2018-1 CSI4117-01
Data Mining

Research Report

Final

Bitcoin price prediction
using Market Indicators & Google Trend

team Nineteen

2011272071 JoonHa Jung
2014154009 Yoojin Kim
2014147588 Jihwan Kim

Index

Keywords

Bitcoin price, Supervised Learning, Unsupervised Learning, PCA, Classification, Linear Regression, Logistic Regression, Support Vector Machine, K-Nearest Neighbor Regression, Recurrent Neural Network

0. Abstract

Since crypto currency has been popular for nowadays despite its instability, researches about predicting its price became important. While there are so many methods for prediction, we developed and found the most outperforming model for bitcoin price prediction in this research. We used bitcoin market indicators and google trend data for our study, and some data preprocessing methods were used. Linear Regression, Logistic Regression, Support Vector Machine, K-nearest neighbor regression, and Recurrent Neural Network(RNN) were used for training model. For regression methods R-square score is used for measuring performance, and RNN seems to be the most outperforming method.

1. Introduction

Problem Definition & Significance

For nowadays, crypto currency is taking center stage under the international spotlight despite its instability. Among many kinds of crypto currencies, the first generation of these, Bitcoin, is the most well known and wide spread currency. Due to its unstable nature, it is harder to predict than usual stock prices. However, since most of people have an interest on this currency, predicting Bitcoin price became popular subject for researchers including computer scientists. By this reason, considerable amount of researches has been done to predict its price but still there are some arguments that which method is the best for prediction.

In this research, we focused on developing and finding the most outperforming model for bitcoin price prediction. We implemented Linear Regression, Logistic Regression, Support Vector Machine, K-nearest neighbor regression, and finally Recurrent Neural Network(RNN). We used Bitcoin price data which has 14 market indices attributes, and Google trend data in recent period, and also perform some data preprocessing methods for more accuracy.

2. Related works

We referred several works for prediction and data exploration. *Stock price forecasting using back propagation neural networks with time and profit based adjusted weight factors* (Khoa & Sakakibara, 2006) suggest the method for using back propagation network to predict stock price. *Using bitcoin ledger network data to predict the price of Bitcoin* (Mern & Anderson) suggest using bitcoin ledger network data, and introduce about ledger network dataset. *An Empirical Study on Modeling and*

Prediction of Bitcoin Prices With Bayesian Neural Networks Based on Blockchain Information (Jang, H., & Lee, J., 2018) introduce modeling and prediction technique of bitcoin prices using Bayesian neural networks based on block chain informations. *Conditional Time Series Forecasting with Convolutional Neural Networks* (Borovykh, A., Bohte, S. & Oosterlee, C. W., 2017) suggests CNN for prediction. *Bitcoin Spread Prediction Using Social And Web Search Media* (Matta, Martina & Lunesu, Ilaria & Marchesi, Michele. 2015) gave us an idea for using trend data in prediction.

3. Method

Entire code used for this project is here

: https://github.com/jooncco/bitcoin_price/blob/master/source.ipynb

Data

1. Data

Used data is from kaggle and has 15 attributes for each daily price data. Total attributes are same as below.

(Date, BTC_Price, DOW_Price, NASDAQ_Price, GOLD_Price, USD_Price,

US_BCI, US_CCI, US2TB, US30TB, ETH, XRP, US_INTR, S&P, BTCCash, LTC, Gsearch)

Leftmost column, Date, is used for only data separation and sorting. BTC_Price is our prediction value so used as Y value. Other 15 columns are attributes for exploration and prediction. Each features' description is as follows.

