PREDICTION OF CRYPTOCURRENCY RETURNS USING MACHINE LEARNING

**Hiridik Rajendran**[[1]](#footnote-1)

MAFinancial Economics, Madras School of Economics, Chennai

|  |  |  |
| --- | --- | --- |
| KEYWORDS |  | ABSTRACT |
| Cryptocurrency  Machine learning  K-nearest neighbor  Decision trees  Random forest  Logistic regression  Naïve Bayes |  | This study examines the predictability of the ten most important cryptocurrencies on a daily basis using several machine learning classification algorithms, such as random forests, k-nearest neighbor, decision trees, logistic regression, and naïve bayes. The models use historical price information and technical indicators as features. The average accuracy of the four algorithms is consistently above 50% for all cryptocurrencies and timeframes, indicating that there is some degree of predictability of price trends in the cryptocurrency markets. On average, machine learning classification algorithms achieve a predictive accuracy of about 50-60% at daily frequencies. |

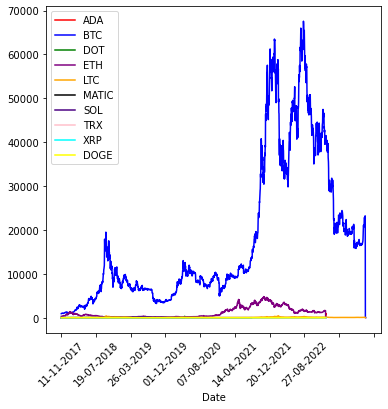
**1.** **Introduction**

In recent years, the significant increase in the value of Bitcoin and other cryptocurrencies has garnered a lot of attention. Specifically, Bitcoin has seen rapid and substantial growth, providing unprecedented earning potential in a short period. Its value has increased by several thousand percent, while other cryptocurrencies have also seen significant growth

Figure 1: Share of total market capitalization

This paper focuses on 10 major cryptocurrencies that account for 67.8% of the total market capitalization[[2]](#footnote-2). Bitcoin makes up 41.3% of the market capitalization. The following coins are analyzed: Bitcoin (BTC), Ethereum (ETH), XRP (XRP), Cardano (ADA), Dogecoin (DOGE), Solana (SOL), Polygon (MATIC), Polka dot (DOT), Litecoin (LTC), and Tron (TRX). Abbreviations are used throughout the paper to refer to these coins.

This study analyses the 10 aforementioned cryptocurrencies at a daily frequency over the period from 2017 to 2023 to identify factors that can predict the direction of price changes, either upward or downward. As a cryptocurrency trader, being able to predict the direction of price changes is more important than the actual direction of the changes, as long as it can be predicted. During an expected period of growth, investors can take a long-term position in these cryptocurrencies, allowing them to reap returns once the prices reach a certain level. On the other hand, if a period of decline is forecasted, investors can utilize margin trading offered by many cryptocurrency exchanges to short-sell these cryptocurrencies and earn higher returns.

 Chart, histogram

Description automatically generated

Figure 2: Closing price of all coins Figure 3:Closing price of coins without bitcoin

The dominance of Bitcoin is evident in Figure 2, which displays the changes in the closing prices of these cryptocurrencies over the years. The changes in Bitcoin's prices are so substantial that the changes in the prices of other cryptocurrencies appear minimal. Figure 3 reinforces this observation by excluding Bitcoin from the analysis.

To predict the direction of price changes, various machine learning algorithms such as Bernoulli Naïve Bayes, Decision Trees, K-Nearest Neighbours, Logistic Regression, and Random Forest are applied to these data cryptocurrencies over the period from 2017 to 2023. The results of these algorithms are then compared to determine the best model for predicting the daily price movements of the cryptocurrency market.

The rest of the paper is structured as follows:

* Section 2 presents the data used in the study.
* Section 3 outlines the methods employed to uncover the predictability patterns.
* Section 4 presents the empirical results.
* Section 5 provides the conclusion.

**2. Data**

In this study, we utilize dollar-denominated cryptocurrency data from Yahoo Finance[[3]](#footnote-3). The initial dataset covers the period from 27th January 2017 to 2023 for most of the coins, except DOT, MATIC, and SOL which entered the market after 2017 but have since acquired a significant market share.

