Barış Özakar
Doğan Parlak
Finance and ML
31.12.2021

“Prediction of Cryptocurrency Returns Using Machine Learning” Project Report

Motivation:

From the perspective of a cryptocurrency trader, it is crucial to determine the direction of prices. In case of upward and downward movements a trader can decide whether to take a long position in cryptocurrencies beforehand or short sell the cryptocurrencies. The question of whether cryptocurrency prices are predictable depend on the notion of Efficient Market Hypothesis (EHM). EHM suggests that in an efficient market, any past information is already embedded in the current prices of the cryptocurrencies. Especially, in the case of the weak-form efficiency, future returns cannot be predicted on the basis of past price changes since the future movement follows a random walk which is equivalent to a martingale process. Most of the research point out that Bitcoin is not weakly efficient, but it tends to gain weak-form efficiency over time. Although there has been extensive research for Bitcoin, the research regarding alternative coins have been lacking. In one study, Wei (2018) states that there is a strong negative relationship between return predictability with cryptocurrency liquidity. With these notions in mind, “Prediction of Cryptocurrency Returns Using Machine Learning” by Akyildirim et al aims to analyze the predictability of the most liquid twelve cryptocurrencies in high frequencies.

Introduction:

In our group project, we wish to replicate the work by Akyildirim et al and achieve similar results that range from 44%–65% predictive accuracy at even more high-frequency at 1-minute level frequency as opposed to Akyildirim et al’s 15-minute level frequency. However, more importantly, we find that their research has a lack of analysis with regards to feature selection and feature importance. In the paper, they have stated the features used for the training (including price, RSI and moving averages) but fail to comment on their effect on prediction. As a group we have further explored the features relevant to predictability. We have applied feature selection and feature extraction methods to increase the performance of the models.

Therefore, in this project, we have done a binary classification task which predicts the direction of “close prices” of several cryptocurrencies utilizing high level frequency (minute level frequency) data. In order to achieve this goal, we have used feature engineering to handcraft the most prevalent technical features used in technical analysis; and used model-dependent and model-agnostic feature selection methods to select the most informative and relevant features. With relevant features, we have trained our baseline decision tree model and aimed to improve the prediction accuracy of returns in varying time horizons using support vector machine, logistic regression, artificial neural network, recurrent neural network and random forest models and observed the performance of distinct models on such a task.

We observed the extent to which the direction of close prices is predictable and examined which model performs best as well as the importance of the features that allow for the prediction. This also allowed us to understand the complexity required for a model to obtain a meaningful outcome.

Dataset:

For this task, we have chosen the minute-level open, close, high, low, amount traded and volume of 5 cryptocurrencies which are comparatively liquid because of their market cap and volume: ADA, BTC, DOGE, ETH and LTC.

Data Preparation and Feature Engineering:

We have 720600 observations for each cryptocurrency. Hence, below mentioned operations are common for all the cryptocurrencies.

Initially, we have timestamp, id, amount, open, high, low, close, count and volume factors for each observation. To extend our analysis and utilize from further financial and technical analysis metrics, we have calculated and included percent change with respect to one timestamp back return-1, similarly return-2, return-3, return-4, return-5, RSI, range, 7-SMA, 14-SMA, 20-SMA, distance from Bollinger up, distance from Bollinger down, 7-EMA, 4-EMA, MACD and our label, which is 1 if the close price went up compared to the previous timestamp and 0 otherwise. In total there are 22 features in our dataset.

Feature Selection:

Furthermore, we have done feature selection among these features. Note that, from now on, since the most liquid cryptocurrency among those 5 is BTC, we have done the analysis on BTC and assumed that same results apply also for the others.

Firstly, we have used the filter-based feature selection using correlation analysis and observed the correlation between the features and the corresponding labels. We have observed the following order of *return-5, return-2, return-4, return-3, distance from Bollinger up, distance from Bollinger down, RSI, return-1, MACD, count, vol, amount, range, 20-SMA, 14-SMA, 7-SMA, 7-EMA, open, high, low* and *close.*

To have a more detailed analysis on feature selection, we have utilized model dependent tree-based (using Random Forest and Light Gradient Boosting Machine) feature selection techniques. We utilized the feature importance calculations that arise from training these models. Essentially, we realize that the reduction in MSE for features used for splitting a tree is used to calculate the feature importances. From our tree-based approach using Random Forest model, we obtained the features *return-1, return-2, return-3, return-4, return-5, amount, count, vol, range, distance from Bollinger up, distance from Bollinger down, MACD* and *RSI*. On the other hand, the Light Gradient Boosting Machine feature importances yielded the features *return-1, return-2, return-3, return-4, return-5, amount, count, vol, range, distance from Bollinger up, MACD* and *RSI*. We have noted that only *distance from Bollinger down* is not found to be important in LBGM feature importances compared to Random Forest. In conclusion, we realized that the ordering of features in filter-based correlation and the chosen features from tree-based feature importances are consistent with only minor differences. We preferred to utilize the features selected by tree-based approach utilizing Random Forest model in our training since it is consistent with the filter-based method and the LBGM.

Cross Validation and Hyperparameter tuning:

Note that, for this section, again we have done the analysis on BTC and assumed that same results apply also for the others. The performance metric used to compare models is cross-validation accuracy.

In order to tune hyperparameters and choose the optimal ones for each model, we have utilized a suitable cross validation technique for time series data called *blocking time series split* for cross validation. From all the data, we have split the last 20% percent for testing. From the remaining data, we have split 80% for training and 20% for validation and this is done for 10 blocks respectively.