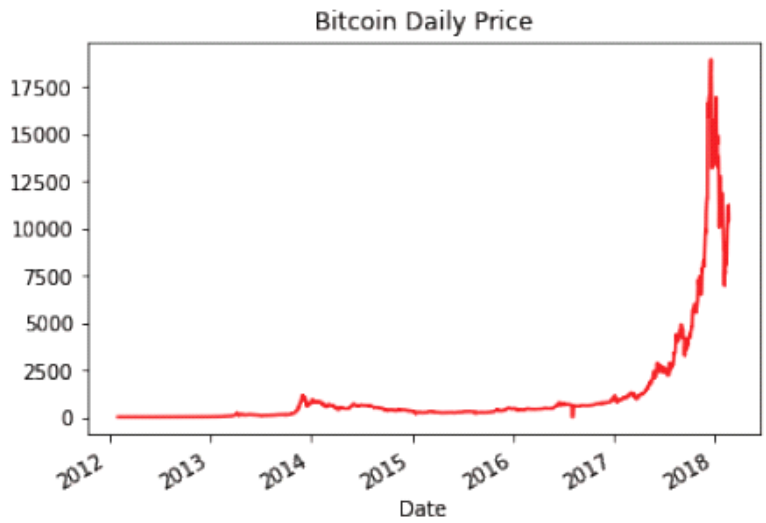
DOW_Price	Dow Jones Industrial Index
NASDAQ_Price	North American Automated stock market price
GOLD_Price	Gold(comex)
USD_Price	USD basket Index
US_BCI	US Business Confidence Index
US_CCI	US Consumer Confidence Index
US2TB	US-2-year treasury Bond futures
US30TB	US-30-year Treasury Bond futures
ETH	other crypto-currency - Ethereum
XRP	other crypto-currency - Ripple
US_INTR	US intr

S&P	S&P
BTCCash	other crypto-currency - BTC cash
LTC	other crypto-currency - Litecoin
Gsearch	Google trends : percentiles of "Bitcoin" search on Google (1: bottom 1%, 100 : top 100% percent of searches)

2. Data exploration, visualization, preprocessing

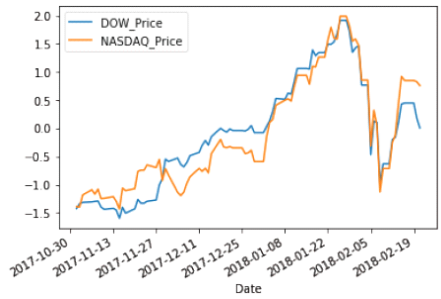
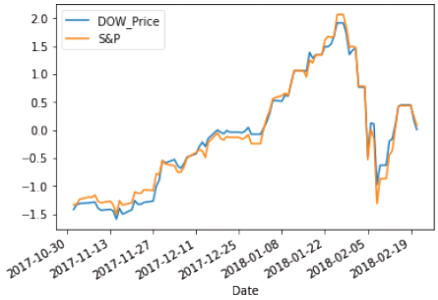
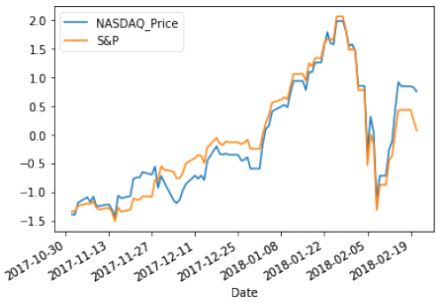
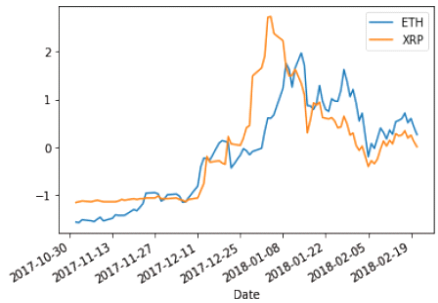
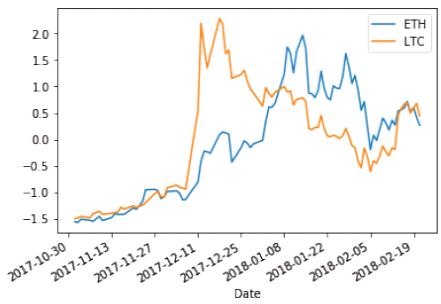
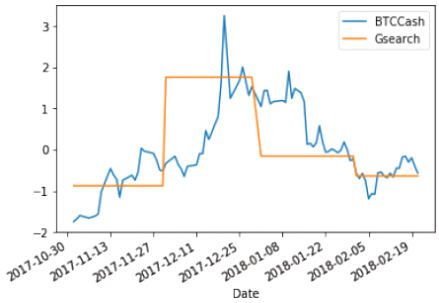
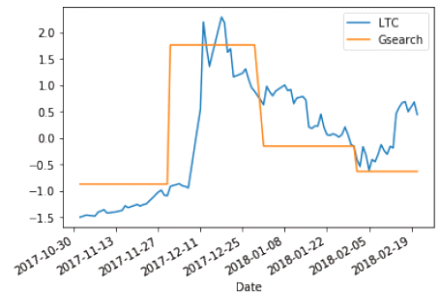
1) Daily price data

