An Empirical Study on Crypto-currencies Price Prediction Using Different Machine Learning Classifiers

Utsab Talukder, Anubrata Sarkar, Samriddhi Sarkar

Department of Computer Science,

Gour Mahavidyalaya

Keywords- Cryptocurrency, Blockchain, Digital currency, Machine learning, Random Forest Regression, LSTM, Gradient Boost, Linear Regression Lasso.

Abstract: In this project, we have tried to predict the price of any cryptocurrency. There are various machine learning tests while making this model. We have found that the LSTM, LASSO, RF Regressor, and Gradient Boost are well suited for our model. The best score of our model is 0.9989. We get our best performance by using the Lasso method.But in the split of 60:40 we get Random forest Regresson model has lowest mean absolute error and mean squared error. Our model helps us to find the future value of any cryptocurrency.

Introduction: Crypto-currency is a digital currency that is an alternative form of payment, created using encryption algorithms. It is monitored and organized by a peer-to-peer network called a blockchain, which also serves as a secure ledger of transactions, e.g., buying, selling, and transferring. The “crypto” in cryptocurrencies refers to complicated cryptography that allows for the creation and processing of digital currencies and their transactions across decentralized systems. Alongside this important “crypto” feature is a common commitment to decentralization; cryptocurrencies are typically developed as code by teams who build in mechanisms for issuance (often, although not always, through a process called mining) and other controls.

Cryptocurrencies were not initially made as an investment option. With some time passes, it became a highly unstable and diversifying asset. There are some risk factors that can make it a sensitive asset.

There’s no denying that some traders have become millionaires thanks to crypto and their successful investments. But also there is a great number of people who have lost a great sum of money by investing in crypto. That is why we try to create an ML method that can predict the rate of crypto currency.

Machine Learning is a part of Artificial Intelligence. ML helps machines learn how to behave like a human.In ML you need some past data to train the machine. Divide the data into train and test sections. The ratio of the train and test data can be 6:4 or 7:3 or 8:2. In ML, we can find the relation between the input data and the result. Machine Learning helps to solve various prediction problems.
We are trying to answer this research question: is it possible to find out the future rate of any cryptocurrency. The result shows that the trained model is predicted to be good in random classification.

Literature review :
Sebastião et al.[1] examine the predictability of three major cryptocurrencies: bitcoin, Ethereum, and litecoin. Their profitability of trading strategies devised upon ML, namely linear models, RF, and SVMs.

These classification and regression methods use attributes from trading and network activity for the period from August 15, 2015, to March 03, 2019, with the test sample beginning on April 13, 2018. The results point out that the best trading strategies are Ensemble 5 applied to Ethereum and litecoin, which achieved an annualized Sharpe ratio of 80.17% and 91.35% and an annualized return, after proportional trading costs of 0.5%, 9.62%, and 5.73%, respectively. These values seem low when compared with the daily minima and maxima returns of these cryptocurrencies during the test sub-sample.

Jaquart et al.[2] analyzed the predictability of the bitcoin market across prediction horizons ranging from 1 to 60 min. They test various machine learning models and find that, while all models outperform a random classifier, recurrent neural networks and gradient boosting classifiers were especially well-suited for the examined prediction tasks. As per the research, they utilized four types of features namely -a comprehensive feature set, including technical, blockchain-based, sentiment-/interest-based, and asset-based features. The results showed that technical features remain most relevant for most methods, followed by selected blockchain-based and sentiment-/interest-based features. Additionally, they pointed out that predictability increases for longer prediction horizons. Although a quantile-based long-short trading strategy generates monthly returns of up to 39% before transaction costs, it leads to negative returns after taking transaction costs into account due to the particularly short holding periods.

Salman et al.[3] analyzed the predictability of the prices of BTC. They use various machine learning methods based on neural networks. They also work on studying stock market trends. They conclude that it is more helpful to use bitcoin as a global currency. Then it will be decentralized. They achieve an accuracy of 94.89%. In April 2020 the prediction decreased by over 13.7% itself for some evolution.

