94867 Decision Analytics for Business and Policy 2022 Fall

Cryptocurrency Price Prediction and Investment Portfolio Optimization

Name	Contributions
Elaine Zhang	Data preprocessing and pipeline, presentation, report
Xi Yan	Prediction using the Machine Learning model, presentation, report
Xintong Zheng	Prediction using the ARIMA model, presentation, report
Wei Xiao	Short term and long term optimization, presentation, report
Jingbo Zhang	Short term and long term optimization, presentation, report

GitHub Link:

https://github.com/Nellyan4/Decision_Crypto_Project

Dec 14, 2022

Resources:

Packages: os, Pandas, csv, numpy, stockstats, Gurobi, Matplotlib, Seaborn, pmdarima, xgboost

Decision Analytics for Business and Policy Course - Portfolio.ipynb

Carnegie Mellon University
HeinzCollege
INFORMATION SYSTEMS • PUBLIC POLICY • MANAGEMENT

Cryptocurrency Price Prediction and Investment Portfolio Optimization

Xintong Zheng, Xi Yan, Elaine Zhang, Wei Xiao, Jingbo Zhang

Table of Contents

1. Problem Statement

As a digital asset, cryptocurrencies are active in the trading market. It is a certain form of money that exists digitally or virtually and uses cryptography to protect transactions. Cryptocurrencies have no central issuer or regulator, but instead, use a decentralized system to record transactions and issue new units. With more and more people starting to invest in cryptocurrencies, it is really meaningful to dive deep into making predictions and optimizing portfolio investments.

Since we cannot make optimization based on the actual price of cryptocurrencies in real deals, we need to first predict the price of cryptocurrencies based on their historical data with ML models and time series forecasting. We can then optimize potential cryptocurrency portfolio returns (maximize the profit in the short-term and minimize the risk in the long term) based on the predicted price and compare the profit with the portfolio of the actual price.

2. Challenges

To better design and build models, we should deal with these challenges:

1) Accuracy of prediction: Due to the high volatility of the cryptocurrencies market and the constantly changing macroeconomics, it is hard to make accurate predictions, which would affect the portfolio investments.

2) Feature engineering: During the process of prediction, solely relying on the performance of cryptos is not enough, we have to do feature engineering to generate quantitative indicators.

3) Model Selection: We have to think about the selection of models since it is not necessarily linear. It is very challenging to find a model that simulates the cryptocurrencies market as most as possible.

3. Data Summary (Sources, Preprocessing, Data Dictionary)

We downloaded the dataset (Top 100 Cryptocurrencies Historical Data) from Kaggle and its usability score is 8.24. The historical prices of each cryptocurrency are stored in separate csv files from their start date.

For each cryptocurrency, we have 7 original features, which are shown in the table:

Features	**Descriptions**
Date	Date of the crypto prices
Open	Opening price of crypto on the respective date
High	Highest price of crypto on the respective date
Low	Lowest price of crypto on the respective date
Close	Closing prices of crypto on the respective date
Volume	Volume of crypto on the respective date
Currency	Type of currency

We build data pipelines to process 93 valid cryptocurrencies:

For prediction, we generate 80 quantitative indicators for each cryptocurrency based on the Open, High, Low, Close (OHLC) price to better fit with models.

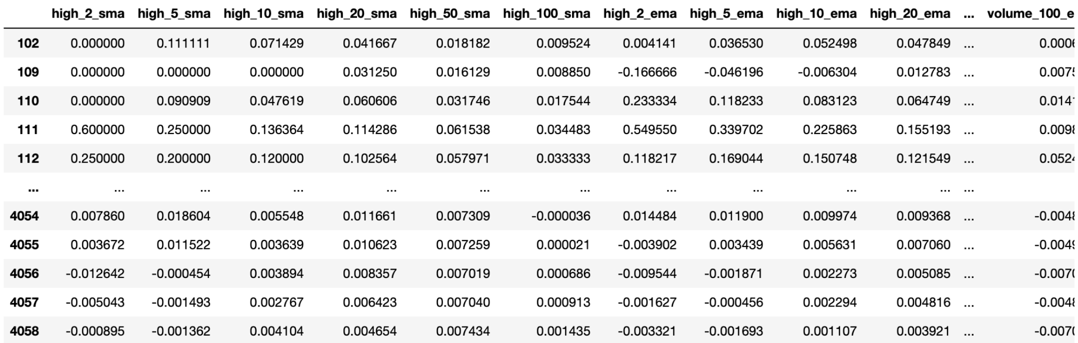
For optimization, we select a one-day Open and Close price for a short-term portfolio and a Close price for a month for long-term optimization.

