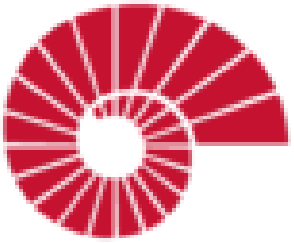
Analysis of Different Machine Learning and Econometric Models to Predict Crypto Currencies High-Frequency Behavior



KOÇ
UNIVERSITY

CSSM 502 - Term project

Oğuzhan PINAR
22.01.2022

Abstract

PURPOSE: The target of this project is to develop machine learning and econometric models to predict whether the price of a coin will increase in the next candlestick or not (label detection) for 21 different coins and analyze the model performances.

DEVELOPED MODULES: In order to do the project, three different modules are developed: Data collection, Feature Extraction, and Modelling.

MODELS: From the machine learning side, XgBoost Classifier and RandomForest Classifier are used; whereas, from the Econometric approach, the Logistic Regression model is used.

RESULTS: The mean accuracy of the XgBoost Classifier is 77% for 21 different coins and it was the best performance. Random Forest Classifier is also performed similarly to XgBoost. However, the Logistic Regression model performed very poorly, with a 50% accuracy score (same as a random assignment). Therefore, for machine learning models, better performance is observed compared to Econometric models.

1. INTRODUCTION

After the popularization of Bitcoin and other Cryptocurrencies, the market has reached billions of dollars in daily volume. With this high trading volume and liquidity, high-frequency trading becomes feasible in those markets. Therefore, algorithmic trading bots are started to be developed by many firms and people. After the popularization of algorithmic trading approaches, predictive algorithms gain significance in that realm. However, Cryptocurrency price prediction is a challenging task due to the highly volatile and unpredictable nature of the market. But, with the advent of machine learning and econometric techniques, it has become possible to develop algorithms that can predict the future price of a cryptocurrency with a certain degree of accuracy. These algorithms use historical price data and other relevant market information to train models that can identify patterns and trends in the data. Once trained, these models can then be used to make predictions about future price movements. Some of the most popular machine-learning techniques used in cryptocurrency price prediction include neural networks, support vector machines, and decision trees. Despite the potential accuracy of these models, it's important to note that cryptocurrency markets remain highly speculative and predictions may not always be accurate.

In order to analyze different model performances on the high frequency behavior of cryptocurrencies, I plan to run a variety of machine learning and econometric models for 21 different coins. These models will take into account various factors such as historical price data, market movements, and dummy variables for seasonality adjustments to make predictions about future price movements. Some of the machine learning models I will use include random forests, and XGBoost. In addition to these, I will also use Logistic Regression as the econometric technique By running multiple models and comparing their results, I hope to gain a more comprehensive understanding of the drivers of cryptocurrency prices and to develop more accurate predictions. Also, machine learning and classical approaches will be compared.

2. LITERATURE REVİEW

Predictive models for cryptocurrency prices have been the subject of much research in recent years. The volatility and lack of historical data make predicting cryptocurrency prices a challenging task. However, many researchers have attempted to develop models that can accurately predict price movements.

One common approach is to use machine learning algorithms, to analyze historical price and trading volume data. These models have been shown to be effective in predicting high-frequency price movements. However, they are not as successful in predicting long-term price movements. (Mallqui, 2018) (Khedr, 2021)

Another approach is to use classical econometric analysis to predict cryptocurrency prices. This approach involves analyzing the behaviour of prices using linear or non-linear fitting techniques. Studies have found that classical techniques can be used to predict long-term price movements, but it is not as effective as machine learning algorithms. (Akyildirim 2021)

There are some researches focusing on comparing machine learning and classical approaches on cryptocurrency pricing behaviour. One research is “Cryptocurrency price prediction using traditional statistical and machine-learning techniques: A survey” written by Khedr et al. They dive deep into the literature on Cryptocurrency models and compare the articles using machine learning and econometric models very successfully. However, comparing the models focusing on different time-frames and different features could be misleading. (Khedr 2021)

