



CRYPTO TRADING ALGORITHM

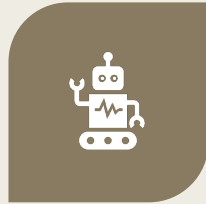
Rice FinTech Bootcamp 2020



Algo Facts, Tools and Goals



TRADE STOCKS AND
CRYPTOCURRENCY.



PULL DATA WITH APIS:
ALPACA, KRAKEN,
BINANCE



TRADING SIGNALS:
SMA, BOLLINGER
BANDS, AND MACD



SIGNAL SIMULATION
AND OPTIMIZATION



MACHINE LEARNING
SUPPORT: LSTM
PREDICTIVE PRICE
MODEL



SUBMIT TRADE
ORDERS ON PAPER
TRADE ACCOUNT



TIMEFRAME: DAILY
SIGNALS

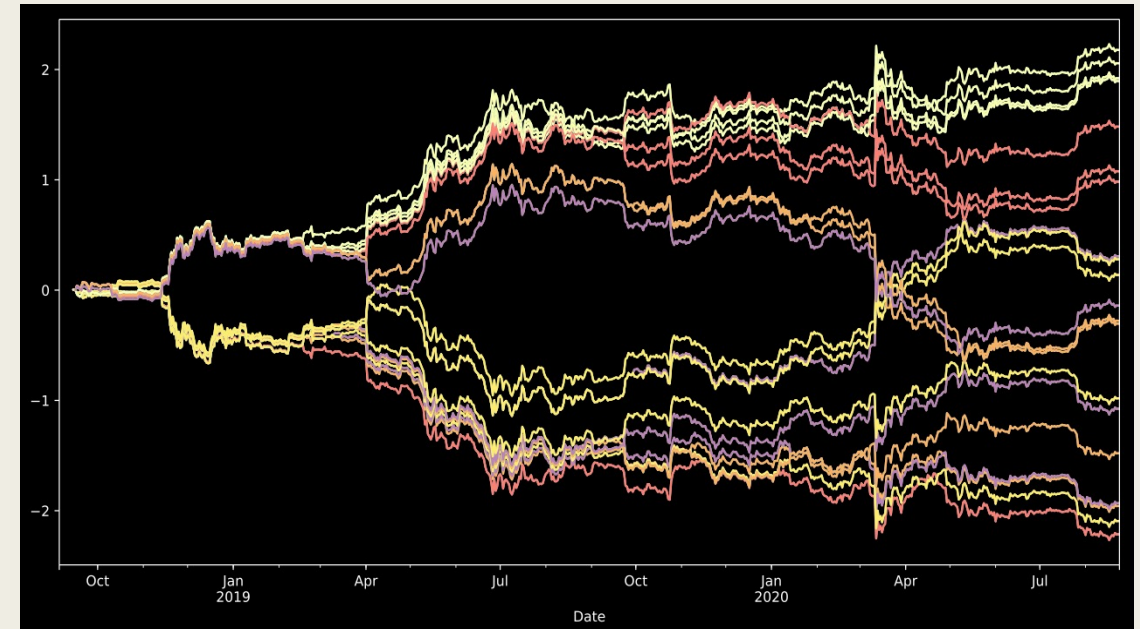
EMA Signal

Optimization & Simulation

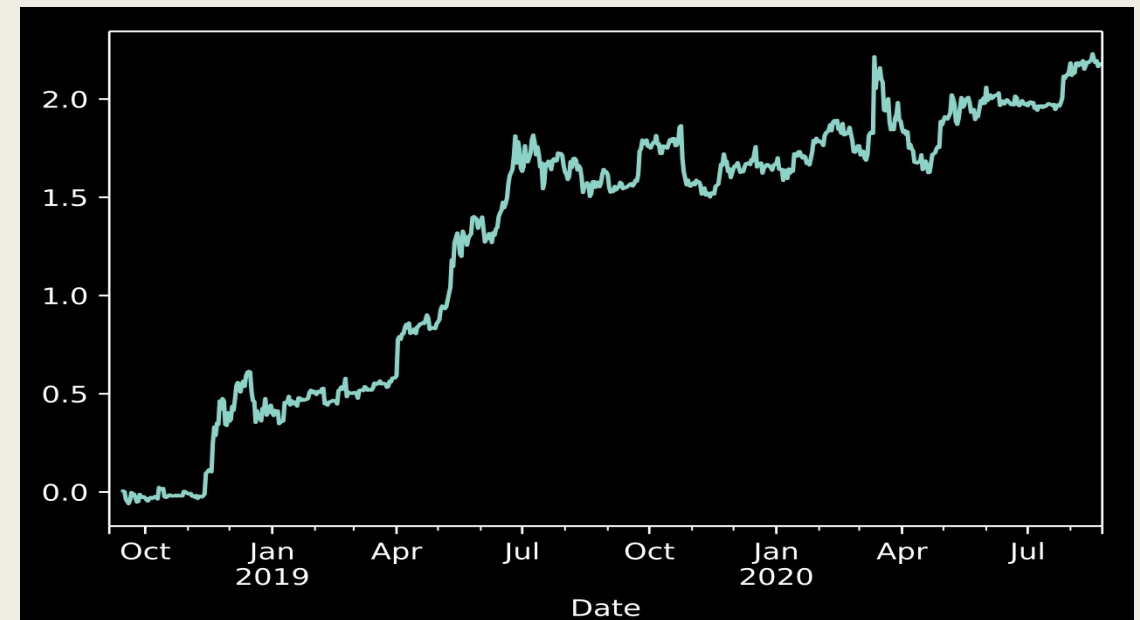
BTC/USD

- Find the Best Signal Inputs
- Set Optimal Parameters (12 H, 32, 10)
- Run Simulation
- View Signals
- Calculated Return (2.17)

