Classifying the Price Movement of Bitcoin

46-927 - MACHINE LEARNING II

Long Vu, Nikhila Kulukuru, Xuanzi Zhao, Zhi Yun Yap

1 Introduction

During the seven years from April 2015 to April 2022, the standard deviation of Bitcoin's daily return rate was 3.85%, which was 3.36 times the standard deviation of the S&P 500 return. Using hourly percentage change as a measure of volatility, maximum volatility on Bitcoin close price is observed on Mar 12, 2020, where the price changes by 21.10% within an hour. Unlike traditional assets, it is more reasonable to access the Bitcoin price movement on an hourly basis due to the sharp fluctuation of the cryptocurrency price. [A]

Machine learning methods have been applied increasingly within the crypto market, due to their ability to learn complex, high-dimensional relationships. Politis et al. (2021) [1] applied a Long-Short Term Memory (LSTM) model to predict the price of Ethereum and achieved an accuracy of 84.2%. Patrick et al. (2021) [2] analyzed the predictability of the bitcoin market across different prediction horizons ranging from 1 to 60 minutes. Aggarwal (2019) [3] studied the relationship of the gold price with Bitcoin price movement using Convolutional Neural Network (CNN), (LSTM), and Gated Recurrent Unit (GRU). Liu et al. (2021) [4] later expanded the range of covariates to 40 explanatory variables including the macro market index (i.e. S&P 500 price) and crude oil price for Bitcoin price prediction.

While previous studies focused on exploring the correlation between Bitcoin and traditional assets, its relationship with investors' sentiment and other cryptocurrencies' prices has been neglected. Here, we assess the predictability of hourly price trends of Bitcoin using social media sentiment, foreign exchange rate, as well as the price movement of other major cryptocurrencies. In this project, we would like to classify the direction of bitcoin price movement in the near future using various economic and financial indicators.

The majority of our models achieved accuracy rates of around 80%. However, the main limitations in our model were scarcity of data points, less relevant features, and highly noisy features. Additionally, the feature selection provided by some models shows us that the price change of other crypto coins affects the price of bitcoin the most. This suggests high correlation and potentially high collinearity between these inputs and our response variable; therefore, extensive assessments are highly suggested in order to improve the current model.

2 Dataset

2.1 Dataset Collection & Preprocessing

We extracted data from Bloomberg, Augmento AI [5], and kucoin.com [6] ranging from Jan 01, 2019, to Nov 29, 2022. Our Bloomberg dataset includes hourly data for the VIX index, USD/GBP exchange rate, and the 10-year bond yield. From Augmento AI, we obtained the hourly cryptocurrency sentiment on Reddit, Twitter, and Bitcointalk. Combined with the hourly Bitcoin (BTC), Ethereum (ETH), Ripple (XRP), and Cardano (ADA) price data from kucoin.com, the dataset comprises 26 covariates. While cryptocurrency markets remain open 24/7, the VIX options and the 10-year Treasury bond are not traded over the weekend. In such cases, we forward-filled the missing values using the latest historical data.

Upon calculating the percentage change of Bitcoin close price, we encoded the price direction with binary labels - 1 as upward movement and 0 as downward movement. This formulates the prediction problem into a binary classification problem. Since the range of the cryptocurrencies' daily closing prices varies significantly, we compute the percentage change of price with respect to each coin. Additionally, to preserve the cyclical nature of weeks, we constructed a datetime feature to represent the day of week information. When decomposing the time series Bitcoin price [B], we see that the price trend is mainly driven by trend and does not possess any obvious seasonality. Additionally, we scaled all the columns and normalized them.

2.2 Evaluation Metrics

To prepare for model training, we split our dataset into a training set with data from 2019 to 2021 and a testing set with data from 2022 [C]. For hyperparameter tuning of neural networks, we follow the conventional 60-20-20 rule for splitting training, validation, and test data. [C]

For comparison, we combined the precision, recall, and F1-score - as a balance of precision and recall score, to evaluate the overall performance of the models. Additionally, we also compare their performance through the accuracy score, the confusion matrix, and the ROC curves.

3 Methodology

To understand how to accurately predict future movements, we employed multiple classification techniques - LDA, QDA, Logistic Regression, Decision Trees, Random Forest, XGBoost, Clustering, and Neural Networks.

