Haiman Wong and Yunting Chiu

Professor Hvitfeldt

STAT-627 Statistical Machine Learning

22 June 2021

Statistical Modeling and Machine Learning for Bitcoin Predictions

Abstract

As Wall Street giants, retail investors, and aspiring cryptocurrency trailblazers continue to flood the cryptocurrency market, the ability to predict the volatility of cryptocurrency stocks has proven to be increasingly invaluable. In this report, we detail our methodology that applies statistical machine learning techniques to predict the direction of Bitcoin stocks. Our work also aims to build upon previous research conducted in anticipating trends within cryptocurrency using statistical machine learning methods. To predict whether the direction of Bitcoin stocks will increase or decrease on a given date, we employ Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), K-Nearest Neighbors (KNN), Logistic Regression Analysis, Random Forest, and Decision Trees. We also perform statistical analyses of our dataset using simple linear regression, multiple linear regression, and summary statistics, and visualize these metrics to gain a more comprehensive understanding of the variables that may contribute to deciding the direction of Bitcoin stocks on a given day. Finally, we analyze our results and discuss opportunities for future work and research.

1. Introduction

Since its inception, cryptocurrency has quickly captured global attention across all sectors as investors, financiers, and aspiring cryptocurrency trailblazers alike continue to debate its merits. Cryptocurrencies are known to rely on blockchain technology and are defined as a network-based exchange medium that employs cryptographic algorithms to secure transactions (Patel et. al, 2020). In other words, cryptocurrencies are virtual currencies that are often used to exchange or transfer assets digitally (Khedr, 2021).

Unique from more conventional forms of digital financing, such as online banking, cryptocurrencies are often appealing to prospective investors due to features such as its speed, lack of physical barriers, ability to incorporate cutting-edge technology and innovation, simplicity, transparency, and increasing adoption rates (Mallqui & Fernandes, 2019). However, the trade-off for these aforementioned benefits includes the high volatility that comes with adopting cryptocurrency. This high volatility rate also means that investors may be at a higher risk of losing assets (Higher Education Review, 2021). The decentralized nature of cryptocurrency and its home within our digital world further means that investors must be wary of the increased risk of being a victim to cyberattacks, scams, and the lack of support from a centralized firm that might otherwise be able to help with refunds, reporting, and resetting login credentials (Hagen, 2021). Despite the known potential risks that come with adopting cryptocurrency, the exponential growth of the cryptocurrency market over the past few years demonstrates investors' opinions largely leaning in favor of cryptocurrency adoption.

Today, most researchers, financial experts, and stakeholders agree that Bitcoin remains the most accepted cryptocurrency globally (Patel et. al, 2020; Khedr, 2021; & Mallqui & Fernandes, 2019). While reports indicated that Bitcoin's prices fell to just over $30,000 per Bitcoin in early-June of 2021 before rebounding to $40,000 per Bitcoin by the early onset of mid-June of 2021, Bitcoin still boasts a market capitalization valued at around $750 billion (Bambrough, 2021). Bitcoin's high market capitalization value and ubiquitous adoption rates affirm its standing as the most accepted cryptocurrency globally now and explain why Bitcoin has been dubbed as "digital gold" by some researchers, financial experts, and stakeholders in more recent years. Still, in a market where investors and experts are always on the lookout for the best cryptocurrency trend to invest in, discussions surrounding Bitcoin remain centered on improving our ability to predict cryptocurrency trends accurately and consistently.

In this report, we present a detailed description of the statistical machine learning techniques that we employ to address the persistent problem of enhancing our ability to successfully predict whether Bitcoin stocks will increase or decrease on a given day. Section 2 of this paper provides an overview of related work that our report draws inspiration from, while Section 3 identifies the problem we aim to solve, the models and data analysis that we conducted, the dataset that we chose, and the available features. Section 4 explains the details behind our statistical machine learning methodology and analyzes the results that we obtained. Finally, we conclude our report in Section 5 by discussing opportunities and recommendations for future work and research.

2. Related Work

The research issue of developing strategies to successfully predict cryptocurrency trends remains emerging within the research and academic communities. In fact, work around this issue has been increasing since 2017 due to its instability, dynamic nature, complexity, and ever-growing need for innovative solutions (Patel et. al, 2020). Researchers also point to the growing adoption rates of cryptocurrency globally and more awareness of the multi-faceted layers that exist in cryptocurrency as other reasons why the subject continues to gain popularity and attention within the research and academic communities.

While the specific objectives of existing studies vary widely, most studies on cryptocurrency prediction employ statistical modeling and machine learning techniques. Specifically, many studies implement the Linear Statistical-Model-Based Approach, Random Forest Classification, Recurrent Neural Networks, and Logistic Regression (Alessandretti et. al, 2018; Mallqui & Fernandes; 2019; Patel et. al, 2020; & Akyildirim et. al, 2021). The type of statistical modeling and machine learning used in a study depended on the researchers' primary goal.

For example, one study conducted by researchers, Dennys C.A. Mallqui and Richard A.S. Fernandes, in 2019 aimed to predict the direction of Bitcoin prices and forecast the Bitcoin exchange rates that included values such as the maximum, minimum, and closing prices. To successfully predict these Bitcoin trends, the researchers analyzed the behavior of Artificial Neural Networks (ANN), Support Vector Machines (SVM), and proposed Ensemble algorithms that were based on Recurrent Neural Networks and k-Means clustering techniques (Mallqui & Fernandes, 2019). The ANN and SVM models were also employed to perform regression analyses of the minimum, maximum, and closing prices of Bitcoin prices (Mallqui & Fernandes, 2019). Furthermore, the regression results were also used as inputs in attempts to improve the price direction predictions. Overall, the results demonstrated that the best machine learning model and selected attributes achieved an improvement of more than 10% in accuracy for the price direction predictions when compared to the results found in other studies published around the same time (Mallqui & Fernandes, 2019). The SVM algorithm also proved to perform the best for all predictions of the minimum, maximum, and closing prices because it obtains the lowest mean absolute percentage rates between 1.52% and 1.58%.

Another study published in 2020 by researchers, Mohil Maheshkumar Patel, Sudeep Tanwar, Rajesh Gupta, and Neeraj Kumar focused on proposing a new model for predicting the price of lesser known cryptocurrencies, such as Litecoin and Monero. The new model proposed by these researchers utilizes a Gated Recurrent Unit (GRU) and Long Short-Term Memory hybrid model. Both LSTM and GRU are variants of the Recurrent Neural Network (RNN), which is classified as a type of feed-forward neural network where inputs and outputs are independent of each other (Patel et. al, 2020). In other words, RNN often appears like multiple neural networks arranged side by side with output from one network as input to another (Patel et. al, 2020). The LSTMs and GRUs are used to address RNN's limitations in learning long-term dependencies that are attributed to its cyclic architecture that cause it to suffer from the problem of vanishing gradient (Patel et. al, 2020). Results from this study prove that the LSTM method performed the best due to its ability to extract and remember temporal features of the data (Patel et. al, 2020). Furthermore, the errors of predictions collected from the researchers' proposed hybrid scheme indicate that the hybrid model outperformed even the LSTM network (Patel et. al, 2020). These results mean that the introduction of more complex modeling that incorporates sentiment data proves invaluable for improving the prediction results for cryptocurrencies.

In contrast, a study conducted by researchers, Erdinc Akyildirim, Ahmet Goncu, and Ahmet Sensoy, in 2021 aimed to analyze the predictability of the twelve most liquid cryptocurrencies at the daily and minute level frequencies using machine learning classification algorithms, such as Support Vector Machines (SVM), Logistic Regression, Neural Networks, and Random Forests. To successfully study the predictability of twelve major cryptocurrencies, the researchers also leveraged the aforementioned machine learning classification algorithms over four different time scales that included daily, 15, 30, and 60 minute returns (Akyildirim et. al, 2021). Results indicate that the SVM model performed the best because it yielded a consistent fit above 50%, low variation across all the products and different timescales, and good generalization ability to different sub-periods consistently (Akyildirim et. al, 2021). The researchers also discovered through their experimentation that while the performance of the other classification algorithms were not uniform across different cryptocurrency coins, a predictive accuracy exceeding 69% was often achieved without additional fine tuning or selecting additional variations of the features for each cryptocurrency coin (Akyildirim et. al, 2021). This suggests that machine learning algorithms and features could quite easily achieve over 70% in predictive accuracy with extra fine tuning of model selection steps (Akyildirim et. al, 2021). In other words, the results from this study present heavy implications that machine learning holds a strong potential to forecast short-term trends in cryptocurrency markets.

While each of the studies detailed above aims to predict a trend within the cryptocurrency market, each study also clearly demonstrates how the same baseline machine learning techniques can be combined and leveraged in different ways to achieve various research objectives. Our approach applies statistical modeling and machine learning techniques described in the related work, such as regression analyses, random forest, and logistic regression, to improve predictions of whether Bitcoin prices will increase or decrease. Since one of the core motivations of our work is to practice applying statistical modeling and machine learning techniques within a cryptocurrency context, we draw inspiration from these related studies to gain a deeper understanding of the work that already exists and the gaps that may continue to persist.

3. Project Methodology and Dataset Description

3.1 Defining the Problem, Motivation, and Methodology

Upon researching previous studies that have applied statistical modeling and machine learning techniques to improve cryptocurrency trend predictions, we were inspired to use Bitcoin data to predict the direction of Bitcoin stocks on any given day.