Azam et al.[4] analyzed the predictability of the future rate of one of the most famous and used cryptocurrencies named Solana. They conclude that the investors are especially investing in Solana over this time. Basically, they use a forecast algorithm for this prediction. They achieve 99.87% accuracy in this project.

Garcia et al.[5] analyzed the growth of the Bitcoin market, the uses of Bitcoin, and the trading strategies. They conclude which signal is more responsible for the changes in bitcoin prices by using different types of strategic methods.

Dataset:

We use the API form of the dataset while making this program. API means Application Programming Interface which provides you the power to monetize a dataset. API helps us to connect all over the world. In an API dataset, you can use the previous dataset made by someone but you can manipulate the dataset as per your needs. An API dataset can update by itself. So if you are working with frequently changeable data then an API dataset is best for you. There are four types of APIs: Open APIs, Partner APIs, Internal APIs, and Composite APIs. In the API, data is stored in array format. Collections are created with the help of the collection() function. Then the saveMatches() function and getMatches() function called to get data matches. For each match, a new record is saved in the database using the Match facade. If the amount of data you get in an API database is not enough for you then API offers to store extra data in your database. In this work we use https://min-api.cryptocompare.com/data/histoday named dataset.

Methodology:

This study examines the predictability rate of different cryptocurrencies by using different machine learning techniques. In this paper basically, we work with bitcoins. We used four various different types of ML methods named LSTM, Lasso, Rf-regressor, and GradientBoost.

- **LSTM:** LSTM is a special type of RNN model. Generally, RNN works well in the case of short term dependency. It means RNN does a very good job of finding out the prediction value of the short-term dependency for which the model has been trained. But in the case of long-term dependency, RNN cannot work properly. So some special changes have been made in RNN so that it can be worked properly based on the long term sequence of data. This is why the model we use is LSTM (Long Short Term Memory). This is a specific type of RNN. The LSTM model helps to solve the problems of RNN such as the **Vanishing Gradient Problem** and **Exploding Gradient Problem**. To solve these types of problems a memory unit has been introduced in LSTM which is also known as a cell unit or cell state. This cell unit is updated for all time zones and all loops at every step. In this case the prediction of the output depends on the current input, previous output, and previous memory. In this case, The previous memory does the most important work, it helps to create long short-term memory.

  Each LSTM state contains five major components. They are-

  1. A Simple RNN Cell,
  2. Cell State→Long Term Memory.
  3. Forget Gate,
  4. Input Gate,
  5. Output Gate.

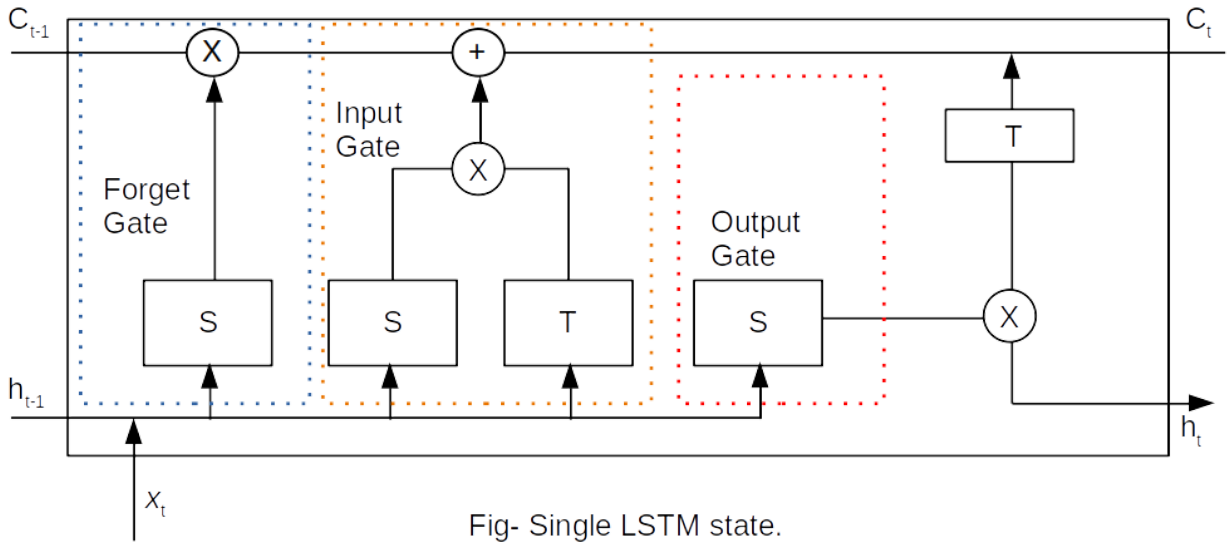


Fig- Single LSTM state.

