Empirical Analysis of Bitcoin Market Volatility using Supervised Learning Approach

Hrishikesh Singh
Department of Computer Science
Jaypee Institute of Information Technology
Noida, 201309, India
hrishikesh.hsk@gmail.com

Parul Agarwal
Department of Computer Science
Jaypee Institute of Information Technology
Noida, 201309, India
parul.agarwal@jiit.ac.in

Abstract – **Crypto currencies are considered as the next model of economics and monetary exchange. In recent years, popular cryptocurrency such as Bitcoin and Ethereum witness an exponential growth in economic sphere. In this paper empirical testing of four conventional machine learning methods is performed to predict the bitcoin prices using last eight years of transactional data. Linear and polynomial regression is implemented using all the features individually. Polynomial regression, Support Vector regression and KNN regression are hyper tuned with grid search logic. Results depicted that KNN regression outperformed others models in attaining mean square error of 0.00021.**

Index Terms – Bitcoin, Machine learning, Cryptocurrency.

I. INTRODUCTION

In year 2008, pseudoonymous engineer Satoshi Nakamoto invented a cryptocurrency, Bitcoin[1]. It has two components, the bitcoin token which is an electronic key proving the digital ownership of the currency. It can be stored and verified digitally only. Second component is the bitcoin-protocol which works on the principle of a hash based shared ledger over the blockchain network.

Bitcoin are maintained on an open and distributed network using the hashing method which is mathematically verifiable. As compared to the federal structures of fiat currency, Bitcoin are limited to maximum count of 21 million which are kept in check by its brilliant self-balancing algorithm which maintains the count in inverse proportionate ratio to the number of customers currently on the network.

It solves the double spending problem by cryptography and check of the blockchain history prior to the transaction[2].

Due to the decentralized structure of the cryptographic currency transaction model, there is no requirement of any third party facilitator to authenticate, monitor or rollback the transaction between various parties. Hence, customers aren't required to validate their identities which maintain the anonymity. To avoid the illegal usage of the bitcoin, law enforcement authorities can trace back the transaction over a blockchain on violation.

Considering the price valuation of the bitcoin over last eight years, the surge from 10$ with average transaction of 2542 per month to the highest valuation at 19,783.21$ with average transactions of 6,112,832 per month, Bitcoin attracted the attention of stock traders and researcher to analyze it further[3].

In the recent years, few empirical studies have been done over the cryptography based currencies. Shah and Zhang [4]implemented a latent source model which works on the Bayesian regression and demonstrated a promising result by attaining high average profitability within a time span of 60 days. Sean et. al [5] performed prediction with recurrent neural network models and further compared its performance with time series based model i.e. Auto Regressive Integrated Moving Average Model, which predicted market volatility with an accuracy of 55-60%. In the studies where features such as Bitcoin Price Index, Volume, etc. are taken in consideration individually shows inferior performance as compared to the feature selection algorithms.

In this paper, an empirical analysis of the conventional machine learning models is performed using dataset consisting of last 8 years of bitcoin transactional details. The work begins with linear and polynomial regression for the benchmark testing. To select the appropriate features among all, Pearson collinearity coefficient matrix is computed. Considering the feature rich dataset, hyper parameter tuning using grid search is used and implemented Ridge and lasso method integrated with polynomial regression to handle multi co-linearity. Further the Support Vector Regressor with linear and *rbf* kernel is tested, K-Nearest Neighbor Regression outperforms all in term of Mean square error and R-Square Score.

This paper is divided into five sections. In section II explains about the conventional machine learning models to predict the bitcoin price value. Section III showcases the experimental setup including the subsections focusing upon the feature selection, data processing and efficiency measurement. The result and analysis of the prediction done by the selected models are empirically presented in section IV and conclusions and future prospects are defined in section V.