CRYPTO EMA OPTIMIZATION



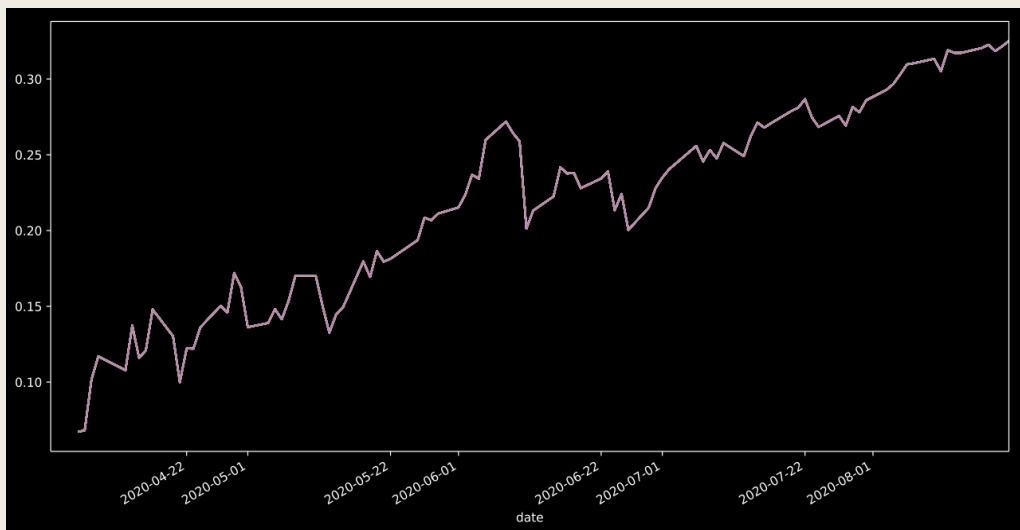
CRYPTO EMA SIMULATION



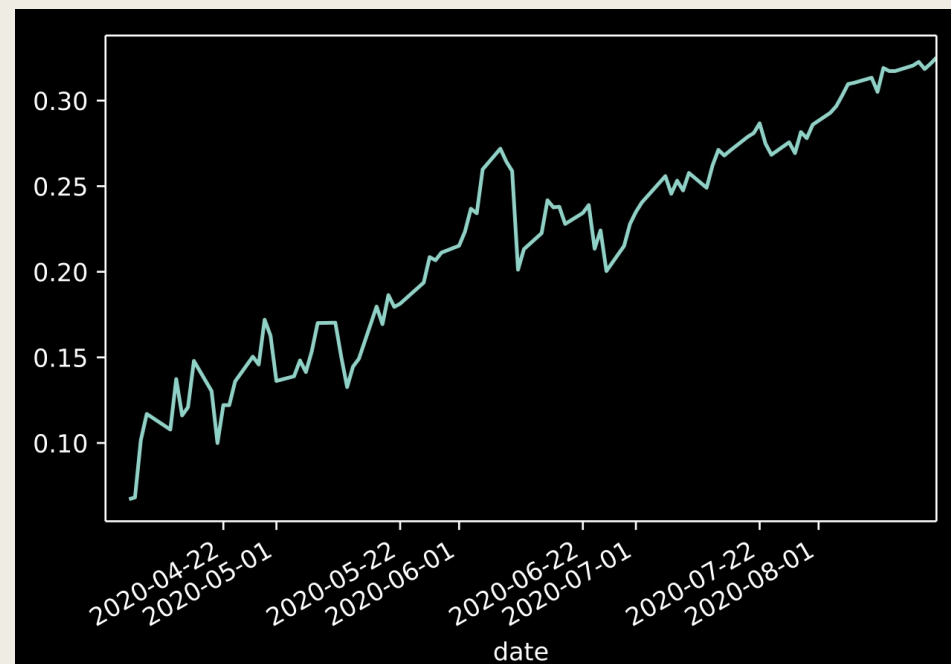
Stock EMA Results – SPY

(12H, 32.0, 10.0, 0.32)