Graph below is a bitcoin daily price in our dataset which to be predicted. We have large dataset spanning from 2012 to 2018 but our main point is to predict next-day bitcoin price, which means our model will learn to detect local pattern in everyday market and trend data which reflects inter-day bitcoin price fluctuation. So we decided to limit period of interest from 20171101 to 20180301, and perform correlation analysis.



2) Correlation analysis on features

By using pandas library's corr method, we found out some strong correlation pairs, (DOW_Price, NASDAQ_Price), (DOW_Price, S&P), (NASDAQ_Price, S&P), (ETH, XRP), (ETH, BTCCash), (ETH, LTC), (BTCCash, LTC), (BTCCash, Gsearch), (LTC, Gsearch). Each pairs' graphs are as follows.

(DOW_Price, NASDAQ_Price)	(DOW_Price, S&P)
	
(NASDAQ_Price, S&P)	(ETH, XRP)
	
(ETH, BTCCash)	(ETH, LTC)
	
(BTCCash, LTC)	(BTCCash, Gsearch)
	
(LTC, Gsearch)	
	

Based on domain knowledge, we decided candidates for PCA(Principle Component Analysis) : DOW_Price, NASDAQ_Price, S&P, BTCCash, and LTC.

3. Preprocess of Data

With PCA candidates from previous data exploration, we performed PCA using sklearn.decomposition library's PCA method. Using parameter n_components for PCA method as 2, we projected 5-dimension features to 2-dimensional features. 2 attributes are represented as pc1 and pc2, and other attributes are normalized. Final dataset attributes are shown as below.

(pc1, pc2, GOLD_Price, USD_Price, US_BCI, US_CCI,
US2TB, US30TB, ETH, XRP, US_INTR, Gsearch)

For training, we used 80% of dataset as training data, and 20% of dataset as test data.

Prediction Model

1. Linear Regression

Linear Regression is a linear approach to modelling the relationship between a scalar dependent variables and one or more independent variables, which are explanatory variables. In this method, the relationships are modeled using linear predictor functions whose unknown model parameters are estimated from the data. In this case, there are 12 predictor variables so it is considered as multiple linear regression. Since the response variable y is a scalar, we decide that this model will be appropriate for our prediction model. For implementation, sklearn.linear_model library's LinearRegression method was imported. Training is performed automatically.

2. Logistic Regression

Logistic Regression is a regression model estimating the parameters of a logistic model. It uses log-odds of the probability of an event in prediction. For more than two values, not 0 or 1 output, which are categorical outputs, it should use ordinal logistic regression. This model itself simply models probability of output in terms of inputs, and appropriate to make a classifier by using a cutoff value. Since our problem needs numerical output, it may work inappropriate, but for comparation this model also be implemented. For implementation, sklearn.linear_model library's LogisticRegression method was imported. Trainig is

performed automatically.

3. Support Vector Machine(SVM)

SVM, Support Vector Machine, is supervised learning model with associated learning algorithm that analyze data used for classification and regression analysis. SVM separate categories by a clear gap that is as wide as possible. For implementation, sklearn.svm library's SVR method was imported, which means Support Vector Regressor, SVM for regression model. By taking parameters epsilon as 0.1 and C as 10, training performed automatically.

4. K-Nearest Neighbour Regression

K-Nearest Neighbour algorithm is a non-parametric method used for classification and regression. The input consists of the k closest training examples in the feature space. In k-Nearest Neighbour regression, the output is the property value for the object. This value is the average of the values of its k nearest neighbours. Which means, for regression, k-NN algorithm is used for estimating continuous variables, uses a weighted average of the k nearest neighbors, weighted by the inverse of their distance. For implementation, sklearn.neighbors library's KNeighborsRegressor method was imported. By choosing number of neighbors as 3, training performed automatically.