978-1-5386-6835-1/18/$31.00 ©2018 IEEE

II. MODELS USED

A. Linear Regression

In this modelling technique, relationship is formed between two variables *X* and *y*, where *X* is a continuous or discrete independent variable and *y* is a continuous dependent variable. We aim to generate a regression line also known as best fitting line, since we try to minimize the distance of every data point distance from this line. These distances are also called residuals.
Mathematically it can be represented matrix notation[6]:

$$y_i = \beta_0 1 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \varepsilon_i = {x_i}^T \beta + \varepsilon_i$$

Where i=1,..,n and further in matrix based form

$$X = {x_i}^T \beta$$
$$y = X\beta + \varepsilon$$

Where *y*and *X*are variable vectors and ε is the residual error.
Similarly, polynomial regression generates a regression line with degree two to better fit the data points while minimizing the residuals.Mathematically it can be represented as: -

$$y_i = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_n x^n + \varepsilon_i$$

In vector form,

$$\vec{y} = x\vec{\beta} + \vec{\varepsilon} \quad (1)$$

Where,

$$\overrightarrow{\beta} = (X^T X)^{-1} X^T \vec{y} \quad (2)$$

B. Grid Search

Parameter search is one of conventional method of hyper parameter tuning using exhaustive searching on the user generated search space of hyper parameters. It's important, since many datasets may provide a real valued or a large search space for a particular algorithm. Hence user defined parameter value set is a mandatory step before searching for reasonable values of the parameters[7][8].

C. Ridge Regression

Multicollinearity among features of any datasets affect the estimation scores such as the variance to be far from the original values while the least square may be unbiased. Ridge regression[9] improves the accuracy by simply increasing the bias for parameters of estimation.

If we represent the regression model as

$$Y' = XB' + e' \quad (3)$$

Where Y', X, B' and e' are dependent, independent, estimated regression coefficient and residuals respectively.

Estimator bias can be defined as:

$$E\left(\widetilde{B'} - B'\right) = [(X'X + kI)^{-1} X'X - I]B' \quad (4)$$

D. Lasso regression

Lasso stands for Least Absolute Shrinkage and Selection Operator[10]. Itaims to reduce the summation of all regression coefficients below a specific absolute value by penalization. Even if few coefficients are required to be equated as zero as compared to the Ridge regression where no coefficients are set as zero. Hence Lasso method may lead to elimination of lesser important features which improves the prediction model.
The objective of Lasso is to minimize the following:

$$\sum_{i=1}^{n} (y_i - \sum_j x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \quad (5)$$

Where λ is data dependent

E. Support Vector Machine

In this model, each instance of the dataset can be represented as a vectors having n-dimensions which can segregated from other instances using hyperplane of n-1 dimensions. Hyper plane with the widest margin from the nearest data instances on both side is considered best i.e. maximum-margin hyperplane. It can be mathematically represented as[11]:

$$\vec{w} \cdot \vec{x} - b = 0 \quad (6)$$

Where $\vec{x}$is a *n* dimensional real vector and $\vec{w}$ is normal vector to the hyperplane.

Similarly, for the regression application we will use Support Vector Regression (SVR)[12] which

$$minimize \ \frac{1}{2} \|w^2\| \quad (7)$$

$$subject\ to \begin{cases} y_i - \langle w, x_i \rangle - b \ \leq \ \varepsilon \\ \langle w, x_i \rangle + b - y_i \ \leq \ \varepsilon \end{cases}$$

Where x_i, y_i are training sample and target value pair.

To improve the computational complexity of inner product of infinite dimensions, kernel methods are integrated with SVM. Considering the data, for *n*features, a matrix defining the similarity of each feature with others (pair form)is a scaler representation of kernel in matrix form.

$$K(x, z) = \phi(x)^T \phi(z) \quad (8)$$

Where φ denotes feature mapping.

Radial basis function(*rbf*) kernel can be expressed as follow:

$$K(x, x') = exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right) \quad (9)$$

Here σ denotes constraint free parameter.

F. K-Nearest Neighbor Regression

In this model, initially continuous features are determined by fundamental KNN algorithm[13]. For a query instance Euclidean distance is computed with respect to the training data. On the basis of this distance, instances of training data are sorted in ascending order. For better performance, optimal value of *K* is computed based on heuristic method using Root Mean Square as a deciding parameter by cross validation. Considering k-nearest multivariate neighbors weighted average of the inverse distance is calculated[14].

III. EXPERIMENTAL SET UP

A. Dataset Processing

Dataset for experiments are collected from coincap market website consisting of last 8 years of bitcoin market data ranging from February 2010 to February 2018. Each instance represents value for a specific date. Initially we had 23 columns with probable features of various categories [3].

To understand the features value range the data is plotted as histogram considering individual features which helps to further understand that features almost similar to the bitcoin price.

To clean the data from the missing value, the null values are replaced with "*bfill*" i.e. next valid observation to fill since bitcoin prices are highly correlated to the previous day prices. The non-numerical type data frames are removed for analysis. Moreover dataset is distributed into training and testing data into the ratio as 80:20.