Though the aforementioned problem that we set out to solve may not be wholly unique from previous studies, our motivation for addressing this problem stems from observations that the pervasiveness of cryptocurrency trend predictions will only continue to grow in coming years and that the issues that persist are complex and require effective solutions pressingly. With the fragility of a global market still on the cusp of a full recovery from the setbacks caused by the COVID-19 pandemic and Bitcoin's volatility, we are motivated to apply existing statistical modeling and machine learning techniques to finetune our understanding of its applications in the context of Bitcoin trend predictions and existing strategies that may be improvable. After all, if statistical models and machine learning techniques can accurately predict the trend of cryptocurrency, investors will be able to make more informed and less risky decisions about their investments and purchases.

3.2 Dataset Description

The Bitcoin cryptocurrency dataset that we use for statistical modeling and machine learning was extracted from the website CoinDesk 20, which filters data from thousands of cryptocurrencies and digital assets to define a core group of 20. To produce the publicly available data, CoinDesk 20 includes assets that represent approximately 99% of the market by volume at eight of the largest and most trustworthy exchanges and employs a research-driven approach to select and rank the top 20 assets based on dollar volume and exchange listings that are verifiable.

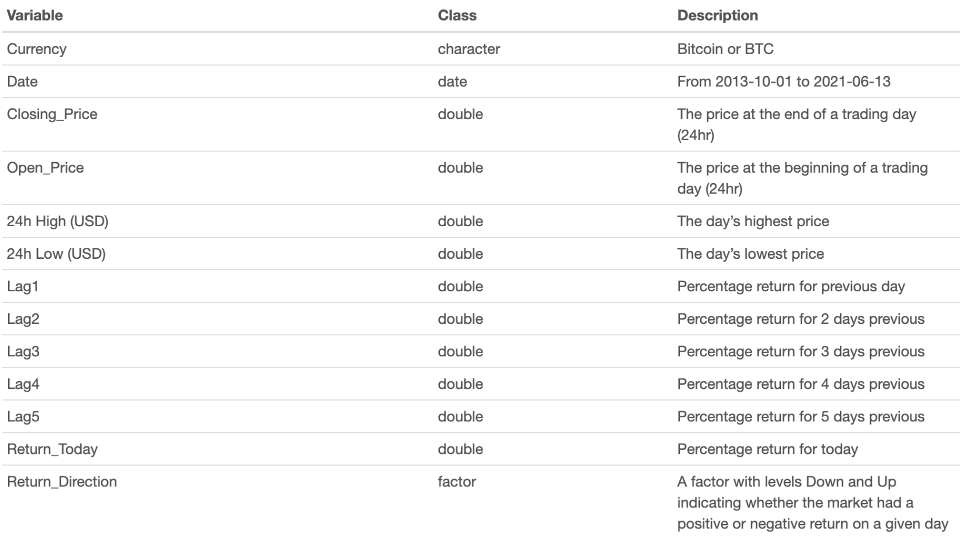
The dataset we used, named "Bitcoin", contains a sample size of 2,808 and 6 unique features. The original 6 features include:

- Currency
- Date
- Closing Price (USD)
- 24h Open (USD)
- 24h High (USD)
- 24h Low (USD)

The dataset and live daily data updates can also be further referenced and downloaded at https://www.coindesk.com/price/bitcoin.

For the purposes of this study, we extract Bitcoin's daily data from CoinDesk 20, ranging from 10/01/2013 to 06/13/2021. Since the dataset that we chose to work with for this report was raw, we implemented preprocessing techniques that included simplifying the names for the "Closing Price (USD)" and "24h Open (USD)" variables to "Closing_Price" and "Open_Price", creating the Return_Today, Direction_Today, and 5 separate Lag variables, and dropping NAs that appeared throughout any variable in the dataset. To conduct these preprocessing steps, we used the dplyr, piping, and drop_na() functions within R. Thus, our pre-processed dataset includes the following 13 features and descriptions, which are all shown in Table 1:

TABLE 1: DATA DICTIONARY



Variable	Class	Description
Currency	character	Bitcoin or BTC
Date	date	From 2013-10-01 to 2021-06-13
Closing_Price	double	The price at the end of a trading day (24hr)
Open_Price	double	The price at the beginning of a trading day (24hr)
24h High (USD)	double	The day's highest price
24h Low (USD)	double	The day's lowest price
Lag1	double	Percentage return for previous day
Lag2	double	Percentage return for 2 days previous
Lag3	double	Percentage return for 3 days previous
Lag4	double	Percentage return for 4 days previous
Lag5	double	Percentage return for 5 days previous
Return_Today	double	Percentage return for today
Return_Direction	factor	A factor with levels Down and Up indicating whether the market had a positive or negative return on a given day

As shown in Table 1, we are also working with labeled data, so our statistical machine learning techniques would classify as supervised machine learning as opposed to unsupervised machine learning.

In *An Introduction to Statistical Learning with Applications in R,* the authors demonstrate how we can use classification methods to predict the daily stock trend by leveraging the Smarket dataset and using the percentage return for the previous two days (James et. al, 2013). We apply a similar approach throughout this report since the trends in the cryptocurrency market and conventional stock markets both present challenges with prediction. However, since the cryptocurrency market is more novel and often more unstable than more conventional stock markets, we create additional variables within our dataset, build more statistical models, and leverage a variety of statistical machine learning techniques to produce more results that we can compare and analyze.

Since our original raw data set has more simplified variables than what was featured in the Smarket data set, we use the mutate() function within R to create additional variables that are better tailored to our research objectives. For example, since we are given the closing price for each day, we can create five new lag variables by using the lag and dplyr functions within R to add columns labeled as “Lag1”, “Lag2”, “Lag3”, “Lag4” and “Lag5”. To create our response variable, Return_Today, we assign the label “Down” to data points that are less than the value zero and “Up” to data points that exceed the value of zero. Finally, we add a new variable named Return_Direction, to save the classifications of “Up” and “Down” so that we can set this variable as a categorical, response variable and begin using statistical methods to predict the direction of Bitcoin prices.

4. Results Obtained

4.1 Data Analysis

Before diving into our statistical modeling and machine learning methodologies, we obtain the summary statistics of all of our quantitative variables to gain a more comprehensive understanding of the data that we are working with. To obtain our summary statistics, we apply the summary() function within R after all of our data pre-processing was completed. Table 2 provides our summary statistic results.

TABLE 2: SUMMARY STATISTICS OF ALL QUANTITATIVE VARIABLES

Variable	**Date**	**Closing _Price**	**Open_Price**	**24h High (USD)**	**24h Low (USD)**	**Lag1**	**Lag2**	**Lag3**	**Lag4**	**Lag5**	**Return_ Today**
Min	2013-10-06	120.7	120.7	121.8	120.4	-27.08	-27.08	-27.08	-27.08	-27.08	-27.08
1st Q.	2015-09-07	468.4	467.7	478.0	454.4	-1.32	-1.30	-1.31	-1.30	-1.30	-1.31
Median	2017-08-09	3259.1	3233.5	3343.4	3157.2	0.16	0.16	0.16	0.16	0.16	0.16
Mean	2019-08-09	6834.6	6820.8	7032.0	6588.5	0.30	0.30	0.30	0.30	0.30	0.30
3rd Q.	2019-07-12	8697.1	8694.5	8905.4	8381.3	1.96	1.96	1.96	1.96	1.96	1.96

Max.	2021-06-13	63346.8	63562.7	64801.8	62094.6	35.85	35.85	35.85	35.85	35.85	35.85

As seen in Table 2, the summary statistics provide a snapshot of the minimum, first quartile, median, mean, third quartile, and maximum values for each of the quantitative variables. Here, we observe that the minimum, median, mean, third quartile, and maximum values for all five Lag variables are the same. The first quartile values for all five of the Lag variables are also within 0.01 of each other.

Furthermore, we can also observe that the minimum values for Closing_Price and Open_Price are exactly the same. The first quartile, median, mean, third quartile, and maximum values are also similar between the Closing_Price and Open_Price, which suggests that there is a relationship between the Open_Price and Closing_Price. This would make sense since a lower Open_Price on a given day would likely cause a lower Closing_Price and vice versa. Similarly, when we analyze the summary statistics for 24h High (USD) and 24h Low (USD), we observe close minimum, first quartile, median, mean, third quartile, and maximum values. This also suggests that there is a relationship between the 24h High (USD) and 24h Low (USD) variables, where a lower value on a given day would cause the other "high" of that day to also be lower than average. Finally, the Return_Today variable appears to mirror the minimum, first quartile, median, mean, third quartile, and maximum values of the five Lag variables. This trend can be attributed to the fact that these summary statistical values are the same even on different dates. For example, if we are reviewing the data for a Friday, the Lag1 variable would represent all the summary statistic values from Thursday. In this case, the Lag2 variable would represent all the summary statistic values from Tuesday, and so forth. This relationship between all five Lag variables explains why the Return_Today variable appears to mirror the same minimum, first quartile, median, mean, third quartile, and maximum values of the five Lag variables.

Another way to gain a better understanding of the data is to evaluate the correlation between each of our quantitative variables. To find the correlation values, we employ the cor() function within R. All of the correlation values are rounded to the thousandth degree in our table. Table 3 captures all of the correlation values between the quantitative variables in our data set.