3.1 LDA & QDA

LDA is used for linear classification problems, while QDA is a variant of LDA that allows for non-linear classification by finding a quadratic boundary that separates different classes. QDA is therefore more flexible and works better with certain data. Our analysis investigates which model works better for this data. Since our data can almost clearly be separated, LDA is a better choice for this problem.

3.2 Logistic Regression

The linear nature of the Logistic Regression model gives us an easily interpretable solution to our classification problem ($p > 0.5$ for 1 and $p < 0.5$ for 0); however, because it does not require tuning/parameter selection, it is hard to find methods to improve upon this model. We apply logistic regression with and without L2-penalty (the model given by the scikit-learn library) and obtain easily interpretable results for feature selection based on the coefficients for each input.

3.3 Clustering

The use of clustering within our analysis is mainly for exploratory data analysis and identification of patterns or relationships within the data, and we want to see whether the process of labeling by 2 main clusters using the k-means clustering model can help improve other predictive methods. Ideally, the model should help if the returned labels have a high enough accuracy compared to our y-test data.

3.4 Tree-based Model

Tree-based classification methods use a series of conditional statements to partition the training data into various final conditions which can be used to predict which class a test data point falls into. In this report, we choose 3 popular tree-based methods - Decision Trees, Random Forests with n_estimators = 500, and XGBoost. Amongst the tree-based models, we observed that XGBoost has the best accuracy and F1 score after tuning it. The best XGBoost model found from the grid search has the following parameters - `('max_depth': 2, 'min_child_weight': 3, 'n_estimators': 40)`.

3.5 Neural Network

To explore the temporal dependencies of intraday Bitcoin price movement, we employed the Long-Short Term Memory (LSTM) framework [F] as the sequential model for the time series classification problem. However, the model is known to struggle from learning long-term trends. Given that our training dataset is small, we need to carefully design the model architecture as deep networks are prone to overfitting. Therefore, instead of stacking multiple LSTM layers to build deeper networks, we use bi-directional LSTM (biLSTM) layers to allow bi-directional information flow during model training while keeping the number of model parameters relatively

small. Leveraging the TensorFlow Keras API, we constructed a 2-layer BiLSTM model (with ReLU activation) and a Dropout layer `(dropout rate = 0.2)`, and the early stopping `(patience = 5)` mechanism. The model is then trained using binary cross entropy loss using Adam optimizer.

4 Results

Traditionally, the Exponential moving average (EMA) is used as a technical indicator to measure trend direction and indicate support and resistance over time. To evaluate the performance of machine learning models, we use EMA (with rolling window = 6) as the benchmark.

LDA vs QDA In Appendix [D], we see that the two classes represented by yellow and purple are separated, in which the centroids are well separated for the binary labels of 0 and 1. However, some data points are overlapping between the centroids. This indicates that the LDA algorithm has difficulties distinguishing between classes with the given set of features. Nevertheless, we can conclude from this graph that the classes are well separated and the data is linearly separable as the data points can be separated by a straight line. As a result, LDA should render a better model for this set of data. This is also supported by LDA's accuracy score which is 7.67% higher than that of QDA. We can also see from the confusion matrix that while 1093 class 1 data points are misclassified as class 0 by QDA, only 600 are misclassified by LDA, which is significantly less. QDA's tendency to predict more 0 labels is also demonstrated through its precision and recall scores, indicating that LDA makes more balanced predictions than QDA.

Tree-based methods Historically, ensemble learning improves model performance. We can conclude that XGBoost performs better than Random Forest which performs better than the Decision Tree. Methods such as Random Forest and XG Boost are also known for feature selection. We observed that from the XGBoost model, only 4 features are required to predict the class with an accuracy rate of 85%. These features were 'eth_change', 'xrp_change', 'ada_change', and 'sum_bullish'. This result is valid since changes in the price of other crypto coins reflects changes in the crypto currency market, which affects the movement of bitcoin. 'sum_bullish' contains keywords drawn from social media indicating optimism or positive outcomes for bitcoin (Figure 2); hence, it is clear that the price movement of bitcoin is dependent on this feature as well.

