University at Buffalo

Team 3: Options Strategies

Mannit Trehan, Lucas Simpson, Brian Scorcia, Eric Velepucha

CSE 478

Dr. Zhen Liu

December 10, 2021

The goal of our group initially was to explore how we could combine our coding knowledge to analyze stock market concepts. Three out of four members of our group were Computer Science undergraduates with no experience in options trading. After hearing the different topic ideas our teammate Mannit, who is a graduate studying Finance, believed options strategies would be a great place to start trying to bridge our two fields.

The first step towards our goal was identifying which options strategies to focus on testing. We decided on Covered Calls, Married Puts and Synthetic Longs. We felt this gave a great balance of utilizing calls and puts. These are all bullish strategies, but we felt that was best to avoid overcomplicating our first experience with options. As a result, our final project was to back test 3 different bullish options strategies on four different stocks and compare the results to simply holding each of the stocks in the same time period. We created graphs and tables based on our findings to evaluate which, if any, was best and in which scenarios one would be preferred over another.

The next step and most cumbersome was data collection. For the standard daily stock data, Yahoo Finance has a Python library which allows very easy access to most stock historical price data. We initially got excited when we also saw that the library had a method called “options_chain” and believed that we could use this to retrieve historical option prices. As a result, we downloaded data for 10 different stocks going back 5 years. Our finance member selected these 10 stocks to cover a variety of industries to get a diverse selection for our study. Our selected stock tickers were AMD, APPL, BBBY, F, GM, M, MSFT, PFE, QCOM, and STZ. We also decided to pull historical data for SPX, the S&P 500, to get a baseline for how the overall market fared compared to our strategies as well.

After collecting all the historical data for daily stock prices, we tried to access historical options pricing only to realize that the yfinance library only offered current options chain data. As a result, we had to search for other sources of finding this information. We ended up finding a site called Barchart.com which offered a free trial to their platinum membership which allowed us to view historical options data for stock tickers. However, the site requires you to manually select a ticker, strike price, option type as well as expiration date. The site also has a daily download limit of 100 reports. As a result, each month required two reports per stock and the time necessary for collecting this much data manually for 10 stocks going back 5 years was not reasonable given the aims of our project. We decided to collect monthly call and put data for "At The Money" strike prices for 4 stocks (MSFT,AMD,GM, and F) going back 2 years.

Now that we had our final dataset, it was time to code our selected trading strategies and also calculate our benchmark returns. The benchmark returns were done first and were fairly straightforward. We decided to have an "account" purchase $100,000 worth of shares at the open price for our earliest month of data, November 2019. We then generated our returns by selling these same shares at the end of October 2021 and then performing a subtraction of our initial value to get our monetary gains. Then we divided this amount by our starting $100,000 to receive our percentage gain. Finally, we then coded the different options strategies in Python. Covered Calls and Married Puts followed a similar process while Synthetic Longs were slightly different in that it required using both call and put data each month. For the first two strategies, the algorithm would check the stock price and the option premium for the corresponding type and then calculate the maximum combination of shares and contracts that could be afforded then purchased or sold them. At expiry, any contracts "in the money" would then be exercised and the new account balance would be recalculated depending on the outcome. For example, for

Covered Calls when the contracts we sold expired in the money that meant that we sold all shares and the account would now be 100% cash. The following month we would repeat the process of buying shares and selling call contracts. If the call contracts sold expired out of the money, the account would be near 100% shares and the following month we would only sell money contracts. The inverse would follow for buying our put contracts for the Married Put strategy.

The last strategy implemented was the Synthetic Long strategy. This algorithm involved reading the call and put prices for a stock at the beginning of each month to buy a quantity of call contracts equal to the quantity of put contracts we sell. At the end of the month if the call contracts expire in the money, we sell for a profit and get to keep the proceeds from the sold put contracts as well. In the case that the put contracts expire in the money, we would have to buy back the put contracts sold for the increased price, leading to a loss and the call contracts we bought would expire worthless.

Our conclusions from running these three strategies on our stocks showed that Married Puts and Covered Calls performed better than simply holding the underlying while Synthetic Long was a boom or bust strategy that usually busted in the time frame tested. In the future, all strategies, especially the Synthetic Long, could be improved by checking daily options prices and setting a percentage gain at which the user is happy to take profits. There were cases where the call options purchased would see substantial swings in value throughout the month before expiring out of the money. Allowing the program to wisely take profits would definitely more closely resemble how a human following these strategies would want to behave.