STOCK EMA OPTIMIZATION



STOCK EMA SIMULATION



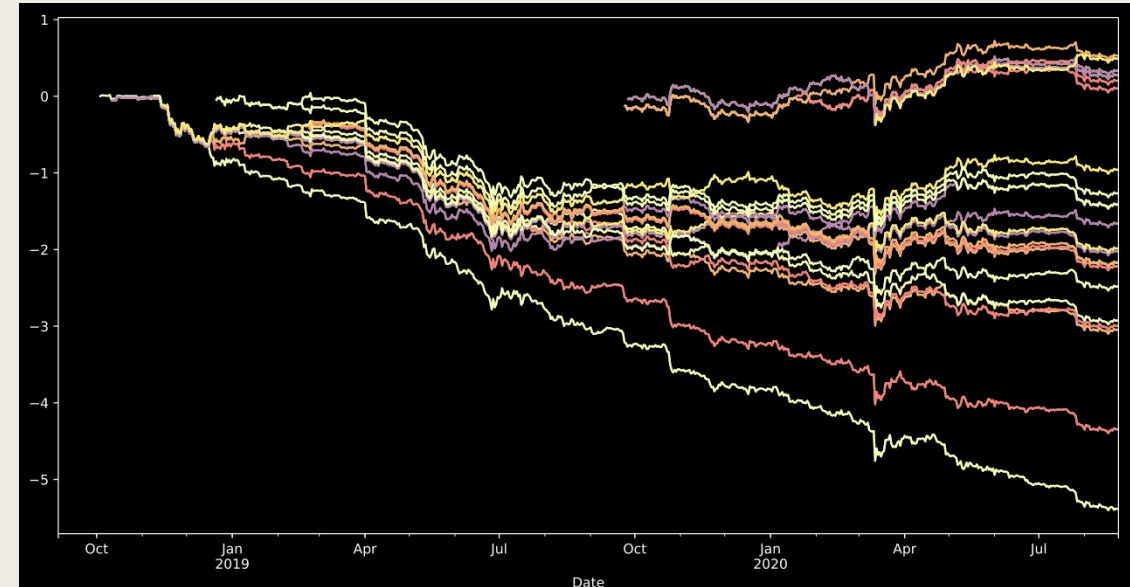
Bollinger Band Signal

Optimization & Simulation

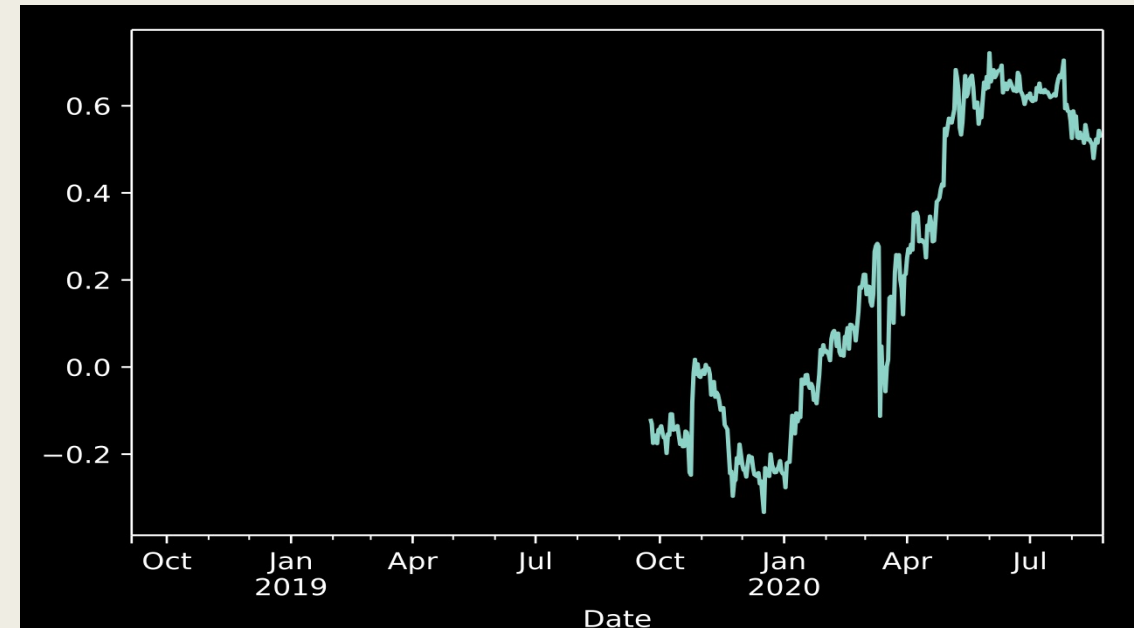
BTC/USD

- Find the Best Signal Inputs
- Set Optimal Parameters (48H, 100, 2.0)
- Run Simulation
- View Signals
- Calculated Return (0.53)

