

Project Objectives

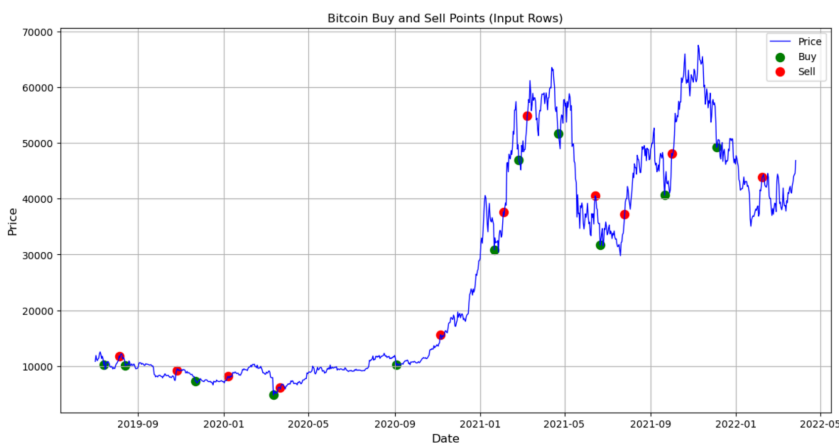
The goal of this project is to develop an automated cryptocurrency trading bot that constantly operates. It executes trades based on user-defined parameters (buy under average percentile, sell over average percentile, and average interval in days). The bot is programmed to buy when the price drops below a user-specified percentage of its moving average, calculated over a time window set by the user. Similarly, the bot will automatically sell when the price rises above the moving average by a user-defined percentage.

Additionally, an algorithm has been implemented to predict future prices. Using these predicted prices as inputs, the bot applies the same trading strategy—buying and selling based on the calculated thresholds and moving averages derived from the predicted data. The objective is for the bot to buy high and sell low while retaining the income that can come from the change in price when buying and selling.

Techniques and tools

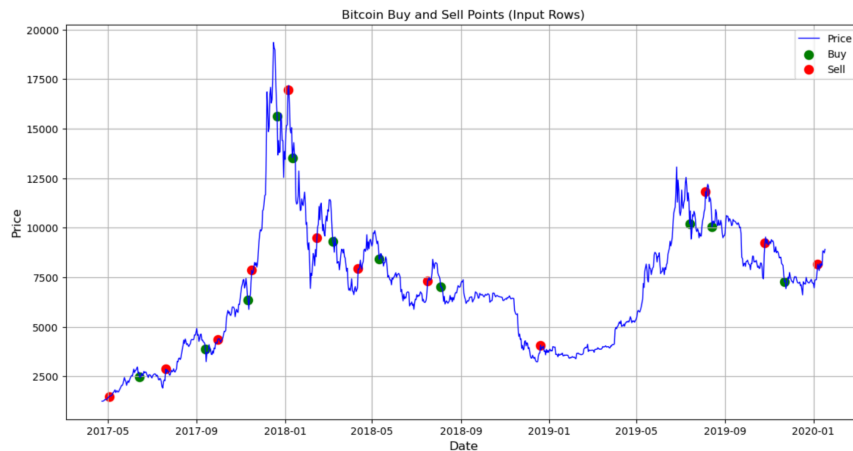
We started by importing the data for the cryptocurrency Bitcoin for 10 years (2014-2024) to test our code. I was able to import the data and set names to the variables. After that, I created a command using the `.rolling[].mean()` to set the rolling average for the set period. With this set, we are able to use the % change equation to find the percent change compared to the average and set the for loop to execute the buy if the average is below the user-set threshold. This works the same way, just opposite for the sell function.

As seen here in the chart, the algorithm overall buys low and sells high. It is the most inaccurate during large bear cycles, but a “stop loss” function may help with reducing the loss in investment. This is illustrated with a set -0.1 or -10% buy threshold and a 0.1 or 10% sell threshold. I used a 10-day average for this illustration. We used the matplotlib, numpy, and pandas to be able to organize and plot the data.



Observations and conclusions

Here we observe that when the market is upward trending, our algorithm has great buy and sell timing. But when the market declines, some major losses are taken with our current



algorithm. This is shown in the partial plot seen here. When there are huge Market bears, we can see significant losses being taken here by the investor. Some ways that can prevent this are implementing stop losses or another addition to our loop that specifies the price the security is bought must be less than

the sale price. Overall, our algorithm seems to be profitable for trading bitcoin specifically with our data here. I think this is a great starting point for an ever-running program to trade cryptocurrency.

In conclusion, our project can help investors trade without having to hawk the market all of the time. Our algorithm calculates the average price using a timeframe set by the user to trade cryptocurrencies. With this, the investor is able to set percentages below the market to buy and percentages above average to sell. Here the algorithm will naturally buy low according to the average and sell high according to the average. We then have a prediction aspect that is able to predict the price of the security on certain days based on this average. With this value, this bot can trade completely with analytical techniques without the requirement of human analysis.

Contributions

I was able to import the data and create the algorithm to calculate when to buy or sell a security. I was then able to create a table to illustrate when the buy and sell transactions take place. With this, I was able to create graphs that represent when the bot bought and sold compared to the price movements of the coin.

My partner James played a vital role in creating the prediction algorithm and applying that to the buy and sell algorithm I developed. He was also able to create a method to find the predicted values based on the average movement throughout time, which would be effective for trading.

Overall, I think we both made vital contributions to this project, and we are both very proud of the experience and knowledge we were able to gain from creating this program.