Decision Tree validation:
For this model, we have done a grid search with tuning max_depth, max_features, min_samples_leaf parameters. The best combination is 10 for max_depth, sqrt for max_features and 4 for min_samples_leaf.

Logistic regression validation:
For this model, we have varied the type of penalty and exclusion of penalty i.e l1 norm, l2 norm and no penalty. The best accuracy is obtained with l1 norm.
Support Vector Classifier validation:
For this model, we have done a grid search with tuning regularization (C) and type of kernel. The best combination is C being 1 with rbf kernel.

Random Forest validation:
For this model, we have done a grid search with tuning n_estimators, max_depth, max_features and min_samples_leaf parameters. The best combination is 200 for n_estimators, 5 for max_depth, log2 for max_features and 5 for min_samples_leaf.

Multilayer Perceptron validation:
For this model, we have done a grid search with tuning hidden_layer_sizes, learning_rate, learning_rate_init and epsilon. The best combination is (150, 75, 30, 2) for hidden_layer_sizes, adaptive for learning_rate, 10^{-3} for learning rate init and 10^{-8} for epsilon.

Reccurent Neural Network validation:
For this model, we have varied the type of build_fn between GRU and LSTM. The best accuracy is obtained with GRU.

Results:

Upon tuning the hyperparameters in cross validation, we have trained each model with their corresponding best performing hyperparameters. In order to judge the predictive quality of the models in high-frequency time series data, we have divided the test set into subsets of first 15, 100, 1000, 10000 minutes which represents a varying prediction time horizon.

Baseline: Decision Tree	Subset of test set			
	First 15min	First 100min	First 1000min	First 10000 min
ADA	0.4000	0.5100	0.5135	0.5049
BTC	0.6667	0.6200	0.5310	0.5167
DOGE	0.4000	0.4800	0.5060	0.5122
ETH	0.8000	0.5300	0.4990	0.5016
LTC	0.5333	0.5400	0.5280	0.5077

Random Forest	Subset of test set			
	First 15min	First 100min	First 1000min	First 10000 min
ADA	0.3333	**0.5300**	**0.5290**	**0.5129**
BTC	0.5333	0.5400	**0.5340**	**0.5265**
DOGE	**0.5333**	**0.5300**	**0.5060**	**0.5193**
ETH	0.7333	**0.5800**	**0.5040**	**0.5118**
LTC	**0.6000**	**0.5400**	0.5160	0.5061

Logistic Regression	Subset of test set			
	First 15min	First 100min	First 1000min	First 10000min
ADA	**0.8000**	**0.5700**	**0.5190**	**0.5062**
BTC	0.5333	0.5700	**0.5500**	**0.5234**
DOGE	**0.5333**	**0.5200**	0.4760	0.5045
ETH	0.6667	**0.5600**	**0.5030**	**0.5080**
LTC	0.4667	0.5000	0.5270	**0.5105**

Support Vector Classifier	Subset of test set			
	First 15min	First 100min	First 1000min	First 10000min
ADA	**0.4667**	**0.6400**	**0.5310**	**0.5085**
BTC	0.5333	0.6100	**0.5480**	**0.5290**
DOGE	**0.5333**	**0.5100**	0.5010	**0.5155**
ETH	0.4667	0.5000	**0.5060**	**0.5163**
LTC	0.4000	0.5200	0.5170	**0.5105**

Multilayer Perceptron	Subset of test set			
	First 15min	First 100min	First 1000min	First 10000min
ADA	**0.6667**	**0.6100**	**0.5240**	**0.5130**
BTC	0.5333	0.5800	**0.5330**	**0.5219**
DOGE	**0.5333**	**0.5500**	0.4900	**0.5153**
ETH	0.4000	**0.5300**	0.4970	**0.5166**
LTC	**0.5333**	**0.5700**	0.5190	**0.5142**

Recurrent Neural Network (GRU)	Subset of test set			
	First 15min	First 100min	First 1000min	First 10000min
ADA	**0.6000**	**0.5500**	**0.5240**	0.4913
BTC	0.5333	0.5100	0.5290	**0.5288**
DOGE	0.3333	**0.5100**	0.4810	**0.5140**
ETH	0.5333	**0.5500**	0.4980	**0.5123**
LTC	**0.5333**	**0.5400**	0.5130	0.5038

The accuracy results for our 1-minute level analysis suggest that using different models, we were able to achieve an accuracy above the expected 0.50 for almost all of the time horizon and cryptocurrency combinations. Moreover, the results also show that by using different models, we have improved the accuracy results acquired by the baseline decision tree model in almost all prediction time horizons (bold numbers indicate better accuracy compared to baseline).

More concretely, all the models were able to improve the baseline model in more than half of the crypto-time horizon combinations. Random forest and support vector classifier models were able to outperform the baseline especially in the longer time horizons (first 1000 and 10000 minute). Specifically, the random forest model was able to achieve both above expectation and above baseline performance for almost all the combinations.

Most models were especially successful in particular time horizons. For instance, for the first 100 minutes and for all the cryptocurrencies, all models except the baseline decision tree model were able to achieve more than the expectation of 0.50. Again, all the models except the recurrent neural network model were able to get an accuracy above 0.50 in the first 10000-minute subset of the test set for all cryptocurrencies (the RNN model gets 0.4913 for ADA).

With our analysis, we were able to conclude by looking at above expectation accuracies of 0.50 for most of the crypto-time horizon combinations, that cryptocurrencies including Bitcoin and many liquid altcoins, can be considered as not yet having weak-form efficiency and that future returns can be predicted on the basis of past price changes and carefully engineered and selected features, even in a high frequency setting.