B. Feature Selection

Among the above 22 features, there is a need to determine the most important contributors in the bitcoin prediction. For that the correlation among features is identified.

Pearson correlation is used to compute the coefficient matrix which can be mathematically represented as [15]:

$$r = \frac{\sum_{i=1}^{n}(x_i-\bar{x}).(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2 \cdot \sum_{i=1}^{n}(y_i-\bar{y})^2}} \quad (10)$$

Where $\bar{x}$,$\bar{y}$are the mean value and *n*indicates the number of feature pairs.

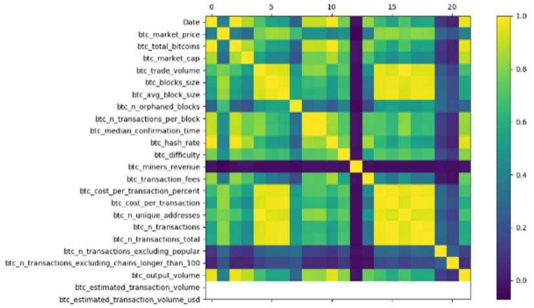

Figure 1: Correlation Graph

The above correlation graph represents the value annotation between the various features and the scale indicate the relative dependencies between the two features under comparison. Lighter indicate highest correlation and vice versa.

Features such as orphan blocks count, transaction fees, cost per transactions, output volume, estimated transaction volume in US Dollar and median confirmation time of a transactions are excluded because of the small value range as compared to the bitcoin price value range which is relatively high. Hence the above mentioned features aren't affecting the price significantly enough.

Further features such as tera-hash rate over bitcoin network, difficulty of transaction, miner's revenue and total bitcoin value under circulation in terms of US Dollar shows significantly higher correlation with bitcoin prices which may result to providing additional information beyond the provided dataset (also known as *data leak*). Similarly, average block size of a bitcoin transaction shows low correlation with target features and hence can be dropped without any noticeable loss in the prediction accuracy.

From the dataset, total number of bitcoin transaction and transactions from the popular IP address are cumulative in nature, hence it will not contribute much in the bitcoin price changes.

Among all features we witnessed a diverse range for each features. So value scaling is done using Min Max Scaling.

C. Efficiency Measurement

For measuring the performance of our model, R squared error is computed

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(X_{obs,i}- X_{model,i})^2}{n}} \quad (11)$$

R square indicates that out of 100% how much is already known to us for making a correct prediction[16].

Further due to various features consideration, it's better to opt for Normalized Root Mean Square Error (NRMSE).

$$\text{NRMSE} = \frac{\text{RMSE}}{X_{\text{obs,max}}-X_{\text{obs,min}}} \quad (12)$$

$$\text{NRMSE} = \frac{\text{RMSE}}{\overline{X_{\text{obs}}}} \quad (13)$$

Sklearn uses OLS(ordinary least square method)[17][18].
Least Square method estimator:

$$\hat{\beta} = \frac{\sum x_i y_i - \frac{1}{n}\sum x_i \sum y_i}{\sum {x_i}^2 - \frac{1}{2}(\sum x_i)^2} = \frac{Cov(x,y)}{Var(x)} \quad (14)$$

$$\hat{\alpha} = \bar{y} - \beta\bar{x} \quad (15)$$

IV. Result & Analysis

Since its inception in 2009 bitcoin price rise from 10$ to highest price i.e.19,783.21$. *Figure 2* below shows the growth of the bitcoin price from 23rd Feb 2010 to 23rd Feb 2018.

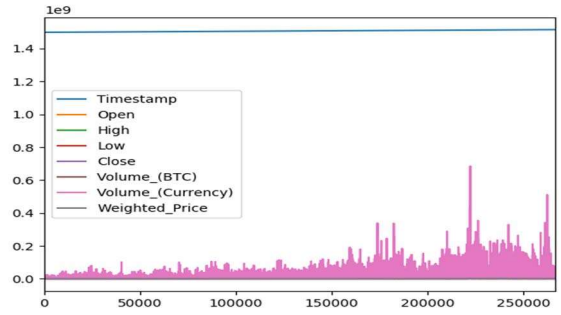

Figure 2: Growth of Bitcoin Price

Initially work begins with linear regression model on bitcoin dataset, calculating intercept and coefficient for all of 22 features individually. Among all of the linear projections, the one with the least mean square error and highest R square value is selected. Linear regression attained R-square score of 0.974 while training and 0.9799 during testing. However, Mean Square Error is obtained as 0.00033.

