The Occidental College Computer Science Comprehensive Project Proposal: Predicting Cryptocurrency Prices for Index Fund Rebalancing

Odelia Putterman
putterman@oxy.edu
Occidental College

Abstract

This paper is the final paper outlining my Occidental College Computer Science Comprehensive Project (COMPs): *Predicting Cryptocurrency Prices for Index Fund Rebalancing*. This paper begins by introducing this project's problem context. Next, it address the technical aspects of realizing this work. This paper also addresses lingering ethical concerns. Replication instructions and an overview of the code developed in this work for future users to engage with this project have been included as GitHub README documents in addition.

1 Problem Context

Investing is a means by which people can grow their money exponentially, but it also holds the potential for significant loss, especially in the context of day trading.

1.1 Index Funds: Explained

As it stands, it's difficult, if not entirely impossible, to predict the exact performance of individual stocks. Accordingly, people invest in index funds: collections of stocks which follow some set of rules for construction. Index funds mirror the whole market rather than a particular holding, with funds mutually pooled (i.e. the stock holdings shared across many holders). Money that goes into an index fund is invested in all the stocks the fund holds, growing the diversification of the fund and minimizing its dependence on individual stock performance. Further, index funds are periodically re-balanced to offset losses and risk, and companies are sold once they leave these predefined rules or parameters discussed.

1.2 Cryptocurrencies: Unique Opportunity and Approach

Index funds are generally constructed using fundamental measures of a company (aka fundamentals), such as cash flow, sales, and dividends. However, cryptocurrencies have none of these intrinsic metrics. The cryptocurrencies themselves are a currency rather than a materialized product, so there are no sales or dividends to be found; their only true measure is price. Accordingly, any approach to creating and maintaining a cryptocurrency index fund must be revolutionary in nature.

Cryptocurrencies are incredibly volatile, making rebalancing especially important. Their instability also offers a unique opportunity for leveraging their turning points to optimize gains if high-accuracy price prediction can be achieved. If we could predict cryptocurrency prices in advance, we could leverage this knowledge to maximize gains while keeping a sense of security with the breadth of the fund.

Typically, index funds are re-balanced on a consistent, preset schedule, such as the third Friday at the end of each calendar quarterly. But, what if we could predict cryptocurrency prices to decide how and when to re-balance such a cryptocurrency fund instead? If this succeeded, the results would be revolutionary in nature and a first-of-its-kind for maintaining a balanced, high-return cryptocurrency fund.

People have previously attempted to predict the behavior of cryptocurrencies with some success, as discussed below in Prior Work. I aim to follow in their steps with a slightly different approach and following implementation.

2 Technical Background

Here, I introduce the technical background required to fully comprehend the methods this project undertakes.

2.1 What Are Cryptocurrencies?

This project involves cryptocurrencies, a relatively new phenomena stemming from advances and uses of blockchain technologies. Cryptocurrencies were first created for their supposed safety and security arising from their basis in blockchain technology. According to IBM (2022) [12], "Blockchain is a shared, immutable ledger that facilitates the process of recording transactions and tracking assets in a business network". Each transaction is recorded as

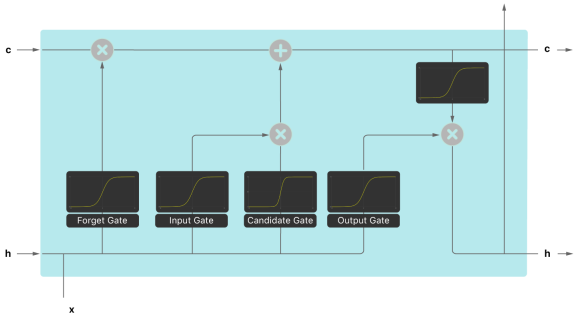


Figure 1: Sketch of the four gates composing LSTMs and their connections. From Apple ()[2].

a "block" of data, connected to the blocks before and after it in an irreversible chain (aka a blockchain).

2.2 Sentiment Data and Sentiment Analysis

Sentiment data is essentially the same as it sounds: data which signifies sentiment. Sentiment analysis, similarly, is the use of computational algorithms to study subjective information (i.e. sentiment). It is particularly important for understanding the social sentiment of a given thing [8]. In the context of this project, the sentiment data is data which can be analyzed to search for crowd feelings on cryptocurrencies, while the sentiment analysis is the analysis of this data to learn attitudes towards cryptocurrencies.

2.3 Long Short-Term Neural Networks

Long short-term neural networks (LSTMs) are a type of neural network or, more specifically, recurrent neural network (RNN), capable of long short-term predictions.

2.3.1 Recurrent Neural Networks

RNNs are a type of neural network capable of learning order dependence, which is key in sequence prediction problems. They differ from traditional feedforward neural networks, or multilayer perceptrons (MLPs), in their inclusion of feedback connections. To be an RNN, a system must:

1. Be able to store information for an arbitrary period
2. Be resistant to noise (i.e. random inputs, outliers, etc.).
3. Have trainable system parameters.

In RNNs, the *context* is learned, meaning an RNN contains cycles which feed the network outputs, or activations, from a prior time step as input to the network for future time steps. However, they fail to "remember" things from far back – RNNs are susceptible to backpropagated error. These neural networks have hidden layers which can decay or blow up when circling around a feedback connection [3]. Essentially, layers in RNNs with small enough gradient updates stop learning, usually in relatively early-on layers, causing short-term memory, or what is known as the *vanishing gradient problem*, while layers with large enough gradients exponentially grow towards infinity, causing a blow-up problem and an unrealistic expectation for reliance on unimportant, or not as important as is implied, information [18].

2.3.2 LSTM Motivation

LSTMs are a subset of recurrent neural networks (RNNs) similar, in most respects, to RNNs. They were first introduced in 1997 by Hochreiter and Shmidhuber [17]. LSTMs differ from RNNs in their ability to solve the vanishing gradients and exploding gradients problems. Like RNNs, they learn order dependence in sequence prediction problems, only they are far more effective at retaining both long-term and short-term memory. They can keep information about the past for a unfixed period, to be decided by the input data weights as feedback connections are formed throughout the neural network operations; they use contextual information flexibly. LSTMs can traverse longer time steps to store important information with a constant error flow, avoiding the RNN problem of exponential blowing-up and decay [3]. These feedback connections, or loops, allow for the persistence of information in traditional RNNs and, correspondingly, LSTMs. RNNs and LSTMs can be thought of as multiple copies of the same network, each passing information to the succeeding network. When the gap in knowledge from the prior input information to the next is relatively small, an RNN may succeed. But substantial research shows LSTMs far outperform typical RNNs when more context is needed [17].

2.4 Index Fund Rebalancing

Index fund rebalancing is the process of selling and buying stocks to balance the distribution across an index fund. There are many ways to re-balance an index fund. The usual choice is **calendar rebalancing**. In calendar rebalancing, the index fund holdings are reviewed on a preset calendar basis and adjusted to their original form. The actual rebalancing is done by examining issues such as transaction costs and drifts. Another type of rebalancing, later discussed, is **percentage-of-portfolio rebalancing**. According to Pinkasovitch (2022) [19], percentage-of-portfolio rebalancing is a "rebalancing schedule focused on the allowable percentage composition of an asset in a portfolio. Every asset class, or individual security, is given a target weight and a corresponding tolerance range." There are many other types of rebalancing algorithms, but these are the two most important rebalancing algorithms to understand for the scope of this project.

3 Prior Work

LSTMs are one of the most widely employed methods for sequential data problems and time-series forecasting. In this section, we will focus on prior work in using LSTMs for stock price prediction and examine existing index fund rebalancing algorithms.

3.1 Sentiment Analysis

According to efficient market hypothesis, stock prices are driven by news rather than past and present prices – they change based on news of changing fundamentals and human psychology (investor reactions to this news) [4]. This suggests our input data to an LSTM for stock price predictions should include fundamentals and sentiment analysis so we can model prices based on changes in news and sentiment which gauge crowd feelings towards a given stock or industry.

Shah, Isah & Zulkernine (2019) [21] recently employed fundamental analysis for stock price prediction, analyzing company profiles, industry, politics, the economy, and social media according to Ching-Ru Ko (2021) [4]. However, as Ching-Ru Ko (2021) [4] mention, this type of fundamental data is usually rather abstract or unstructured, making it hard to fit into a form LSTMs can use. Instead, Ching-Ru Ko (2021) [4] suggest using NLP methods to distill relevant information. Sentiment analysis is a strong candidate currently used to analyze text for emotional sentiments and reactions to "understand" human feeling. From this "understanding", we can examine how a stock may perform based on human feelings towards this stock. If, for example, negative sentiment about bitcoin is spreading, this may signal to us a coming drop in the bitcoin stock price. As such, sentiment analysis of crowd psychology can monitor public opinion [4].

3.2 RNNs and LSTMs

RNNs, particularly their LSTM derivative, have been shown to outperform traditionally used models, such as support vector machines (SVMs), in accurately predicting values for time-series problems [4]. By feeding a combination of fundamental and sentiment analysis to LSTMs to analyze, researchers have made promising strides towards success in predicting stock prices.

Ching-Ru Ko (2021) [4] used a long short-term memory neural network with text sentiments and historical stock transactions to predict stock prices similar to my approach for this project. The authors argued in the paper's introduction for the importance of aggressive investing strategies and, hence, the usefulness of accurate stock trend forecasting. They used fundamental analysis and technical analysis to forecast stock prices in addition to an LSTM, which was applied to this forecasting. Using text sentiment from blog articles, social media, online forum posts, and product reviews, the authors aimed to understand online public opinions. They used an existing NLP algorithm (BERT) to classify the texts into sentiment categories by counting the emotional words that appeared in a given sentence, after which they calculated a score for each emotional word, and, with these individual scores, outputted the condition of the sentence. Further, this article addressed pre-processing data to get it into a form where it can be passed to the related ML algorithm, which was very useful for me in learning how to apply my project's own product algorithm.

3.3 Other Models

Helder Sebastio (2021) [9] examined the predictability of three major crypto contenders: Bitcoin, Ethereum, and Litecoin. They focused on times of extreme upheaval and used linear models, random forests, and support vector machines for their predictive work. Of the 18 models they built, 13 models had a success rate over 50%, affirming the promise of ML algorithms for cryptocurrency stock price prediction. In their introduction, they explained the proposal of sentiment analysis for stock price prediction, which is what motivated me to choose the same for my project.

3.4 Trading Algorithms

There are many possible trading algorithms to use for index fund rebalancing.

3.4.1 Mean-Variance Optimization

One such trading algorithm is mean-variance optimization. Lewinson (2019) [15] implores this algorithm, implementing Markowitz's portfolio optimization (MPT) and the Modern Portfolio Theory in python as trading strategies with the *zipline* library. The general goal of MPT is to maximize returns while minimizing risk. This method ignores transaction costs, which, for an index fund, need to be considered, and disallows short-selling.

3.4.2 Constant Rebalanced Portfolios

Constant Rebalanced Portfolios are another trading algorithm, achieved by *Constant Rebalanced Portfolios - some simulations with numpy* [6]. The idea explained in this article is:

Say we have three stocks called A, B, and C we believe will perform equally well. So, we split our money equally amongst the three stocks, making our holdings

$\{\frac{1}{3}A, \frac{1}{3}B, \frac{1}{3}C\}$. Let's say stock A does significantly better than the rest, making our holdings: $\{\frac{1}{2}A, \frac{1}{4}B, \frac{1}{4}C\}$. The idea here is: we believe this disproportionate return was just chance, and so we sell off some of stock A and re-balance to our original $\{\frac{1}{3}A, \frac{1}{3}B, \frac{1}{3}C\}$.

The use of this trading algorithm has historically allowed for cutting losses and maximizing gains dynamically.

4 Methods

This project predicts future cryptocurrency prices using an LSTM neural network based on historical price and sentiment analysis data inputs. Because, as previously mentioned, cryptocurrencies do not have fundamentals, we must evaluate and try to predict their behavior using other metrics for input data. Here, I motivate the use of sentiment analysis. (Note, my language of choice for this project was Python for its abundant scraping and machine learning libraries/tools in addition to its simplicity of use. Accordingly, all libraries discussed are Python libraries.)

4.1 Sentiment Data

Originally, I thought to use both sentiment analysis and current events data for cryptocurrency price prediction. However, data collection was a far more difficult endeavor than I had anticipated. After realizing the complexity involved in data collection alone, I chose to source sentiment data exclusively. Since sentiment ideally mirrors current events, any major disarray influencing cryptocurrency prices should likely be expressed in sentiment towards the currency, further justifying the use of sentiment data alone.

4.1.1 News Headlines

Citing all news data found has the potential to overwhelm the size of the data sets and slow the computational step of sentiment analysis, as it incurs more compute. Accordingly, I narrowed the sentiment data chosen for this project to news headlines and social media posts. That said, moving forward, sentiment data would ideally be sourced from news headlines alone.

News headlines typically encompass the summary of the respective article's contents. So, I expect such headlines to convey a condensed version of their respective article's overall sentiment. Such sentiment, in this case, should convey feelings towards or about cryptocurrencies. Since we are interested in feelings towards cryptocurrencies for this price prediction, we chose to further narrow our sentiment search to news headlines containing cryptocurrency names in them. Moreover, I sought to find data specific to each coin so I could predict trends in prices for each coin individually rather than as a whole, as their individual price predictions are crucial for successful index fund rebalancing.

Disclosure It was difficult to source data, so I narrowed my search to sourcing data for Bitcoin, Ethereum, and Solana alone. As this project is simply a proof of concept, having data for three coins only sufficed. However, for a fully functional rebalancing pipeline, we would need to expand the index fund to include a broader range of coins. In future iterations of this project, data should be sourced to include this broader range of coins.

Specifics News headlines were collected for Bitcoin, while social media postings were collected for Ethereum and Solana. For Ethereum, news headlines were difficult to find, but tweets were readily available. Tweets are relatively short, so this did not pose any noticeable harm to the computational cost of sentiment analysis. Similarly, for Solana, news headlines were difficult to find, but a data set of comments from the Solana Subreddit on Twitter was available. Comments have size restrictions, which also suggests these do not pose any noticeable harm to the computational cost of sentiment analysis.

4.2 Sourcing Data

To amass data sets for training my price prediction models, I searched for existing data sets and examined scraping data from scratch and using data scraping APIs. For this model, I needed both historical cryptocurrency prices and news headlines around these respective cryptocurrency prices with corresponding dates.

4.2.1 Existing Datasets

Existing data sets for historical cryptocurrency prices appeared in abundance. Conversely, existing historical news headlines data sets were more difficult to source. Further, upon finding existing news headlines data sets, often the dates were limited to a few days, which is unhelpful for training a model for long-term prediction. Searching through Kaggle, I managed to find a few useful data sets. I pulled the following data sets from Kaggle for inclusion in my final data sets for sentiment analysis:

Price Datasets

- Bitcoin Historical Data: Historical Bitcoin prices from January 1st, 2012 to March 31st, 2021.
- Ethereum Crypto Price: Historical Ethereum prices from November 11th, 2017 to May 24th, 2022.

- Ethereum Cryptocurrency Historical Data 09.2022: Historical Ethereum prices from July 30th, 2019 to September 1st, 2022.
- Solana Data: Historical Solana prices from April 10th, 2020 to March 25th, 2022.

News Datasets

- Bitcoin-News Dataset: Bitcoin news data from the past year.
- Bitcoin Price Prediction: Bitcoin news data from July 1st, 2015 to June 12th, 2021.
- Ethereum Tweets: Ethereum tweets starting from January 1st, 2021.
- r/solana Comments 2020.01-2022.04: Solana comments on the Solana Subreddit from January 1st, 2019 to April 29th, 2022.

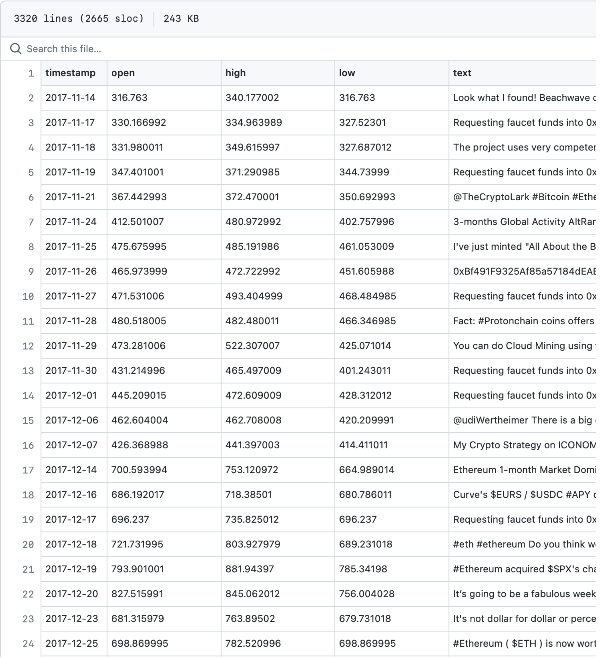
4.2.2 Scraping Data

In my search for existing data sets, I found a lot of price data but limited news data. I sought to supplement the news data found with scraped data to, in turn, optimize the price prediction model training. Accordingly, I attempted to use existing scraping APIs to collect more data. I tried using the APIs:

- News API; and
- Crypto News API.

This was largely unsuccessful because of these APIs' limits on querying; the free version of News API limits querying to search articles up to a month old, while the Crypto News API requires a paid subscription after a 14-day free trial period. I did not want to pay any subscription fees, so after the 14-day free trial period for Crypto News API, I decided not to move forward attempting to collect news data with this tool. Similarly, I did not want to pay for an upgraded version of News API, so I was limited to this 30-day look-back period. This made the News API largely useless for collecting data for training these price prediction models, as these models require data on a much larger scale. But, this News API is very useful for the end-to-end pipeline of price prediction and rebalancing, as my final price prediction model was created to require 30 days of historical data, as allotted by the API, to predict the next 15 days of prices.

I also attempted to create a scraper from scratch to scrape news data. But, this, too, had poor results for collecting historical data due to the customization required for each news site to handle differing HTML architectures and the work-around necessary for dynamic loading to get historical results.



3320 lines (2665 sloc) | 243 KB

timestamp	open	high	low	text
2017-11-14	316.763	340.177002	316.763	Look what I found! Beachwave c
2017-11-17	330.166992	334.963989	327.52301	Requesting faucet funds into 0x
2017-11-18	331.980011	349.615997	327.687012	The project uses very competer
2017-11-19	347.401001	371.290985	344.73999	Requesting faucet funds into 0x
2017-11-21	367.442993	372.470001	350.692993	@TheCryptoLark #Bitcoin #Ethe
2017-11-24	412.501007	480.972992	402.757996	3-months Global Activity AltRar
2017-11-25	475.675995	485.191986	461.053009	I've just minted "All About the B
2017-11-26	465.973999	472.722992	451.605988	0xBf491F9325Af85a57184dEAE
2017-11-27	471.531006	493.404999	468.484985	Requesting faucet funds into 0x
2017-11-28	480.518005	482.480011	466.346985	Fact: #Protonchain coins offers
2017-11-29	473.281006	522.307007	425.071014	You can do Cloud Mining using
2017-11-30	431.214996	465.497009	401.243011	Requesting faucet funds into 0x
2017-12-01	445.209015	472.609009	428.312012	Requesting faucet funds into 0x
2017-12-06	462.604004	462.708008	420.209991	@udiWertheimer There is a big
2017-12-07	426.368988	441.397003	414.411011	My Crypto Strategy on ICONOM
2017-12-14	700.593994	753.120972	664.989014	Ethereum 1-month Market Domi
2017-12-16	686.192017	718.38501	680.786011	Curve's $EURS / $USDC #APY c
2017-12-17	696.237	735.825012	696.237	Requesting faucet funds into 0x
2017-12-18	721.731995	803.927979	689.231018	#eth #ethereum Do you think w
2017-12-19	793.901001	881.94397	785.34198	#Ethereum acquired $SPX's cha
2017-12-20	827.515991	845.062012	756.004028	It's going to be a fabulous week
2017-12-23	681.315979	763.89502	679.731018	It's not dollar for dollar or perce
2017-12-25	698.869995	782.520996	698.869995	#Ethereum ($ETH) is now wort

Figure 2: Pictured here is a sample of the cleaned Ethereum data set, showing the kept columns: timestamp, open, high, low, and text. Note, the text column was cut short for visualization purposes.

4.3 Cleaning Data

The data sets collected from Kaggle were cleaned and processed so they could be forwarded to the sentiment analyzer. This cleaning required merging the price and news data sets on the timestamp column, with all the news headlines combined and stored as a single column grouped by this date. All the data sets were additionally formatted according to a preset paradigm, with all the column headers converted to lowercase and the following columns being kept: timestamp, open, high, low, and text. In these columns, "open", "high", and "low" correspond to the open, high, and low prices, respectively, for the relevant coin, "timestamp" corresponds to the date associated with that row of data in the 'yyyy-MM-dd' format, and "text" is the combined headlines from that day. A sample from one of these cleaned data sets is pictured in Figure 2.

4.4 Sentiment Analysis

After pre-processing the data, sentiment analysis was performed. The built-out sentiment analyzer ingested these output files from the cleaning step and calculated the following sentiment scores for each row of the textual news column:

- subjectivity

- polarity
- compound
- negative
- neutral
- positive

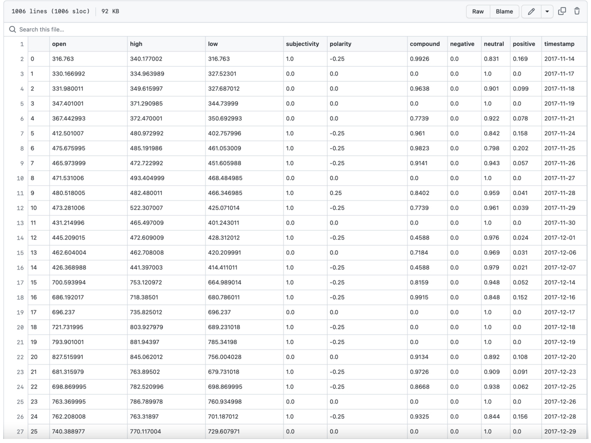
According to Singh (2020) [22], compound is “the sum of positive, negative & neutral scores which is then normalized between -1 (most extreme negative) and +1 (most extreme positive)”. In other words, as Hutto and Gilbert (2014) [11] put it, compound is a “normalized, weighted composite score”. Meanwhile, subjectivity, polarity, negative, neutral, and positive are measures of their exact meaning. A range of scores were computed to expand the span of quantitative analysis and capture a breadth of features.

This sentiment analysis was done using **VADER Sentiment Analysis**, the tool created by Hutto and Gilbert (2014) [11], following the YouTube tutorial Stock Market Sentiment Analysis Using Python & Machine Learning, with refactoring added to organize the code more logically and ingest and output data in accordance with the other steps of this pipeline. I chose to use this VADER Sentiment Analysis tool for its extensive documentation and its specific tuning to sentiment expressed in social media. Further, it has been used by over 4.4 thousand GitHub users, which speaks to its ease of use and solidity [11]. Because of this tuning, I believed the tool would be well-equipped to capture the features present in the training set, which was collected across news outlets and social media sites (Twitter and Reddit).

These sentiment scores were then appended to the originally ingested data set as new columns and the previous text column was removed, as only the sentiment calculations are needed as input to the price prediction algorithm and excess columns may add unnecessary computational slowdown. The resulting data set was outputted as a new file, maintaining the input data sets intact. This outputted file was the input to the LSTM price prediction model.

4.5 LSTM Price Predictions

A long short-term neural network (LSTM) was used to perform price predictions for the cryptocurrencies with available data. As mentioned in the sections 3.2 and 2.3, LSTMs are a prime choice for time-series price predictions because they factor for both long-term and short-term trends in their analysis while concurrently solving the vanishing gradient problem and long-term dependence issues to outperform tradition recurrent neural networks (RNNs) in such time-series problems. Since LSTMs typically outperform RNNs in time-series predictions, it was clear to proceed with these neural networks for addressing cryptocurrency price predictions.



	open	high	low	subjectivity	polarity	compound	negative	neutral	positive	timestamp
0	316.763	340.177002	316.763	1.0	-0.25	0.9926	0.0	0.831	0.169	2017-11-14
1	330.166992	334.963989	327.52301	0.0	0.0	0.0	0.0	1.0	0.0	2017-11-17
2	331.980011	349.615997	327.687012	0.0	0.0	0.9638	0.0	0.901	0.099	2017-11-18
3	347.401001	371.290985	344.73999	0.0	0.0	0.0	0.0	1.0	0.0	2017-11-19
4	367.442993	372.470001	350.692993	0.0	0.0	0.7739	0.0	0.922	0.078	2017-11-21
5	412.501007	480.972992	402.757996	1.0	-0.25	0.961	0.0	0.842	0.158	2017-11-24
6	475.675995	485.191986	461.053009	1.0	-0.25	0.9823	0.0	0.798	0.202	2017-11-25
7	465.973999	472.722992	451.605988	1.0	-0.25	0.9141	0.0	0.943	0.057	2017-11-26
8	471.531006	493.404999	468.484985	0.0	0.0	0.0	0.0	1.0	0.0	2017-11-27
9	480.518005	482.480011	466.346985	1.0	0.25	0.8402	0.0	0.959	0.041	2017-11-28
10	473.281006	522.307007	425.071014	1.0	-0.25	0.7739	0.0	0.961	0.039	2017-11-29
11	431.214996	465.497009	401.243011	0.0	0.0	0.0	0.0	1.0	0.0	2017-11-30
12	445.209015	472.609009	428.312012	1.0	-0.25	0.4588	0.0	0.976	0.024	2017-12-01
13	462.604004	462.708008	420.209991	0.0	0.0	0.7184	0.0	0.969	0.031	2017-12-06
14	426.368988	441.397003	414.411011	1.0	-0.25	0.4588	0.0	0.979	0.021	2017-12-07
15	700.593994	753.120972	664.989014	1.0	-0.25	0.8159	0.0	0.948	0.052	2017-12-14
16	686.192017	718.38501	680.786011	1.0	-0.25	0.9915	0.0	0.848	0.152	2017-12-16
17	696.237	735.825012	696.237	0.0	0.0	0.0	0.0	1.0	0.0	2017-12-17
18	721.731995	803.927979	689.231018	1.0	-0.25	0.0	0.0	1.0	0.0	2017-12-18
19	793.901001	881.94397	785.34198	1.0	-0.25	0.0	0.0	1.0	0.0	2017-12-19
20	827.515991	845.062012	756.004028	0.0	0.0	0.9134	0.0	0.892	0.108	2017-12-20
21	681.315979	763.89502	679.731018	1.0	-0.25	0.9726	0.0	0.909	0.091	2017-12-23
22	698.869995	782.520996	698.869995	1.0	-0.25	0.8668	0.0	0.938	0.062	2017-12-25
23	763.369995	786.789978	760.934998	0.0	0.0	0.0	0.0	1.0	0.0	2017-12-26
24	762.208008	763.31897	701.187012	1.0	-0.25	0.9325	0.0	0.844	0.156	2017-12-28
25	740.388977	770.117004	729.607971	0.0	0.0	0.0	0.0	1.0	0.0	2017-12-29

Figure 3: Pictured here is a sample of the output of the sentiment analysis process for Ethereum with the kept columns: timestamp, open, high, low, subjectivity, polarity, compound, negative, neutral, positive, timestamp.

There are many flavors of LSTMs. Since I wanted this LSTM to consider an array of sentiment scores in addition to historical prices for this time-series problem, I needed to use an LSTM that accepts multiple features (aka a multivariate LSTM). For this prime multivariate LSTM model, I followed the tutorial Predicting the price of Bitcoin with multivariate PyTorch LSTMs. In this tutorial, the author used the metrics “high”, “low”, and “close”, and “volume” grouped by date as the input features to predict future “open” prices. I kept my output feature the “open” price as well. Similarly, I used an Adam optimizer and the mean squared error for the loss, which measures how well neural network models the training and is aimed to be minimized [26]. The mean squared error is a measure of the difference between the actual and predicted values, or, rather, prices in this case – we seek to minimize this value so that our predicted price are as close as possible to the actual prices (see Price-Prediction Evaluation for further explanation). However, since my input features were different, I had to refactor the code to accept my additional and differing input features. I also had to refactor the code to accommodate shorter input and output series. Additionally, I refactored the code for clarity, building out a class from this base code, grouping snippets into logical functions, and commenting all the code with complete method stubs.

4.6 Index Fund Rebalancing Algorithm

The index fund re-balancing algorithm used in the available pipeline was a slight variation on that created in Teo (2021) [24], which revolves through a preset number of stocks based on their performance across a fixed period of time. The algorithm implemented here was nearly identical, with variations to support the shorter re-balancing period allotted by the future price predictions. This algorithm, however, was implemented to be used in comparison against a novel trading algorithm, which is still being hashed and included in theory for future work. This algorithm (currently in production) aims to utilize the predicted prices and their respective accuracies, depending on the coin, in conjunction with the predicted values and our aim to maximize profits in the re-balancing business logic, at which point the varied implementation of the algorithm from Teo (2021) [24] can be used solely for evaluation purposes.

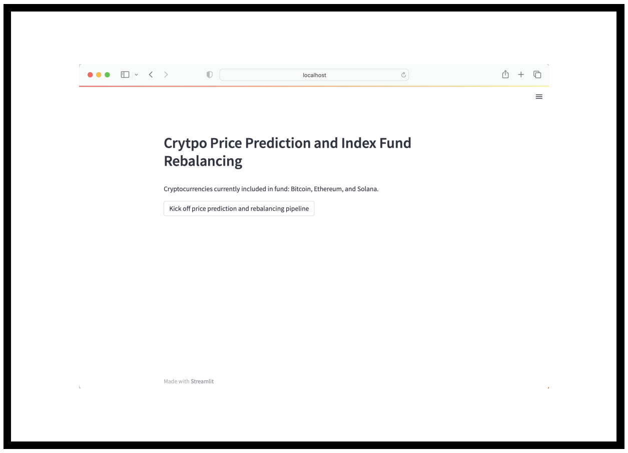
4.7 End-to-End Pipeline

To fully realize my project's goals, I built out an end-to-end pipeline. This pipeline follows:

1. For each coin:
   (a) Collect its price across the past 30 days.
   (b) Collect related news from the past 30 days.
   (c) Combine and format the price and news data into a cohesive data set compatible for input to the sentiment analyzer.
   (d) Pass the formatted combined data set to the sentiment analyzer.
   (e) Perform sentiment analysis and save the output.
   (f) Pass the output of the sentiment analysis process to the trained LSTM model for price predictions.
   (g) Retrieve predicted prices and save the predicted prices.
2. Combine and format the individual coins' predicted prices into a cohesive data set compatible for input to the re-balancing algorithm.
3. Re-balance the index fund according to these predicted prices.

Collecting Previous Prices To collect each coin's price from across the past 30 days, I used the **yfinance** library [27], specifying the end date for downloading the prices as today and the start date as today minus 30 days (the day 30 days ago). The dataframe returned from the yfinance query is altered to change the column headers to follow the standard naming convention employed across this project. This information is saved for each coin as a separate csv file.

Collecting Corresponding Related News Related news for the past 30 days is collected for each coin using the **News API**. The API is used to query news sources for all headlines containing the coin's name in them. For each coin this news file is saved as a separate csv file.



Crytpo Price Prediction and Index Fund Rebalancing

Cryptocurrencies currently included in fund: Bitcoin, Ethereum, and Solana.

Figure 4: Streamlit app home page for this project's end-to-end pipeline.

Combining Price and News Data The price and news data are then combined, with the textual news column grouped by timestamp so there is only one row per date. If no news was found for a given date, the text "neutral" is stored as the value corresponding to the text column for that row. This is saved as a single csv file.

Sentiment Analysis The sentiment analyzer reads in the the clean, combined price and news data frame and performs sentiment analysis on it. The result of this sentiment analysis is saved as a new, separate csv file for each coin.

LSTM Price Predictions The LSTM model is called on the sentiment analysis output file to predict future prices for each coin 15 days into the future. The predicted prices are saved as a new, separate csv file for each coin.

Combining Predicted Prices After all the future prices have been predicted for each coin, the results are combined into a cohesive csv file and formatted according to the pattern required for the index fund re-balancing algorithm. This is saved as a single, cohesive csv file.

Rebalancing the Index Fund With the cohesive price predictions csv file, the re-balancing algorithm re-balances the index fund. The output of this is displayed to the user.

4.7.1 Streamlit App

For proof of concept, I used **streamlit** [1], an open-source app framework, to build out a very simple UI from where this end-to-end process can be kicked-off. The home page for this simple UI is pictured in Figure 4.

5 Evaluation Metrics

Were the rebalancing algorithm to work based on bad predictions, regardless of its ingenuity and logic, we would expect its results be poor. Similarly, if the price predictions are highly accurate but the rebalancing algorithm is defective, we may see poor overall results. Essentially, it may be impossible to discern where fault lies in poor results unless we evaluate these two components separately. As such, the evaluation of this project was split into two general categories: evaluation of the price predictions and evaluation of the rebalancing algorithm.

5.1 Price-Prediction Evaluation

To evaluate the multivariate LSTM model, I took two approaches:

1. To evaluate the sanity of the tutorial model's logic itself, I built out adjacent models.
2. To evaluate whether the sentiment input data affected the accuracy of the price predictions, I compared the sentiment data sets to data sets with constant sentiment scores.

To evaluate a model, Data Scientists typically split their data set into two parts: a training set and a testing set. The training set is used to train the model, while the testing set is used to evaluate the model's performance. The split between testing and training sets varies, but it was broadly kept between 70% to 95% training and 5% to 30% testing of the original data set for this project – in this project, the train/test split varied from coin to coin and model to model based on the amount of available data and the performance metrics from varying this split.

Using this testing data, I compared predicted results (from using the testing set's input data) against the corresponding actual prices (the testing set's output data). For machine learning models outputting binary decisions, evaluation typically consists of calculating a set of metrics such as precision and recall. In this case, the model's output is not a binary decision but a discrete value instead. Achieving perfect accuracy in the predicted prices is likely impossible, but we only need to capture the general features and trends of the price movements to have good results. So, we must measure the performance of these models with other means. I chose to measure the relative accuracy of each model by calculating the mean squared error (mse), given by

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{1}$$

where n is the number of data points, Y_i are the observed values and $\hat{Y}_i$ are the predicted values [16]. In our case, the number of data points n are the number of data points in the test set, Y_i are the actual prices, and $\hat{Y}_i$ are the model's corresponding predicted prices. The mean squared error looks at the average difference between the actual prices and the predicted prices, giving a sound measurement for calculating performance. With this metric of performance, we seek to minimize the mean squared error – a smaller mean squared error value corresponds to better predictions.

5.1.1 Adjacent Models

The adjacent models for the first point of comparison were:

- a price-based LSTM model built with TensorFlow;
- a multivariate sentiment and price-based model with different logic built with TensorFlow; and
- a sentiment and price-based Linear Discriminant Analysis model.

These models were built to evaluate the soundness of the code for the sentiment and priced-based multivariate PyTorch LSTM based off the tutorial model. The price-based LSTM model built with TensorFlow was made to compare a TensorFlow model to a PyTorch model and to see if the sentiment inputs added much to the overall prediction accuracy. This was by far the most accurate model next to the prime model. The multivariate sentiment and price-based model built with different logic using TensorFlow was used to compare the prime model against a sample multivariate model to evaluate the logic of the code with a rather 1:1 comparison. The sentiment and price-based Linear Discriminant Analysis model was used to compare a non-LSTM model to an LSTM model to constrain whether the LSTM model was a sound choice.

The evaluation against these models generally employed the use of mean squared error as a means for measuring accuracy. The Linear Discriminant Analysis model was excepted from using the mean squared error method of evaluation since it outputs 0/1 values signifying only a predicted decrease/increase in price (0 = decrease, 1 = increase) – we cannot compare decreases/increases in price to actual price data points. Instead, for this model, I followed the typical evaluation for binary models and used the metrics: precision, recall, f1-score, support, accuracy, macro average, and weighted average. According to *Classification: Precision and Recall* [5], precision is a measure of the "proportion of positive identifications [that are] actually correct" and recall is a measure of the "proportion of actual positives... identified correctly". Similarly, f1-score is a value that combines precision and accuracy to give a composite score.

5.1.2 Constant Sentiment

To evaluate whether the sentiment data had any impact on the accuracy of the predicted prices, for each coin I took the cleaned data sets outputted from the sentiment analyzer and converted all the sentiment scores to the value ‘1’. I then compared the prime model’s performance after being trained on the actual sentiment files versus these constant sentiment files to see if the sentiment scores impacted the accuracy of the predicted prices.

5.2 Index Fund Rebalancing Evaluation

To evaluate the index fund, I took 15 days worth of past actual prices for the cryptocurrencies in the Bitwise 10 Crypto Index Units Beneficial Interest (BITW) fund and treated them as “predicted future prices”. This was done to eliminate the price predictions affecting the perceived validity of the re-balancing algorithms and, instead, evaluate the re-balancing algorithms assuming the price predictions were entirely accurate. Essentially, I evaluated the re-balancing algorithm under the assumption that the predicted prices inputted are entirely correct. I compared the temporary index fund’s returns, created by Teo (2021) [24], to the returns produced by BITW, using the metrics of compound annual growth (CAGR), volatility, the sharpe ratio, and the maximum drawdown with the implementations by Teo (2021) [24]. Once the pending index fund re-balancing algorithm is fully finished, I plan to evaluate its performance against both the simple re-balancing algorithm implemented here and the BITW fund results using the same evaluation metrics.

6 Results and Discussion

The section follows the split between the evaluation of the predicted prices and re-balancing algorithm introduced under Evaluation Metrics, with the results of each initially discussed separately. The joint results are then viewed in the context of this project’s overall goals.

6.1 Predicted Prices

The predicted prices were evaluated using the two methods above: alternative models and constant sentiment. The results of comparisons of the prime model against alternative models are first discussed.

6.1.1 Alternative Models Comparison

Price-Based TensorFlow LSTM Comparison The prime model was compared against a price-based LSTM model built with TensorFlow based on the tutorial by Valkov (2019) [25]. This TensorFlow model, which inputted historical prices alone in its prediction of future prices, was the most accurate model next to the prime model. The outputs of this can be viewed under the project folder located under *src.models.outputs.graphs* by viewing the related files:

- PriceLSTMModel_comparison_bitcoin.png
- PriceLSTMModel_comparison_ethereum.png
- PriceLSTMModel_comparison_solana.png

Note, this model had a higher input length, which increased the accuracy of the price predictions. With an increased input length, the prime model’s accuracy also increased. This model did not prove that the business logic driving the prime model was necessarily superior, but the comparison did suggest the respective logic driving each model is comparable.

Multivariate Sentiment and Price-Based Differing Logic TensorFlow Comparison The multivariate sentiment and price-based TensorFlow LSTM model (based on differing logic from *How can I create Multiple Input One Output LSTM Model with Keras?* [10]) was fairly unsuccessful; the trained model failed to capture the trends of price prediction. Comparing this model to the prime model concluded the prime model to be considerably more successful. As this was a rather 1:1 comparison seeing as they had the same input data, this further affirmed the validity of the logic behind this prime model’s code. The outputs of this can be viewed under the project folder located under *src.models.outputs.graphs* by viewing the related files:

- SentimentLSTMModel_comparison_bitcoin.png
- SentimentLSTMModel_comparison_ethereum.png
- SentimentLSTMModel_comparison_solana.png

Sentiment and Price-Based Linear Discriminant Analysis Comparison The results obtained from analyzing the Linear Discriminant Analysis, based off the model developed in this tutorial *Stock Market Sentiment Analysis Using Python & Machine Learning* [23], performance according with the outlined evaluation metrics were very poor. For Bitcoin, which had the largest data set by far, spanning the course of nearly seven years, the results were still fairly deplorable. These results can be pictured in the table Sentiment and Price-Based Linear Discriminant Analysis Comparison.

Linear Discriminant Results for Bitcoin				
	precision	recall	f-score	support
0	0.38	0.08	0.13	211
1	0.50	0.88	0.64	221
accuracy			0.49	432
macro avg	0.44	0.48	0.38	432
weighted avg	0.44	0.49	0.39	432

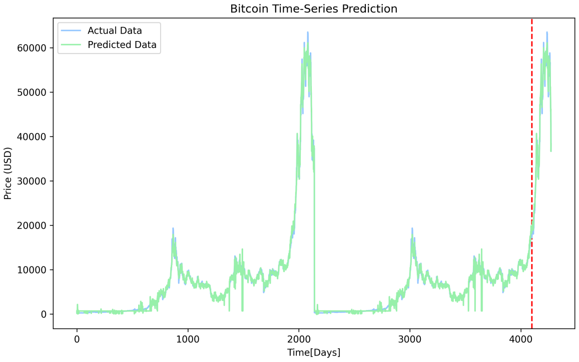


Figure 5: This image shows the prime model's time-series predictions against the actual prices from the same time.

The relative high accuracy of the prime LSTM model compared with this Linear Discriminant Analysis model promoted the choice to use an LSTM, affirming the approach backing price-prediction in this project.

6.1.2 Constant Sentiment Comparison

The comparison between the predicted prices outputted from the same prime model trained on the actual sentiment input data versus the constant sentiment input data showed that the model trained on the sentiment data was more accurate. The outputs of this constant sentiment evaluation can be viewed by comparing the project files located at *src.evaluation.outputs.graphs*, representing the constant sentiment outputs, with those under *src.models.outputs.graphs* beginning with *SentimentPriceLSTMModel*, which represent the outputs of the prime model.

The improved accuracy in using a model trained on sentiment data proves that the sentiment data is useful in predicting prices. This result suggests that sentiment is, indeed, a good indicator for cryptocurrency prices.

6.2 Rebalancing Algorithm

The final rebalancing algorithm has yet to be fully implemented, so the evaluation compared the varied implementation by Teo (2021) [24] against the performance of the BITW fund. Surprisingly, the algorithmic trading algorithm outperformed the BITW fund on the same stocks, suggesting this re-balancing algorithm, though simple, has promising results.

6.3 Collective Results

Overall, the price predictions generated by the prime model were fairly accurate. Despite the predictions never being exact, they captured the general trend of price movement. The results of this project are promising. This project was made as a proof of concept, succeeding in its goal to build a fully-functional end-to-end pipeline. Further, it showed promising initial results for cryptocurrency price prediction alone and in conjunction with re-balancing index funds, suggesting future versions of this work could have increased success in leveraging the volatility of these currencies and maximizing gains in investments.

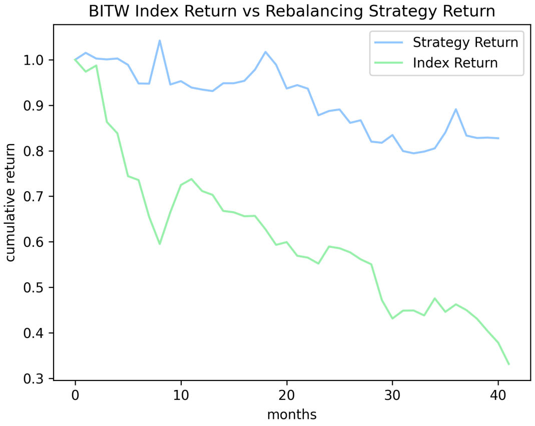


Figure 6: This image shows the algorithmic re-balancing algorithm's returns compared to the returns given by the BITW fund on the same coins during the same period. This comparison was developed by Teo (2021) [24].

7 Ethical Considerations

When considering the ethical considerations of predicting cryptocurrency prices for stock trading using machine learning, we must break the question into sub-tasks. Since this project advocates for the use of cryptocurrencies, we must consider the ethics of cryptocurrencies themselves. Further, we must look into the ethics of stock trading and the investor obligations that come into play when building an index fund, especially one built with machine learning.

7.1 Nature of Cryptocurrency

Due to the immutability of blockchain, and, hence, cryptocurrencies, the risk of tampering by bad actors is eliminated, creating a trustworthy network. Traditionally, information storage and retrieval systems were susceptible to cyberattacks. Moreover, without transparency on behalf of the record-management company or system, record tampering would be harder to catch and verify. The open-book nature of blockchain allows for widespread verification of all transactions. Coupled with the blockchain's immutable nature, with no user nor managing power having the ability to alter or delete a transaction, blockchain technologies, and, hence, cryptocurrencies, are extremely safe from a tampering standpoint [12]. This addresses transparency on behalf of the cryptocurrency operations, but it does not cover the ethical considerations with regard to the security of such assets.

A second consideration is the instability of where cryptocurrencies are hosted. One Washington Post Article, *Tracking stolen crypto is a booming business: How blockchain sleuths recover digital loot*, details the commonality of crypto heists. Hackers have previously hacked into cryptocurrency trading networks, getting away with massive sums. In August of 2021, Poly Network, a trading platform for popular cryptocurrencies, was hacked, with the hackers taking $610 million in crypto, which they quickly converted to a "stable coin". Luckily, the "stable coin" Tether worked with authorities to release the holdings to the rightful owners. Nonetheless, this heist broke the belief that cryptocurrency is impossible to trace (public ledgers do not retain account holder information) [7]. The anonymity of cryptocurrencies may be ethical, guarding the identity of its users or not, allowing for bad actors to use crypto for illicit activities, but it is certain that their online holdings make them susceptible to being stolen, further questioning the security of their nature.

Despite the secure theoretical ideas behind cryptocurrencies, their value is often in flux. Ethicists have previously questioned the ethics of cryptocurrencies themselves. Professor Tobey Karen Scharding shared her evaluation of bitcoin from an ethical perspective in an interview at Rutgers Business School:

> With bitcoin, I felt there were too many uncertainties... [The] evaluation was pretty negative because bitcoin didn't really have any way to secure its value, and so it wouldn't be able to fulfill the ethical purpose of currency on Fichte's account of stabilizing and securing these exchanges over generations. [20]

Professor Scharding refers here to the Fichte account, which explains that currency should allow people to live their lives with secure access to basic goods and services and allows them to enjoy life and exchange goods with other people. By Professor Scharding's logic, cryptocurrency falls short of being an ethical currency because its often-changing value means that one day it could secure these assets and more for a given holder while the next day it could be deemed worthless, leaving the holder unable to meet these basic needs. That said, she believes cryptocurrencies could become ethical if an entire nation takes on cryptocurrency and stabilizes its value [20].

7.2 Investor Obligations

Managing an index fund holds a moral obligation to the holders to aim to grow their investment. Cryptocurrency price prediction carries inherit risk. A major concern is the volatility of cryptocurrencies. Though we seek to leverage this volatility, it could also be a major pitfall if prices are incorrectly predicted. In this proposed fund, if the crypto prices are wrongly predicted, funds will likely get misallocated and, in turn, lose the investor money. If the machine learning algorithm wrongly predicts the coming crypto prices, we risk losing investor money and being an unethical project, as it promotes investing with no return. To uphold our ethical obligation to investors, we must strive for the highest level of accuracy in this work.

7.2.1 Model

Cryptocurrency prices are extremely volatile, making their price prediction a uniquely difficult challenge. The likelihood that this comps project were to succeed in accurately predicting cryptocurrency prices consistently is incredibly low. This leaves a high likelihood of inaccurate predictions at some point, if not frequently, which has greater implications on the ethical consideration of investor obligations, model transparency, and security. Promoting the practice of placing a monetary commodity in cryptocurrencies is ethically dubious, as it holds high risks of loosing this money, threatening the financial security of such investors.

To make this project more ethical, there is a need for a lot of transparency on the risks involved in investing in this comps project intended fund due to the accuracy rates of the developed machine learning model and the extremely volatile nature of the assets involved.

7.3 Investing

To explore the ethics of this investing-based project, we ask, *is investing itself fair?* In investing, we subscribe to a system where the wealthy have an infinitely easier time to build more wealth than the poor do. The stock market is an inherently unfair playing field. Due to the compound nature of investing and investments, it is infinitely easier to become wealthy if you are already wealthy than it is to turn a little into a lot; the stock market favors those who start off with more. By this logic, it appears to be an unethical system.

We can, however, approach ethical investing from another standpoint. Rather than concern ourselves with the ethics of investing itself, we can ask how to ethically invest money. There is a growing field of learning around *ethical investing*. Ethical investing has many different approaches, but the idea is to use investing as a tool to do good. Manisha Thakor, a financial planner and consultant, says, according to Lauren Migaki (2022) [14], “The broad idea behind this style of investing is a belief that you can generate meaningful, measurable, societal outcomes while also generating a healthy profit”. You can put your money towards companies addressing an ethical issue, such as climate change, racism, or workplace inequality while also generating wealth. The three overarching areas for ethical investments are environment issues, societal issues, and governance issues [14]. Cryptocurrencies do not fall under environment issues to the best of my knowledge, but they can, arguably, fall under societal and governance issues.

For starters, cryptos can help in enabling financial inclusion. As Jill Carlson (2021) [13] says, “The crypto economy is leading to the development of an alternative financial and technological infrastructure that is global, open source, and accessible to all who have access to the internet, regardless of nationality, ethnicity, race, gender, and socioeconomic class... not enough of us are rolling up our sleeves and getting involved with building this new global and inclusive open financial system”. By this framework, cryptocurrencies have the power to do extreme good, and investing in them allows us to ”[roll] up our sleeves and [get] involved with building this new global and inclusive open financial system”, allowing for greater financial inclusion and opportunity worldwide, a pivotal societal and governance problem.

7.4 Accessibility

Another ethical concern this project faces is accessibility. We must consider:

- Is it fair that only some have access to crypto price predictions?
- Is it ethically right to make such crypto price predictions widely available?
- If crypto price predictions are made widely available, will the prediction itself alter the crypto prices, creating self-fulfilling, ‘announcement-driven’ outcomes?

We navigate each of these questions in the below sections.

7.4.1 Power Distribution

As it stands, “the rich get richer”. If this comps project achieved any level of success in predicting cryptocurrency prices, there would be a question of *who* gets access to these predictions. If these predictions were offered as a subscription service with a high buy-in, the buyers would most likely be those wealthier individuals, giving the elite investors a tool not readily available to the average investor, making it continuously easier for the wealthy to keep get wealthier while the poor get poorer. If these predictions or this fund were expensive and exclusive, we risk contributing to the wealth divide and creating further disparity in power distribution. If instead we choose to make this tool openly accessed, we run into other ethical problems.

7.5 Self-Fulfillment

Imagine now that everyone had access to this crypto price prediction tool and that it operated with some meaningful level of accuracy. If these predictions were so easily accessed, it is entirely possible that the predictions would become self-fulfilling cycle where:

1. Price predictions are publicized;
2. People buy stock when its predicted to be rising in price and sell it when it is predicted to fall in the coming future;
3. These prediction-based purchases drive up or down the crypto.

If this were to happen, no *actual* predicting would be helpful, and all people would not be benefited by such predictions. As such, it is important that the price prediction component of this comps project be kept private, with only the fund performance made available.

References

[1] *A faster way to build and share data apps*. Last accessed December 5, 2022. 2022. URL: `https://streamlit.io`.

[2] Apple. “Using Long Short-Term Memory Layers (LSTM)”. In: *Apple Developer Documentation* (). DOI: `https://developer.apple.com/documentation/accelerate/bnns/using_long_short-term_memory_layers_lstm`.

[3] Brownlee, Jason. *A Gentle Introduction to Long Short-Term Memory Networks by the Experts*. 2017. URL: `https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/` (visited on 03/31/2022).

[4] Ching-Ru Ko, Hsien-Tsung Chang. “LSTM-based sentiment analysis for stock price forecast”. In: *National Library of Medicine* (2021). URL: `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7959635/`.

[5] *Classification: Precision and Recall*. Last accessed December 5, 2022. July 2022. URL: https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall.

[6] *Constant Rebalanced Portfolios - some simulations with numpy*. 2015. URL: https://www.chrisstucchio.com/blog/2015/constant_rebalanced_portfolios.html.

[7] Dalvin Brown. *Tracking stolen crypto is a booming business: How blockchain sleuths recover digital loot*. 2021. URL: https://www.washingtonpost.com/technology/2021/09/22/stolen-crypto/.

[8] Gupta, Shashank. *Sentiment Analysis: Concept, Analysis, and Applications*. 2018. URL: https://towardsdatascience.com/sentiment-analysis-concept-analysis-and-applications-6c94d6f58c17 (visited on 04/25/2022).

[9] Helder Sebastio, Pedro Godinho. “Forecasting and trading cryptocurrencies with machine learning under changing market conditions”. In: *SpringerOpen* (2021). URL: https://jfin-swufe.springeropen.com/articles/10.1186/s40854-020-00217-x.

[10] *How can I create Multiple Input One Output LSTM Model with Keras?* Last accessed December 14, 2022. URL: https://stackoverflow.com/questions/59457567/how-can-i-create-multiple-input-one-output-lstm-model-with-keras.

[11] Hutto, C.J. and Gilbert, E.E. *VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text*. Ann Arbor, MI, June 2014.

[12] IBM. *What is Blockchain Technology?* 2022. URL: https://www.ibm.com/topics/what-is-blockchain.

[13] Jill Carlson, Chrintine Moy. “Cryptocurrencies can enable financial inclusion. Will you participate?” In: *World Economic Forum* (2021). URL: https://www.weforum.org/agenda/2021/06/cryptocurrencies-financial-inclusion-help-shape-it/.

[14] Lauren Migaki, Andee Tagle. *Understanding the promises and limits of ethical investing*. 2022. URL: https://www.npr.org/2022/01/11/1072207126/ethical-investing-with-esg-funds.

[15] Lewinson, Eryk. “Algorithmic trading based on mean-variance optimization in Python”. In: *Towards Data Science* (2019). URL: https://towardsdatascience.com/algorithmic-trading-based-on-mean-variance-optimization-in-python-62bdf844ac5b.

[16] *Mean squared error*. Last accessed December 5, 2022. URL: https://en.wikipedia.org/wiki/Mean_squared_error.

[17] Olah, Christopher. *Understanding LSTM Networks*. 2015. URL: https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (visited on 03/31/2022).

[18] Phi, Michael. “Illustrated Guide to LSTM’s and GRU’s: A step by step explanation”. In: *Towards Data Science* (2018). DOI: https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21.

[19] Pinkasovitch, Arthur. *Types of Rebalancing Strategies*. 2022. URL: https://www.investopedia.com/articles/stocks/11/rebalancing-strategies.asp#toc-what-is-a-percentage-of-portfolio-rebalancing-strategy (visited on 04/25/2022).

[20] Rutgers Business School Newark and New Brunswick. *Is bitcoin an ethical currency? Not according to professor’s new research*. 2018. URL: https://www.business.rutgers.edu/business-insights/bitcoin-ethical-currency-not-according-professors-new-research.

[21] Shah, Isah & Zulkernine. “Stock Market Analysis: A Review and Taxonomy of Prediction Techniques”. In: *International Journal of Financial Studies* 2019.7 (2019), pp. 1–22. DOI: https://www.mdpi.com/2227-7072/7/2/26.

[22] Singh, Falak. *Sentiment Analysis Made Easy Using VADER*. Last accessed December 4, 2022. Dec. 2020. URL: https://analyticsindiamag.com/sentiment-analysis-made-easy-using-vader/.

[23] *Stock Market Sentiment Analysis Using Python & Machine Learning*. URL: https://www.youtube.com/watch?v=4OlvGGAsj8I.

[24] Teo, Bee Guan. *How to Improve Investment Portfolio with Rebalancing Strategy in Python*. Last accessed December 5, 2022. Aug. 2021. URL: `https://python.plainenglish.io/how-to-improve-investment-portfolio-with-rebalancing-strategy-in-python-a58841ee8b5e`.

[25] Valkov, Venelin. *Cryptocurrency price prediction using LSTMs*. Last accessed December 14, 2022. Apr. 2019. URL: `https://towardsdatascience.com/cryptocurrency-price-prediction-using-lstms-tensorflow-for-hackers-part-iii-264fcdbccd3f`.

[26] Yathish, Vishal. *Loss Functions and Their Use in Neural Networks*. Last accessed December 5, 2022. Aug. 2022. URL: `https://towardsdatascience.com/loss-functions-and-their-use-in-neural-networks-a470e703f1e9`.

[27] *yfinance*. Last accessed December 5, 2022. Nov. 2022. URL: `https://pypi.org/project/yfinance/`.