TABLE 3: CORRELATION OF ALL QUANTITATIVE VARIABLES

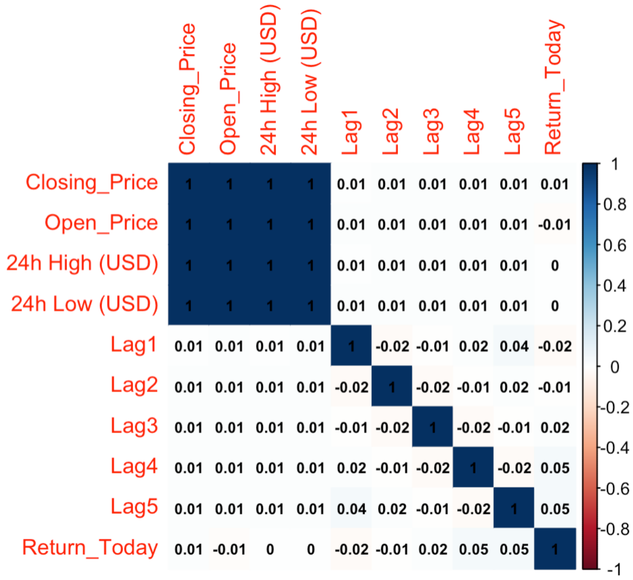
Variable	Closing_Price	Open_Price	24h High (USD)	24h Low (USD)	Lag1	Lag2	Lag3	Lag4	Lag5	Return _Today
Closing_Price	1.000	0.999	0.999	0.999	0.010	0.012	0.012	0.014	0.011	0.013
Open_Price	0.999	1.000	0.999	0.999	0.011	0.010	0.012	0.013	0.011	-0.014
24h High (USD)	0.999	0.999	1.000	0.999	0.010	0.010	0.012	0.013	0.011	0.000
24 Low (USD)	0.999	0.999	0.999	1.000	0.013	0.0123	0.012	0.014	0.011	0.003
Lag1	0.010	0.011	0.010	0.013	1.000	-0.024	-0.010	0.016	0.044	-0.023
Lag2	0.012	0.010	0.010	0.012	-0.021	-1.000	-0.024	-0.010	0.017	-0.009

Lag3	0.013	0.012	0.012	0.012	-0.010	-0.024	1.000	-0.024	-0.010	0.015
Lag4	0.014	0.013	0.012	0.014	0.016	-0.010	-0.024	1.000	-0.023	0.046
Lag5	0.011	0.011	0.011	0.011	0.044	0.017	-0.010	-0.023	1.000	0.045
Return_Today	0.013	-0.014	0.000	0.003	-0.022	-0.009	0.015	0.046	0.045	1.000

As shown in Table 3, the correlation values appear to be the highest between Closing_Price and Open_Price with a value of 0.999. The 24h High (USD) and 24h Low (USD) also boast a high correlation value of 0.999. These correlation results also affirm the conclusions about the potential relationships between variables drawn from our summary statistics shown before in Table 1.

While obtaining the raw correlation values is helpful, creating a correlation table is often another effective way to efficiently analyze the relationship between variables. To create the correlation table, we employed the corrplot() function in R and adjusted the color and size of our plot. Table 4 displays the correlation table that was created.

TABLE 4: CORRELATION TABLE



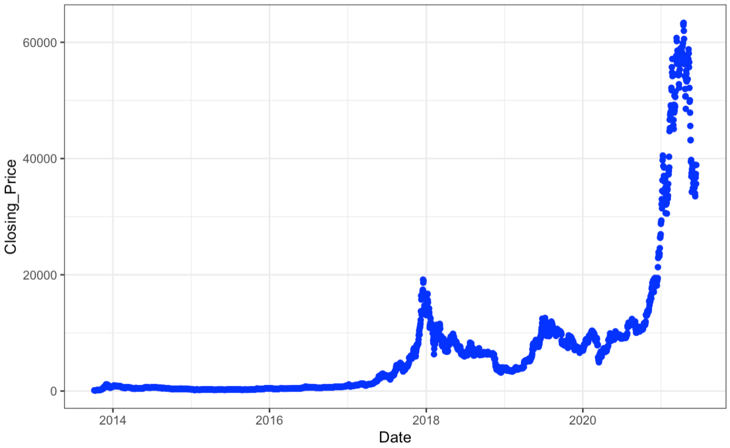
	Closing_Price	Open_Price	24h High (USD)	24h Low (USD)	Lag1	Lag2	Lag3	Lag4	Lag5	Return_Today
Closing_Price	1	1	1	1	0.01	0.01	0.01	0.01	0.01	0.01
Open_Price	1	1	1	1	0.01	0.01	0.01	0.01	0.01	-0.01
24h High (USD)	1	1	1	1	0.01	0.01	0.01	0.01	0.01	0
24h Low (USD)	1	1	1	1	0.01	0.01	0.01	0.01	0.01	0
Lag1	0.01	0.01	0.01	0.01	1	-0.02	-0.01	0.02	0.04	-0.02
Lag2	0.01	0.01	0.01	0.01	-0.02	1	-0.02	-0.01	0.02	-0.01
Lag3	0.01	0.01	0.01	0.01	-0.01	-0.02	1	-0.02	-0.01	0.02
Lag4	0.01	0.01	0.01	0.01	0.02	-0.01	-0.02	1	-0.02	0.05
Lag5	0.01	0.01	0.01	0.01	0.04	0.02	-0.01	-0.02	1	0.05
Return_Today	0.01	-0.01	0	0	-0.02	-0.01	0.02	0.05	0.05	1

The correlation table shown in Table 4 affirms the conclusions drawn before that were based on our raw correlation values and the ones we extrapolated from our summary statistics in Table 2. The correlations between the Lag variables and today's returns are close to zero. In other words, there appears to be little association between today's returns and those from previous days. The Closing_Price, Open_Price, 24 High (USD), and 24 Low (USD) variables have substantial correlations, so we exclude these four as predictors in our later statistical modeling and machine learning. Furthermore, since the formula for the Return_Today variable is (Closing_Price - Open_Price) / Open_Price*100, it makes little sense to use Return_Today as a predictor variable because this is the target we want to predict. Here, our main objective is to find the related coefficients that will allow us to best predict Bitcoin's trends by using the percentage return over the previous five days. Now that we have a better understanding of the variables that are in the dataset and the relationships that may exist between them, Section 4.2 will feature additional data visualizations that also shed further light on trends and relationships that exist between variables.

4.2 Data Visualization

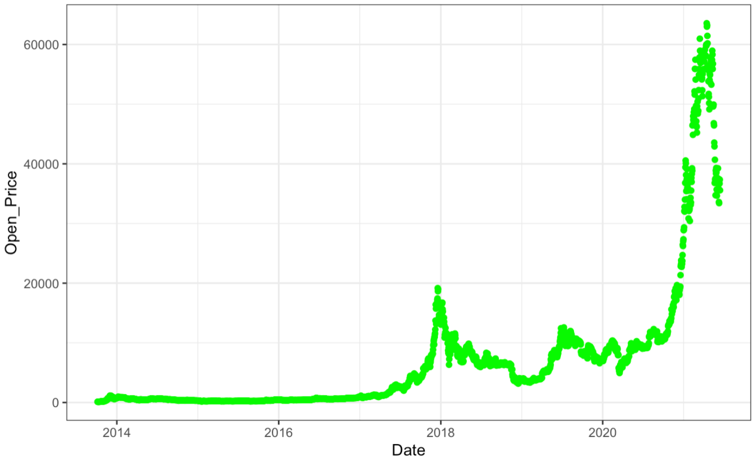
To analyze the relationship between the Closing_Price variable and the Date variable, we apply the ggplot function within R to create the scatter plot that is featured below in Table 5.

TABLE 5: SCATTERPLOT BETWEEN DATE AND CLOSING PRICE



Similar to the scatterplot created above, we use the ggplot function again to produce a scatter plot between the Open_Price and the Date variables. The scatter plot is shown below in Table 6.

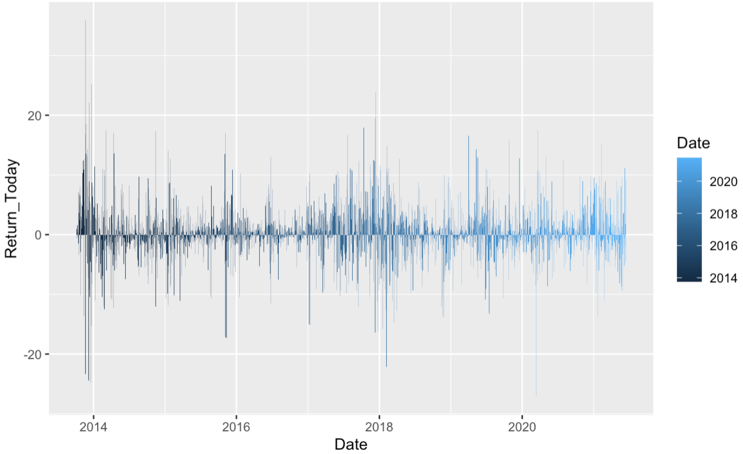
TABLE 6: SCATTERPLOT BETWEEN DATE AND OPENING PRICE



When we compare both Tables 5 and 6, we observe that the trends align. This affirms that when the Open Price on a given day is lower, the Closing Price will match that trend and also be lower. Likewise, when the Open Price on a given day is higher, the Closing Price will align with that trend and also be higher. We can also observe that there are three noticeably visible peaks in Opening and Closing Price in 2018, the end of 2019, and mid-2021.