In this study, 15 features are used to predict the direction of price changes. These features are based on the work of Akyildirim et al. ([2021](#akyil)). Kara et al. ([2011](#kara)) , Huang et al. ([2005](#huang)) , Guresen et al. ([2011](#guresen)) and Kim ([2013](#kim)) utilize the past price information and similar set of technical indicators as features to predict the asset returns in various other markets except the cryptocurrency markets. The initial dataset contained only four features: open, close, low, and high. The remaining variables were calculated using the methods described by Achelis ([1995](#achelis)) in Microsoft Excel.

In the dataset, "open" refers to the opening price, which is the price at which the cryptocurrency opens for trading in a day, and "close" refers to the closing price, which is the price at which it closes for the day. "Low" represents the lowest price that the currency reached during a given day, and "high" represents the highest price that it reached. "Volume" refers to the number of tokens or contracts traded for that cryptocurrency in a single day.

"High-Low" is the difference between the highest and lowest closing prices of the currency in a single day. "rt-1," "rt-2," "rt-3," "rt-4," and "rt-5" are the lagged log-returns of the currency over the last one, two, three, four, and five periods, respectively. These variables are calculated using the following formula:

Log returns at t (rt ) =

Where "Pt" is the closing price of the current period, and "Pt-1" is the closing price of the previous period. The lagged log returns for two, three, four, and five can be calculated similarly, by adjusting the "t" subscripts and inserting the appropriate values.

The "log returns (last 3 days)" and "log returns (last 5 days)" represent the sum of the log returns over the last three and last five lagged periods, respectively. This is calculated using the following formula:

Log returns (last 3 days) = rt-1+rt-2+rt-3

Log returns (last 5 days) = rt-1+rt-2+rt-3+rt-4+rt-5

The "last 5 - last 3" feature represents the difference between the sum of log returns over the last five periods and the sum of log returns over the last three periods.

Finally, Simple moving average (SMA) is calculated using the following formula:

SMA =

where pt is the closing price of the current period and pt-1, pt-2, pt-3, and pt-4 are the closing prices of the previous four periods.

**3. Methodology**

The target variable in the classification algorithm is defined as 1 if the next time step return is positive, and -1 if the next time step return is negative. This represents a binary classification problem, where the goal is to predict the direction of the cryptocurrency price movements, either up or down, based on the 15 features discussed earlier. The target variable is calculated using the formula for log returns over daily intervals.

Movement =

For anticipating price movement, five distinct categorization algorithms are applied. The well-known Python scikit-learn package is used to construct these algorithms, such as logistic regression, Bernoulli naive bayes, K-nearest neighbors, decision trees, and random forest algorithms[[4]](#footnote-4). We quickly go over the use of these categorization techniques in this section.

**3.1 Logistic Regression**

The logistic regression model is used to predict the probability of a binary outcome based on the values of the input features. The model uses the sigmoid function to calculate the probability that the output belongs to a particular class. The sigmoid function outputs a value between 0 and 1, which can be interpreted as the probability of the positive class. The model is trained using labeled data and can be used to make predictions on new, unseen data. The simplicity of the model and its ease of use make it a popular choice for solving binary classification problems.

In addition to the benefits mentioned, logistic regression also has the advantage of interpretability. The model provides the coefficients of each feature, allowing the user to understand the impact each feature has on the final prediction. This information can be used to inform decision-making or further feature selection. Here we have used the scikit-learn library of python to build the logistic regression model and find the accuracy of the model.

**3.2 Random Forest Classifier**

Random Forest is an ensemble of decision trees, which combines the outputs of multiple trees and uses bagging to create a diverse set of trees by training each tree on a random subset of the data. The final prediction is made by taking the majority vote on all the trees. This technique helps to reduce overfitting, improve the model's accuracy and stability, and increase the model's ability to generalize to unseen data. Although the use of random forest directly in the return classification is less common compared to the support vector machines or artificial neural networks, promising results have been reported for a few stocks from the US equity market in the recent study by Khaidem et al. ([2016](#khaidem)).This algorithm has been used for prediction in Chen et al. ([2020](#chen)) , Akyildirim et al. ([2021](#akyil)) , Khedr et al. ([2021](#khedr)) and Ranjan et al. ([2022](#Ranjan)).