4. Methodology

4.1. Machine Learning

With 93 different cryptos, we are not able to try out each one of the crypto one by one for the machine learning methodology. Our strategy is to try on one of the crypto and then apply it to the rest of them, as the data structure of each of the crypto is the same - they are OHLC data.

The goal of our machine learning regression is to predict the percentage change of the next day's closing price compared to the closing price the day before. With this prediction and today's closing price, it means that we can forecast tomorrow's crypto closing price. However, we cannot directly use the OHLC table to do prediction, since if we use simply the OHLC table as our input for the model, we are never going to be able to "peek" the future and input tomorrow's Open, High, Low and predict the closing price in advance. There is no way we will be able to get these numbers a day before. Therefore, for machine learning, we can use only the knowable data, which we determine as the moving average of High, Low, and Volume. We can always convert the data into [2, 5, 10,20, 50, 100] days of [simple, exponential, standard deviation, variance] Moving Average since they are an aggregation of past statistics. Therefore, we engineer our features to be like this. After this step, we also convert the number in each cell to the percentage change compared to the previous day. The reason why we do this is that our output is also in the percentage change format.



	high_2_sma	high_5_sma	high_10_sma	high_20_sma	high_50_sma	high_100_sma	high_2_ema	high_5_ema	high_10_ema	high_20_ema	...	volume_100_e
102	0.000000	0.111111	0.071429	0.041667	0.018182	0.009524	0.004141	0.036530	0.052498	0.047849	...	0.0006
109	0.000000	0.000000	0.000000	0.031250	0.016129	0.008850	-0.166666	-0.046196	-0.006304	0.012783	...	0.0075
110	0.000000	0.090909	0.047619	0.060606	0.031746	0.017544	0.233334	0.118233	0.083123	0.064749	...	0.014
111	0.600000	0.250000	0.136364	0.114286	0.061538	0.034483	0.549550	0.339702	0.225863	0.155193	...	0.0098
112	0.250000	0.200000	0.120000	0.102564	0.057971	0.033333	0.118217	0.169044	0.150748	0.121549	...	0.0524
...	...	...	...	...	...	...	...	...	...	...	...	
4054	0.007860	0.018604	0.005548	0.011661	0.007309	-0.000036	0.014484	0.011900	0.009974	0.009368	...	-0.0048
4055	0.003672	0.011522	0.003639	0.010623	0.007259	0.000021	-0.003902	0.003439	0.005631	0.007060	...	-0.0049
4056	-0.012642	-0.000454	0.003894	0.008357	0.007019	0.000686	-0.009544	-0.001871	0.002273	0.005085	...	-0.0070
4057	-0.005043	-0.001493	0.002767	0.006423	0.007040	0.000913	-0.001627	-0.000456	0.002294	0.004816	...	-0.0048
4058	-0.000895	-0.001362	0.004104	0.004654	0.007434	0.001435	-0.003321	-0.001693	0.001107	0.003921	...	-0.0070

After feature engineering, we also do standard scaling and perform a standard data cleaning (clear NaN values) for machine learning to make sure the train test dataset is clean. Then, we would also need to set up the evaluation metrics for our regression model. Statistical evaluation such as R^2 and RMSE is important since it generally tells us whether this is a good model for this type of dataset. But we think that what is important for these models, since we are predicting the price for tomorrow's crypto price, we should focus more on the "winrate" of each of the models. We define the win rate as 1. result: the profit (loss) resulting from betting one unit in the direction of the sign we predict compared to the actual sign; and 2. residual: the absolute value of prediction compared to the true value.

For the model selection, we try different types of regression: linear, bagging, and boosting. We are not sure which model performs better, but we are aware of the pros and cons for each type of the model, such as the bagging will have lower variance and higher bias, and vice versa for boosting. Also, we use models that can perform feature selection on their own. Therefore, in the end, we choose [Random Forest Regression, LASSO, Support Vector Machine Regression, Stochastic Gradient Descent, Gradient Boosting, and XGboosting Regression] for machine learning prediction, the result is shown below:

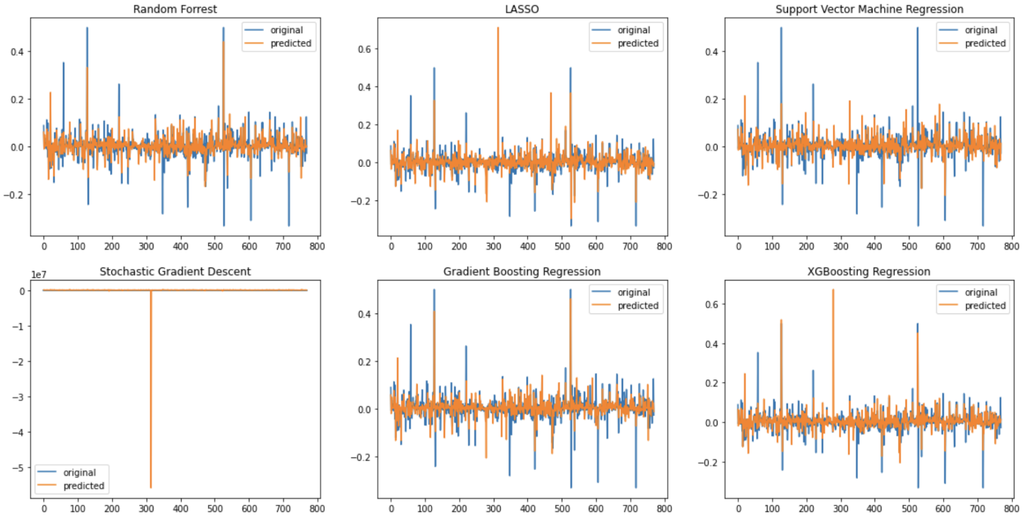


Figure Test model in the test-set

Based on the result, we can see that five models except stochastic gradient descent regression can properly fit the test set. As we dig further, we find that linear models such as LASSO and SVM do not fit as well as bagging and boosting regressions. We also found that Random Forest has the best "winrate", this is because of the high variance nature of the model. As we try on other cryptocurrencies, we found that RandomForest does not always have the lowest sum of "residuals" we defined previously.

In conclusion, when we apply a machine learning method to all of the cryptos, we let the machine itself decide which model to use as the final prediction for the percentage change for tomorrow's price based on which model can get the smallest sum of "residuals" for all the test data.

4.2. ARIMA

ARIMA, also known as autoregressive integrated moving average, is a regression analysis model widely used for time series data. The final objective of the model is to predict future time series movement by examining the differences between values in the series. The inputs are the 93 crypto files, and each file includes Date, Open price, High price, Low price, Close price, Volume, and

Currency attributes. The output is one single csv file, including the crypto name, actual open price on Aug 28th, and predicted close price on Aug 28th.

There are seven steps to apply the ARIMA model to the datasets. First, we used the ADF test to test for stationarity. Second, we separated the trend and decomposed the series. Third, we eliminated trends. Fourth, we split data into training and testing sets. Fifth, we applied the auto-ARIMA function embedded in Python to identify the most optimal parameters. The parameters we focused on tuning were p (the number of lag observations included in the model), d (the number of times that the raw observations are different), and q (the size of the moving average window). Sixth, we built models based on the selected optimal parameters. Last, we performed the forecasting.

Similar to what we did during machine learning, before we automated the forecasting procedure, we randomly chose one dataset to test if our approach was applicable and how it performed. The ARIMA model and the automated procedure turned out to work well on our datasets.

4.3. Short-term Optimization (Maximize One-Day Expected Profit)

4.3.1 Without prediction

Available data: 93 cryptos' open prices and close prices on Aug 28th

Other concerns: the investor is risk-averse so he does not want more than 30% of his investment funds in any single crypto.

Decision variables: Xi represent the amount of crypto i purchased on Aug 28th, 2021

Parameters:

1) $budget$ represents the amount of money invested (budget)
2) $open_i$ represents the open price for each crypto i
3) $close_i$ represent the close price for each crypto i
4) $percentage$ represents the percentage of investment funds in any single crypto (30%)

Objective: maximize expected profits Z = $\sum_{i=1}^{93} (close_i - open_i) * Xi$

Constraints:

1) The total amount of cryptos purchased should be less than the budget:

   $\sum open_i * Xi <= budget$

2) Each kind of crypto purchased should be less than 30%
   $open_i * Xi <= budget * percentage$, for i in {1, 2, 3… 93}
3) Xi > 0, for i in {1, 2, 3… 93}

4.3.2 With prediction

Decision variables: Xi represent for the amount of crypto i purchased on Aug 28th, 2021

Parameters:

1) $budget$ represents the amount of money invested (budget)

2) $open_i$ represents the open price for each crypto i
3) $predict\_close_i$ represents the predicted close price (ML/ARIMA results) for each crypto i
4) $percentage$ represents the percentage of investment funds in any single crypto (30%)