Another article on a similar topic is “Prediction of cryptocurrency returns using machine learning” by Akyildirim et al. They develop different models to predict price cryptocurrency returns. They find “Machine learning classification algorithms reach about 55–65% predictive accuracy on average at the daily or minute level frequencies, while the support vector machines demonstrate the best and consistent results in terms of predictive accuracy compared to the logistic regression, artificial neural networks and random forest classification algorithms” (Akyildirim 2021). In our research, we will use a similar approach to Akyildirim; in addition, it will be shown that further accuracy scores could be reached by applying different feature sets.

Some recent studies have also explored the use of sentiment analysis techniques on social media data to predict cryptocurrency prices. The study found that social media sentiment can be a useful indicator of future price movements. Even though sentiment analysis is not used in this research, integrating Twitter data could be a possible further improvement to the research. (Abraham, 2018)

Overall, while several studies have attempted to develop predictive models for cryptocurrency prices, there is still a lack of consensus on the most effective approach. More research is needed to develop models that can accurately predict short and long-term price movements, as well as models that can handle the volatility of the cryptocurrency market.

3. METHODOLOGY

In order to do the analysis 3 different modules are developed: Data gathering, Feature Extraction and Modelling. Briefly, the flow of the project is like following:

- **Data gathering** module takes start and end dates as input and makes the required connections with Binance API and return the raw data according to given start - end datas and frequency.
- **Feature extraction** module takes the raw data as input and creates the required features. As output it returns the Features and target variable data for both train and test sets.
- **Modelling** module takes train and test data and fits the models. It returns the fitted models.

The flow is illustrated in figure 1.