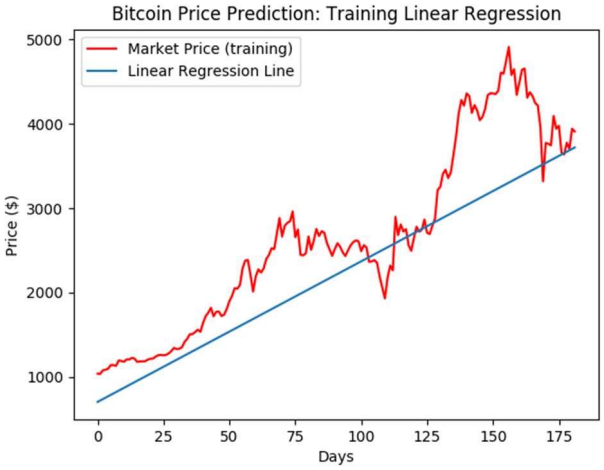

Figure 3: Training Bitcoin Prediction using Linear Regression

To improve this approach, the appropriate value of hyper parameters is needed i.e. C, gamma and epsilon. For hyper parameter tuning, grid search logic is applied.

The second model is Ridge Regression with hyper parameter tuning using grid search logic. Grid best score is obtained as -186.0165 and best estimator (alpha) is computed as 100.0 out of all alpha values ranging from 0.0001 to 100 with tenfold increment including 0. R-square values are gained as 0.7463 and 0.1147 while training and testing respectively.

Similarly, Lasso Regression is also implemented for further improvements. Gris search logic is also used for hyper parameter tuning. Lasso model obtained grid best score of -87.872 and best estimated alpha value was 0.0001 out of the same range used for Ridge regression model.

Lasso model depicted some improvements in terms of R-Squared Scores with 0.9731 and 0.9774 while training and testing respectively.

Additionally two variants of polynomial regression model of second degree are implemented. Polynomial model without any hyper parameter training results in the overfitting, hence gives perfect R-Squared score of 1.00 for both training and testing. To rectify this, Ridge polynomial regression is executed which results in R-squared score of 0.997 and 0.996 for training and testing respectively.

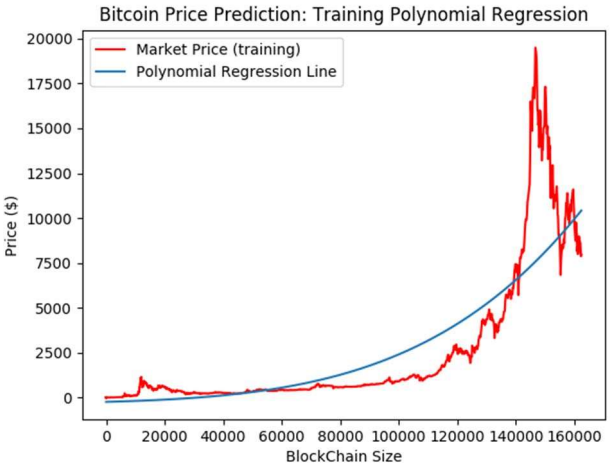

Figure 4: Training Bitcoin Prediction using Polynomial Regression

Before implementing Support Vector Regression Model, the best value of hyper parameter is determined for better fitting using grid search logic. Ranges for C, epsilon and gamma were [0.01,10], [0.001,1] and [0.001,1] with 10-fold increase in each value, respectively.

Support Vector Regressor is implemented with linear and rbf kernel. Hyper parameter tuning for SVM with linear kernel, outputs values of C as 0.01, epsilon as 0.001 and gamma as 0.001. R-Squared Score is 0.91193 and 0.37062 for training and testing respectively, Mean Square error obtained is 0.001269. SVM with rbf kernel improved with R-square score of 0.95281 and 0.61858842. Mean square error also dropped to 0.0007261. Hyper parameter tuning gives best value for above analysis, C as 0.1, epsilon as 0.01 and gamma as 0.01 using the same parameter grid range used for linear kernel.

Out of all dataset variables, most of them are non-volatile in nature and ranges between small values. These variables have variance in proportionate degree with respect to each other's. KNN model works similar to local method and seems a better trade off because of the lower bias advantage offer over other models.