K-means Clustering Before proceeding with other supervised learning models, we performed clustering to see if our dataset can be labeled using an unsupervised learning algorithm. Unfortunately, labels returned by k-means clustering when compared with our y-test data shows a low accuracy score (50.74%). As a result, we assessed all included models without the assistance of clustering.

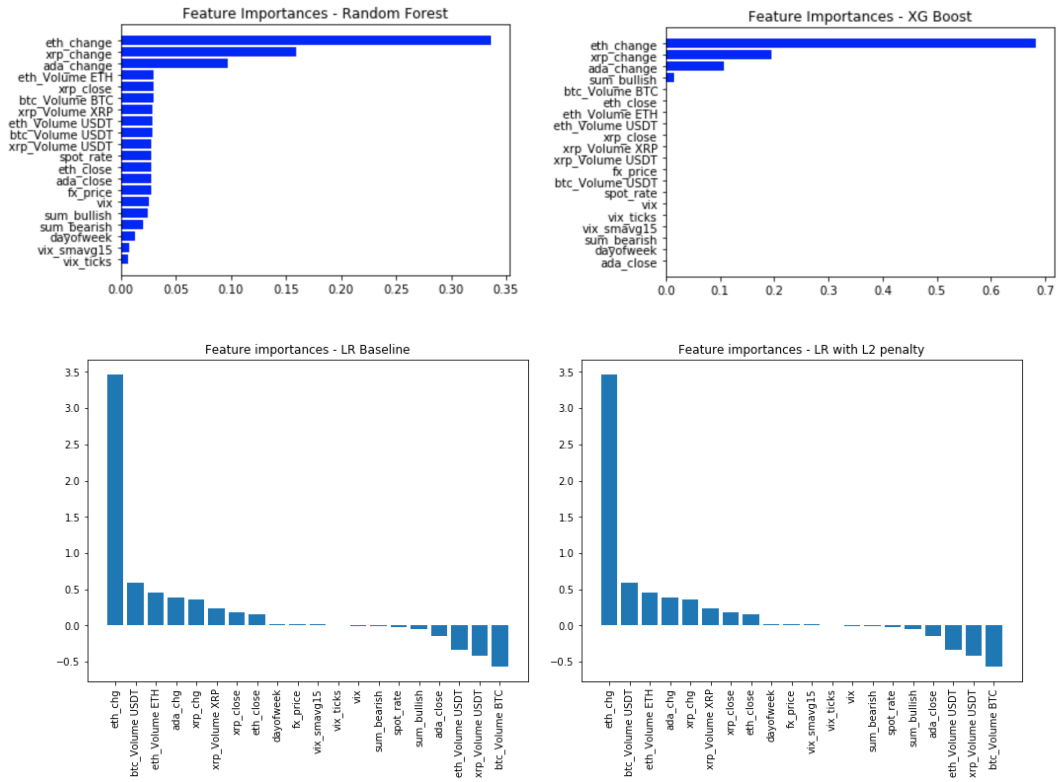


Figure 2. Features importance in logistic regression (bottom) and tree-based (top) method

Logistic Regression This model also gives a close result with other supervised learning methods, which is obtained by collecting the coefficients for each variable and ranking them from largest to smallest. Variables such as ‘eth_change’, ‘btc_Volume’, ‘eth_Volume’, ‘ada_change’, and ‘xrp_change’ are good predictors, whereas other macroeconomic variables such as ‘vix’, ‘fx_price’, and ‘spot_rate’, even though theoretically do affect the cryptocurrency market, do not contribute as much to our model (Figure 3). This also suggests that these variables have high correlation and high collinearity, and thus require more extensive analysis.

Neural Networks The biLSTM model achieves similar training, validation, and test accuracy, which indicates no obvious overfitting problem. Despite having a lower accuracy and F1-score as compared to other models, biLSTM makes a balanced prediction - a small discrepancy in the precision and recall score.

