ML Methods for Return Predictions and Cryptocurrency Portfolio Construction

Project Report

Machine Learning Under a Modern Optimization Lens

Jaupi, Megi
jaupimeg@mit.edu

Rocafort, Roland
rolandr@mit.edu

Fall 2022

1 Introduction

Asset return prediction has always been the core of financial research and remains the basis of constructing trading strategies and investment portfolios. The literature is separated into two main branches, namely using fundamental (financial statements, market sentiment) or technical (price related measures) indicators to extract signals as to how the asset prices will move. Considering the highly volatile environment in which cryptocurrencies operate and the speculative nature of it as an asset class, we hypothesize the latter approach as an interesting realm to explore. Building upon this, our goal is to explore whether ML models are of use in predicting asset returns. Since machine learning models have been extensively used recently for traditional asset classes such as stocks [1], we are rather interested in making use of these methods in cryptocurrencies. Further, while tree-based methods have been used in the literature to predict asset returns [2], to the best of our knowledge, optimal regression trees are yet to be applied to the problem.

2 Data and Feature engineering

We used data from Investing.com, which has daily historical prices for all cryptocurrencies since their inception. We ended up selecting 13 different cryptocurrencies in total in the time-frame starting from July 1, 2018, to November 1, 2022., as it was the maximum amount of time for which we could find this many coins in circulation.

The coins chosen (along with their market tickers) are the following: Ripple - XRP, Neo - NEO, Stellar - XLM, Dogecoin - DOGE, Tether - USDT, NEM - XEM, Tron - TRX, Nxt - NXT, Ethereum - ETH, Peercoin - PPC, Binance - BNB, Bitcoin - BTC, and Litecoin - LTC. The raw dataset consists of daily Open, Close, Low, High levels of prices and daily Volume.

One of the key parts of this project is the construction of the technical indicators that will serve as features for the prediction of future returns. All technical indicators are function of historical price or volume. There is a vast literature on the construction of indicators with claimed predictive power. We focus on constructing a set of indicators that capture different aspects of the price dynamics:

- Momentum: indicators such as Exponentially Weighted Moving Average or Average Directional Movement (ADX), Absolute Price Oscillator (APO) etc aim to measure the rate of increase or decrease of a price.
- Volatility: such as or Bollinger Bands (BB) or Average True Range (ATR) measure the spread of price movements.
- Volume: such Chaikin Accumulation Distribution Line measure the cumulative flow of money into and out of an asset and indicate changes in liquidity in the markets.
- Cycles: such as Hilbert Transformation of prices aim to measure phase transitions.
- Patterns: such as Three Inside Up/Down aim to discover pattern reversals, i.e. points in time when current trends may be ending and movements in the other direction may start.

In total, we construct 47 transformations for each of the crypto assets in the basket, or in other words, 47 input features for our models.

3 Methodology

As mentioned, returns prediction is one of the most difficult undertakings in the financial market's realm. Even with the construction of a large number of technical indicators it is important to note that market signals are short-lived. The large amount of participants in the markets make these signals transitory.

That being said, we will first focus on predicting whether technical indicators can provide us with useful signals if the price of an asset will go up or down, namely a classification problem, with focus on interpretability. Afterwards we will try to use models to predict the actual value of the future returns. Rather than interpretability, in this case we are more interested in accuracy. We will test whether predicting future returns brings real life value by using the predicted returns to construct optimal portfolios.

We compute daily returns out of the daily prices as follows:

$$r_t = \frac{p_t}{p_{t-1}} - 1 \tag{1}$$

Then we construct classes of returns as:

$$r_t^c = \begin{cases} 1, & \text{if } r_t \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

For the classification problem, per crypto asset, we will fit various ML models of the form:

$$\hat{r^c}_{t+1} = f^c(\tilde{X}_t) \tag{3}$$

For the regression problem, per crypto asset, we will fit various ML models of the form:

$$\hat{r}_{t+1} = f(\tilde{X}_t) \tag{4}$$

where $\tilde{X}_t$ is the feature vector at time t containing all the engineered technical indicators explained in the previous section.