CRYPTO BOLLINGER OPTIMIZATION



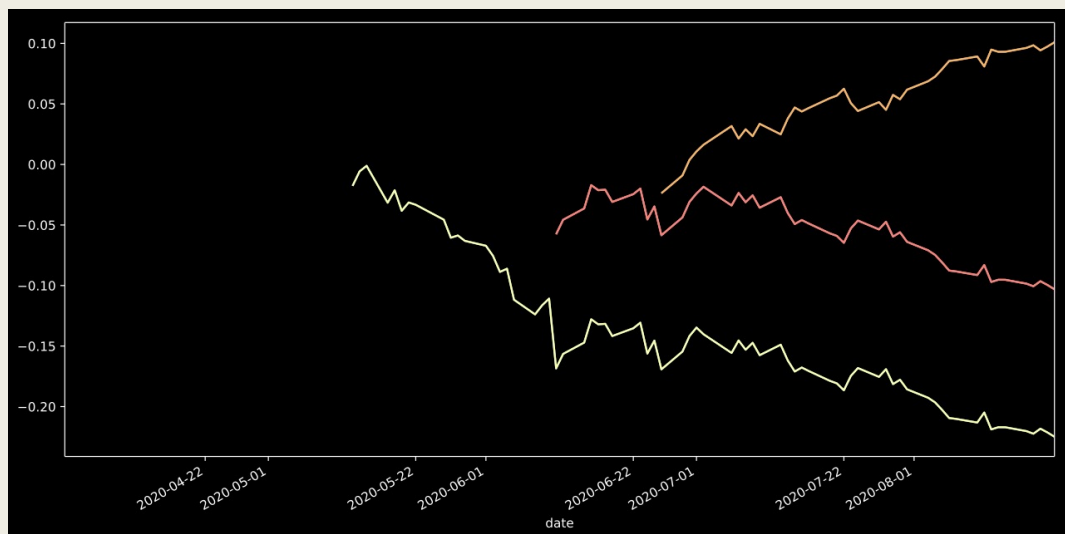
CRYPTO BOLLINGER SIMULATION



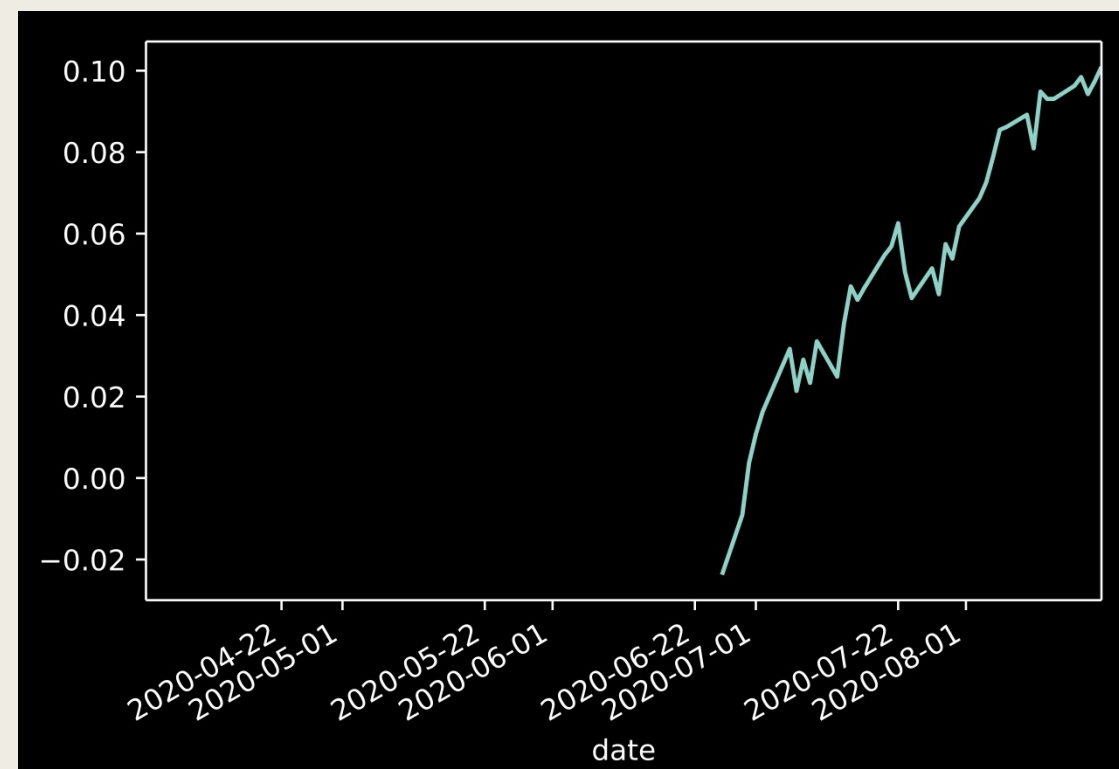
Stock Bollinger Results – SPY

(12H, 10, 2.0, 0.10)