C_{t-1} = Previous Cell State,

C_t = Output Cell State,

h_{t-1} = Previous Hidden State,

h_t = Output Hidden State,

x_t = Input,
S = Sigmoid Function,
T = tanh Function,
X = Multiplication,
+ = Addition.

Equation of Gates used in a single LSTM state-

$$i_t = \sigma(w_i [h_{t-1}, x_t] + b_i)$$

$$f_t = \sigma(w_f [h_{t-1}, x_t] + b_f)$$

$$o_t = \sigma(w_o [h_{t-1}, x_t] + b_o)$$

Here,

i_t = input gate,

f_t = forget gate,

o_t = output gate,

σ = sigmoid function,

w = weight of a respective gate,

b = bias of a respective gate.

The equation of cell state and final output-

$$\tilde{c}_t = \tanh(w_c [h_{t-1}, x_t] + b_c)$$

$$c_t = f_t \times c_{t-1} + i_t \times \tilde{c}_t$$

$$h_t = o_t \times \tanh(c^t)$$

Here,

$\tilde{c}_t$ = Cell state's candidate at timestamp(t),

c_t = Cell state at timestamp(t).

h_t = Hidden state at timestamp(t).

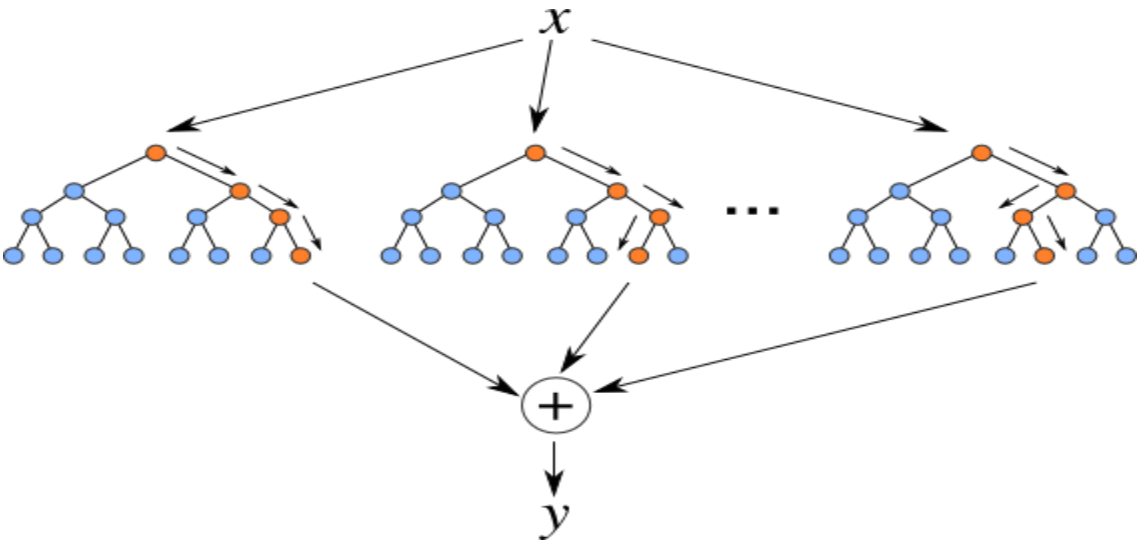
- **LASSO:** The word 'LASSO' would be abbreviated as **'Least Absolute Shrinkage and Selection Operator'**.The LASSO is a modified version of linear regression. We know that in linear regression, the model is not penalized in terms of weight choice. But here in LASSO, the model is penalized for the sum of absolute values of the weights. That's why it is a very famous regularization technique. Compared to linear regression Lasso is more accurate at prediction.