For the construction of optimal portfolios, we we make use of Portfolio Theory and the Markowitz Efficiency Frontier. The goal of our optimized portfolio is to minimize risk while maximizing expected returns that is why we choose to construct Maximum Sharpe Ratio portfolios. In order to achieve this, we optimize over the weights assigned to each of assets in the portfolio, which in this case are the different coins. Thus, we describe an optimal portfolio as the solution to the following optimization problem:

$$\max_w \frac{w^T \hat{r}_{t+1}}{w^T \Sigma w} \tag{5}$$

$$\text{s.t} \sum_{i=1}^{n} w_i = 1 \tag{6}$$

$$w_i \geq 0 \quad \forall i \in n \tag{7}$$

where $\Sigma \in \mathbb{R}^{n \times n}$ is the covariance matrix for all asset returns, and $\hat{r}_{t+1} \in \mathbb{R}^n$ is an array with the predicted returns for all n coins. For the purpose of this exercise, we

Coin	Logistic Regression	XGBoost	OCT
XRP	0.5306	0.5789	0.5263
NEO	0.5714	0.6315	0.6842
Stellar	0.5102	0.5263	0.5510
Dogecoin	0.4693	0.5263	0.4736
Tether	0.6938	0.5789	0.7368
XEM	0.5102	0.6315	0.5263
TRX	0.5510	0.2631	0.5789
NXT	0.5306	0.4736	0.5263
PPC	0.5510	0.5263	0.6315
BNB	0.5102	0.5789	0.5789
Ethereum	0.5306	0.3684	0.5510
Bitcoin	0.4897	0.4736	0.591
Litecoin	0.5510	0.3684	0.5306

Table 1: Test set performance of the classification problem. Evaluating models based on Accuracy.

abstract from the risk-free rate, and optimize over returns instead of excess returns. Furthermore, we construct a long-only portfolio, meaning that we do not consider the shorting of assets, which is implied by our constraint that the weights have to be equal to or greater than zero.

4 Experiments and Results

4.1 Predicting return movements

For the classification experiment, we split the data into 1300, 50 days respectively for the train and test set. We train our models using cross-validation via grid search, tune the hyperparameters and use the fitted models to predict on the test set. We do not retrain the models on a rolling window for this experiment.

The results of the models on the test set as measured by Accuracy are given in Table 1 and AUC in Table 2. Notably all models perform little over 50% accuracy. In fact, logistic regression seems to not being able to learn much as AUC scores are all 0.5, i.e. we might as well just predict randomly and flip a coin. However, if we observe the AUC scores, OCTs perform consistently above 0.5. By the nature of the financial markets, we would expect that the data is mostly noise and very little signal, however using the models seems to be a bit better than random. Surprisingly, OCT perform better than the ensemble method XGBoost. Even though cross-validated, it appears that an XGBoost with 10000 estimators tends to overfit and does not generalize well on the test set.

Coin	Logistic Regression	XGBoost	OCT
XRP	0.5	0.5611	0.52
NEO	0.5	0.7307	0.5
Stellar	0.5	0.4464	0.4642
Dogecoin	0.5	0.573	0.5
Tether	0.5	0.5	0.6785
XEM	0.5	0.6477	0.55
TRX	0.5	0.2784	0.5
NXT	0.5	0.4772	0.5
PPC	0.5	0.4545	0.55
BNB	0.5	0.5773	0.5789
Ethereum	0.5	0.3555	0.5475
Bitcoin	0.5	0.4602	0.60
Litecoin	0.5	0.4935	0.5

Table 2: Test set performance of the classification problem. Evaluating models based on AUC.

On to the element that induced our interest in this project. It is common for technical traders to consult a great number of indicators. In practice, the technical indicators are compared to heuristic benchmarks based on previous experiences to understand if the market is bearish or bullish. Hence we try here to assist a trader's judgement with OCTs.

