Bitcoin price prediction with ML

Yewon Bin

Introduction

The increasing variability in Bitcoin prices has prompted the exploration of machine-learning models for price prediction. In this project, I aim to forecast the future price of Bitcoin by leveraging the Bitcoin price dataset from 1 January, 2021 to 12 May 2021 in 1-minute intervals. We have implemented three different algorithms, namely Linear Regression, Decision Tree Regression, and XGBoost Regression, to compare their predictive performance.

Problem Formulation

The dataset used for this analysis was obtained from https://www.kaggle.com/datasets/aakashverma8900/bitcoin-price-usd/data.
Input: The input features for our machine learning models include historical attributes such as Open, High, Low, and Volume.
Output: The output, or the target variable, is the closing price of Bitcoin.
Dataset: The dataset comprises various features, including Open Time, Close Time, Quote Asset Volume, Number of Trades, Taker Buy Base Asset Volume, Taker Buy Quote Asset Volume, and more.
Number of Samples: The dataset consists of 3229 samples, from 2014-09-17 to 2023-07-20 with each entry representing a specific time frame.

Baseline

Decision Tree Regression

- Hyperparameters: I conducted a grid search for hyperparameter tuning, exploring options such as max_depth, min_samples_split, and min_samples_leaf.
- Tuned Parameters: The best parameters obtained from the grid search were used to initialize the Decision Tree Regression model, from max_depth = [3, 5, 7], minimum ample split = [2, 5] and minimum sample leaf = [1, 2, 4].

Ridge Regression

- Hyperparameters: The alpha parameter, controlling the regularization, was optimized through grid search.
- Tuned Parameters: The best alpha value obtained from the grid search was employed in training the Ridge Regression model, from alpha = [0.001, 0.1, 1].

XGBoost Regression

- Hyperparameters: I performed a grid search for parameters like learning_rate, n_estimators, max_depth, subsample, and colsample_bytree.

- Tuned Parameters: The optimal parameters determined from the grid search were utilized to configure the XGBoost Regression model, from learning_rate = [0.01, 0.1], n_estimators = [30, 50], maximum depth = [3, 5, 7], subsample = [0.8, 0.9, 1.0] and colsample by tree = [0.8, 0.9, 1.0].

Evaluation Metric

The primary measure of success for our models is based on their predictive accuracy. We have employed two evaluation metrics:

Mean Squared Error (MSE):

MSE is a measure of the average squared difference between the predicted values and the actual values. So the primary goal is to minimize the MSE, as it directly reflects the accuracy of our predictions in terms of how close they are to the true values. This squaring of the errors emphasizes larger discrepancies, providing a comprehensive assessment of predictive performance.

R-squared (R^2):

R-squared is a statistical measure representing the proportion of the variance in the dependent variable (Bitcoin closing price) that is predictable from the independent variables (features). So a higher R-squared value indicates a better fit of the model to the data. Using R^2 helps quantify the explanatory power of the model, offering insights into the percentage of variability captured by our predictions.

Result

Decision Tree Regression

- Best Parameters:
  Maximum depth = 7
  Minimum sample leaf = 4
  Minimum sample split = 5
- Evaluation Metrics:
  Mean Squared Error: = 949354.4859525769
  R^2 = 0.9963598745814027

Ridge Regression

- Best Parameters:
  Alpha = 1
- Evaluation Metrics:
  Mean Squared Error = 147094.00078947496
  R^2 = 0.9994359950691551

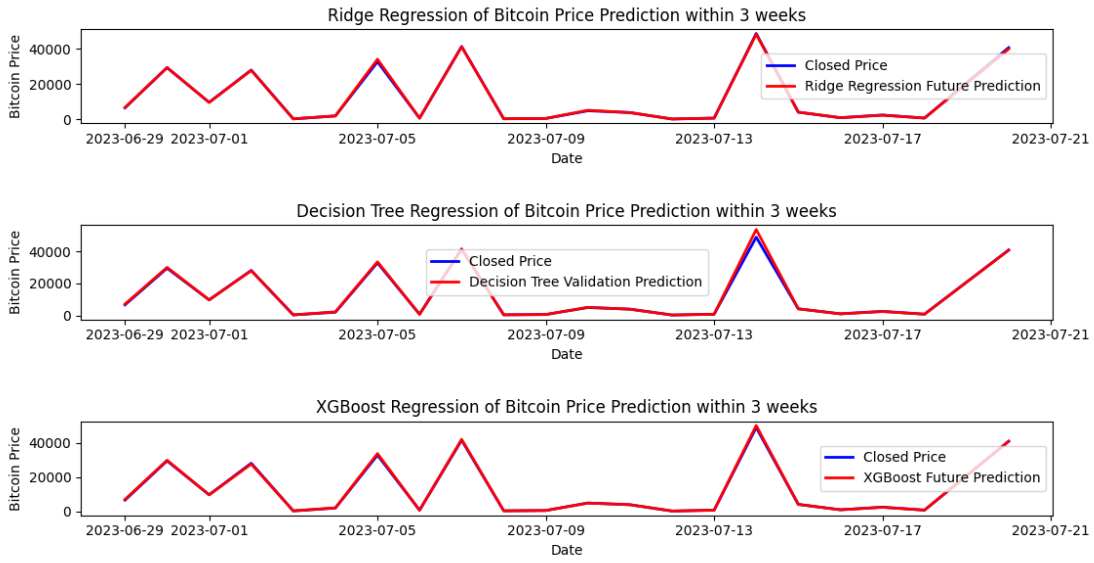
XGBoost Regression

- Best Parameters:
  Column Sample = 0.8

Learning Rate = 0.1
Maximum depth = 5
N Estimators = 50
Subsample = 1.0

- Evaluation Metrics:
  Mean Squared Error = 57671.892447557875
  R^2 = 0.9997788677203897

The following hyperparameters work the best with the machine learning models.



In comparison with the Baseline, XG Boost Regression works the best for Bitcoin price prediction with the lowest MSE and the highest R-squared value indicating that XGB regression prediction is a better fit of the model to the data. Ridge Regression is comparatively optimistic but trails XGBoost, while Decision Tree Regression shows less accuracy compared to both XGBoost and Ridge.

From the Visual analysis, the plot illustrates a strong alignment between XGBoost predictions and actual closed prices, indicating a robust fit. Conversely, the Decision Tree prediction deviates from actual prices, highlighting its limitations.

In conclusion, this project contributes to the understanding of Bitcoin price prediction using machine learning algorithms. The comparative analysis of Ridge Regression, Decision Tree Regression, and XGBoost Regression provides insights into their respective strengths and weaknesses.