STOCK BOLLINGER OPTIMIZATION



STOCK BOLLINGER SIMULATION



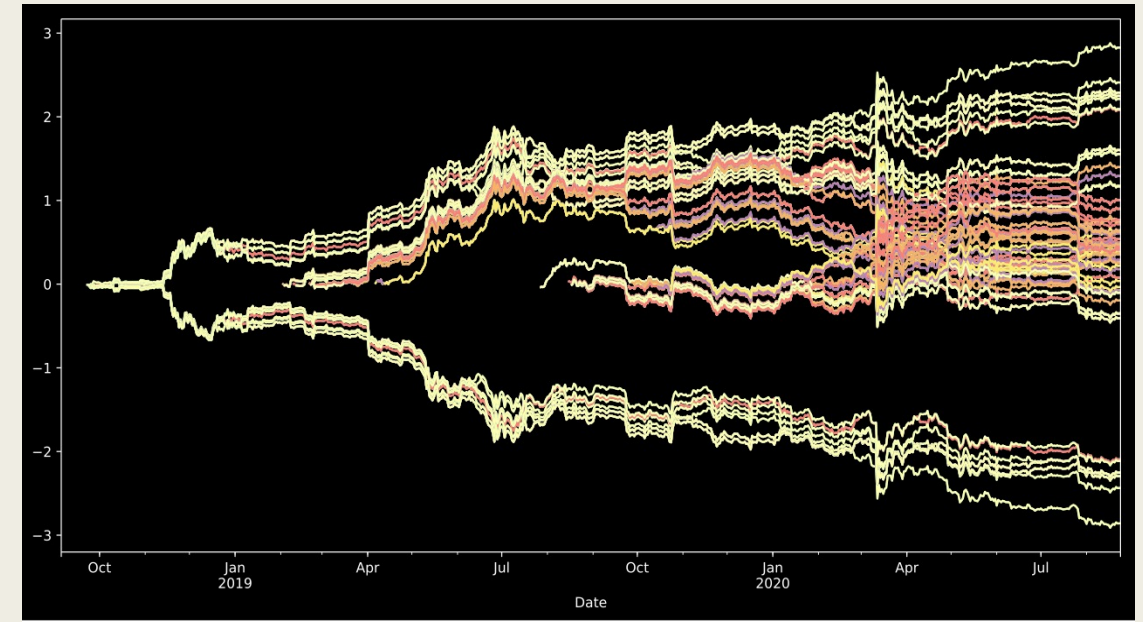
MACD Signal

Optimization & Simulation

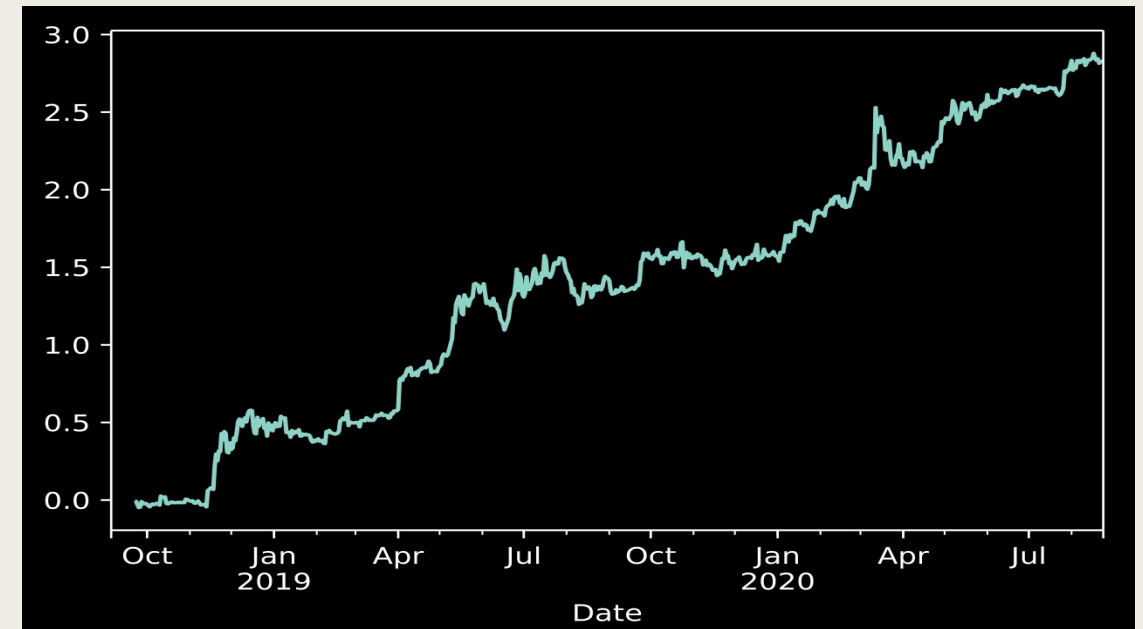
BTC/USD

- Find the Best Signal Inputs
- Set Optimal Parameters
- Run Simulation
- View Signals
- Calculated Return