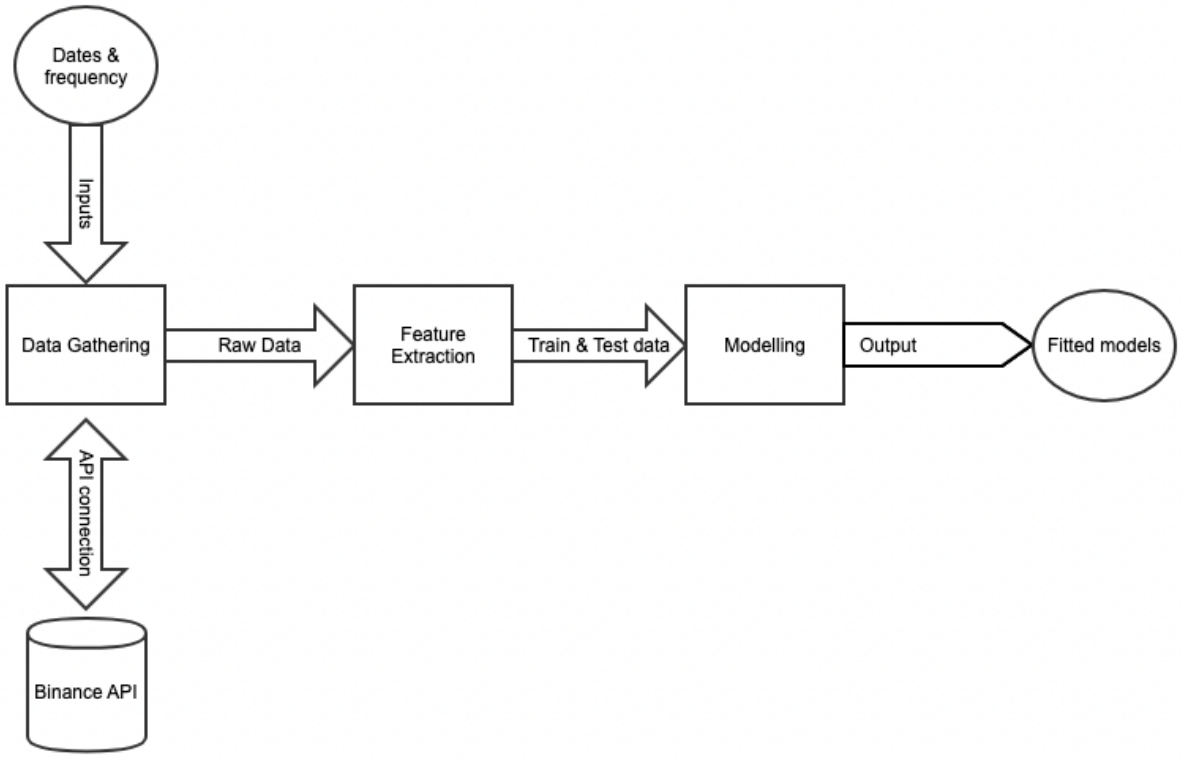


Figure 1: Flow diagram

3.1. Module 1: Data Gathering

Binance allows trading of over 600 digital currencies and tokens, such as BTC, ETH, LTC, DOGE, and BNB, through exchange of one cryptocurrency for another. Luckily they provide all required data for all of the coins tradable in Binance via its API. In this research candlestick (klines) data will be used as the source data which have open, close, high, low, and volume information at given frequencies. The models will be developed for 21 different coins and the target coins are: "BTC", "ETH", "BNB", "DOGE", "ADA", "MATIC", "DOT", "TRX", "LTC", "SOL", "UNI", "AVAX", "LINK", "XMR", "ATOM", "ETC", "XLM", "ALGO", "VET", "NEAR", "HBAR".

Since we use 21 different coins all processes has to be time efficient otherwise it would be very time costly to do the analysis. In order to increase the time efficiency of the project. Parallel processing for all coins is applied for both data gathering and feature extraction modules. The details of parallel processing will be shared later in this chapter.

Similar with all API's Binance API has some restrictions on requests. The main constraint here was API limit which returns maximum 500 rows of data per request. To handle it, multiple requests are sent with a lambda function for a coin. To handle the restriction divide and conquer approach is used which works in the following manner.

- Divide the total duration to 500 rows sub-durations.
- Make a request for each sub-duration.
- Merge all results.

For the research I used 1 minute frequency data from 2022-11-01 00:00:00 to 2022-12-23 00:00:00 which makes 75 000 rows of data. It is known that financial markets exhibits different characteristics during and after the Christmas. Therefore, the duration of data is chosen as the most current dates before the Christmas. As mentioned above to increase the code performance parallel processing is used for data gathering process which will be mentioned now.

Parallel Processsing:

To make a more comprehensive analysis the models are developed for 21 different coins in this research. However, it is not easy to tackle with the processing cost of gathering data, feature extraction and modelling. The proposed solution to this problem is parallel processing.

The modern computers have multiple cores in their CPU; however, with standart methods of coding operations can be processed in only one core since the code is read line by line. With the help of parallel processing multiple tasks can be pooled together and processed simultaneously in different cores of the CPU. Which significantly reduces the time cost of the operations.

In the research data gathering, feature extraction and modelling processes are applied using the parallel processing approach. Meaning that 21 different task is pooled for a module and processed simultaneously. The codes used for this operation can be found below:

```
begin_time = dt.datetime.now()

start = dt.datetime(2022,11,1)
end = dt.datetime(2022,12,23)

my_data_constructor = data_constructor()
coin_list = ["BTC", "ETH", "BNB", "DOGE", "ADA", "MATIC", "DOT", "TRX", "LTC", "SOL", "UNI",
             "AVAX", "LINK", "XMR", "ATOM", "ETC", "XLM", "ALGO", "VET", "NEAR", "HBAR"]

klines_data_dict = {}
threads_dic = {}
pool = ThreadPool(processes=5)

#Start pooling:
for coin in coin_list:
    symbol = coin + "USDT"
    async_result = pool.apply_async(my_data_constructor.get_klines_data, args = (symbol, "1m", start, end))
    threads_dic[symbol] = async_result


#Get results:
for coin in coin_list:
    symbol = coin + "USDT"
    returned_df = threads_dic[symbol].get()
    klines_data_dict[symbol] = returned_df


time_cost_pooling = dt.datetime.now() - begin_time
```

By applying parallel processing for each coin, the module provided a good performance which takes around 5 minutes to get data for all 21 coins where every coin has aroud 75 000 rows of data. Considering that API has 500 rows limit and therefore 150 different requests have to be sent to API for each coin this could be interpreted as a good performance improvement.