Let us choose Bitcoin to interpret as the most well-known cryptocurrency. Inferring from the OCT, one may say that in a Dominant Cycle Phase (HT_DCPHASE) less than 36 degrees, with positive returns of Lag 4 and a Directional Movement Rating (ADXR) smaller than 22, we would expect the returns tomorrow to be negative. This is somewhat intuitive, a low value of a cycle phase for practicioners represents a regime shift, and a small Directional Movement Rating indicates that a bullish market is weakening. On the contrary, following the OCT at the same path but with a Directional Movement Rating larger than 22, we would expect positive returns tomorrow. Perhaps because there is still the momentum from the bullish market that will delay the regime shift. This is just one example of the many patterns that can be extracted from the OCT.

4.2 Predicting return values

The aim of our regression experiment is to predict the next day returns for each of the cryptocurrencies, $r_{t+1,i}$, given returns data from the past 100 days. In other words, we use a 100-day rolling window to predict returns for coin i. We use four different regression models: (1) Linear Regression, (2) LASSO, (3) Random Forest, and (4) XGBoost.

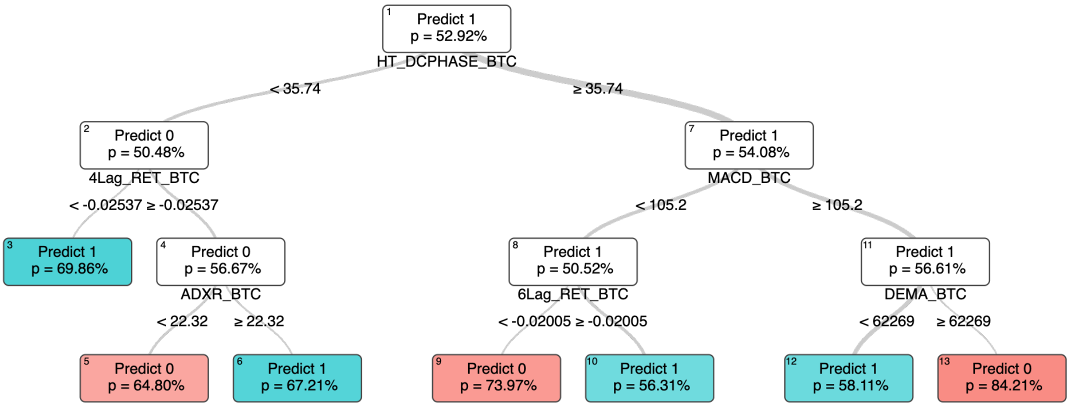

Figure 1: Cross-Validated OCT for BTC. We observe that out of the 47 constructed indicators, the classifier only selected 6, making the process of judging over a bull or bear market easier.

XGBoost Model	OSR2	RMSE	MAE
16 Features	-0.02201	0.06399	0.05024
47 Features	0.0117	0.06292	0.04901

Table 3: Mean validation set performance of the simplified and normal regression problem on XGBoost. Evaluating models based on OSR2, RMSE and MAE.

In order to train our models, we use cross-validation via grid search and tune the relevant hyper-parameters. We only tune the hyper-parameters once, for the first 100 days in which our data is available, and use these parameters for the remaining 100-day windows. We further experiment with $\tilde{X}_t$, the feature vector at time t, and perform the original regression problem, using all 47 features, and a simplified regression, using only features for lag returns and moving averages, totalling 16 features. This way we can also look at how much more information we can extract from additional technical indicators. We explore this with an XGBoost regression. Overall, we saw a some improvement using all factors, as evidenced by Table 3, even though MAE and RMSE are still very similar for both regression problems. Still, due to larger improvements in OSR2 we use all features for following experiments.

We validate each of our model's performance on data from June 14, 2022 to August 23, 2022, and test our model performance on data from August 24, 2022 to November 1, 2022. We do this such that we can select the best model from the validation set to use in our construction of the optimal portfolio, which will use data from the test set.