Objective: maximize expected profits Z = $\sum_{i=1}^{93} (predict\_close_i - open_i) * Xi$

Constraints:

1) The total amount of cryptos purchased should be less than the budget:

   $\sum open_i * Xi <= budget$

2) Each kind of crypto purchased should be less than 30%:
   $open_i * Xi <= budget * percentage$, for i in {1, 2, 3… 93}
3) Xi > 0, for i in {1, 2, 3… 93}

4.4. Long-term Optimization

Available data: 93 cryptos' open prices, close prices, and risks for one month (July 28th, 2021 to August 28th, 2021). Risk is represented using standard deviation.

Decision variables: Xi represent for the amount of crypto i, i is from {1, 2, 3…93}

Parameters:

1) $budget$ represents the amount of money invested (budget)
2) $open_i$ represents the open price for each crypto i on July 28th 2021
3) $close_{it}$ represent the close price for each crypto i on t, from July 28 th 2021 to August 28th 2021; $\overline{close_{it}}$ is the average value of crypto i for the 30 days.
4) $percentage$ represents the percentage of investment funds in any single crypto (30%)

Objective:

minimize portfolio risk Z = $\sum_{i=1}^{93}\sum_{j=1}^{93} Xi * Xj$ * Cov[($close_{it}$- $\overline{close_{it}}$),($close_{jt}$- $\overline{close_{jt}}$)]

Constraints:

1) The total amount of cryptos purchased should be equal to the budget:

   $\sum open_i * Xi = budget$

2) Each kind of crypto purchased should be less than 30%:
   $open_i * Xi <= budget * percentage$, for i in {1, 2, 3… 93}
3) Xi > 0, for i in {1, 2, 3… 93}

5. Prediction Results

5.1. Machine Learning

The Machine Learning output generates the percentage change of tomorrow's closing price, on August 28, 2021, compared to August 27's price. Since the results are percentage change, we multiply it by August 27's closing price to get the forecast price for August 28's closing price. This prediction result generated from the machine learning model becomes the input of our short-term optimization model.

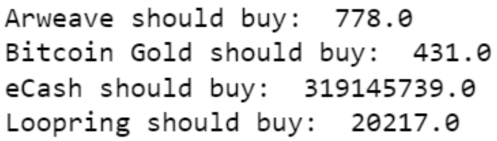
5.2. ARIMA

The ARIMA model is mainly used to predict the close price for each cryptocurrency on August 28, 2021. The prediction results are stored in one single csv file named *output_ARIMA*, including the crypto name, actual open price on Aug 28th, and predicted close price on Aug 28th. And the prediction results generated from the ARIMA model become the inputs of our short-term optimization model.

6. Optimization Results

6.1. Short-term Optimization

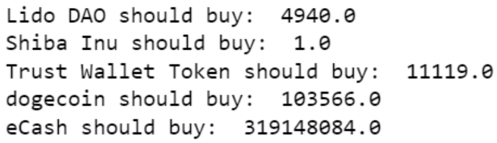
With a budget of $100,000, the optimal value (maximum expected profit on August 28th) is $31,386 without using any prediction results. The corresponding optimal investment plan:



```
Arweave should buy:  778.0
Bitcoin Gold should buy:  431.0
eCash should buy:  319145739.0
Loopring should buy:  20217.0
```

This profit is the most ideal scenario but not practical. So we replace the actual close prices with the predicted prices from ARIMA and machine learning results.

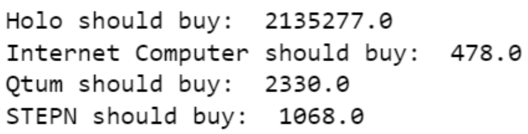
With the predicted close prices from machine learning and the same budget, the optimal value (maximum expected profit on August 28th) is $7,971, less than the ideal number. The corresponding optimal investment plan:



```
Lido DAO should buy:  4940.0
Shiba Inu should buy:  1.0
Trust Wallet Token should buy:  11119.0
dogecoin should buy:  103566.0
eCash should buy:  319148084.0
```

$7,971 is calculated based on the predicted price from the machine learning model. If we actually purchase cryptos according to the optimal investment portfolio, the actual expected return would be $18,088. In other words, we can actually generate profits using the machine learning model.

With the predicted close prices from ARIMA and the same budget, the optimal value (maximum expected profit on August 28th) is $3,666, much less than the ideal number. The corresponding optimal investment plan:

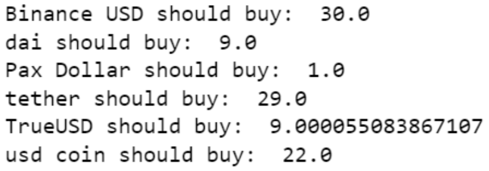


```
Holo should buy:  2135277.0
Internet Computer should buy:  478.0
Qtum should buy:  2330.0
STEPN should buy:  1068.0
```

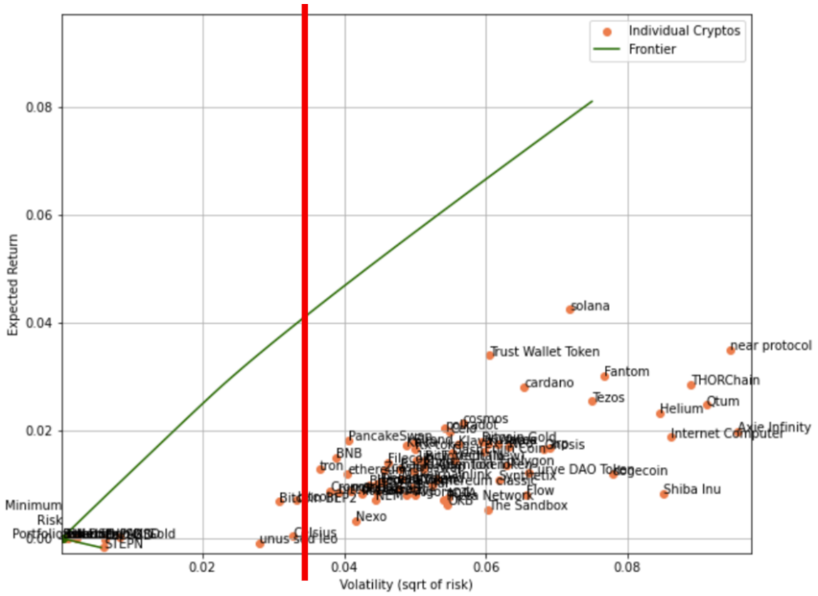
$3,666 is calculated based on the predicted price from the ARIMA model. If we actually purchase cryptos according to the optimal investment portfolio, the actual expected profit is $18,088. We can actually make profits using the ARIMA model.

6.2. Long-term Optimization

With a budget of $100, the optimal value (minimum risk) is 0.00026. The following picture shows the corresponding optimal investment plan.



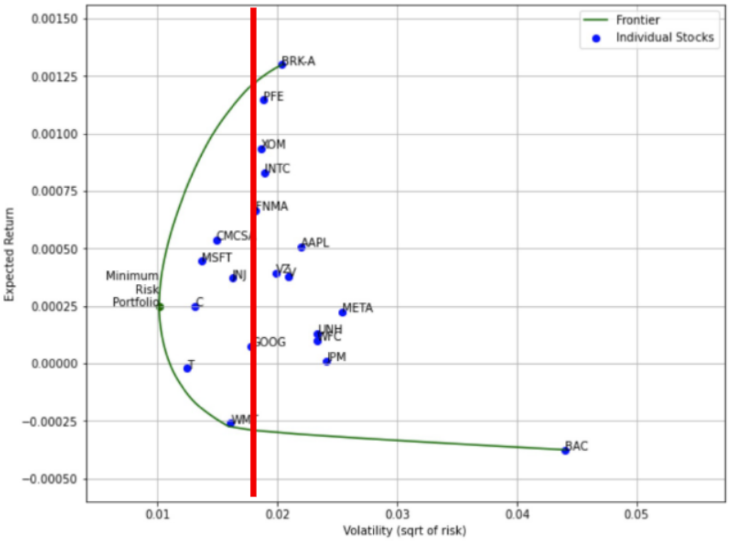
```
Binance USD should buy:  30.0
dai should buy:  9.0
Pax Dollar should buy:  1.0
tether should buy:  29.0
TrueUSD should buy:  9.000055083867107
usd coin should buy:  22.0
```



Besides, we also plotted the frontier of expected return and volatility. As demonstrated in the graph, the frontier is basically a straight line, which basically means that the more risks an investor can take, the more rewards an investor can get.

This straight frontier is apparently different from the curved frontier for stocks generated in the class. We can draw a vertical line for each of the two frontiers. For the stock frontier, it is possible that multiple stocks with different rewards have the same risk. Therefore, for a reasonable investor and what is taken into consideration for optimization, the stock with higher rewards is desired. However, such a thing does not happen in the crypto frontier.