3.2. Module 2: Feature Extraction

The second module built for the research is Feature extraction which gets klines data of the coins as input and creates the required features. After that train test split is done in this module and it returns the features and target variable data for both train and test sets. Used features and target variables are as follows:

Target variable:

As the target variable I used the dummy variable showing wheter the coin is profitable or not in the given candlestick:

$$y_t = \begin{cases} 1 & \text{if} \quad Close > Open \\ -1 & \text{if} \quad Close \leq Open \end{cases}$$

Making predictions in the financial markets is difficult as the market charactersictics. Therefore, targeting a continuos variable would be more challenging and almost impossible to create a good model due to socalled efficient markets hypothesis. In order to make the job a little bit easy a binary variable is used as the target in this research.

Features:

I used following features in the analysis:

1. Price_level(open) - profit (last 5 lags) - volume (last 5 lags) - range (last 5 lags)
2. The labels and profit of 5 min - 15 min - 1 hour - 12 hours - 1 day candlesticks
3. Other currencies (market) weighted labels (weight with volume)
4. Weekend - weekday dummies
5. Day time dummies (divide the day into 6 parts where each part is 4 hours)
6. Trend

The most important part of feature extraction was preventing the data leakage from future. To handle it, I created a module (extract_features) which only creates the features for each rows without any lags etc. In addition to that, I developed another superior class which handles data leakage problems (by dropping future data) and adds lags.

3.3. Module 3: Modelling

The modelling module is used to fit the required models. It takes train and test data returned from feature extraction module and makes a hyperparameter tuning with cross validation and fits the models using the best parameters found by tuning.

I have tried XgBoost classifier and Random Forest classifier models from the machine learning approach. On the other hand, Logistic Regression is used from classical econometric side. For the Logistic Regression, we have to have stationarity assumption to hold. In order for stationarity to hold we have to decompose trend and seasonality from our data. Note that we have trend and time dummies (is_weekend and day time) in our feature set. Therefore, those variables will allow us to control for stationarity. As can be seen from correlation heatmap in Figure 2, the volumes and ranges are seem to be correlated with each other. This correlation is not a significant problem for machine learning algorithms; however, it poses a thread for regression since it might cause multicollinearity problem.

