Crypto Trading Algorithm Using YahooFinance Data

Disclaimer : I am not a Financial Advisor and I am not advocating the trading algorithm below. This is an example.

The example will apply a trading algorithm to a set of historical data and publish the resulting returns. It is based on using standard Python libraries as well as using Yahoo Finance for acquiring market data. There are no market data licenses or costs involved, because the data was provided with a 15 minute delay and publicly available from YahooFinance.

I did not write all the code with this bot in its entirety. I started with a framework from Alex Reed and refurbished it to meet my needs. I highly suggest you check out some of the work of Alex Reed. The source code for this project can be found at the link below –

https://github.com/jeffj0110/Crypto_Trading_Bot

Core Topics

1. Review of trading strategy
2. Back Testing and Results
3. Lessons Learned

Review of the trading strategy

Indicators Used -

EMA 9, 50 and 200 – The Exponential Moving Average of the closing price based on the last 9, 50 or 200 candles. The 'exponential' prefix indicates that the most recent price in the candle is given more weight. Again, a candle can be any amount of time, but in this example we are using daily prices.

Stoch RSI is an abbreviation for Stochastic Relative Strength Indicator and it ranges from zero to 1. I won't go into the math, but the key is that it helps identify overbought and oversold situations. Our rule indicates that an RSI indicator below .30 is signaling an oversold condition. The K line is the fast moving RSI moving average (usually < 3 candles) and the D line is the slow moving average (> 3 candles).

The strategy we are using will employ the following rules.

- EMA 9 is higher than EMA 50 and EMA 50 is higher than EMA 200
- STOCH RSI is below 30 trending and STOCH line K cross above D
- Entry price: close of previous candle (previous day)
- Exit price when STOCH RSI line K crosses below D and STOCH RSI is greater than .9
- There is a stop loss if the position goes 20% below the purchase price.

Back Testing and Results

I ran the bot with 2 years of daily prices from Yahoo Finance for 25 cryptocurrencies. I traded one unit of a crypto and did not use any leverage. The results of those back tests are given in the table below. The variations in returns are not for the faint of heart. However, with this example algorithm, you are better off buying and holding (ie. be a hodler) rather than trying to trade them. This makes sense as the most crypto's did increase in value which is where buying and holding outperform a trading strategy which comes in and out of that upward trend. There are several cases below where the algo loss was 100%, but the overall average across all of them is an annualized return of 160% (over two years). The average return for buying and holding all 25 crypto's was 702% (again, over two years). Please don't take these returns as indicative of future results. There are several special situations where a crypto started at very small fraction of a cent and ended up a larger fraction of cent.

Ticker	Description	2 year Annualized Bot_Algo_Returns	2 Year Annualized Buy_Hold_Returns	Bot Transaction Count
BTC-USD	Bitcoin	59.61	103.97	16
ETH-USD	Ethereum	263.17	287.53	20
USDT-USD	Tether	-0.09	-0.06	6
BNB-USD	Binance Coin	-100	358.74	20
USDC-USD	USD Coin	-0.57	-0.11	4
ADA-USD	Cardano	86.73	393.13	14
SOL-USD	Solana	805.69	1349.08	8
HEX-USD	Hex	-100	4249.54	22
XRP-USD	XRP	-100	63.61	20
LUNA1-USD	Terra	2206.27	1710.43	28
DOT-USD	Polkadot	92.1	284	8
DOGE-USD	Dogecoin	594.42	672.65	16
AVAX-USD	Avalanche	-100	449.46	16
MATIC-USD	Polygon	-100	891.88	16
SHIB-USD	Shiba	-100	5522.69	2
CRO-USD	Crypto.com	53.81	175.78	12
DAI-USD	Dai	-2.05	0.06	4
ATOM-USD	Cosmos	-34.93	171.71	24
LINK-USD	Chainlink	197.14	154.98	20
LTC-USD	Litecoin	23.84	40.8	18
UNI1-USD	Uniswap	-100	161.63	6
ALGO-USD	Algorand	256.69	137.35	14
TRX-USD	TRON	-13.36	87.43	18
FTT-USD	FTX Token	206.46	303.58	18
BCH-USD	Bitcoin Cash	-100	-4.19	18

<u>Lessons Learned</u>

1. If Algorithmic trading was easy, everyone would be into it. Regardless, it is fun to correlate indicators to prices with the quantitative development tool sets in Python. This is very fertile ground for Machine Learning in my opinion. There are 2-3 extra steps involved to implement a model and you have to be careful not to overtrain the model for one thing.

2. Trading crypto’s isn’t difficult, if you can mentally and financially cope with the volatility. Don’t risk anything you can’t afford to lose.
3. Back testing is not indicative of future results, but does tell a story. It is very interesting to see the traps and/or rewards of an emotionless algorithm.
4. Make sure you are using source code control. Changing algorithms requires record keeping that needs to be referenced to a version of code and an input dataset.