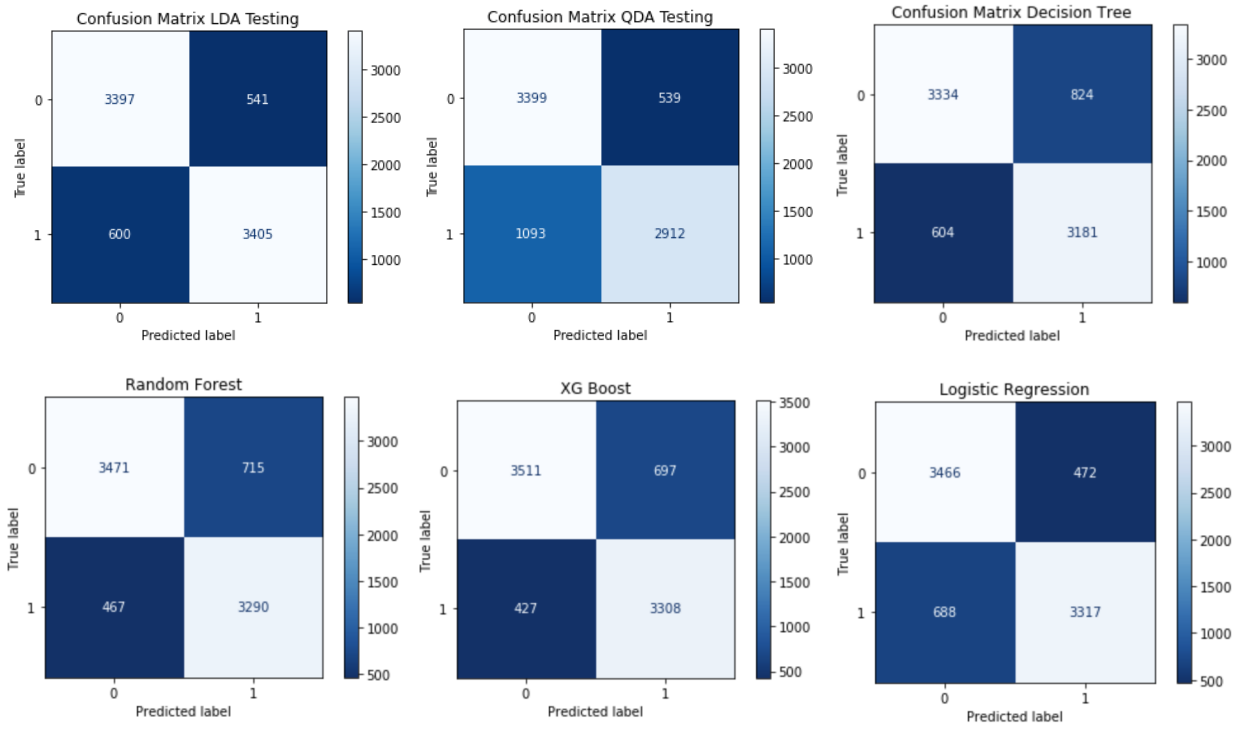
Evaluation among all models While all 8 models outperform the EMA baseline, we see that Logistic Regression and XGBoost have the highest accuracy scores. Based on recall, XGBoost has the highest recall score and Logistic Regression has the highest precision score. Thus, we can safely say that XGBoost and Logistic Regression with Penalty are our 2 best predictors.

We visualize the sensitivity-specificity tradeoff using the ROC curve by comparing the models with random guessing. From Figure 4, we see that the XGBoost model has the best tradeoff

between sensitivity and 1-specificity amongst all tree-based models while Logistic Regression with and without penalty gives us the best tradeoff between sensitivity and 1-specificity.

#	Model	Precision	Recall	F1-score	Accuracy
0	EMA	54.41%	55.45%	54.92%	54.51%
1	Logistic Regression	86.52%	84.06%	85.27%	85.37%
2	Logistic Regression (with L2 penalty)	86.67%	84.02%	85.32%	85.43%
3	LDA	86.26%	84.99%	85.62%	85.60%
4	QDA	86.31%	75.67%	80.64%	79.50%
5	Decision Tree	80.18%	84.66%	82.36%	82.02%
6	Random Forest	82.91%	88.14%	85.45%	85.11%
7	XG Boost	83.43%	89.15%	86.20%	85.84%
8	bi-LSTM	83.85%	85.77%	84.80%	84.77%