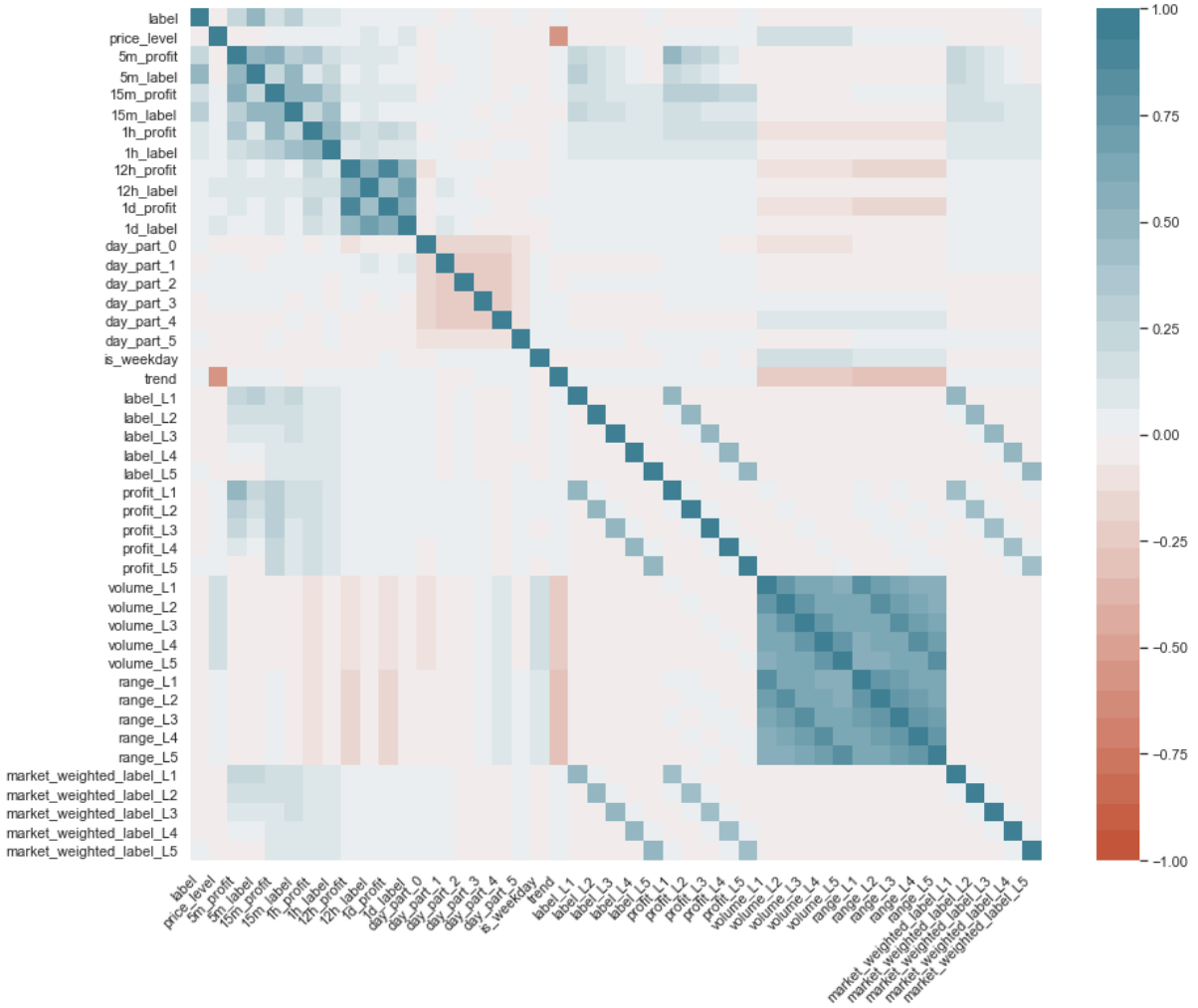


Figure 2: Correlation heatmap

As we can see from figure 3 we have balanced data with 49.8 % of the candlesticks the trade is profitable while it is not profitable for 50.2 % of the time. Having a balanced data is important for most of the machine learning classifiers since they are learning with the subsets of the given data. Therefore, do not need to make operations like down-sampling etc.

