BITCOIN PRICE DETECTION WITH PYSPARK USING RANDOM FOREST

Yakup Görür

Department of Computer Science
Ozyegin University
yakup.gorur@ozu.edu.tr

ABSTRACT

Cryptocurrencies are digital currencies that have garnered significant investor attention in the financial markets. The aim of this project is to predict the daily price, particularly the daily closing price of the cryptocurrency Bitcoin. This plays a vital role in making trading decisions. There exist various factors which affect the price of Bitcoin, thereby making price prediction a complex and technically challenging task. To perform prediction, random forest model was trained on the historical time series which is the past prices of Bitcoin over several years. Features such as the opening price, highest price, lowest price, closing price, volume of Bitcoin, volume of currencies, and weighted price were taken into consideration so as to predict the closing price of

the next day. Random forest model designed and implemented on both of pyspark and scikit learn frameworks to build predictive analysis and evaluated them by computing various measures such as the RMSE (root mean square error) and r (Pearson's correlation coefficient) on test data. Pyspark framework was used to make parallelize the creating trees when training the random forest to handle bigdata.

Code has been made available at:

https://github.com/ykpgrr/Price-Prediction-with-Random-Forest

***Index Terms*—** Bigdata, Random-Forest, Bitcoin, Pyspark, Time series, Predictive Analysis

1. INTRODUCTION

A. Motivation

Bitcoin (BTC) [1] is a novel digital currency system which functions without central governing authority. Instead, payments are processed by a peer-to-peer network of users connected through the Internet. Bitcoin users announce new transactions on this network, which are verified by network nodes and recorded in a public distributed ledger called the blockchain. Bitcoin is the largest of its kind in terms of total market capitalization value. They are created as a reward in a competition in which users offer their computing power to verify and record transactions into the blockchain. Bitcoins can also be exchanged for other currencies, products, and services. The exchange of the Bitcoins with other currencies is done on the exchange office, where "buy" or "sell" orders are stored on the order book. "Buy" or "bid" offers represent an intention to buy certain amount of Bitcoins at some price while "sell" or "ask" offers represent an intention to sell certain amount of Bitcoins at some price. The exchange is done by matching orders by price from order book into a valid trade transaction between buyer and seller. Cryptocurrency has a total market cap of around $600 billion USD by the end of December 2017 with Bitcoin having a market cap of around $300 billion USD share in the cryptocurrency market [2]. Not only the investors but also brokers and private investors are finding cryptocurrency as an investment tool. In this regard, it is very much necessary to predict the future values of cryptocurrency so as to take correct trading decisions.

B. Goal and Technical Objective

The goal of this project is to predict the closing price of Bitcoin on a given day based on the Bitcoin data. This is basically a time series prediction problem. If the bitcoin data is considered from 2008 to 2019, it makes 468 weeks, 3.285 days, 78.840 hours, 4.730.400 minutes. If the total markets which are selling and buying bitcoin are considered we have to multiply these values with 27.000. There is not only Bitcoin in cryptocurrency world, there are also nearly 5.000 other cryptocurrency coins. That makes nearly 60.000 weekly data, 440 billion daily data, 10 trillion hourly data and finally 600 trillion minutely data.

In order to handle these huge data, spark environment is chosen to parallelize prediction. In this work, data is chosen as daily bitcoin prices. In order to make the one day ahead

prediction of closing price of Bitcoin, features such as the opening price, highest price, lowest price, closing price, volume of Bitcoin, volume of currencies, and weighted price are taken into consideration. To predict the closing price on a day, the random forest

model is trained with data over the past historical data and it is tested over the next quarter. Model accuracy is presented as RMSE (root mean square error) and r (Pearson's correlation coefficient).