Table 4 and Table 5 show the out of sample R2 for the validation and test sets for each of the models and all cryptocurrencies. Looking at both of these tables, it is clear that model performance is very low for this time period, as the score is usually close to zero and is sometimes negative. Given the recent volatility of the crypto

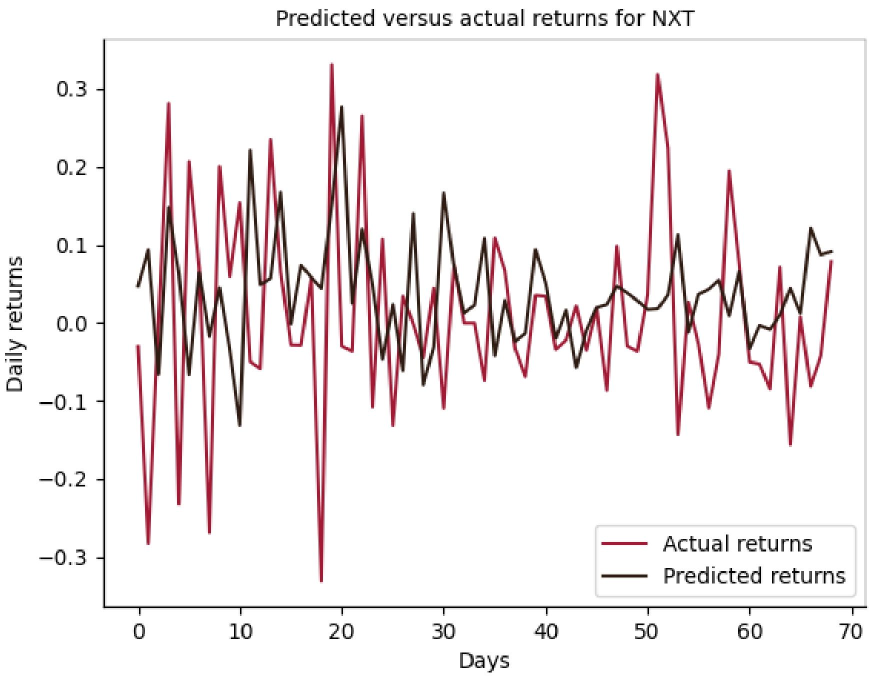

Figure 2: Plot of predicted versus actual returns of NXT coin over the 70 day horizon in the test set

markets, and the speculative nature of this asset class, we hypothesized that this was likely to happen. It is possible that in the future, once crypto markets have stabilized more, predictive models could have better performance. Still, we see that both XGBoost and LASSO have the best performance on 5 out of 13 coins each on the validation test (Table 4). However, since LASSO has negative R2 values on some coins for which it performs the best on, we say that XGBoost is overall the best performing model in the validation set, and choose it for constructing our portfolio.

Figure 2 graphs compares our prediction for NXT crypto returns to the actual values for the test set.

4.3 Portfolio Construction

From looking at the performance on the validation set (Table 4), we select the XGBoost model to construct our portfolio, as it had the best individual performance on the most coins. As our XGBoost model predicts daily returns, we run the optimization problem, as described in equation (5) for each day in the test set. For simplicity, we further assume that transaction costs associated with rebalancing the portfolio are negligible, such that we can re (i.e. update weights) every day.