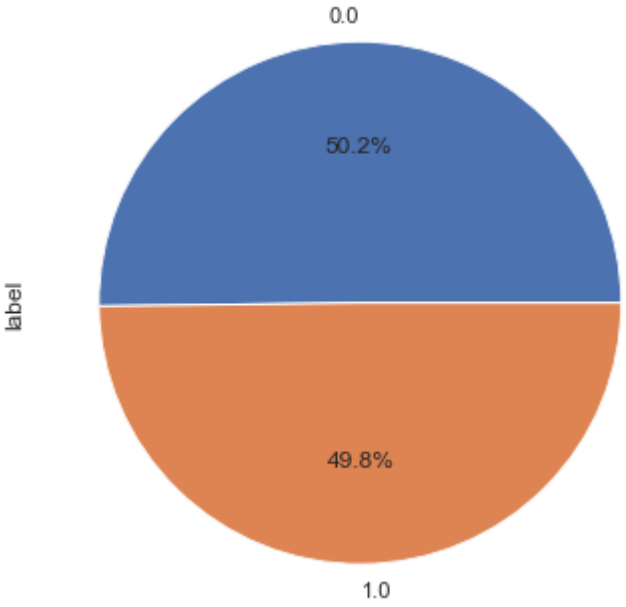
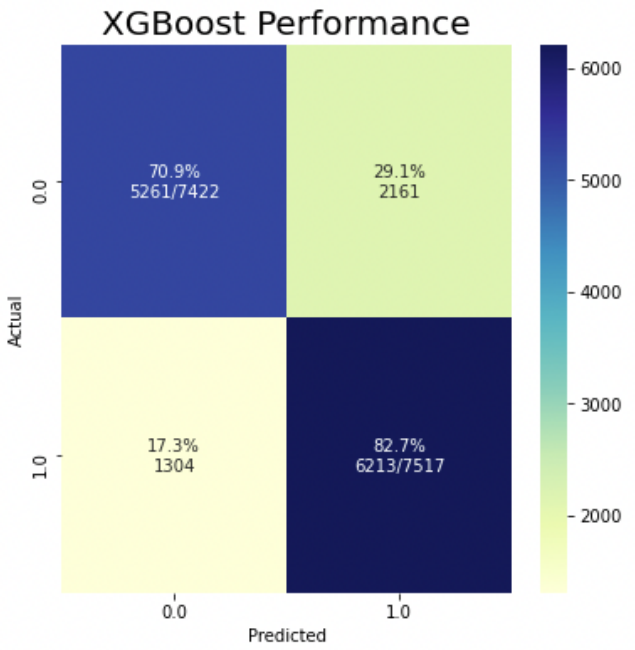