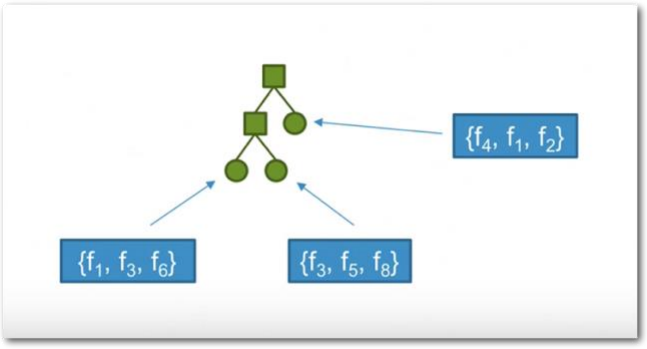

Fig. 1. Random Forest

2. BACKGROUND STUDY AND RELATED WORK

A. *Literature Survey*

Various approaches have been used in the past to carry out the price prediction task. There are mainly two sets of literature that are highly relevant to this work. One is financial data analysis; the other, time series data analysis

Financial Data Analysis

Several approaches are described in the literature including, one called technical analysis also known as "charting" that forecasts future prices [3]. According to it, stock market prices do not follow random walks, that is – the price movements follow a set of patterns. These price movements can be used to predict the future price [4]. There exist some other empirically designed patterns such as heads-and-shoulders, double-top-and bottom that can be used to predict future prices.

Time Series Data Analysis

In the context of future price predictions, classical methods are quite popular. Autoregressive integrated moving average (ARIMA) models are a popular choice for forecasting over a short term. It works very well when the data exhibits consistent or stable pattern over time with least possible outliers. The ARIMA methodology works well only when the data exhibits "stationarity", which means that the series remains almost constant. But this is not always possible in the real time scenario, where the data fluctuates drastically, and it is highly volatile. Ediger and Akar used the seasonal ARIMA model to estimate the future fuel energy demand in Turkey over certain years [5].

However, this scenario is not guaranteed to work for unseasonal or non-linear data. To solve the real time prediction problems on big data, random forest model is very much useful to increase the speed of computation due to its ability to handle nonlinearities in the data.

B. *Methods*

In this project, random forest model is trained to predict the closing price on next day. At a high level, random forests are collections of decision trees used for classification and regression tasks. Once grown, each decision tree classifies an unlabeled point by casting a vote, and the random forest reports the label or value with the most votes [6].

A random forest is a type of ensemble model, which averages the predictions of many different "reasonably good" models to produce a prediction that better estimates the true hypothesis. Ensemble models are highly successful as machine learning tools, because they avoid the chance-dependent pitfalls of many singular models. For example, gradient descent methods can get stuck in local minima, but combining many models increases the chance that one will find the global minimum. Alternatively, even if none of the models in the ensemble produce the true hypothesis, averaging every prediction can lead to a prediction that more closely matches the underlying truth [7].

For this project, I used the Spark Machine Learning (ML) implementation of random forest regressing using python programming language with pyspark framework. Apache Spark is a scalable data processing system that provides an engine for processing big data workloads. At its core is a structure called the resilient distributed dataset (RDD), which can be distributed over a cluster of machines and is fault-tolerant. A series of libraries run on Spark and take advantage of its cluster-computing capabilities. One such library is MLlib, which contains a wide array of machine learning tools. Random Forests in spark use a different subsample of the data to train each tree. Instead of replicating data explicitly, we save memory by using a TreePoint structure which stores the number of replicas of each instance in each subsample. This can be seen in figure-

Random forest also can be summarized in the perspectives of Map-Reduce step by step like that:

1. Read train data from a CSV file.

2. For n Trees call n mappers

3. Create 90% subset of the training data with replacement

4. When mapper is running it takes these data

5. After receiving data, each mapper starts to build tree and produce prediction for test dataset.

6. Pass the test data and label as key and value to Reducer

7. Reducer counts the majority label according to key.

8. Write results to output file.

This representation can be seen at the figure-2.