5. Recurrent Neural Network(RNN)

Recurrent Neural Network is a class of artificial neural network where connections between nodes form a directed graph along a sequence. It is a supervised learning for classification and regression. These nodes allow it to exhibit dynamic temporal behavior for a time sequence. For implementation, we used keras library. Each layers are LSTM(Long Short Term Memory networks) cells, which adds or removes input data to cell state using gate so that information can be flow selectively. The first gate layer in one LSTM cell is called forget layer. It decide to maintain or remove the input data by sigmoid layer. Then in input gate layer, it decide which value to be update, and tanh layer create candidate value to be added on cell state and add these two value so that it can influence to the next layer. Finally apply tanh function to this new state and decide output value by sigmoid layer.

For our implementation, we construct 3 layered RNN model with LSTM cell by using Keras. Dropout rate is 0.25 and use mse as loss function to calculate

gradient. ReLu function is used for activation function, and adam optimizer is used as an optimizer for back-propagation. Each layer has 512 neurons, and training is performed with batch size 12 and epoch 30. Training performed automatically by using fit() method.

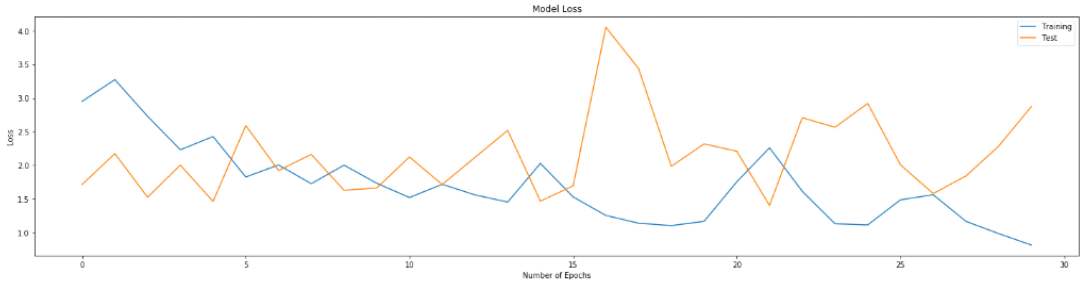
4. Result and Analysis

Four regression methods, linear regression, logistic regression, Support vector machine(regressor), K Nearest Neighbour regression, was measured its performance by R-square score.

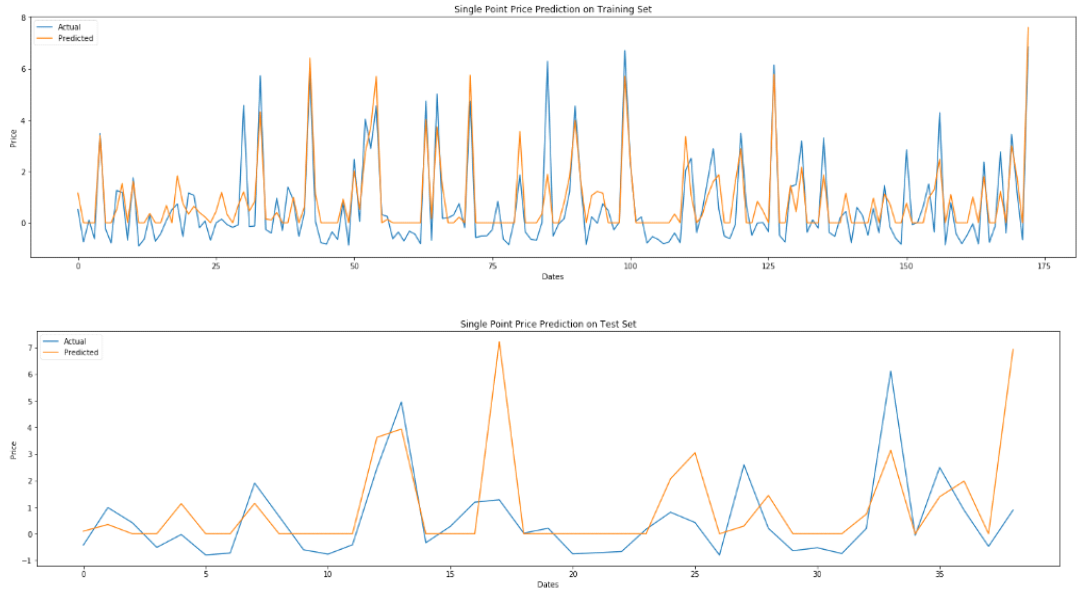
Regression Method	R-square score with random training data	R-square score with serial training data
Linear Regression	0.961	0.426
Logistic Regression	0.000	0.000
Support Vector Regression	0.864	0.351
K-nearest neighbour regression	0.977	-1.873