Figure 2: Target variable distribution

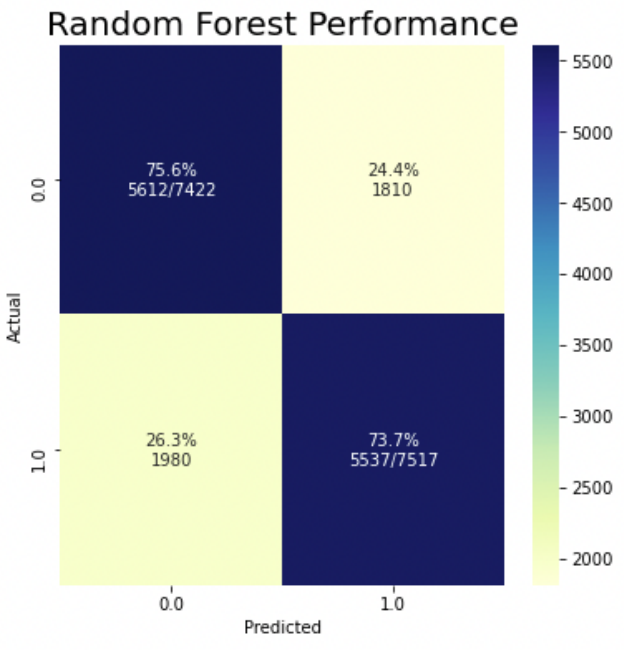
4. RESULTS

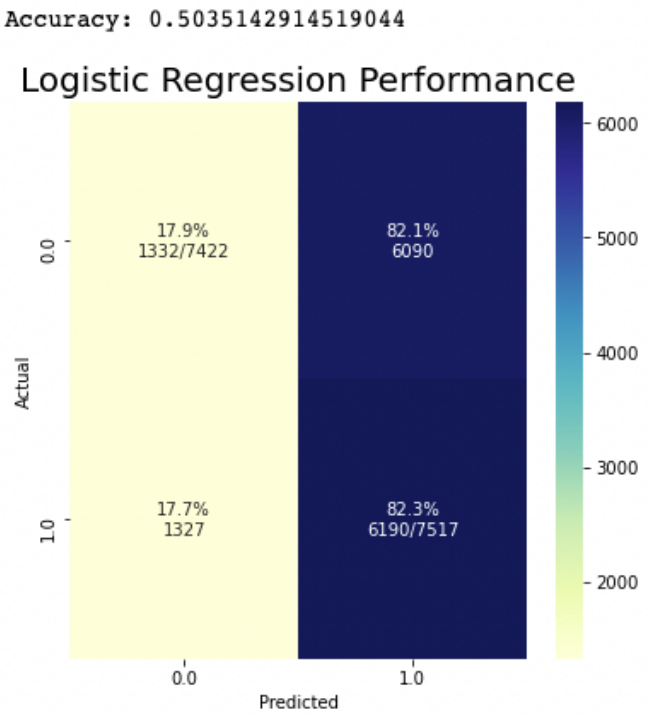
As performance metric the accuracy is used in this research. Using accuracy is suitable in this context since we are making profit with True positives and we are protected from loses with true negatives. Since we are not considering the profit rate in our target variable, using accuracy rate as the performance metric seem to be intuitive. Her I will provide the confusion matrices to compare the results of different models. Also, even though we build different models for 21 different coins; I will present the performance metrics of models predicting BTCUSD model.

```
Accuracy: 0.7680567641743089
```




```
Accuracy: 0.7463016266149006
```





As we can see from confusion matrices, best performing model is XGBoost with 77% accuracy. Its performance for detecting the 1's is better with 82.7 % True Positive rate. However for detecting price falls (0's) the XGBoost model performs poorer than Random Forest model.

As a result XGBoost model can be used for a better performance. But RandomForest model provides us a safer decisions (less false positives) even though its accuracy is lower than XGBoost. However, Picking XGBoost model seems reasonable to me since it is more accurate. The safety of model could be increased by arranging the probability treshold.

Logistic Regression model performs very poorly with 50 % accuracy (same with random assignment). The reason behind it is probably the multicollinearity problem stated in the modelling section. The issues in logistic regression should be handled by better feature selection and decomposition of trend and stationarity. However, it seems impossible for it to go beyond XgBoost model.

XgBoost Model accuracies in test sets for each coin is like following:

'BTCUSDT': 0.77,
'ETHUSDT': 0.77,
'BNBUSDT': 0.74,
'DOGEUSDT': 0.78,
'ADAUSDT': 0.76,
'MATICUSDT': 0.76,
'DOTUSDT': 0.64,
'TRXUSDT': 0.77,
'LTCUSDT': 0.77,
'SOLUSDT': 0.72,
'UNIUSDT': 0.69,

'AVAXUSDT': 0.77,
'LINKUSDT': 0.76,
'XMRUSDT': 0.81,
'ATOMUSDT': 0.76,
'ETCUSDT': 0.78,
'XLMUSDT': 0.83,
'ALGOUSDT': 0.78,
'VETUSDT': 0.81,
'NEARUSDT': 0.77,
'HBARUSDT': 0.88}

For all coins accuracy levels are good and can be improved with better parameter tuning. Espacially, performance of HBARUSDT is 88% which is very good. I believe it has some potential to be used in live trading after some backtesting and beter tuning.

5. CONCLUSION

In conclusion, recent studies have shown that machine learning models have outperformed econometric models in predicting cryptocurrency pricing behavior. This is due to their ability to handle large amounts of data and identify complex patterns, which econometric models struggle to do. The use of machine learning models in cryptocurrency prediction is a growing field and shows promise in providing more accurate predictions in the future.

Bibliography

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

2. Khedr, A.M. *et al.* (2021) “Cryptocurrency price prediction using traditional statistical and machine learning techniques: A survey,” *Intelligent Systems in Accounting, Finance and Management*, 28(1), pp. 3–34. Available at: https://doi.org/10.1002/isaf.1488.

3. Akyildirim, E., Goncu, A. and Sensoy, A. (2020) “Prediction of cryptocurrency returns using machine learning,” *Annals of Operations Research*, 297(1-2), pp. 3–36. Available at: https://doi.org/10.1007/s10479-020-03575-y.