Now, using the ggplot function again in R, we create the bar graph featured in Table 7 to visualize the Return_Today variable's relationship with the Date variable:

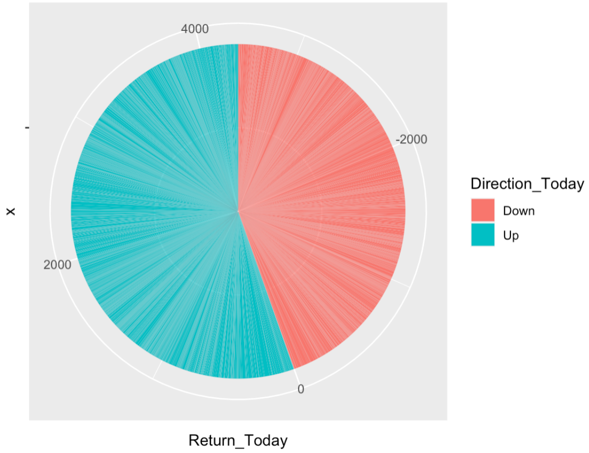
TABLE 7: BARPLOT FREQUENCY BETWEEN DATE AND RETURN_TODAY



The legend shown above on Table 7 indicates that the overall returns day to day had a consistent trend of increases and decreases over the years. While some decreases and increases significantly outweighed others, the overarching trend on this bar graph shows that the frequency of returns increasing and decreasing everyday is variable.

Next, we turn to the pie chart featured in Table 8. To create the pie chart featured in Table 8, we employed the ggplot functions within R. Table 8 successfully visualizes the frequency that the direction of Bitcoin stocks increases and decreases overall:

TABLE 8: PIE CHART FREQUENCY OF DIRECTION_TODAY



Based on Table 8, we see that while the pie chart is almost a 50/50 split between increases and decreases, there are still slightly more days where the direction of Bitcoin stocks increases as opposed to decreases.

4.3 Regression Analyses

Regression analyses are used to determine whether there is a relationship between the response and predictor variables, how strong existing relationships are, whether there is linearity, and how accurately predictions can be made. In this section, we focus on simple linear regression and multiple regression techniques.

Table 9 shows the results of five independent simple linear regression models. Each independent simple linear regression model featured Return_Today as the response variable and one of the Lag variables as a predictor variable. Since there are five Lag variables in our dataset, we feature five independent simple linear regression model variations. For example, the first simple linear regression model features the Return_Today variable as the response and Lag1 as the predictor variable. In contrast, the second simple linear regression model features Lag2 as the predictor variable instead of Lag1, but maintains the Return_Today variable as the response. This strategy is repeated for the third, fourth, and fifth simple linear regression models.

TABLE 9: SIMPLE LINEAR REGRESSION SUMMARY RESULTS

	Estimate	**Std. Error**	**t-value**	**p-value**	**Residual Standard Error**	**Multiple R-Squared**	**Adjusted R-squared**	**F-Statistic**	**Overall p-value**
M1 Intercept	0.30557	0.08052	3.795	0.000151	4.256 on 2806 degrees of freedom	0.0005078	0.0001516	1.426 on 1 and 2806 DF	0.2326
M1 Lag1	-0.02255	0.01889	-1.194	0.232566					
M2 Intercept	0.301610	0.80539	3.745	0.000184	4.257 on 2806 degrees of freedom	8.164e-05	-0.0002747	0.2291 on 1 and 2806 DF	0.6322
M1 Lag2	-0.009037	0.018881	-0.479	0.632238					
M3 Intercept	0.29437	0.08052	3.656	0.000261	4.257 on 2806 degrees of freedom	0.0002343	-0.000122	0.6576 on 1 and 2806 DF	0.4175
M3 Lag3	0.01528	0.01884	0.811	0.417479					
M4 Intercept	0.28518	0.8045	3.545	0.000399	4.253 on 2806 degrees of freedom	0.002138	0.001782	6.011 on 1 and 2806 DF	0.01428
M4 Lag4	0.04616	0.01883	2.452	0.014277					
M5 Intercept	0.28564	0.08045	3.551	0.000391	4.253 on 2806 degrees of freedom	0.002048	0.001692	5.758 on 1 and 2806 DF	0.01648
M5 Lag5	0.04523	0.01885	2.400	0.016482					

As shown in Table 9, the first simple linear regression model results are shown on the rows that correspond with “M1 Intercept” and “M1 Lag1”. The summary results of our first simple linear regression model yield an adjusted r-squared value of 0.0001516 and a multiple r-squared value of 0.0005078, our results suggest that the model is not doing the best at explaining the variability in the response variable. Our low adjusted r-squared value also indicates that our model is not telling us very much about how much of the variability for the response variable is due to explanatory variables that have impact on the response variables. Our p-value of 0.2326 is also greater than our standard significance level of 0.05, which suggests that we fail to reject the null hypothesis and our model is not significant. We can also observe that the predictor variable is not significant here because its p-value is 0.232566, which is greater than our standard significance level of 0.05.

In comparison, the summary results of our second simple linear regression model boast an adjusted r-squared value of -0.0002747 and a multiple r-squared value of 8.164e-05, our results suggest that the model is not doing the best at explaining the variability in the response variable. Our low adjusted r-squared value also indicates that our model is not telling us very much about how much of the variability for the response variable is due to explanatory variables that have impact on the response variables. Our p-value of 0.6322 is also greater than our standard significance level of 0.05, which suggests that we fail to reject the null hypothesis and our model is not significant. We can also observe that the predictor variable is not significant here because its p-value is 0.632238, which is greater than our standard significance level of 0.05.

The third simple linear regression model summary results indicate that the adjusted r-squared value is -0.000122 and that the multiple r-squared value is 0.0002343. These results suggest that the model is not doing the best at explaining the variability in the response variable. Our low adjusted r-squared value also indicates that our model is not telling us very much about how much of the variability for the response variable is due to explanatory variables that have impact on the response variables. Our p-value of 0.4175 is also greater than our standard significance level of 0.05, which suggests that we fail to reject the null hypothesis and our model is not significant. We can also observe that the predictor variable is not significant here because its p-value is 0.417479, which is greater than our standard significance level of 0.05.

In contrast, the fourth simple linear regression model produces an adjusted r-squared value of 0.001782 and a multiple r-squared value of 0.002138, our results suggest that the model is not doing the best at explaining the variability in the response variable. Our low adjusted r-squared value also indicates that our model is not telling us very much about how much of the variability for the response variable is due to explanatory variables that have impact on the response variables. Our p-value of 0.01428 is less than our standard significance level of 0.05, which suggests that we can reject the null hypothesis and our model is significant. We can also observe that the predictor variable is significant here because its p-value is 0.014277, which is less than our standard significance level of 0.05.

Finally, the fifth simple linear regression model yields an adjusted r-squared value of 0.001692 and a multiple r-squared value of 0.002048, our results suggest that the model is not doing the best at explaining the variability in the response variable. Our low adjusted r-squared value also indicates that our model is not telling us very much about how much of the variability for the response variable is due to explanatory variables that have impact on the response variables. Our p-value of 0.01648 is less than our standard significance level of 0.05, which suggests that we can reject the null hypothesis and our model is significant. We can also observe that the predictor variable is significant here because its p-value is 0.016482, which is less than our standard significance level of 0.05.

Overall, the fourth and fifth simple linear regression models performed the best when compared to all of the simple linear regression models because these were the only two models that proved to be significant based on the p-values. Still, there is room for the models all across the board to be better fits for our data-set since the adjusted r-squared and multiple r-squared values were low across all variations of our simple linear regression models. There also appears to be a trend where the simple linear regression models are significant when our predictor variables are significant and insignificant when our predictor variables are insignificant. By evaluating the relationships between each predictor variable and our response variable independently, we can better ascertain how each predictor variable may influence our other statistical machine learning models.

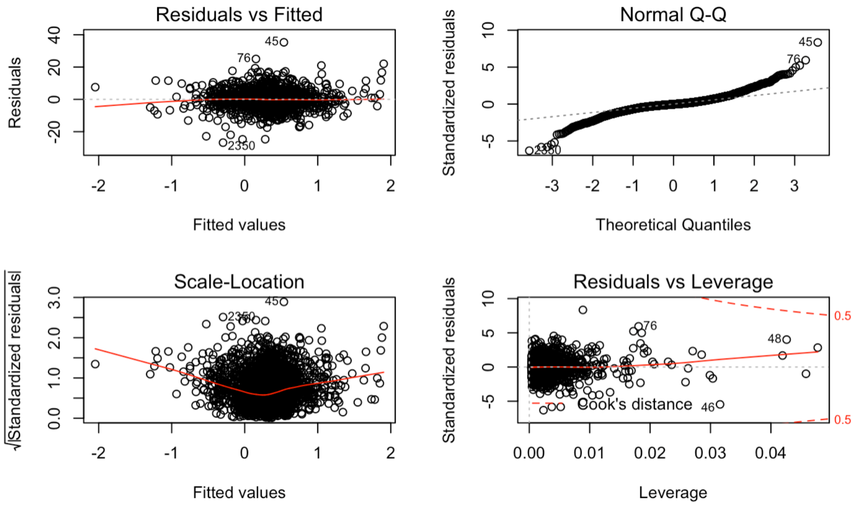
To more holistically evaluate how all the predictor variables relate to the response variable, we also perform multiple linear regression. The summary results of our multiple linear regression model are shown in Table 10.