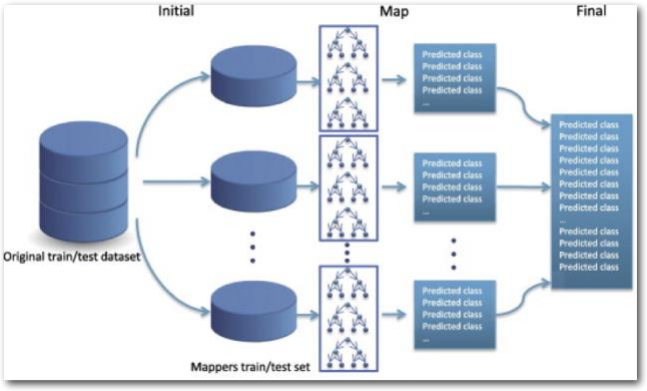

Fig. 2. Random Forest Map-Reduce

3. DATASET AND REPRESENTATION

Several Bitcoin data sets are available online to download for free. Most of them provide the data related to price of Bitcoin on a day to day basis. So, in this project daily prediction is chosen. However, the goal of this project is to make prediction of closing price of Bitcoin with using bigdata representation. to work on larger data, there are lots of available source to download with extra payment. In this project Kaggle Bitcoin Historical Data is used. The dataset is starting from January 2012 to 2018 (present).

This dataset gives access to Bitcoin exchanges and daily Bitcoin values. These values:

OPEN PRICE: The open represents the first price traded during the candlestick.

HIGH PRICE: The high is the highest price traded during the candlestick.

LOW: The low shows the lowest price traded during the candlestick.

CLOSE: The close is the last price traded during the candlestick.

Volume (BTC): Volume, in BTC traded in the stock market during a given measurement interval.

Volume (Currency): Volume, in USD, traded on stock market during a given measurement interval.

Weighted Price: Measure of the average price.

To predict the closing price of Bitcoin one day ahead, in each of the day data sets, columns close are shifted up by one (1) unit. And also, the features column is shifted down by seven (7) unit to use 7-days historical values for prediction. The data representation can be seen at figure-3.

	Open	High	Low	Close	Volume_(BTC)	Volume_(Currency)	Weighted_Price	Open_b_1	High_b_1	Low_b_1	...
Timestamp											
2014-12-08 00:00:00+00:00	378.00	378.00	378.0	378.0	0.000	0.0000	0.000000	378.00	378.00	378.0	...
2014-12-09 00:00:00+00:00	375.01	375.01	375.0	375.0	0.235	88.1251	375.000426	378.00	378.00	378.0	...
2014-12-10 00:00:00+00:00	398.00	398.00	398.0	398.0	0.010	3.9800	398.000000	375.01	375.01	375.0	...

Fig 3. Dataset

To calculate model score, the dataset is setting in two parts as training data and test data. Percentage of test data over all data is nearly 10%. The dataset representation as training and test sets can be seen at figure-4.

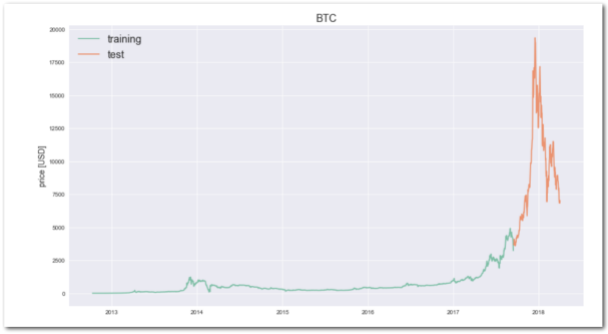

Fig 4. Test and Training Set

4. RESULT

A. Performance Metrics

RMSE (Root Mean Square Error)

Root mean square error (RMSE) is a frequently used measure of the differences between values (sample or population values) predicted by a model or an estimator and the values observed. RMSE is a measure of how spread out prediction errors are. It can be formulated as follows:

$$RMSErrors = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$$

In the above formula, n is the number of exemplars in the data set.

r (Pearson's Correlation Coefficient)