In order to assess the performance of our optimized portfolio, we look at the cumulative returns over the the time horizon in the test set. We compare the cumulative returns from our optimized weights to an equally weighted portfolio, i.e. a portfolio

Coin	Linear Regression	Lasso	Random Forest	XGBoost
XRP	-0.20669	**0.02011**	0.01766	-0.02601
NEO	-0.04679	0.01767	0.06899	**0.18242**
Stellar	-0.03644	**0.00314**	-0.07168	-0.17038
Dogecoin	-0.7983	**0.00886**	-0.06951	-0.25174
Tether	-2.41325	**-0.00396**	-0.02014	-36.28501
XEM	-0.06198	0.00518	**0.17291**	0.06544
TRX	-0.36164	**-0.00304**	-0.09713	-0.06743
NXT	-0.02821	0.0071	0.21053	**0.45577**
Ethereum	-0.13568	-0.10478	0.12163	**0.26334**
PPC	0.08855	0.00088	**0.10203**	0.01829
BNB	-0.79325	-0.04421	-0.02437	**0.0509**
BTC	-0.10166	0.00054	**0.03663**	-0.13097
Litecoin	-0.04627	-0.00432	0.02614	**0.07124**

Table 4: Validation set performance of the regression problem. Evaluating models based on OSR2.

Coin	Linear Regression	Lasso	Random Forest	XGBoost
XRP	-0.112	0.01628	0.03257	-0.04692
NEO	-0.04287	-0.00633	0.11255	0.1065
Stellar	-0.1414	0.00073	-0.00529	0.01108
Dogecoin	-0.93677	0.00498	0.2358	0.37535
Tether	-5.69631	-1.08582	-1.97348	-379.68553
XEM	-0.11068	-0.02276	-0.35908	-0.55305
TRX	-0.62079	-0.07032	-0.0987	-0.26783
NXT	0.00617	0.00467	-0.08199	-0.0185
Ethereum	-0.14812	-0.02301	0.01804	-0.00485
PPC	0.06412	0.03302	0.02444	0.09811
BNB	-0.30573	0.01991	0.03894	0.01363
Bitcoin	-0.14352	-0.0569	-0.0134	-0.09737
Litecoin	0.01503	0.03567	0.03078	-0.22615

Table 5: Test set performance of the regression problem. Evaluating models based on OSR2.

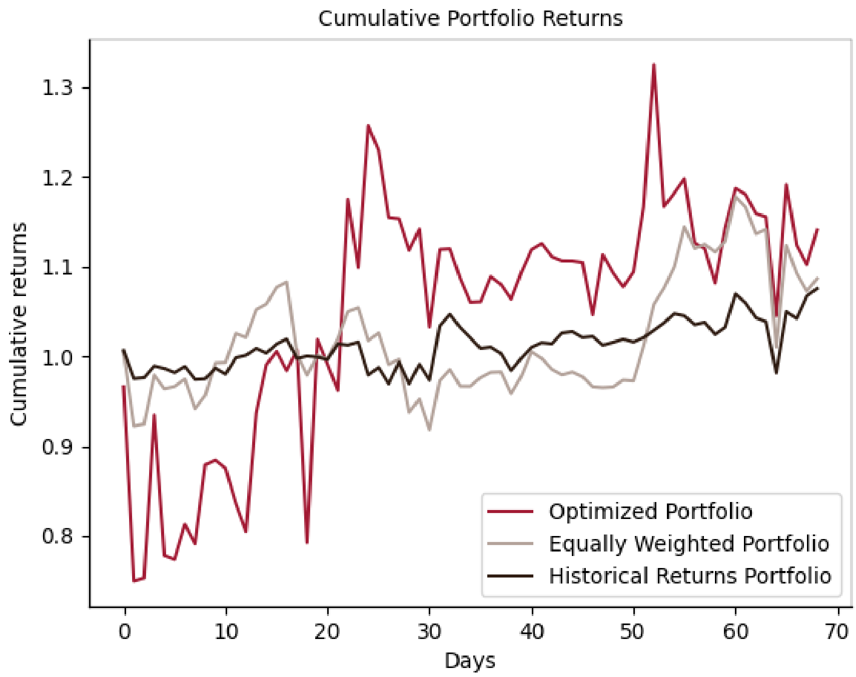

Figure 3: Plot of cumulative returns for optimized portfolio, equally weighted portfolio and historical returns portfolio over the 70 day horizon in the test set

where $w_i = \frac{1}{n}$, for all $i \in n$, and to an optimized portfolio where we instead use $\overline{r}$, which is estimated using the historical mean of the asset return in the training set, instead of our XGBboost prediction $\hat{r}$.