TABLE 10: MULTIPLE LINEAR REGRESSION SUMMARY RESULTS

	Estimate	**Std. Error**	**t-value**	**p-value**	**Residual Standard Error**	**Multiple R-Squared**	**Adjusted R-squared**	**F-Statistic**	**Overall p-value**
Intercept	0.276195	0.081172	3.403	0.000677	4.249 on 2802 degrees of freedom	0.005297	0.003522	2.984 on 5 and 2802 DF	0.01084
Lag1	-0.025475	0.018883	-1.349	0.177409					
Lag2	-0.009517	0.018859	-0.505	0.613849					
Lag3	0.016422	0.018823	0.872	0.383055					
Lag4	0.047975	0.018825	2.548	0.010874					
Lag5	0.047803	0.018860	2.535	0.011313					

With an adjusted r-squared value of 0.003522 and a multiple r-squared value of 0.005297, our multiple linear regression results suggest that the model is not doing the best at explaining the variability in the response variable. Our low adjusted r-squared value also indicates that our model is not telling us very much about how much of the variability for the response variable is due to explanatory variables that have impact on the response variables. However, our p-value of 0.01084 is less than our standard significance level of 0.05, which suggests that we can reject the null hypothesis and our model is significant. We can also observe that only two of our predictor variables, Lag4 and Lag5, are less than our standard significance level of 0.05 and significant. Using the plot() function within R, we now plot our fitted multiple regression model to obtain the four diagnostic plots featured below in Table 11:

TABLE 11: MULTIPLE LINEAR REGRESSION DIAGNOSTIC PLOTS

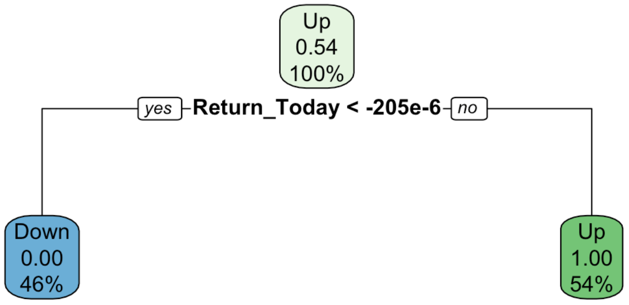


Overall, the four diagnostic plots shown in Table 11 suggest that our model is a moderately good fit for our data. There is evidence of some unequal variance and distribution in the Scale-Location and Residuals vs. Leverage plots, but the points still largely follow the fitted line and exhibit some strength in randomization. The QQ plot and Residuals vs. Fitted plots also indicate that our points are fairly normally distributed. This affirms that our multiple linear regression model is a fair fit for our data.

4.4 Decision Trees

To create our first decision tree model using the default hyperparameters, we begin by setting a seed and creating the initial split, training data set, and testing data set. Next, we build the decision tree model specification by setting the engine equal to rpart and the mode equal to classification. This then allows us to fit our decision tree model to our training data set. Finally, we plot our first decision tree model using the rpart.plot() function in R. Table 12 shows the decision tree plot that we obtained from this first decision tree model.

TABLE 12: DECISION TREE MODELS 1 AND 2 PLOT

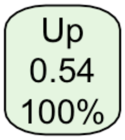


Based on Table 12, we see that the decision tree has one visible split and that values that are less than -205e-6 are classified as a downward direction, while values that are greater than -205e-6 are classified as an upward direction. We also see that our results align with the frequency that we observed in our pie chart because there are slightly more classifications of Bitcoin stocks increasing at 54% than decreasing at 46%.

Next, we build a second decision tree model by tuning the hyperparameter where cost_complexity is equal to 0.5. After tuning the hyperparameter, we can fit and plot the decision tree model against the training data as we did before with the first decision tree model. In doing so, we observe that the decision tree plot produced by our second decision tree model is the same as the first decision tree model. Thus, we can draw the same conclusions that we detailed above under Table 12.

Our third decision tree model features a tuned hyperparameter where cost_complexity is set equal to 3. Again, after we tune the hyperparameter, we can fit and plot the decision tree model against the training data. Table 13 below shows the decision tree plot from our third model:

TABLE 13: DECISION TREE MODELS 3 AND 4 PLOT



Finally, we build our fourth decision tree model by tuning the hyperparameter cost_complexity to equal 1.5, which is a more moderate value than the cost_complexity value of 3 that we featured in our third decision tree model. Table 13 above also shows the decision tree plot from our fourth model. Based on Table 13, we observe that models three and four of our decision tree showed the most dramatic effects.

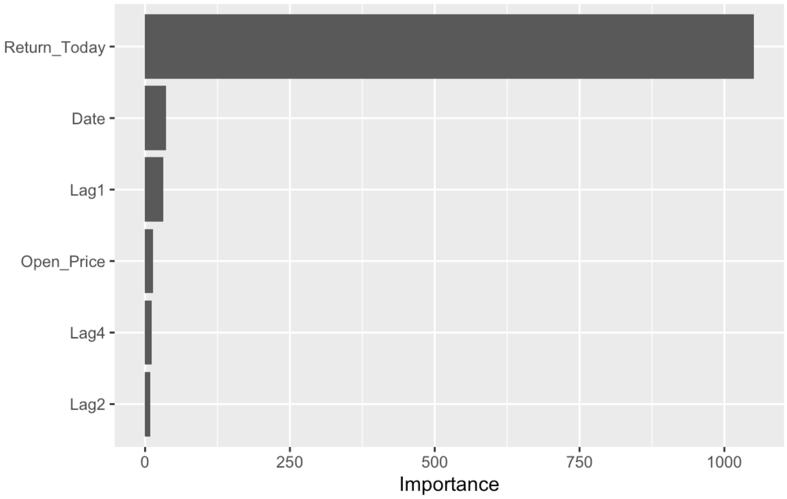
Here, it is clear that tuning the hyperparameter of cost_complexity to 3 dramatically affected our decision tree model results. There is barely a visible split here and only the 54% increase is shown. This may also suggest that the tuned hyperparameters of this decision tree model are not that good of a fit for our data since it fails to yield results that meaningfully predict our Bitcoin stock directions. Even when we tried to choose a more moderate hyperparameter of cost_complexity with a value of 1.5, the decision tree plot showed the same results of the hyperparameter of cost_complexity with a value of 3.

Now that we've run four variations of the decision tree model, we can use the vi() function to find the variable importance for the first decision tree model that we created because it appears to have one of the best fits with the default hyperparameters. Table 14 shows the raw variable importance values, while Table 15 features the variable importance plot.

TABLE 14: DECISION TREE VARIABLE IMPORTANCE VALUES

Variable	**Importance**
Return_Today	1050.90
Date	36.56
Lag1	31.34
Open_Price	13.58
Lag4	11.50
Lag2	9.40

TABLE 15: DECISION TREE VARIABLE IMPORTANCE PLOT



Based on both Tables 14 and 15, the Return_Today variable is the most important. The results in the plot also align with the variable importance values in Table 14 that we obtained when we used the vi() function and applied our best fitted decision tree model in R.

4.5 Random Forest

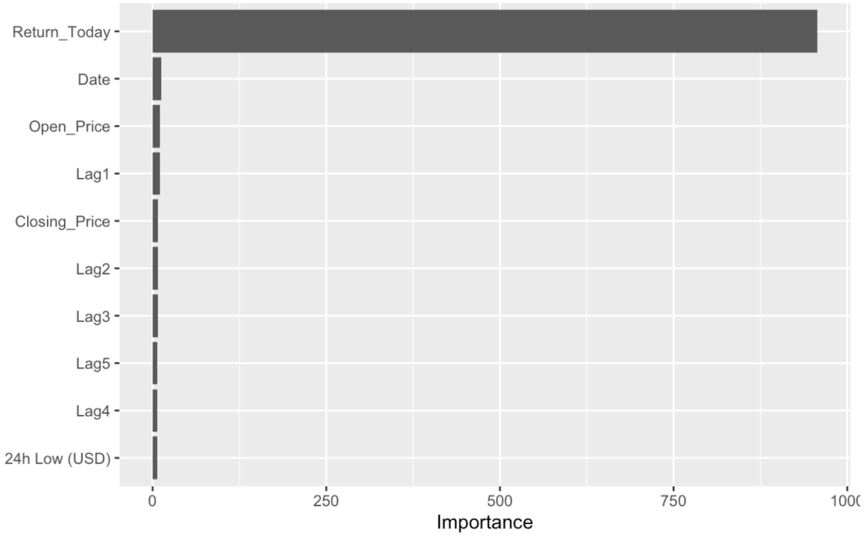
Similar to the decision tree models, we set a seed and use the default split in both the training and testing datasets for constructing our random forest model. To create the random forest specification, the mode was set to classification and the engine was set to ranger. When fitting the random forest, the gini criterion was used as the splitrule and the variable importance mode was impurity. The target node size was 10, and the number of trees in this random forest model was 500.

Overall, the predictor error was 0.001833006, which proves to be significantly small and suggests that this model is likely a good fit for our data and for predictions. After fitting the random forest model, we calculated the variable importance and plotted the variable importance, which is shown below in Tables 16 and 17.

TABLE 16: RANDOM FOREST VARIABLE IMPORTANCE VALUES

Variable	**Importance**
Return_Today	956.34
Date	12.49
Open_Price	10.51
Lag1	10.18
Closing_Price	7.80
Lag2	7.72
Lag3	7.50
Lag5	6.99
Lag4	6.50
24h Low (USD)	6.25
24h High (USD)	6.20
Currency	0.00

TABLE 17: RANDOM FOREST VARIABLE IMPORTANCE PLOT

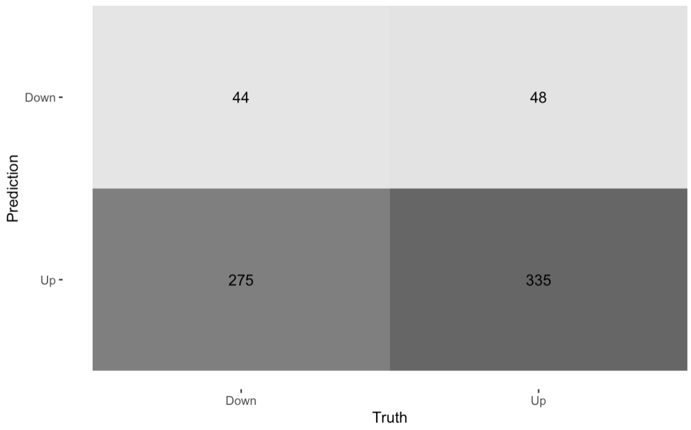


Based on the plot shown in Table 17, we can observe that our response variable, Return_Today holds a significantly higher "importance" value than any of our predictor variables. The results in the plot also align with the variable importance values in Table 16 that we obtained when we used the vi() function and applied our fitted random forest model in R. This result affirms that the random forest model is a likely good fit for our data.