Table 1. Comparison of model performance on test data



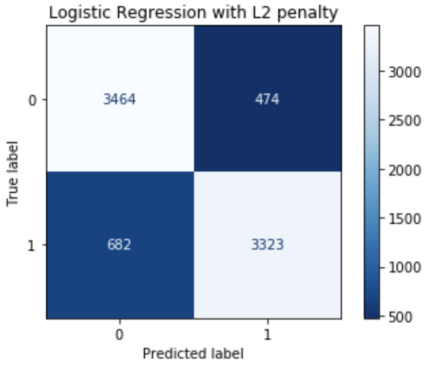


Figure 3. Confusion matrices for prediction on test set

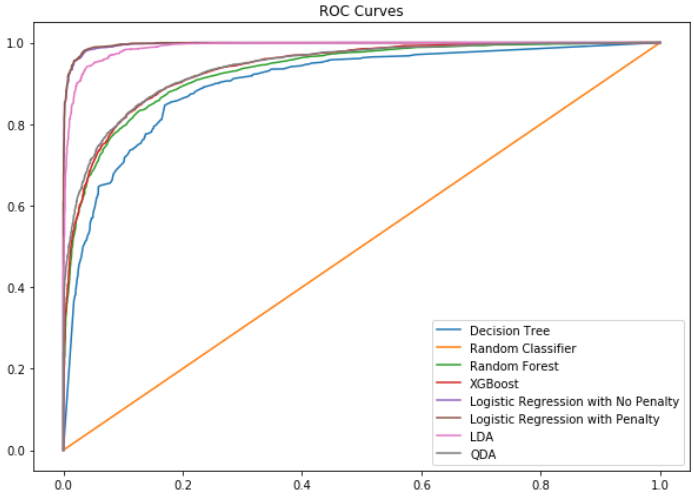


Figure 4. Comparison of ROC curves

5 Discussion

In our empirical analysis, we analyze the short-term predictability of the bitcoin market, leveraging different machine learning models. Based on the evaluation metrics - ROC, F1-score, we conclude that Logistic Regression and XGBoost slightly outperformed other methods. From the feature importances graphs, we see that technical indicators (change of other major cryptocurrencies price) remain prevalent in predicting the Bitcoin price movement.

Traditional time series models like ARIMA were not used as the time series Bitcoin price does not show obvious seasonality during time series decomposition [B]. While more complex models like QDA random forest are known to better capture high-dimensional relationships in the data, they (especially deep learning models) are notoriously data-hungry. Data augmentation techniques and generative models like Generative adversarial networks (GAN) can be used to generate synthetic data for further model training. With a larger dataset and more relevant covariates, more techniques (e.g. attention mechanism in NN) and frameworks (e.g. transformer,

temporal convolutional network) can also be employed to better capture the long-term temporal dependencies of the financial data.

Since hourly data can be highly noisy, we recommend applying smoothing methods like simple exponential smoothing and adaptive bandwidth local linear regression to reduce the noise in the training data. In addition, the data preprocessing method can also be further enhanced by implementing a rolling window approach for train and test sets instead of using a fixed number of data points, which could potentially improve the models significantly given the time series nature of our data.

References

[1] Politis, A.; Doka, K.; Koziris, N. Ether price prediction using advanced deep learning models. In Proceedings of the 2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Sydney, Australia, 3–6 May 2021; pp. 1–3.

[2] Patrick Jaquart, David Dann, and Christof Weinhardt (2021, March 18). *Short-term bitcoin market prediction via machine learning*. The Journal of Finance and Data Science. Retrieved February 28, 2023, from https://www.sciencedirect.com/science/article/pii/S2405918821000027

[3] Aggarwal, Apoorva, Isha Gupta, Novesh Garg, and Anurag Goel. 2019. Deep Learning Approach to Determine the Impact of Socio-Economic Factors on Bitcoin Price Prediction. Paper presented at 2019 Twelfth International Conference on Contemporary Computing (IC3), Noida, India, August 8–10.