  Lasso Regression = Loss + α ||w||
  Here,
  Loss = Difference between the predicted value and the actual value.
  α ||w|| = Penalty. (w represents the weight value).

- **Gradient Boost:** In the Gradient Boosting algorithm we build a sequence of models where we try to reduce the errors of a subsequent model compared to its previous model. Gradient Boost is used in both classification and regression. As our work is based on regression, that's why here we used **Gradient Boosting** Regressor. Actually, the Gradient Boost is generally used when the bias error needs to be decreased. Gradient Boosting algorithm has three elements-
    1. Loss function,
    2. Weak Learner,
    3. Additive Model.

  Here, in our work, we have used MSE(Mean Squared Error) as a loss function. Mainly Decision Trees are used as weak learners in Gradient Boost. In this model, the Gradient Boosted trees are built in a stage-wise method and it allows the optimization of an arbitrary differentiable loss function. In our work, we have tried to teach our model to predict values in the form of $\hat{y} = F(x)$ by minimizing the MSE. For doing so the following algorithm has been complied-
  Step-1: The average of the target label is calculated,
  Step-2: The errors or residuals are calculated (residuals = actual values - predicted values),
  Step-3: Decision trees are constructed to predict the residuals,
  Step-4: Now the target label is predicted using all of the trees within the ensemble,
  Step 5: The computation of new residuals are made,
  Step 6: Repeat steps 3 to 5 until the number of iterations matches the number specified by the hyperparameter. (Here, Hyperparameter denotes nothing but the number of estimators),
  Step-7: The final prediction is made using all of the trees in the ensemble.

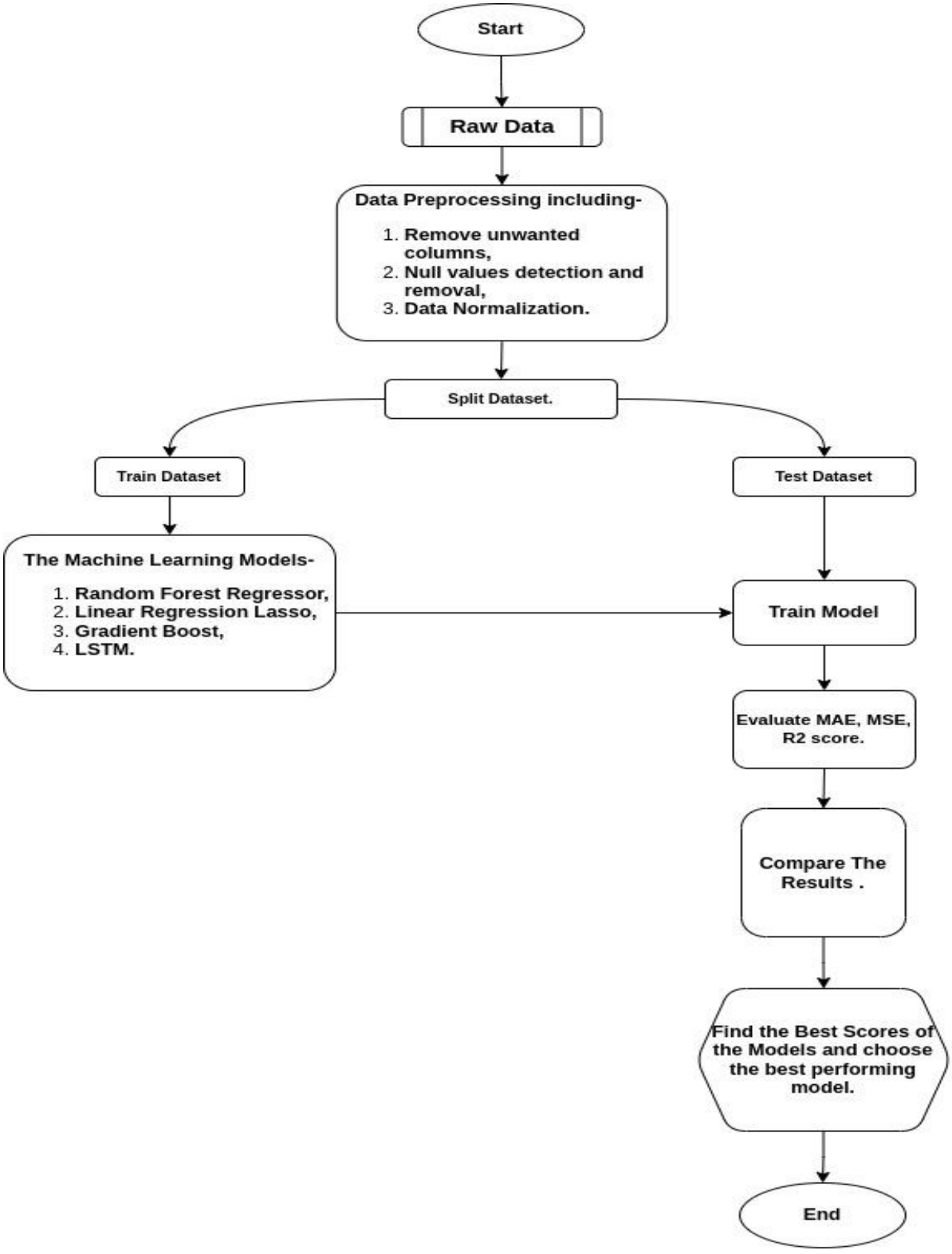
- **Random Forest Regression**: It is a supervised learning algorithm that mainly uses the **ensemble learning method** in the case of regression prediction. This method is a technique that combines predictions from many types of machine learning models to make it a more precise prediction than a single model.



