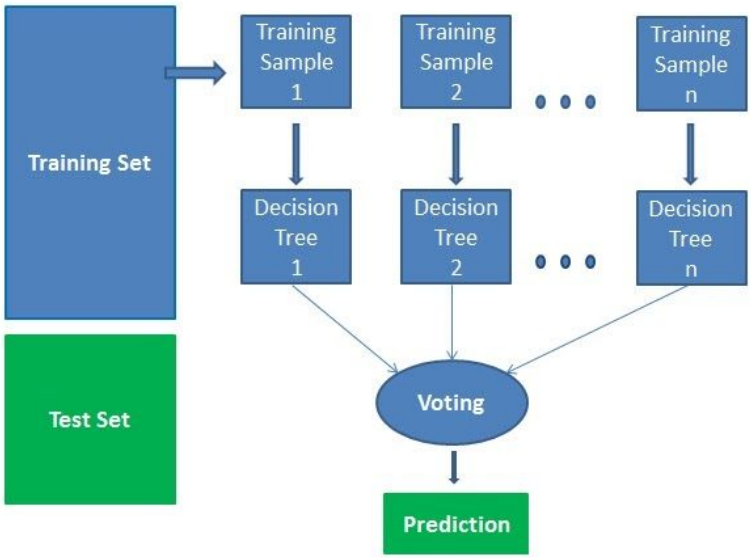
$$(1- \sum_{k=1}^{p} \alpha_k L^k)\ (1-L)^d\ X_t = (1- \sum_{k=1}^{q} \beta_k L^k)\ \varepsilon_t$$

In the above equation, L is the lag operator, and p,d,q are the hyperparameters over which we will optimize. At each time t, we will train a model using the history of price to predict the price at time t and use the sign of the change in price as a prediction.

I. Random Forest:

Random forest is a supervised learning algorithm. It can be used both for classification and regression. Random forests create decision trees on randomly selected data samples, get a prediction from each tree, and select the best solution through voting.
The algorithm works in four steps:

1. Select random samples from a given dataset.
2. Construct a decision tree for each sample and get a prediction result from each decision tree.
3. Perform a vote for each predicted result.
4. Select the prediction result with the most votes as the final prediction.



J. Gradient boosting Algorithm:

Boosting is a method of converting weak learners into strong learners. In boosting, each new tree is a fit on a modified version of the original data set. Gradient boosting involves three elements:

1. A loss function to be optimized.
2. A weak learner to make predictions.
3. An additive model to add weak learners to minimize the loss function.

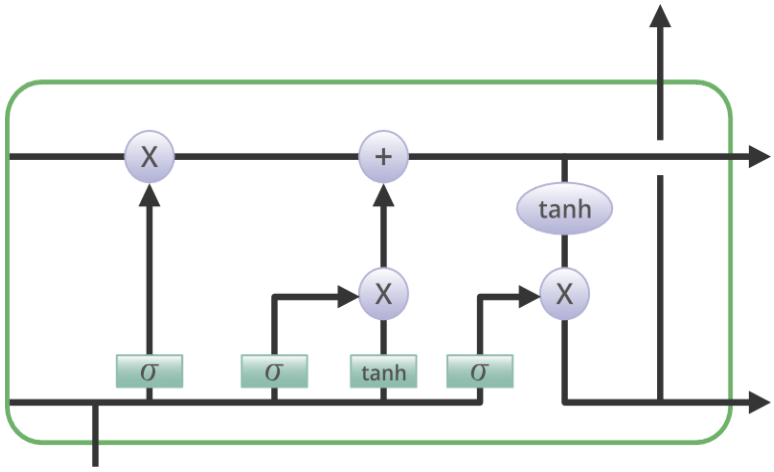
Gradient boosting is a greedy algorithm and can overfit a training dataset quickly. It can benefit from regularization methods that penalize various parts of the algorithm and

generally improve the performance of the algorithm by reducing overfitting. We can use four enhancements to basic gradient boosting:

1. Tree Constraints
2. Shrinkage
3. Random Sampling
4. Panelized learning

K. Long-Short Term Memory (LSTM):

Long-Short Term Memory (LSTMs) is a type of Recurrent Neural Networks(RNNs). LSTM solves the problem of long term dependencies in RNN as RNN can give more accurate information only from the recently stored information.LSTM is designed to store information for an extended period through memory blocks called cells. It consists of four neural networks in a chain pattern and different cells.

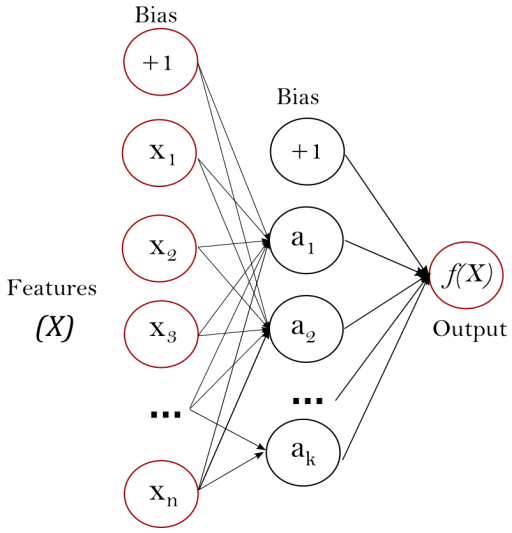


Three types of gates are used for memory manipulations which are Forget gate, Input gate, and Output gate. Forget gate removes the information which is no longer useful. The input gate adds valuable information to the cell state and the output gate extracts valuable information from cells which are considered as output.

L. Multilayer Perceptron (MLP):

Multi-layer perceptron(MLP) is based on Neural networks. Unlike logistic regression, between input and output layers, there are many hidden layers, and each layer is fully connected with its adjacent layers. MLP classifier trains using backpropagation, i.e. it uses gradients for training which are calculated using backpropagation. For multi-class classification, it uses Softmax as an output function. Softmax maps the non-normalized output of a network to a probability distribution over predicted output classes. The advantage of MLP is that it can learn models in linear time and can learn non-linear models very efficiently while the cons of using MLP are that it has to tune a no. of

hyperparameters and it is susceptible to feature scaling. The following figure shows one layer:



Expected results

After performing the different algorithms on the processed data, we expect to arrive at the Bitcoin price predictions. Certain algorithms will give better results than others. We shall do a comparative analysis to understand the outputs generated by the different algorithms and their functioning. We shall produce these results in both numeric formats and pictorial for clear and better visualization of the results. We expect that on completion of the project, we would be able to have a good understanding of the data mining process and the various techniques that we use in the project.