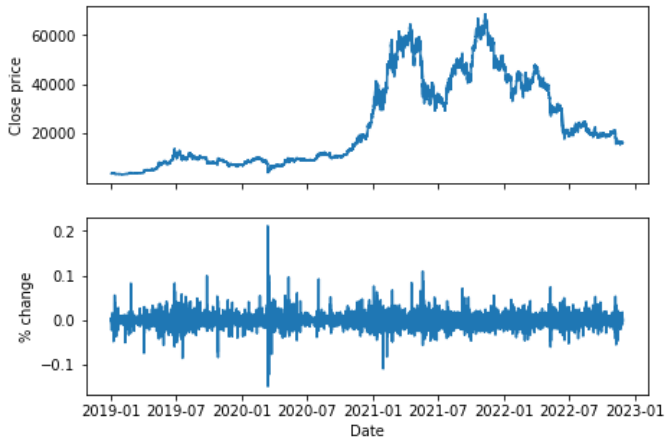
[4] Liu, Mingxi, Guowen Li, Jianping Li, Xiaoqian Zhu, and Yinhong Yao. 2021. Forecasting the price of Bitcoin using deep learning. Finance Research Letters 40: 101755.

[5] “Bitcoin Sentiment – Bull & Bear Index.” Augmento, 14 Apr. 2020, https://www.augmento.ai/bitcoin-sentiment/.

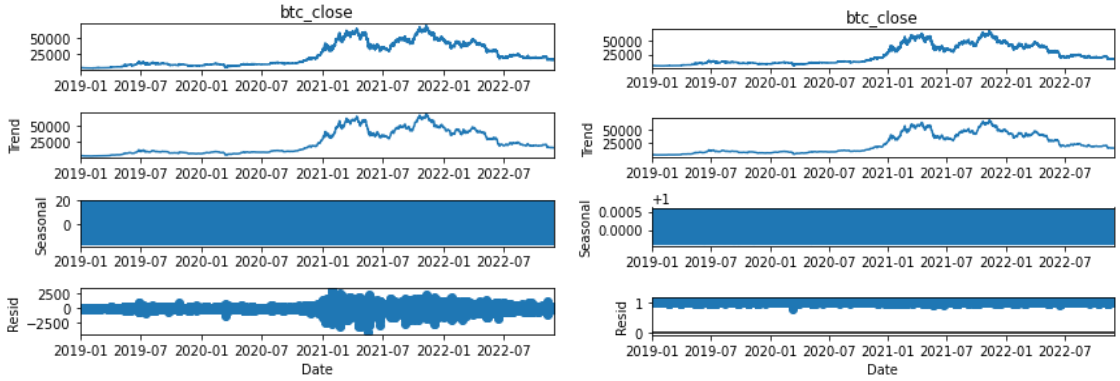
[6] *Crypto Exchange: Bitcoin Exchange: Bitcoin trading*. KuCoin. (n.d.). Retrieved February 28, 2023, from https://www.kucoin.com/

Appendix

[A] Time trend of Bitcoin close price and its hourly percentage change

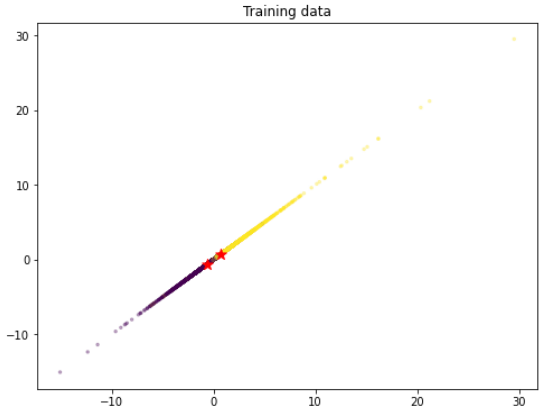


[B] Decompose time series Bitcoin price into the trend, seasonality, and noise using additive (left) and multiplicative (right) models



[C] Partition of samples into training, validation (for NN), and test subsamples.

[D] Visualization of LDA predictions



[E] Model summary for biLSTM framework

```
Model: "sequential_6"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
bidirectional_6 (Bidirectional)  (None, 24, 16)            1664
dropout_4 (Dropout)          (None, 24, 16)            0
bidirectional_7 (Bidirectional)  (None, 8)                 672
dense_2 (Dense)              (None, 1)                 9
=================================================================
Total params: 2,345
Trainable params: 2,345
Non-trainable params: 0
_________________________________________________________________
```

[F] Training and validation history of biLSTM model