Intra-day Bitcoin Price Prediction with Machine Learning and Ensemble Methods

John Tsang

University of Ottawa

Fall 2021

MAT 4900 – Undergraduate Research Project

Outline

Introduction: Project Goal

Project Goal:

Predict the average Bitcoin prices on Coinbase from August 5th, 2021 to August 31st, 2021 for every five-minute interval.

Introduction: Highlights

- The performance of some basic machine learning models to forecast intra-day Bitcoin price
- Can ensemble methods improve the forecasting performance?
- Is there other possible approaches to improve the forecasting performance?
- Lessons I learnt from this project

Introduction: Stage 1

Stage 1: One-step ahead Predictions with Base Models

1. the random walk (RW) model
2. the linear regression (LR) model
3. the support vector regression (SVR)
4. the *k*-nearest neighbour regression (*k*-NN)
5. the decision tree regression (DT)

Introduction: Stage 2

Stage 2: Use Ensemble Methods to Improve Predictions

1. Stacking
2. Random Forest
3. Bagging

Introduction: Forecast Evaluation Criteria

1. Forecast Accuracy: Mean Absolute Percentage Error (MAPE)

$$\text{MAPE} = \frac{1}{T}\sum_{t=1}^{T}\left|\frac{\text{Predicted Value}_t - \text{Actual Value}_t}{\text{Actual Value }_t}\right|$$

2. Direction Accuracy

The relative frequency (in %) of correctly predicting the direction of changes in Bitcoin prices

Introduction: Forecast Evaluation Criteria

3. Profitability: "buy low, sell high"

If the magnitude of the predicted Bitcoin price change in the next period is more than 1.0%, there will be a transaction.

- Predicts Bitcoin price $\uparrow \implies$ purchase 1 Bitcoin $\implies$ Sell 1 Bitcoin in the next period
- Predicts Bitcoin price $\downarrow \implies$ sell 1 Bitcoin $\implies$ Buy 1 Bitcoin in the next period

Average Rates of Return=Simple arithmetic average of the rates of return of all transactions

Predictors: Traditional Demand- and Supply-side Factors

	Predictor	Frequency
1.	Moving Average Bitcoin Prices	5, 10, 30 and 60 minutes
2.	Moving Averages of Total Bitcoin Transaction Volume	5, 10, 30 and 60 minutes
3.	Percentage of the Volume of Buy Transactions	5, 10, 30 and 60 minutes
4.	Change in Daily Close Nasdaq Bitcoin Reference Price (NQBTC)	Daily

Predictors: Social Media-related Variables (Daily)

Google Trends:
Web & News Searches

Google Trends:
YouTube Searches

New Posts &
New Comments

New Pageviews

New Tweets

Stage 1: Methodology I

Input: Rolling window size $w_R > 0$; a list of hyperparameters `k-list` $\neq \emptyset$

Input: `model` $\in$ [LR,SVR,k-NN,DT]

Output: A list for storing each prediction `prediction`

foreach *rolling window* $R = (\boldsymbol{x}, y)$ *of size* w_R *(Figure 1)* **do**

  Initialize dictionary `error_dict[]` as a list for each key k in `k-list`

  foreach *training set* T_R *and test set* $t_R = (\boldsymbol{x}_{t_R}, y_{t_R})$ *in* R **do**

    foreach k *in* `k-list` **do**

      `fitted-model` $\leftarrow$ the `model` fitted with T_R and k

      $y_k \leftarrow$ prediction from `fitted-model` using $\boldsymbol{x}_{t_R}$

      `abs-pct-error` $\leftarrow \left| \frac{y_k - y_{t_R}}{y_{t_R}} \right|$

      Append the list `error_dict[`k`]` with `abs-pct-error`

    end

  end

  `optimal-k` $\leftarrow$ the key k where $\min_k$ average(`error_dict[`k`]`)