Figure 3 shows the cumulative retruns of our optimized portfolio, an equally weighted portfolio and historical mean portfolio. Here, we see that over our 10 week time horizon, our optimized portfolio performs better than the baseline equally weighted portfolio by 4.99% and better than the historical returns portfolio by 6.07%. Furthermore, our equally weighted portfolio performed better than our optimized historical means portfolio by around 1.03%. Overall, our optimized portfolio saw an 18.08% increase in the span of 70 days, compared to an 8.19% increase and a 6.87% increase for the equally weighted and historical returns portfolios, respectively.

5 Conclusions

Our first goal of this exercise was to predict whether a crypto asset's return will go up or down in the next time period. Model performance varied a lot from coin to coin, but overall we saw (little) improvement over the baseline (of 0.5) for both AUC and accuracy. Still, we were able to gain valuable insights from our OCT, which makes very interpretable predictions while performing in a more stable way than XGBoost. In fact, given a better technical indicators and assuming that those indicators would provide a better prediction of whether markets will increase or decrease, OCTs can automatically generate trading strategies. A natural extension to this project would precisely be a backtesting of the trading strategies generate by following the branches of the trained OCT.

After predicting the movement of the crypto asset, we then attempted to predict its returns. All models show poor out of sample performance and very little predictive ability. Overall, we note that the technical price indicators used are not good predictors of returns for this particular asset class. We knew this was the likely outcome of this experiment, as the speculative nature of cryptocurrencies, as well as the increasingly volatile environment, would make crypto much harder to predict, compared to other asset classes.

Finally, we built an optimal portfolio based on predicted returns. Surprisingly, we saw that even though our predictive ability was very poor, our optimized portfolio beat both the equally weighted portfolio and the optimized portfolio built from mean historic returns. Our portfolio saw an increase in cumulative returns of 18.08% over the 70 day span, which was also quite surprising. Still, we are quite wary of these results, due to our model's poor performance.

The experiments conducted help to prove that while some valuable insights can be gained from technical price indicators, they are overall bad predictors of future returns, particularly for this asset class. Still the few insights gained from using optimal trees show that there is a lot of potential for their use in making machine learning in financial markets more interpretable. We also believe that our experiments may have been hindered by the current market environment and high volatility in cryptocurrencies, which is still a fairly new asset class. In the meantime, it is important to continue experimentation with more technical and fundamental indicators of crypto returns, to continuosly improve our modeling.

6 Student Contributions

The work was equally distributed between both members of the team. With her background in finance and financial markets, Megi took the lead in terms of coming up with the project methodology and scope. For the implementation, Roland handled the creation of the dataset, (downloading individual data files and merging) and Megi then created the technical indicators using the TA-Lib module in python. Afterwards, Megi and Roland worked on the classification and regression problem, respectively, while helping each other out with any roadblocks encountered. Megi and Roland kept a github repository and build upon each others code where possible. Megi ran Logistic, XGBoost, and OCT for classification, while Roland ran Linear, LASSO, XGBoost and RF for regression. Megi wrote the script for portfolio optimization, with some help from Roland, and then Roland used the results from his regression problem to run the optimization. The writing was also distributed pretty equally, both working together and continuously discussing their parts of the write-up. Megi also did most of the presentation, while Roland kept trying to improve the models as long as time permitted.

References

[1] Shihao Gu, Bryan T. Kelly, and Dacheng Xiu. "Empirical Asset Pricing via Machine Learning". In: *Chicago Booth Research Paper No. 18-04, Yale ICF Working Paper No. 2018-09* (2019).

[2] L. Fiévet D. Sornette. "Decision trees unearth return sign predictability in the SP 500". In: *Quantitative Finance* (2018).