4.6 Logistic Regression Model 1

Based on the results of our exploratory data analysis, regression analysis, decision trees, and random forest model above, we should choose Lag1, Lag2, Lag3, Lag4, and Lag5 as predictors. As shown in the confusion matrix in Table 18, the logistic regression model operates the five Lag variables as predictors.

TABLE 18: LOGISTIC REGRESSION 1 CONFUSION MATRIX

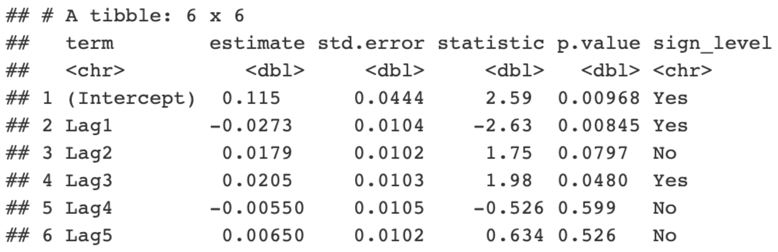


The predictions are represented by the rows of the confusion matrix, while the ground truth is represented by the columns. The upper-left corner represents true Down, and the lower-right area represents true Up; these two values are leading indicators for determining whether the model is right or wrong. The true Down is 44, which is less than the false Down 48. In contrast, the true Up is 335, which is greater than the false negative value of 275. When we compare these results to the observed data points, we observe that the predictions have room for improvement. The model's overall accuracy is 54.0%. To improve this model, we can check the coefficient table.

4.7 Coefficients Table

In the coefficients table of logistic regression shown below in Table 19, only two predictors are significant. We know that predictors with a high p-value often cause an increase in variance without a corresponding decrease in bias, so we opt to only keep the Lag1 and Lag3 variables as the model’s predictors.

TABLE 19: COEFFICIENTS TABLE IN LOGISTIC REGRESSION



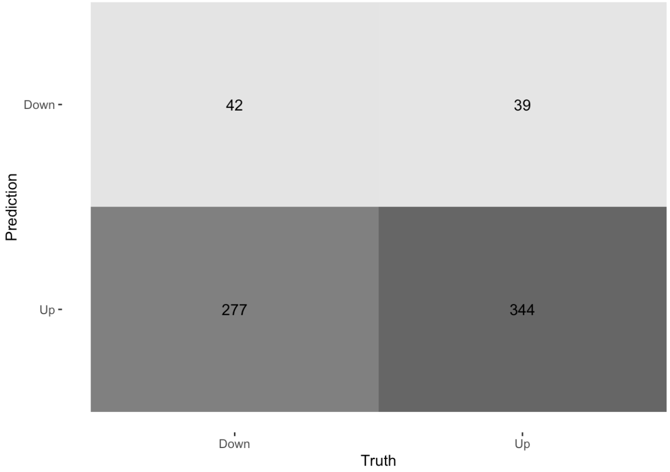
```
## # A tibble: 6 x 6
##   term        estimate std.error statistic p.value sign_level
##   <chr>          <dbl>     <dbl>     <dbl>   <dbl> <chr>
## 1 (Intercept)  0.115      0.0444     2.59  0.00968 Yes
## 2 Lag1        -0.0273     0.0104    -2.63  0.00845 Yes
## 3 Lag2         0.0179     0.0102     1.75  0.0797  No
## 4 Lag3         0.0205     0.0103     1.98  0.0480  Yes
## 5 Lag4        -0.00550    0.0105    -0.526 0.599   No
## 6 Lag5         0.00650    0.0102     0.634 0.526   No
```

As shown in Table 19, there is also no clear evidence that a real association exists between the Lag2 variable and the Return_Direction variable, the Lag4 variable and the Return_Direction variable, or the Lag5 variable and the Return_Direction variable.

4.8 Logistic Regression Model 2

Based on the analysis and reasoning provided in Sections 4.6 and 4.7, this second logistic regression model will only operate the two Lag variables, Lag1 and Lag3, as predictors. Table 20 also features the confusion matrix associated with this second logistic regression model.

TABLE 20: LOGISTIC REGRESSION 2 CONFUSION MATRIX



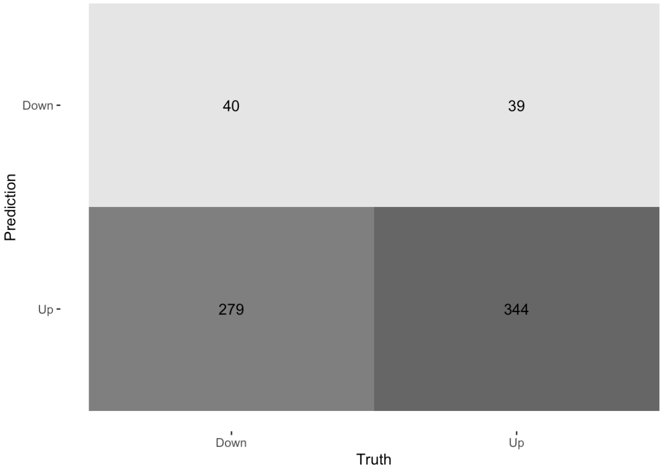
Prediction \ Truth	Down	Up
Down	42	39
Up	277	344

Based on Table 20, we observe that the true Down is 42, which is greater than the false Down value of 39. The true Up value of 344 is also greater than the false negative value of 277. Keeping only the significant predictors in this logistic regression model, the model's overall accuracy increased to 54.9%.

4.9 Linear Discriminant Analysis (LDA)

To conduct our LDA analysis, we only retain the Lag1 and Lag3 predictor variables to fit the models on the remaining statistical models. In other words, all of the other predictor variables, Lag 2, Lag 4, and Lag 5, were all omitted due to the insignificance that was observed surrounding these predictors in earlier data analyses. Here, Table 21 captures the confusion matrix results for the LDA model.

TABLE 21: LDA MODEL CONFUSION MATRIX



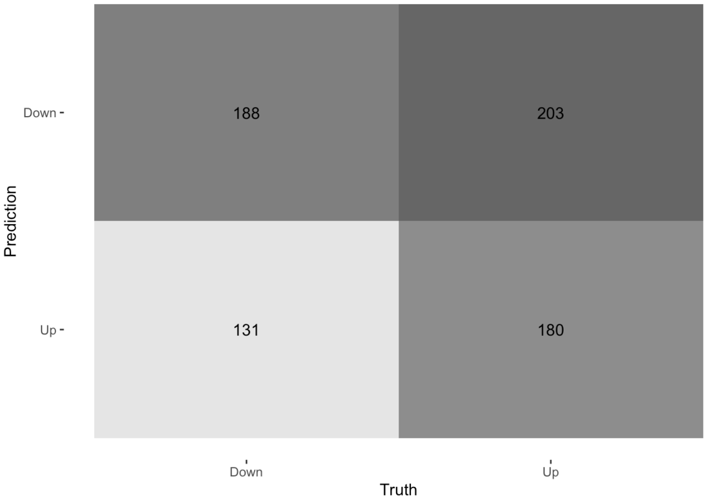
Prediction \ Truth	Down	Up
Down	40	39
Up	279	344

According to the LDA confusion matrix shown in Table 21, the true Down is 40, which is greater than the false Down value of 39. The true Up value of 344 is also greater than the false negative value of 279 in the LDA model. Here, the model demonstrates a fair fit and a comparable accuracy when compared to the testing data set because it yields an accuracy rate of 54.7%, which is very close to the outcome of Logistic Regression Model 2 that also performed well overall.

4.10 Quadratic Discriminant Analysis (QDA)

To conduct our QDA analysis, we employ the same methodology that was described above. Here, Table 22 captures the confusion matrix results for the QDA model.

TABLE 22: QDA MODEL CONFUSION MATRIX



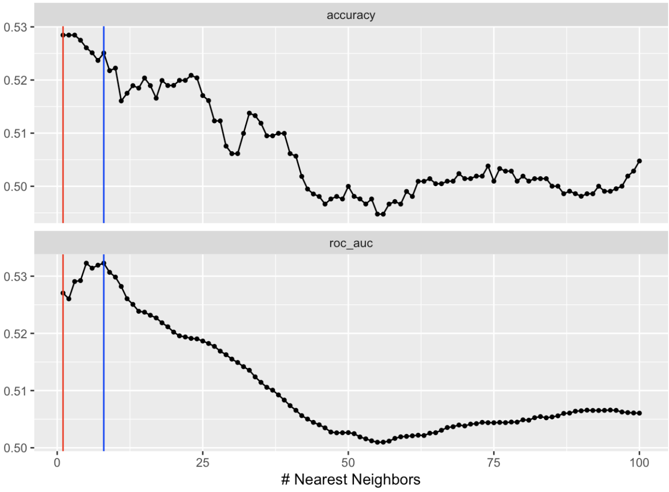
Prediction \ Truth	Down	Up
Down	188	203
Up	131	180

Based on the results shown in Table 22, we observe that the true Down value is 188, which is less than the false Down value of 203. The true Up value of 180 is also greater than the false negative value of 131. Therefore, the confusion matrix shows that the four values are evenly distributed in the matrix. This means that the decision boundary does not follow a quadratic trend. The model's overall accuracy is 52.4%.