K-Nearest Neighbour based regression is implemented with all the parameters hyper tuned using grid logic. Initially K value is taken in range of 1 to 10, where optimal results are obtained

for K=10. For computing the best value of nearest neighbours, the auto mode is enabled which check iteratively for Ball Tree, KD Tree and Brute force algorithms and further uses the best performing one automatically.

KNN Regression model determines best grid score of -1.66906 and best value of alpha as 10. R-squared score is estimated as 0.98778 and 0.988105 while training and testing respectively.

TABLE 1: SUPERVISED LEARNING MODELS RESULTS ON BIT COIN DATASET

Model	R2 Training	R2 Test	MSE
Linear Reg	0.97410	0.97990	0.000329
KNN	0.98778	0.98810	0.000213
Poly Reg(2 + Ridge)	0.99723	0.99611	0.000892
SVM(kernel = linear)	0.91193	0.37062	0.001269
SVM(kernel = rbf)	0.95281	0.61858	0.000726

TABLE 2: RIDGE AND LASSO REGRESSION WITH GRID SCORE ON BIT COIN DATASET

Model	R2 training	R2 Testing	MSE	Grid Score	Alpha
Ridge Reg	0.7463194	0.1146961	0.0037794	-186.0165	100.0
Lasso Reg	0.9731032	0.9773951	0.0003565	-87.8717	0.0001

V. CONCLUSIONS

Comparative analysis of linear regression, polynomial regression, ridge and lasso regression, SVM with linear and rbf, kernel and KNN regression model is made. It indicates that KNN is the most suitable model out of selected one, with MSE of 0.0002. In contrast to conventional stock market, crypto currencies have smaller daily transaction capital which leads to higher volatility in the market prices. Further completely independent transactional network which is least affected by federal decision, demands more features for accurate prediction.

REFERENCES

[1] S. Nakamoto, " Bitcoin: A peer-to-peer electronic cash system," 2008.

[2] G. A. E. a. C. S. Karame, "Double-spending fast payments in bitcoin," in *Proceedings of the 2012 ACM conference on Computer and communications security. ACM., 2012, pp.906-917.*

[3] "Coinbase," Coinbase , May 2018. [Online]. Available: https://www.coinbase.com/. [Accessed Feb 2018].

[4] D. a. Z. K. Shah, "Bayesian regression and Bitcoin," in *Communication, Control, and Computing (Allerton), 2014 52nd Annual Allerton Conference, IEEE., 2014.*

[5] S. McNally, "Predicting the price of Bitcoin using Machine Learning," Doctoral dissertation, Dublin, National College of Ireland, 2016.

[6] D. P. E. a. V. G. Montgomery, "Linear regression," in *Introduction to linear regression analysis (Vol. 821).,* John Wiley & Sons., 2012.

[7] P. Lerman, "Fitting segmented regression models by grid search.," in *Applied Statistics*, 1980, pp. 77-84.

[8] J. a. B. Y. Bergstra, " Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, 13, pp. 281-305, Feb, 2012.

[9] S. a. V. H. J. Le Cessie, "Ridge estimators in logistic regression," in *Applied statistics*, 1992, pp. 191-201.

[10] C. Hans, "Bayesian lasso regression," *Biometrika, 96(4)*, pp. 835-845.

[11] I. a. C. A. Steinwart, Support vector machines, Springer Science & Business Media., 2008.

[12] D. P. S. a. P. D. Basak, "Support vector regression," *Neural Information Processing-Letters and Reviews*, 11(10), pp. 203-224, 2007.

[13] D. Larose, " k-nearest neighbor algorithm," in *Discovering knowledge in data: An introduction to data mining*, 2005, pp. 90-106.

[14] S. Dudani, "The distance-weighted k-nearest-neighbor rule," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 4, pp. 325-327, 1976.

[15] J. C. J. H. Y. a. C. I. Benesty, "Pearson correlation coefficient," *Noise reduction in speech processing, no.* Springer Berlin Heidelberg., pp. 1-4, 2009.

[16] A. a. W. F. Cameron, "An R-squared measure of goodness of fit for some common nonlinear regression models," *Journal of Econometrics*, vol. 77(2), pp. 329-342, 1997.

[17] "Scikit Learn," scikit-learn, May 2018. [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html. [Accessed May 2018].

[18] S. a. V. H. J. C. Le Cessie, "Ridge estimators in logistic regression," in *Applied statistics*, 1992, pp. 191-201.