In this result, K-nearest neighbour regression and Linear Regression seems to be most outperformed prediction model, and K-nearest neighbour is the best. However, with further consideration, it seems to have many problem. First, logistic regression shows meaningless result. Since it is for classification, it requires only integer value for output variable. For this reason, we used label encoder for preprocessing but it dropped accuracy hardly. Secondly, three other methods shows meaningful R-square score only for randomly chosen training dataset. This problem emerged because of our dataset. Bitcoin price increases remarkably day by day. So these simple regression methods cannot learn for the classes with high prediction values. As a result, it cannot predict high bitcoin prices for recent days. Also when we calculated RMSE(Root-Mean squared Error) value for each methods, it was too big so it cannot be considered as well-performing training model.

Contrastively, RNN shows fairly good result. Graph below shows model loss depending on umber of epochs. Blue line is for training set and the other one is for test set. In this result, we model loss is optimal in around 27 epoch. Since its loss is remarkably smaller than previous four methods, we decide RNN as the most outperforming model and check prediction result.



Graphs below are single point price predictions on Training set and test set. Blue line represents the actual value, and the other line represents a predicted value. For test set, trained model shows quite accurate prediction data obviously. With test set, it cannot predict absolutely same values as actual values but it shows relatively similar trend with actual data. The mean absolute error was 1.10%, which is reasonably meaningful.



As a result, we concluded that RNN model shows the most accurate result for Bitcoin price prediction. Other four regression methods, linear regression, logistic regression, support vector machine regression, and k nearest neighbour regression shows disappointed result. Since these methods are meaningful when attributes are deeply related to prediction value, inappropriate dataset could be the reason for our result. Result of our RNN model is meaningful that our model predicts an daily Bitcoin price data reasonably, which took center stage in nowadays. Since our model used 3 LSTM layers, it still can be improved for the future works.

5. Reference

[1] Jang, H. & Lee, J. (2018). An Empirical Study on Modeling and Prediction of Bitcoin Prices With Bayesian Neural Networks Based on Blockchain Information. IEEE Access, 6, 5427-5437

[2] Shah, D. & Zhang, K. (2014, September). Bayesian regression and Bitcoin. In Communication, Control, and Computing (Allerton), 2014 52nd Annual Allerton Conference on (pp. 409-414). IEEE

[3] Khoa, N. L. D., Sakakibara, K., & Nishikawa, I. (2006, October). Stock price forecasting using back propagation neural networks with time and profit based adjusted weight factors. In SICE-ICASE, 2006. International Joint Conference (pp. 5484-5488). IEEE

[4] Mern, J., Anderson, S., & Poothokaran, J. (2017) Using Bitcoin Ledger Network Data to Predict the Price of Bitcoin: Milestone

[5] Baumann, A., Fabian, B. & Lischke, M. Exploring the Bitcoin Network, vol. 1 (2014)

[6] Kondor, D., Csabai, I., Szu'le, J., Po sfai, M. & Vattay, G. Inferring the interplay between network structure and market effects in Bitcoin. New J. Phys. 16, 125003 (2014). DOI 10.1088/1367-2630/16/12/125003. 1412.4042.

[7] Dixon, M., Klabjan, D. & Bang, J. H. Classiffication-based financial markets prediction using deep neural networks. CoRR abs/1603.08604 (2016). URL http://arxiv.org/abs/ 1603.08604. 1603.08604.

[8] Borovykh, A., Bohte, S. & Oosterlee, C. W. Conditional Time Series Forecasting with Convolutional Neural Networks. ArXiv e-prints (2017). 1703.04691.

[9] Abadi, M. et al. TensorFlow: Large-scale machine learning on heterogeneous systems (2015). URL https://www.tensorflow.org/. Software available from tensorflow.org.

[10] Matta, Martina & Lunesu, Ilaria & Marchesi, Michele. (2015). Bitcoin Spread Prediction Using Social And Web Search Media.

[11] Data : https://www.kaggle.com/acornec/bitcoin-predictions