  `fitted-model` $\leftarrow$ the `model` fitted with R and `optimal-k`

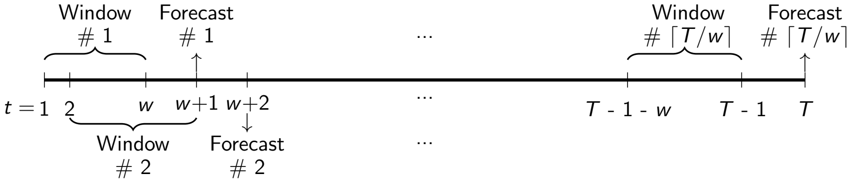
  $y_{\texttt{optimal-k}} \leftarrow$ prediction from `fitted-model` using the last observation's $\boldsymbol{x}$

  Append the list `prediction` with $y_{\texttt{optimal-k}}$

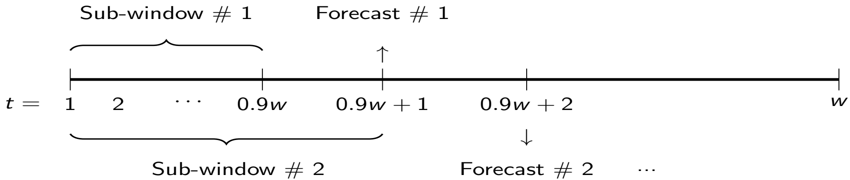
end

Stage 1: Methodology II

Figure 1: Illustration of the Rolling Window Strategy

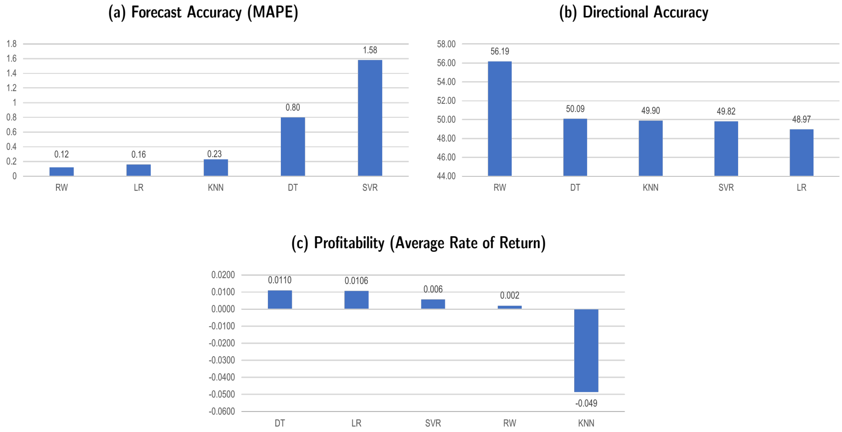


In the first rolling window,



Stage 1: Forecast Evaluation Results

Figure 2: Stage 1 Forecast Evaluation Results, per cent



Methodology: Stage 2

Stacking

Feed predictions from Stage 1 as predictors to each base model for prediction

Bagging

A prediction is the average of predictions from each model trained on each bootstrap sample

- Bagged linear regression: use LASSO to avoid extreme predictions

Random Forest

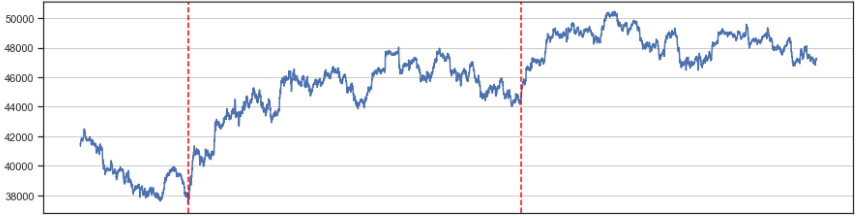
Bagged decision tree + random choice of predictors

Results: Stage 2

	MAPE	DA	Profitability
(a) *Stacking*:			
k-NN	0.2	46.71	0.1500
DT	0.8	49.68	0.0113
SVR	1.5	49.67	-0.0026
LR	0.2	43.84	0.0289
(b) *Bagging*:			
k-NN	0.2	50.50	-0.0220
DT	0.8	49.80	0.0130
SVR	1.6	49.95	0.0055
LR with LASSO	0.2	46.64	-0.1428
(c) Random Forest	0.3	49.81	0.0376