The above diagram shows us the structure of a Random Forest. There you can notice that different trees run in parallel with no collision among them. A random forest operates by constructing several decision trees during training time and outputting the means of the class as the prediction of all the trees.
To get a better understanding of the Random Forest algorithm, let's walk through the steps:

1. Pick at random *k* data points from the training part.
2. Build a decision tree associated with these *k* data points.
3. Choose the number *N* of trees you want to build and repeat steps 1 and 2.
4. For a new data point, make each one of your *N*-tree trees predict the value of **y** for the data point in question and assign the new data point to the average across all of the predicted **y** values.

The approach we have followed in this project is shown below-



Here, first of all, we have read the raw data through API. Then we modified the dataset using various data preprocessing methods. In this case, we have removed the unwanted columns, detected and removed the null values, and performed data normalization using MinMaxScaler. Next, we have divided the dataset into the train and the test portion. We have trained the models with training data and we have tested the models with test data. Then we evaluated MAE, MSE, and R2 scores for each model. Based on this, the best-performing model has been selected by comparing the results of the models.

Result: Here we have divided the dataset into 3 ways for Train_test_split - 80:20, 70:30, and 60:40. On this basis, we have evaluated MAE, MSE, and R2 scores for each model. With the same 'train test split' ratio we calculated the 'Grid Search Best Score' of the mentioned models. But in the case of LSTM we used ‘Grid Search Accuracy’. Here, we have discussed the outcomes of our models regarding BTC.

	Crypto prediction rates				
	Train-test-split			**GS.best_score_(cv=5)**	
		MAE	**MSE**	**Train part**	**Test part**
Random forest regressor	80:20	0.007832	0.000169	0.998125	0.997195
	70:30	0.007960	0.000149	0.997987	0.997632
	60:40	0.007213	0.000124	0.997787	0.997743
	Train-test-split			**GS.best_score_(cv=5)**	
		MAE	**MSE**	**Train part**	**Test part**
Linear regression Lasso	80:20	0.274308	0.091189	0.998952	0.999021
	70:30	0.273553	0.090258	0.998964	0.998844
	60:40	0.271493	0.088999	0.998913	0.998949
	Train-test-split			**GS.best_score_(cv=5)**	
		MAE	**MSE**	**Train part**	**Test part**
Gradient boost	80:20	0.007914	0.000157	0.998363	0.997902
	70:30	0.008264	0.000154	0.998287	0.998042
	60:40	0.007831	0.000139	0.998123	0.997721
	Train-test-split			**GS._Accuracy_(cv=5)**	
		MAE	**MSE**	**Train part**	**Test part**
LSTM	80:20	0.016559	0.000522	0.002496	0.014925
	70:30	0.018267	0.000799	0.002861	0.003333
	60:40	0.015514	0.000476	0.001666	0.002506