A random forest can be used to assess the importance of features in a machine-learning model by measuring the improvement in the model accuracy after permuting the values of each feature. This allows us to determine which features have a significant impact on the model's predictions. By dropping the less significant features, we can simplify the model and improve its overall performance. The random forest algorithm in the sci-kit-learn library provides a feature importance attribute that can be used for this purpose.

**3.3 K-Nearest Neighbours (KNN)**

In this case, it becomes ambiguous as to which group the new data point should belong to. The selection of the value of K is crucial as it affects the model's accuracy. A smaller value of K will lead to a more complex model, while a higher value of K will simplify the model. It is essential to conduct experiments to determine the optimal value of K that provides the best results for a given problem.

The optimal k value is determined using techniques such as cross-validation, grid search, or the elbow method. These techniques aim to find the best value of k that minimizes the prediction error and enhances the model's accuracy. Once the optimal value of k is found, the KNN algorithm is trained using the optimal k value and used to make predictions.

**3.4 Decision trees**

The decision tree algorithm recursively splits the data into subsets based on the feature that gives the greatest information gain, until the tree is fully grown, or a stopping criterion is met. The CART algorithm splits the data into different subsets by determining the most significant feature and dividing the data accordingly. The process continues until the data in the subsets is homogeneous, and the tree cannot be further divided. The resulting tree represents a series of decisions that can be used to classify a target value for new, unseen data

This process of following the branches of the decision tree and making predictions based on attribute values continues until the algorithm reaches a leaf node, which represents the final prediction made by the model. The predictions made at each node are based on the relationships between the features in the dataset and the target variable, as represented by the structure of the decision tree.

**3.5 Naïve Bayes[[5]](#footnote-5)**

Naive Bayes is a probabilistic algorithm that makes classifications based on Bayes' theorem, which states that the probability of a class given a set of features is proportional to the likelihood of the features given the class, multiplied by the prior probability of the class. In the case of Naive Bayes, it assumes that the features are conditionally independent given the class, hence the term "Naive." This algorithm is widely used in text classification, spam filtering, and sentiment analysis due to its simplicity and high efficiency.

Bernoulli Naive Bayes is well suited for binary classification problems where the dependent variable takes on binary values (i.e. 0 or 1, yes or no, etc.). In these cases, the algorithm assumes that the features in the data are conditionally independent given the class variable. This allows it to estimate the probabilities required for classification with a high degree of efficiency and ease.

Table 1: Confusion Matrix

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Predicted Value** | | |
|  |  | positive | negative |
| **Real Value** | positive | True positive | True negative |
|  | negative | False positive | False Negative |

**3.6 Evaluation Indicator**

This research utilizes evaluation indicators to assess the performance of machine learning models, as outlined by Ranjan et al. ([2022](#Ranjan)) and Chen et al. ([2020](#chen)). One of these indicators is the confusion matrix, which is a table that represents the performance of machine learning models and calculates their accuracy in a classification framework. Table 1 shows that if both the actual and predicted outcomes are negative, it is considered a False Negative. If the actual outcome is negative but the predicted outcome is positive, it is classified as a False Positive. If the actual outcome is positive but the predicted outcome is negative, it is classified as a False Negative. Finally, if both the actual and predicted outcomes are positive, it is considereda True Positive.

The formulas for accuracy, precision, recall, and F1-score based on the confusion matrix are as follows:

Accuracy = (TP + FN) / (TP + TN + FP + FN)

Precision= TP / (TP + FP)

Recall = TP / (TP + FN)

F1 score = 2 × Precision × Recall / (Precision + Recall)

where TP stands for True Positive, TN for True Negative, FP for False Positive, and FN for False Negative.

Accuracy is a simple and intuitive measure of performance that is calculated as the proportion of correctly predicted observations to the total number of observations. A higher accuracy score indicates a better model. Precision is another performance metric used to evaluate machine learning algorithms. It represents the ratio of correctly predicted positive values to the total number of predicted positive values. Recall, also known as sensitivity or true positive rate, is defined as the proportion of correctly predicted positive observations to the total number of actual positive observations. The F1 score is the weighted average of precision and recall, taking into account both false positives and false negatives simultaneously.

**4. Empirical results**

This summary of the results from using different classification algorithms on the daily time intervals data will help you understand which algorithm performs the best for your specific problem. It is important to keep in mind that choosing the right algorithm depends on the characteristics of the data, such as its size, complexity, and target variable. It's always good practice to try out multiple algorithms and compare their performance to make an informed decision on which one to use for your problem.

**4.1 Logistic Regression**

The first model that was used in this study achieved an accuracy of 0.5897 in predicting cryptocurrency prices using the logistic regression algorithm.

Table 2: Confusion matrix on daily price movement using logistic regression.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Predicted (− 1) | Predicted (1) | |
| Actual (− 1) | 998 | 1166 |
| Actual (1) | 598 | 1538 |

This means the algorithm made 998+1538=2536 correct predictions and 598+1166=1764

incorrect predictions.

**4.2 Random Forest Algorithm**

The accuracy of predicting the daily price movement using the random forest classifier algorithm is 0.5777.

Here random forest algorithm is used to check the significance of the features in price movement prediction.

Table 3: Features and their significance in daily price movement prediction

|  |  |
| --- | --- |
| Features | Significance |
| rt-1 | 0.0850969 |
| rt-2 | 0.0776344 |
| rt-3 | 0.0764743 |
| rt-4 | 0.0762735 |
| Volume | 0.0760749 |
| Log returns (last 3 days) | 0.0753326 |
| rt-5 | 0.0740476 |
| last5-last3 | 0.072861 |
| Log returns (last 5 days) | 0.0717311 |
| High-Low | 0.0616001 |
| Close | 0.0540957 |
| Open | 0.0519562 |
| SMA | 0.0512808 |
| Low | 0.0478075 |
| High | 0.0477333 |

The algorithm found that all the features included in the analysis were significant and had almost equal weight for significance. As a result, all the features were retained in the model. Additionally, the log returns with a lag of one period were found to have the highest significance among all the features.

Table 4: Confusion matrix on daily price movement using random forest.

|  |  |  |
| --- | --- | --- |
|  | Predicted (− 1) | Predicted (1) |
| Actual (− 1) | 1237 | 927 |
| Actual (1) | 889 | 1247 |

This means the algorithm made 1237+1247=2484 correct predictions and

889+927=1816 incorrect predictions.

**4.3 K-Nearest Neighbours**

Here initially the optimal k is found and the optimal k=11. After that, the algorithm is used for the daily price movement predictions.

The accuracy with which KNN predicts the price movement is 0.5502.

Table 5: Confusion matrix on daily price movement using KNN.

|  |  |  |
| --- | --- | --- |
|  | Predicted (− 1) | Predicted (1) |
| Actual (− 1) | 1136 | 1028 |
| Actual (1) | 962 | 1174 |

This means the algorithm made 1136+1028=2164 correct predictions and 962+1028=1990 incorrect predictions.

**4.4 Decision Tree**

Table 5: Confusion matrix on daily price movement using decision trees.

|  |  |  |
| --- | --- | --- |
|  | Predicted (− 1) | Predicted (1) |
| Actual (− 1) | 1134 | 1030 |
| Actual (1) | 1023 | 1113 |

This means the algorithm has made 1134+1113=2247 correct predictions and 1030+1020=2050 incorrect predictions. The accuracy of predicting the daily price movement using decision trees algorithm is = 0.5225.

**4.5 Bernoulli Naïve Bayes**

The study found that some of the features in the dataset were multicollinear, meaning they had a high correlation with one another. Specifically, the features High, Low, Close, Volume, and High-Low were highly correlated with Open, with a correlation value greater than 0.6. Additionally, the features rt-4, rt-5, and log returns (last 5 days) were highly correlated with last5-last3. To address this issue, these features were dropped from the dataset. After dropping these features, a Bernoulli naïve Bayes model was built, and it achieved an accuracy of 0.5287 for predicting the daily price movements.

Table 5: Confusion matrix on daily price movement using Bernoulli naïve bayes.

|  |  |  |
| --- | --- | --- |
|  | Predicted (− 1) | Predicted (1) |
| Actual (− 1) | 895 | 814 |
| Actual (1) | 807 | 924 |

This means the algorithm has made 895+924=1819 correct predictions and 807+814=1621 incorrect predictions.