4.11 KNN Model Using V-Fold Cross Validation

Since we are using a K-nearest neighbor model, it is important that the variables are centered and scaled to make sure that the variables have a uniform influence. Before we fit the KNN model, we should normalize all predictors. One of the advantages of cross validation is that it resamples without replacing data, resulting in smaller surrogate data sets than the original. In this model, we set the V-fold to 10, then use grid search to tune the best neighbors from 1 to 100 to find the optimal K-nearest neighbor in the model.

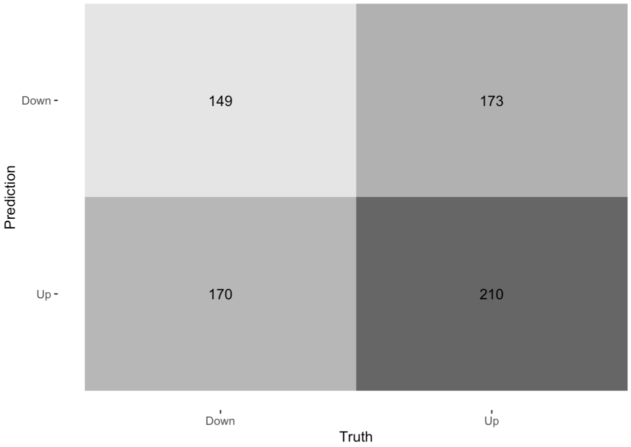
TABLE 23: THE OPTIMAL K IN V-FOLD CROSS VALIDATION



Based on the plot shown above in Table 23, we observe that if we use accuracy as the metric, the best result occurs when the neighbor is equal to 1 (red line). If we compute the area under the ROC curve, the best neighbor is 8 (blue line). Here, it is also important to note that the higher the area is under the ROC curve, the better the model will be at predicting Up classes as Up and Down classes as Down. We will now focus on the model accuracy in the following sections.

To fit the model, we will choose based on the best accuracy. As a result, we use 1 as the k-nearest neighbors to fit the model. Table 24 captures the confusion matrix results that our model yields.

TABLE 24: KNN MODEL (WITH CV) CONFUSION MATRIX

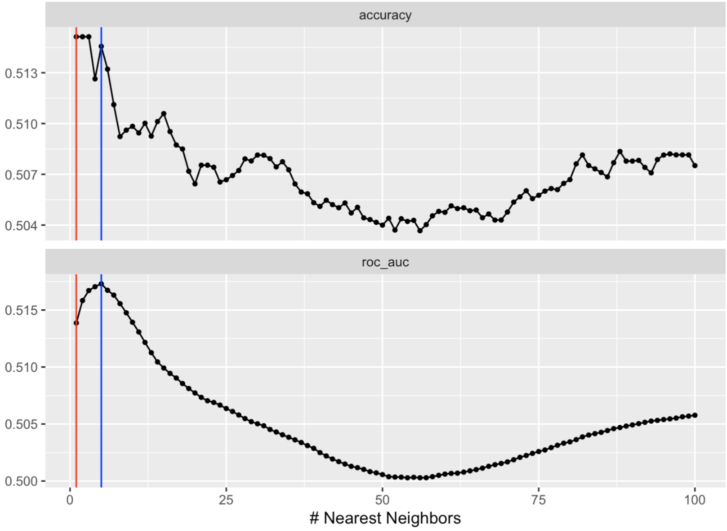


Now, the KNN model shows that the true Down is 149, which is less then the false Down value of 173. The true up value of 210 is also greater than the false negative value of 170. Overall, the model's accuracy is 51.1%.

4.12 KNN Model Using Bootstrap Sampling

Here, we utilize 25 bootstraps as the resamples data set. The method is similar to the previous one; however, the resample method is changed from cross validation to bootstrap sampling. Table 25 captures a plot that shows the optimal K.

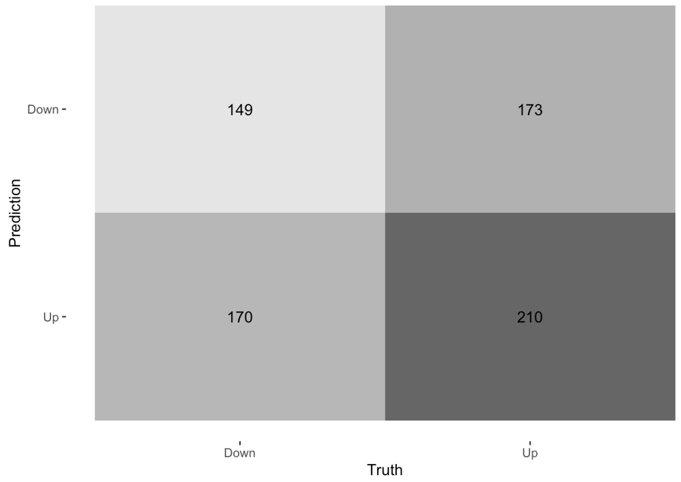
TABLE 25: THE OPTIMAL K IN BOOTSTRAP SAMPLING



Based on the plot shown above in Table 25, we can see from the red line's placement that if the nearest neighbor is equal to 1, the accuracy is the highest in bootstrap resampling. Furthermore, the blue line indicates that the best area under the ROC curve is 5. Thus, we can conclude that the best hyperparameter of this model may be 1 because we observe the same optimal nearest neighbor when using cross validation and bootstrap in the KNN model.

Finally, Table 26 features the results from the confusion matrix for the second KNN model with bootstraps.

TABLE 26: KNN MODEL 2 (WITH BOOTSTRAPS) CONFUSION MATRIX



Prediction \ Truth	Down	Up
Down	149	173
Up	170	210

In the confusion matrix, the true Down is 149, which is less than the false Down value of 173. The true Up value is 210, which is greater than the false negative value of 170 in KNN with 1 neighbor model. We observe that this outcome is the same as the outcome from the KNN model of Cross-Validation. The model's accuracy is 51.1%, which is the lowest one so far.

4.10 Comparative Results

Regression Analyses:

In this section, we set the numerical variable, Return_Today, as a response variable, and set Lag1, Lag2, Lag3, Lag4, and Lag5 as predictor variables with various, diverse combinations in single linear and multiple linear regression formula. Results from our regression analyses indicate that the R-squared values do not surpass 0.01 in every linear model, which means that the models explain less than 1% of the variability in the response data around its mean. Furthermore, the results find that there are only a few regressors that are significant, such as the Lag4 and Lag5, predictors in multiple linear regression. In contrast, the multiple linear regression model plots show that there are outliers in the Residuals vs Leverage plot. There is also evidence against homoscedasticity when we view the Scale-Location plot. The multiple linear regression model also does not totally follow a normal distribution in the normal QQ plot. Thus, this evidence suggests that Bitcoin's price increases and decreases do not totally follow a linear relationship with the previous days.

Decision Trees and Random Forest:

The results from both our decision tree and random forest models indicate that the most important variable is Return_Today. This makes sense because the Return_Today is generated from and corresponds with the direction indicated by the Direction_Today variable. For example, when a given point in our data corresponds with an upward direction within the Direction_Today variable, the Return_Today variable is most likely a positive, numerical value. Thus, none of the other predictors can directly explain the importance of the Direction_Today in our random forest or decision tree models as well as our Return_Today variable.

Based on these results, we observe the underlying relationship between the Return_Today and Direction_Today variables and discover that all five Lag variables correspond with lower values of variable importance in comparison. Still, the variable importance values of each of the five Lag variables prove to be quite similar, but again, are low in comparison to the Return_Today variable.

Overall, the split shown in the first and second decision tree models indicate that values that are less than -205e-6 are classified as a downward direction, while values that are greater than -205e-6 are classified as an upward direction. We also see that these results align with the frequency that we observed in our pie chart in the data visualization section because there are slightly more classifications of Bitcoin stocks increasing at 54% than decreasing at 46%. In contrast, the third and fourth decision tree models with the tuned cost_complexity hyperparameter to the values of 1.5 and 3 did not perform as well due to failure to establish a visible split into our upward and downward directions on our decision tree plot. Finally, since the random forest predictor error was 0.001833006, which is significantly low, our results suggest that this model is likely a good fit for our data and predictions.

Classification Methods:

When analyzing the results yielded by six of our models, we observe that the results do not vary significantly from one another. Here, Table 27 shows the results we obtained from our logistic regression, QDA, KNN, and LDA models side-by-side to allow for better comparison and further analysis.

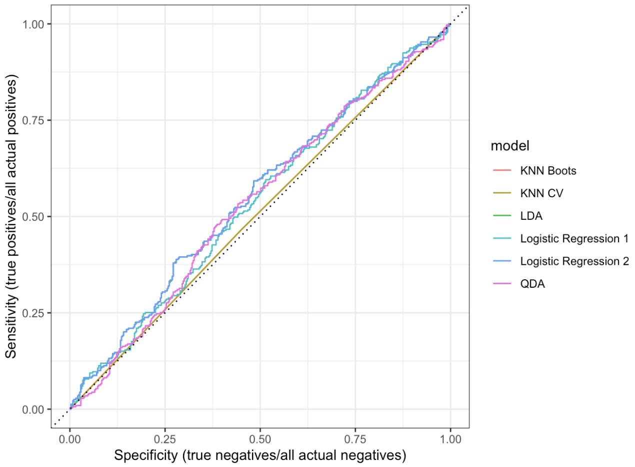
TABLE 27: COMPARATIVE RESULTS

	Estimator	**Metric**	**Estimate**
LR Model with Lag1 and Lag3	Binary	Accuracy	0.5498575
LDA Model with Lag1 and Lag3	Binary	Accuracy	0.5470085
LR Model with Lag1 to Lag5	Binary	Accuracy	0.5398860
QDA Model with Lag1 and Lag3	Binary	Accuracy	0.5242165
KNN Model with Lag1 and Lag3 (CV)	Binary	Accuracy	0.5113960
KNN Model with Lag1 and Lag3 (Bootstrap)	Binary	Accuracy	0.5113960

In Table 27, it’s clear that the logistic regression model proves to be the best technique because simply including the Lag1 and Lag3 variables as predictors causes the model to correctly predict the direction of Bitcoin stocks on a given day in 55% of cases overall. In other words, when the Lag1 and Lag3 variables are used as predictors, the logistic regression model yields the best results of the methods that we have examined so far. Based on these results, we can conclude that the decision boundary may be linear because the logistic regression and LDA models would have performed better than the others if the decision boundary was non-linear. This is also attributed to the fact that the LDA and logistic regression models create a linear line that separates two classes. In other words, since the logistic regression and LDA models did not perform better, our results suggest that the decision boundary may be linear.

Table 28 now features the ROC Curve plots for all six of the aforementioned models.

TABLE 28: COMPARATIVE ROC CURVES



Based on this plot, we can observe that the ROC Curves largely overlap with each other, which suggests that there is not a significant or remarkable difference between the performances of each of the six models. This affirms the conclusion that we drew based on the accuracy estimates shown in Table 27. However, it's important to note that the ROC curve still shows that the logistic regression model with Lag1 and Lag3 predictors and the QDA model has slight differences in sensitivity and specificity. Still, since our logistic regression model performs the best and successfully predicts Bitcoin's trend with a 55% accuracy rate, our results suggest that this method may be able to serve as a reliable and useful option in cryptocurrency markets.

In this section, we detailed our methodology for running each of our statistical modeling and machine learning techniques and analyzed possible factors that influenced why we obtained the results that we did. To find the best model, we use logistic regression, LDA, QDA, and KNN with significant predictors, and employ cross validation and bootstrap techniques to test the performance of our models. However, there is seldom a truly perfect model that can be obtained in real-world examples. In many situations, a set of independent variables are readily available, but the response variable cannot be easily obtained. In other situations, the expected value of Y will not be a perfect estimate for Y, but we can still use many statistical methods to optimize coefficients.

Opportunities and recommendations for future work that expands on this report include advising researchers to try adjusting the proportion used between the simple training data set and the testing data set splits. Alternatively, researchers aiming to build upon the work compiled in this report could use a specific year as the foundational data for the training and testing split. Modifying the proportion that is used to split our dataset into an initial split, training data set, and testing data set could change our results in such a way that the most influential predictors are no longer the Lag1 and Lag3. Similarly, adjusting the range of dates that are included in our data set could change which statistical model and machine learning method performs the best overall.

5. Discussion

With global cryptocurrency adoption rates expected to increase well into 2021 and beyond, we must continue to finetune our ability to apply statistical modeling and machine learning techniques to develop better strategies for successfully predicting the direction of cryptocurrency, such as Bitcoin. Throughout this report, we have presented a collection of statistical analysis, modeling, and machine learning techniques and demonstrated the factors that influence a model’s ability to produce competing results. These results provide significant insight into how the statistical modeling and machine learning processes can be tuned to improve our ability to predict whether Bitcoin stocks will increase or decrease successfully.

Opportunities and recommendations for future work include conducting more comprehensive research on the strengths and weaknesses of other emerging cryptocurrencies, such as Cosmos, Dogecoin, Monero, Polkadot, Litecoin, and Compound, and evaluating if and how existing cryptocurrency trend prediction strategies can be applied to these emerging cryptocurrencies. Furthermore, moving towards an interdisciplinary research model that aims to synthesize across the fields of machine learning, economics, psychology, statistics, and political science could allow future work targeted at better understanding the factors that influence investor behavior and how those factors can be predicted or even controlled to better predict the volatility surrounding cryptocurrency trends. Assuming an interdisciplinary research approach could also yield results that better inform financial, political, and global policies and best practices governing cryptocurrency to allow for more effective regulation. Finally, extracting and analyzing how early cryptocurrency trends relate and compare to the current state and future state trends could prove invaluable to expanding our understanding of how cryptocurrency trends start and fade out, which could better inform our adoption and development around cryptocurrency in the future. As cryptocurrency continues to expand globally, we must also remain committed to evolving our research strategies, tools, and techniques to produce the most relevant prediction results that help inform critical decisions across all sectors.

References

Ahmed M Khedr, Ifra Arif, Pravija Raj P V, Magdi El‐Bannany, Saadat M Alhashmi, & Meenu Sreedharan. (2021). Cryptocurrency price prediction using traditional statistical and machine‐learning techniques: A survey. *Intelligent Systems in Accounting, Finance & Management*, *28*(1), 3–34. https://doi.org/10.1002/isaf.1488

Akyildirim, E., Goncu, A., & Sensoy, A. (2021). Prediction of cryptocurrency returns using machine learning. *Annals of Operations Research*, *297*(1-2). https://doi.org/10.1007/s10479-020-03575-y

Alessandretti, L., ElBahrawy, A., Aiello, L. M., & Baronchelli, A. (2018). Anticipating Cryptocurrency Prices Using Machine Learning. *Complexity (New York, N.Y.)*, *2018*, 1–16. https://doi.org/10.1155/2018/8983590

Bambrough, B. (2021, April 29). *Bitcoin Price Prediction: Why Bitcoin Could Be About To Soar To $100,000*. Forbes. https://www.forbes.com/sites/billybambrough/2021/04/28/bitcoin-price-prediction-why-bitcoin-could-be-about-to-soar-to-100000/?sh=7202ff1558fb.

Bambrough, B. (2021, June 11). *Crypto Price Alert: JPMorgan Issues Stark Bitcoin Warning As Ethereum, Binance's BNB, Cardano And Dogecoin Slide*. Forbes. https://www.forbes.com/sites/billybambrough/2021/06/11/crypto-price-warning-jpmorgan-issues-serious-bitcoin-alert-as-ethereum-binances-bnb-cardano-and-dogecoin-slide/?sh=2cf1cc2353a2.

Bambrough, B. (2021, June 15). *Crypto Price Prediction: Is Ethereum About To 'Flip' Bitcoin?* Forbes. https://www.forbes.com/sites/billybambrough/2021/06/15/crypto-price-prediction-is-ethereum-about-to-flip-bitcoin/?sh=73f250d6416c.

Brockman, K. (2021, June 15). *Is Cryptocurrency Investing or Gambling? 3 Things You Need to Know*. The Motley Fool. https://www.fool.com/investing/2021/06/15/is-cryptocurrency-investing-or-gambling-3-things-y/.

Bulao, J. (2021, May 20). *43+ Cryptocurrency Statistics You Need To Know In 2021*. TechJury. https://techjury.net/blog/cryptocurrency-statistics/#gref.

Chaudhuri, A. P., Narula, C., & Khandelwal, A. (n.d.). *The Higher Education Review*. TheHigherEducationReview. https://others.thehighereducationreview.com/news/5-pros-and-5-cons-of-investing-in-bitcoins-in-2021-nid-1803.html.

Crawley, J. (2021, June 16). *Most UK Financial Advisers Would Steer Clear of Crypto and Meme Stocks: Poll*. CoinDesk. https://www.coindesk.com/opinium-survey-ifa-crypto-investments.

Ebiefung, W. (2021, June 15). *2 Reasons to Sell Coinbase Stock*. The Motley Fool. https://www.fool.com/investing/2021/06/15/2-reasons-to-sell-coinbase-stock/.

Hagen, K. (2021, May 24). *4 Surprising Ways You Can Lose Money Investing in Cryptocurrency*. The Motley Fool. https://www.fool.com/investing/2021/05/24/4-surprising-ways-you-can-lose-money-investing-in/.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An Introduction to Statistical Learning with Applications in R*. Springer.

Locke, T. (2021, April 7). *Thinking of buying bitcoin? What experts say about big crypto concerns: 'You have to be mentally prepared'*. CNBC. https://www.cnbc.com/2021/01/09/what-experts-say-about-cryptocurrency-bitcoin-concerns.html.

Mallqui, D. C. ., & Fernandes, R. A. . (2019). Predicting the direction, maximum, minimum and closing prices of daily Bitcoin exchange rate using machine learning techniques. *Applied Soft Computing*, *75*, 596–606. https://doi.org/10.1016/j.asoc.2018.11.038

Patel, M. M., Tanwar, S., Gupta, R., & Kumar, N. (2020). A Deep Learning-based Cryptocurrency Price Prediction Scheme for Financial Institutions. *Journal of Information Security and Applications*, *55*, 102583–. https://doi.org/10.1016/j.jisa.2020.102583

Skaff, E. (2018, November 27). *Before Bitcoin. History of Cryptocurrency- Arizona Tax Advisors (602)274-7770*. Arizona Tax Advisors. https://arizonataxadvisors.com/bitcoin-tax-and-information/before-cryptocurrency-the-history-of-bitcoins-predecessors/.

Stewart, E. (2021, June 16). *GameStop. Dogecoin. Now AMC. Do meme traders need to be protected from themselves?* Vox. https://www.vox.com/policy-and-politics/22528238/gme-amc-robinhood-stock-market-reddit.

Willing, N. (2021, June 17). Online Trading with Smart Investment App. https://capital.com/ethereum-price-prediction-2021-will-eth-go-up.