Discussion: The popularity of cryptocurrency is slowly increasing day by day in many countries of the world. The Central African Republic and El Salvador have already recognized cryptocurrencies such as BTC as legal tender. In this context, it can be said that countries like the USA, Russia, UAE, Turkey and Brazil are also showing their interest in cryptocurrency. Even our country India has recently taken some special steps in this regard. As a result, cryptocurrency is becoming a reliable option for investors. But by reviewing various datasets during our project, we realized that this cryptocurrency market is very volatile. Therefore, to protect investors from losses, it is necessary to accurately predict the rate of cryptocurrency. We have learned from our observations that it is difficult but not impossible to predict the rate of cryptocurrency. We have noticed in the case of our data visualization that the correlations of certain features with the target feature are very high. By using these features, it is possible to make accurate predictions of the target feature by training different machine learning models. The models that we are using, in this case, are - LSTM, Lasso Regression, Random Forest Regressor, and Gradient Boost. The outcome of each model has given us the expected results. After reviewing and comparing the results of the models, we can conclude that the prediction of the 'LASSO Regression' model (on the basis of Grid Search CV) performed the best. But in the case of train_test_split of 60:40 ratio, we get that the Random Forest Regression model has the lowest mean absolute error and mean squared error. However, the difference in results between each model is very small. We can further say in the light of this small observation that by taking the right steps, it is possible to eliminate the volatility of the crypto market and also bring it under various regulations like legal tenders.

Conclusion: The main objective of our project is to create a price prediction model by reviewing the characteristics of different types of cryptocurrencies to make its price prediction more accurate. Here we have worked with four machine learning models in total - *Random forest regressor, Linear regression Lasso, Gradient boost, and LSTM.* If we only look at the GS_Best_Score in case of the results of the models- Random Forest Regressor's best score is 0.99743, in the case of Linear Regression Lasso it will be 0.999021, on the other hand, Gradient Boost's best score is 0.996042. In the case of LSTM, the GS_Accuracy has been found out which is 0.014925. By reviewing the results of these models, we realize that price prediction is the most accurate for Linear Regression Lasso. But in the case of train_test_split of 60:40 ratio, we get that the Random Forest Regression model has the lowest mean absolute error(0.007213) and mean squared error(0.000124). We have given preference to Bitcoin as a cryptocurrency here. This study has been prepared by reviewing the different values of the mentioned features- ‘high’, ’low’, ‘open’, ‘volumefrom’, ‘volumeto’ & ‘close’. Here, our target feature is ‘close’. In our observation, we see that the correlation of these features - ‘high’, ‘low’, and ‘open’ is the highest with the 'close' feature.

The way technology is advancing in the present age and the demand for cryptocurrency is increasing, it can be inferred that the use of machine learning in the crypto market will increase in the near future. In this context, we have tried our best to make the slightest contribution through this project.

References:

1. Sebastião, H., & Godinho, P. (2021). Forecasting and trading cryptocurrencies with machine learning under changing market conditions. *Financial Innovation*, *7*(1), 1-30.
2. Jaquart, P., Dann, D., & Weinhardt, C. (2021). Short-term bitcoin market prediction via machine learning. *The journal of finance and data science*, *7*, 45-66.
3. Ibrahim, A. A. (2020, November). Price prediction of different cryptocurrencies using technical trade indicators and machine learning. In *IOP Conference Series: Materials Science and Engineering* (Vol. 928, No. 3, p. 032007). IOP Publishing.
4. Azam, S., & Kumar, R. (2021). Crypto Currency Price Prediction Using Machine Learning. *International Journal of Recent Advances in Multidisciplinary Topics*, *2*(12), 56-58.

5. Garcia, D., & Schweitzer, F. (2015). Social signals and algorithmic trading of Bitcoin. *Royal Society open science*, *2*(9), 150288.