Further Discussion: Trend Changes

Figure 3: MA Bitcoin Price Every 5 Minutes, Aug 1 – Aug 31 2021



Two Trend changes:

- the 153^{rd} five-minute interval on August 5
- the 166^{rd} five-minute interval on August 19

Further Discussion: Trend Changes

Forecast Evaluation: Entire Timeframe - Periods without the Breaks

	MAPE	DA	Profitability
Stage 1			
KNN	0.0	0.1	0.2468
DT	-0.2	0.2	0.0000
SVR	-0.4	0.3	-0.0040
LR	0.0	-0.3	-0.1050
Stage 2			
(a) *Stacking*			
KNN	0.0	0.1	0.0482
DT	-0.2	0.1	-0.0003
SVR	-0.2	0.2	0.0042
LR	-0.0	-0.1	-0.1233
(b) *Bagging*			
k-NN	0.0	0.5	0.1286
DT	-0.3	0.1	-0.0045
SVR	-0.4	0.3	0.0040
LR with LASSO	0.0	0.0	-0.1803
(c) Random Forest	0.0	-0.1	0.0000

Further Discussion: Using Subsets of Predictors

- **Group 1**: the last 5-minute, 10-minute, 30-minute and 60-minute moving average Bitcoin prices and the daily change in the close Nasdaq Bitcoin Reference price (NQBTC)
  - Most relevant to forecast accuracy and profitability
- **Group 2**: the last 5-minute moving average Bitcoin prices
  - Most relevant to forecast accuracy and profitability
  - Using only Group 2 can achieve good forecast accuracy and profitability at the expense of some directional accuracy

Further Discussion: Using Subsets of Predictors

- **Group 3**: moving average total Bitcoin transaction volumes and percentages of the volume of buy transaction for the last 5-minute, 10-minute, 30-minute and 60-minute intervals
  - Most relevant to directional accuracy

Further Discussion: Replacing Standard Bootstrap

- Use a modified version of the standard bootstrapping to account for serial dependence:
  - the seasonal-trend decomposition using locally estimated scatterplot smoothing (STL decomposition)
  - the moving block bootstrap
- Improves profitability of bagged k-NN (-0.07% to 0.003%) and bagged DT models (-0.01% to 0.003%)

Summary: Research Results

- The performance of some basic machine learning models to forecast intra-day Bitcoin price
- Can ensemble methods improve the forecasting performance?
- Is there other possible approaches to improve the forecasting performance?

Lessons Learnt

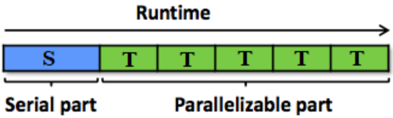
Aside from learning more about machine learning and ensemble methods, I learnt two main lessons about working on a project with a large data size:

- Lesson 1: Always write scalable programs
- Lesson 2: Plan computational time ahead

Also, this is my first set of slides in $\LaTeX$!

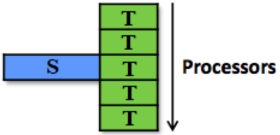
Lessons 1: Always Write Scalable Program

One processor:



Total Runtime $= S + 5T$

Multiple processors:

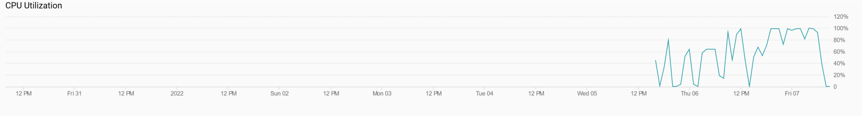


Total Runtime $= S + T$

Modified from: https://skirt.ugent.be/skirt8/_parallel_performance.html

Lessons 2: Plan Computational Time Ahead

Virtual machine 1:



Virtual machine 2:

Lessons 2: Plan Computational Time Ahead

Virtual machine 3: