Crypto For Underserved Populations: Technical Report

Group A1: Brendan Byrnes, Griffin Lester, Vish Nalagatla

5/10/2022

**Introduction**

The purpose of this report is to provide deeper context into the technical concepts applied to our cryptocurrency final project. These concepts include data sources, metrics used, machine learning models/methods used, and others. The sections of this report are organized in chronological order of completion. We will also walk through our workflow in Databricks step by step.

**Data Collection/Exploration**

The first main task of our project was the data collection phase. This step was crucial in ensuring that we could easily and efficiently access and update the data on a regular basis. For different parts of the project, we used two different data sources. They are outlined below:

* *Tidyquant (R Package)*: This package provides simple ways to load and analyze financial data in R. The function used to pull data from Yahoo Finance via this package is *tq\_get()*, which allows for stock, index, and cryptocurrency calls. You can also specify which specific symbols to pull and the periodicity of each. We used this data source for the purposes of data exploration and some brief feature engineering. More information about this package can be found [here](https://cran.r-project.org/web/packages/tidyquant/vignettes/TQ00-introduction-to-tidyquant.html).
* *LunarCrush (Python API)*: This API gives much information about various cryptocurrencies and the social interactions associated with each. The main benefit of using LunarCrush is that many of the features are already included, displaying information about Twitter, Reddit, or URL activity. They also include various ‘scores’ that are admittedly arbitrary and should not be naively trusted without further research. More information about this API can be found [here](https://lunarcrush.com/about).

**Feature Engineering**

The next phase of the project was to engineer meaningful features into our data. Although most of our features were already present via the LunarCrush API, we needed more information about specific movement in the crypto space to provide deeper actionable insights. Listed below are the features we engineered:

* *Moving Average*: For each crypto we analyzed, we created a 7-day and a 50-day moving average of the closing price. This means that we averaged the closing price over each time frame and moved that window forward with each new day. We engineered this feature to acquire a short-term and a long-term understanding of a crypto’s closing price. We also used these features when analyzing MACD, which is explained later in this section. More information about moving averages can be found [here](https://www.investopedia.com/terms/m/movingaverage.asp#:~:text=our%20editorial%20policies-,What%20Is%20a%20Moving%20Average%20(MA)%3F,commonly%20used%20in%20technical%20analysis.).
* *Relative Strength Index (RSI)*: The RSI is a metric often used for financial evaluation. Its values are within a 0 to 100 scale. An RSI value below 30 often means that an investment is underbought and undervalued, pushing people to buy the asset. An RSI value over 70 often means that an investment is overbought and overvalued, pushing people to sell the asset. We used this metric to determine optimal times to buy and sell cryptocurrencies. More information about RSI can be found [here](https://www.investopedia.com/terms/r/rsi.asp).
* *Moving Average Convergence/Divergence (MACD)*: The MACD is a financial evaluation method that indicates when best to buy or sell an asset. This method involves the use of both a short-term and a long-term moving average. When the short-term moving average crosses and moves below the long-term moving average, it is recommended that a user sells the given asset. When the short-term moving average crosses and moves above the long-term moving average, it is recommended that a user buys the given asset. More information about MACD can be found [here](https://www.investopedia.com/terms/m/macd.asp#:~:text=Moving%20average%20convergence%20divergence%20(MACD)%20is%20a%20trend%2Dfollowing,from%20the%2012%2Dperiod%20EMA.).

**Databricks Workflow**

*Environment Initialization*

This section initializes the Lunar Crush API, and allows us to call information from this API.

*Coins*

The next five sections are titled: Ethereum (ETH), Bitcoin (BTC), Cardano (ADA), Dogecoin (DOGE), and Litecoin (LTC). Each of these sections contains the code to call certain specified columns of information for that specific coin. We will take you through Ethereum as an example. The first block of code simply specifies the cryptocurrency that you want to pull, how many days of information you would like to pull, and the time interval that you want to pull. The data that is pulled is in json format. The next block of code converts this json format into a spark dataframe. The final block of code creates the schema that you would like, and it allows you to select the columns of information that you would like to pull for that specific coin. This process remains the same for all the other coins as well.

*Further Feature Engineering*

The first part of this section was to create our *Rolling Means* for each different type of cryptocurrency. We imported 3 different modules that helped us with the calculation of these rolling means. Those modules were avg, col, and Window. These functions allowed us to carry out the rolling mean functions as shown. I will again use ethereum as an example. The first code block shows how to get the 7 day rolling mean and the second code block shows how to get the 50 day rolling mean. The process is fairly simple. This process was repeated for all cryptocurrency. I should also mention that this function creates a column for each of those rolling means.

The second part of this section is titled *RSI* or Relative Strength Index. We create another column in each different crypto dataframe showing its RSI. There is a function that we used, which is shown in the code. It is important to note that this code does not work for a spark dataframe. This is why we had to convert the data frames to pandas data frames to use the function, then change them back to spark dataframes. This process was repeated for all 5 cryptocurrencies.

*Make Combined Crypto Data Frame*

The first block combines all the data frames into one using the union function. The next block is used to turn the pandas dataframe back into a spark dataframe. Finally we turned this data frame into a csv file so that we could attempt to merge these predictions with our chatbot.

**AutoML**

Included in the zip file submission are various notebooks produced by the AutoML feature in Databricks. We used this feature to produce time series forecasts over a 7-day period for both the RSI and the Galaxy Score. There were five notebooks produced for each response; each notebook represents a model trained by the AutoML feature. The resulting forecasts would then be used to aid our chat bot’s decision making and recommendation processes. Please refer to the notebooks included in the zip file for more information.d