The reason for the obvious difference between crypto frontier and the stock frontier is probably due to the fact that the risk of investing in cryptos is much higher than in stocks. In order to earn decent returns from cryptos, taking risks is inevitable. Free lunch like what happens in the stock market is impossible for the crypto market.



7. Conclusion and Discussion

Before making the final conclusion, we want to discuss how investing in Crypto is different from investing in any other assets such as stocks. The reason why we specifically choose to invest in Cryptocurrency is that Cryptocurrency is more volatile than any other asset. When we are making short-term investment predictions and optimization, we are only predicting one day in advance of the market. In traditional markets such as the stock market, it is much less volatile, typically within a range of 5% of fluctuation except on the earning announcement day. However, Cryptocurrencies markets are far more volatile than stock markets, where it is typical to see a 10% change in one day. Both our Machine Learning and ARIMA prediction models are prone to have more volatile data where the change is more obvious for observation and significant for training.

Move to the conclusion, for short-term optimization and a total budget is $100,000 on 8/28/2021, the optimal value without prediction is $31,386 (31.4% up in 1 day), the optimal value with machine learning prediction is $18,088 (18.1% up in 1 day), and the optimal value with ARIMA prediction is $2,741 (2.74% up in 1 day).

According to the optimal values, on 8/28/2021, both the machine learning model and time series model can generate profit. On that day, Machine learning prediction can generate more profit than ARIMA prediction.

For long-term optimization and a total budget is $100 from July 28th, 2021 to August 28th, 2021, the optimal value is 0.00026.

8. Business Impacts

Based on the analytics results and this project experience of providing an optimal portfolio suggestion, we came up with the following business impacts for investors and portfolio managers.

As an investor, profitability or losing less money is what they all ultimately care about. Our model can make profits (or at least did not lose money). Being able to rely on models to forecast prices in the future is preferable in the investment decision-making process than “guessing” what the price will be only by looking at the market fundamentals and market sentiments.

As a portfolio manager, optimizing the performance based on a specific objective (maximizing profit in the short-term or minimizing risk in the short-term) can satisfy the client’s needs and is able to find the optimal portfolio given the information and knowledge we have. The better the model, the more trust from our clients.

9. Potential Future Work

This project is a good opportunity for our group to work on cryptocurrency data and the related financial field. In the future, we anticipate more work to be done. Improvements could be made in the following directions. First, include more features on macroeconomic aspects (GDP growth, interest rates, etc). Second, try out time series methods, and neural networks such as LSTM and compare the performance. Third, if computational power permitted, hyperparameter tuning for each of the models for each of the crypto to find the best parameters. Fourth, if computational power is permitted, increase the long-term optimization budget. Last, for the time-series model, test on more days to see if model performance variates.

References

[1] Top 100 Cryptocurrencies Historical Data
https://www.kaggle.com/datasets/kaushiksuresh147/top-10-cryptocurrencies-historical-dataset
[2] "Stock Prediction with ML: Model Evaluation — The Alpha Scientist." Accessed December 1, 2022. https://alphascientist.com/model_evaluation.
[3] Saha, Raj. "Forecasting the Stock Market Using ARIMA in Python." *Medium* (blog), April 27, 2022.
https://medium.com/@raj.saha3382/forecasting-of-stock-market-using-arima-in-python-cd4fe76fc58a.
[4] Zhuang, Cedric. "Stockstats: DataFrame with Inline Stock Statistics Support." OS Independent, Python. Accessed December 1, 2022. https://github.com/jealous/stockstats.
"Portfolio.Ipynb: Decision Analytics for Business and Policy." Accessed December 1, 2022.
https://canvas.cmu.edu/courses/30486/files/8510975?module_item_id=5273179.

Appendix

When choosing the machine learning approach to predict the cryptos' close prices, we tried Random Forest Regression, LASSO, Support Vector Machine Regression, Stochastic Gradient Descent, Gradient Boosting, and XGboosting Regression. The program will choose the best model with the lowest accuracy automatically.

When choosing the objective function for long-term optimization, we first considered maximizing the expected revenue of the investment portfolio while using risk as a constraint. Risk can be represented by standard deviation or variance. However, during actual implementation, we found out that we could not use risk as a constraint, since this will make the program choose none of the cryptos. Therefore, we shifted to the methods the professor mentioned in class. We eventually decided to minimize the portfolio risk for long-term optimization and produce a risk-return frontier graph.