**4.6 Comparing the accuracy score of the different models.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Models | Accuracy | Precision | Recall | F1-score |
| Logistic regression | 0.5897 | 0.6253 | 0.3935 | 0.483 |
| Random forest | 0.5777 | 0.5818 | 0.4979 | 0.5366 |
| KNN | 0.5502 | 0.5414 | 0.4917 | 0.5153 |
| Decision Trees | 0.5225 | 0.5257 | 0.5046 | 0.5149 |
| Bernoulli Naïve Bayes | 0.5287 | 0.5258 | 0.4921 | 0.5084 |

The logistic regression gives the best accuracy for predicting the price movement on daily

frequency level.

**5. Conclusion**

The cryptocurrency markets are characterized by high volatility and have a shorter history when compared to traditional financial markets. As a result, there is limited data and a lack of understanding of the underlying economic forces driving cryptocurrency prices. Despite these challenges, many researchers are investigating the use of machine learning algorithms to analyze trends in the cryptocurrency markets and make informed investment decisions. By using these algorithms, researchers can overcome some of the limitations of traditional financial analysis, allowing them to identify patterns and relationships in the data that are difficult to discern through human analysis alone.

This study provides evidence that machine learning algorithms can be used to predict cryptocurrency prices and highlights the significance of considering the cryptocurrency market as a valuable asset class for both traders and researchers. The logistic regression algorithm was found to be the best-performing in the study, achieving a prediction accuracy of over 59% frequently. These results are consistent with previous studies on cryptocurrency prediction using machine learning, which further strengthens the potential for using these techniques in future cryptocurrency market analysis and predictions. The findings show that the logistic regression model is the best among the five algorithms evaluated in the study, and this conclusion aligns with the conclusions drawn in other studies by Akyildirim et al. ([2021](#akyil)), Chen et al. ([2020](#chen)), Ranjan et al. ([2022](#Ranjan)), and Khedr et al. ([2021](#khedr)).

**References**

Achelis, S. B. (1995). *Technical analysis from A to Z* (2nd ed.). New York: McGraw-Hill.

Akyildirim, E., Goncu, A., & Sensoy, A. (2021). Prediction of cryptocurrency returns using machine learning. *Annals of Operations Research*, *297*, 3-36.

Chen, Z., Li, C., & Sun, W. (2020). Bitcoin price prediction using machine learning: An approach to sample dimension engineering. *Journal of Computational and Applied Mathematics*, 365, 112395.

Guresen, E., Kayakutlu, G., & Daim, T. (2011). Using artificial neural network models in stock market index prediction*. Expert Systems with Applications*, 38, 10389–10397.

Huang, W., Nakamori, Y., & Wang, S.-Y. (2005). Forecasting stock market movement direction with support vector machine. *Computers & Operations Research*, 32, 2513–2522

Kara, Y., Acar, M., Boyacioglu, O., & Baykan, K. (2011). Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange. *Expert Systems with Applications*, 38, 5311–5319

Kim, K.-J. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, 55, 307–319.

Khaidem, L., Saha, S., & Dey, S. R. (2016). Predicting the direction of stock market prices using random forest.

Khedr, A. M., Arif, I., El‐Bannany, M., Alhashmi, S. M., & Sreedharan, M. (2021). Cryptocurrency price prediction using traditional statistical and machine‐learning techniques: A survey. Intelligent Systems in Accounting, *Finance and Management*, 28(1), 3-34.

Ranjan, S., Kayal, P., & Saraf, M. (2022). Bitcoin price prediction: A machine learning sample dimension approach. *Computational Economics*, 1-20.

1. Email address: hiridikrajendran99@gmail.com [↑](#footnote-ref-1)
2. See <https://coinmarketcap.com> for recent market capitalization [↑](#footnote-ref-2)
3. See <https://finance.yahoo.com/crypto/> for the data [↑](#footnote-ref-3)
4. The logistic regression and other classification algorithms are implemented in Python with the scikit-learn package available on the following website: https://scikit-learn.org/stable/. [↑](#footnote-ref-4)
5. See <https://medium.com/analytics-vidhya> for a more detailed understanding of the algorithms. [↑](#footnote-ref-5)