The Pearson's correlation coefficient depicts the linear association between two variables. It helps to figure if two sets of data move in the same direction. It is denoted by r. It can take the values from -1 to +1. If X is the network output and D is the desired output, then r is given by:

$$r = \frac{\sum_i (xi - \bar{x})(di - \bar{d})}{\sqrt{\sum_i \frac{(di - \bar{d})}{N}} \sqrt{\sum_i \frac{(xi - \bar{x})^2}{N}}}$$

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} xi \quad \text{and} \quad \bar{d} = \frac{1}{N}\sum_{i=1}^{N} di$$

If r = 0, it indicates there is no correlation between X and D

If r > 0, it indicates there is positive association between X and D i.e if value of one variable increases, the other variable increases.

If r < 0, it indicates there is negative association between X and D i.e if value of one variable increases, the other variable decreases.

The following diagram shows the possible correlation between two variables.

Fig 5. Pearson's correlation coefficient

B. Experimental Result

The result of random forest model trained on data set from 2012 to 2018 (starting from January 2012 to June 2018) and tested on second quarter of 2018(June 2018 – December 2018). Model use window size 7 and 1000 epochs during training and testing.

The illustration of result can be seen figure-6.

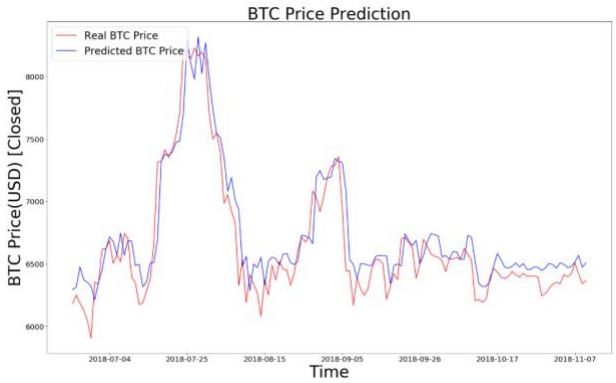

Fig 6. Result

Root Mean Square Error: 193

r: 0.822

3. CONCLUSION

This work presents an application of random forest model using pyspark framework for making one day ahead prediction of closing price of cryptocurrency Bitcoin. It is observed that bitcoin market is not stable. Since the Bitcoin market is open to manipulation, price of bitcoin may be hard decrease or hard increase. It may be necessary to provide data that the model can predict manipulations to better capture these ups and downs. Examples of these data are tweets about bitcoin, bitcoin news, and telegram bitcoin groups.

12. REFERENCES

[1] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. (2008). http://bitcoin.org/bitcoin.pdf

[2] Amjad, M. J. & Shah, D. (2017). "Trading Bitcoin and Online Time Series Prediction".Proceedings of the Time Series Workshop at NIPS 2016, (pp. PMLR 55:1-15).

[3] Areekul, P., Senjyu, T., Urasaki, N., & Yona, A. (2010). Next day price forecasting in deregulated market by combination of Artificial Neural Network and ARIMA time series models. 2010 5th IEEE Conference on Industrial Electronics and Applications. doi:10.1109/iciea.2010.5514828

[4] Lo, A. W. & MacKinlay, A. C. (1988). Stock market prices do not follow random walks: Evidence from a simple specification test. The Review of Financial Studies, Volume 1, 41–66.

[5] Areekul, P., Senjyu, T., Urasaki, N., & Yona, A. (2010). Next day price forecasting in deregulated market by combination of Artificial Neural Network and ARIMA time series models. 2010 5th IEEE Conference on Industrial Electronics and Applications. doi:10.1109/iciea.2010.5514828

[6] Leo Breiman Statistics and Leo Breiman. Random forests. pages 5–32, 2001.

[7] Thomas G. Dietterich. Ensemble methods in machine learning. In Proceedings of the First International Workshop on Multiple Classifier Systems, MCS '00, pages 1–15, London, UK, UK, 2000. Springer-Verlag.