CRYPTO MACD OPTIMIZATION



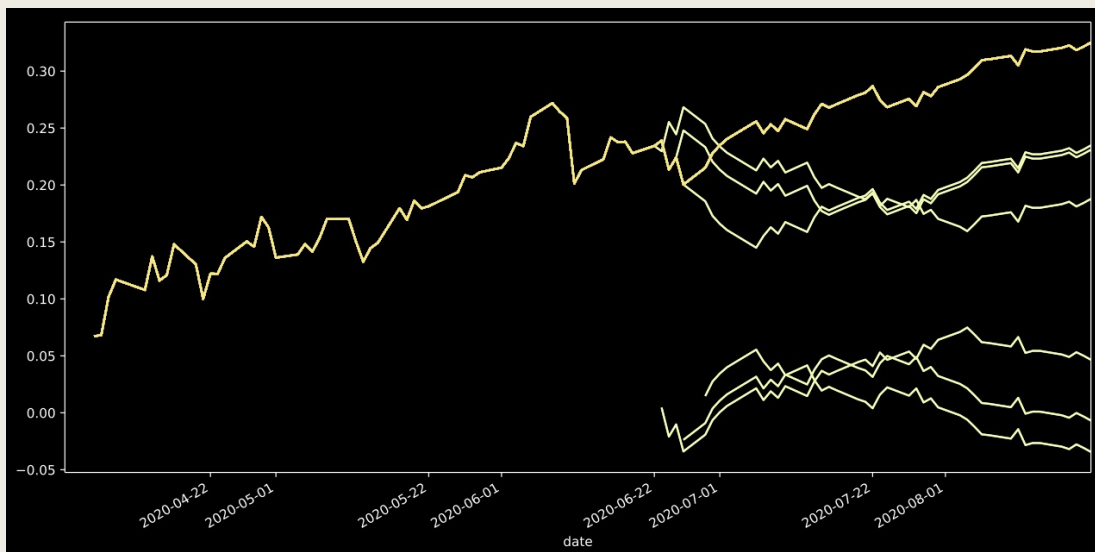
CRYPTO MACD SIMULATION



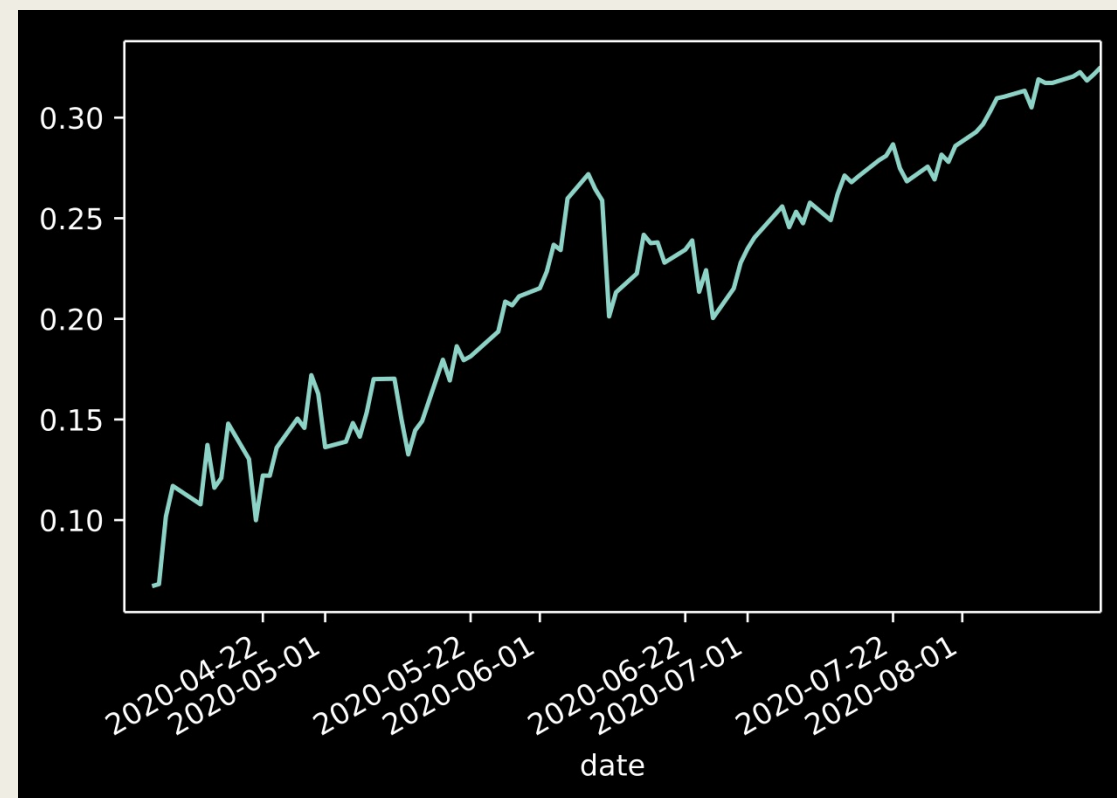
Stock MACD Results – SPY

()

STOCK MACD OPTIMIZATION



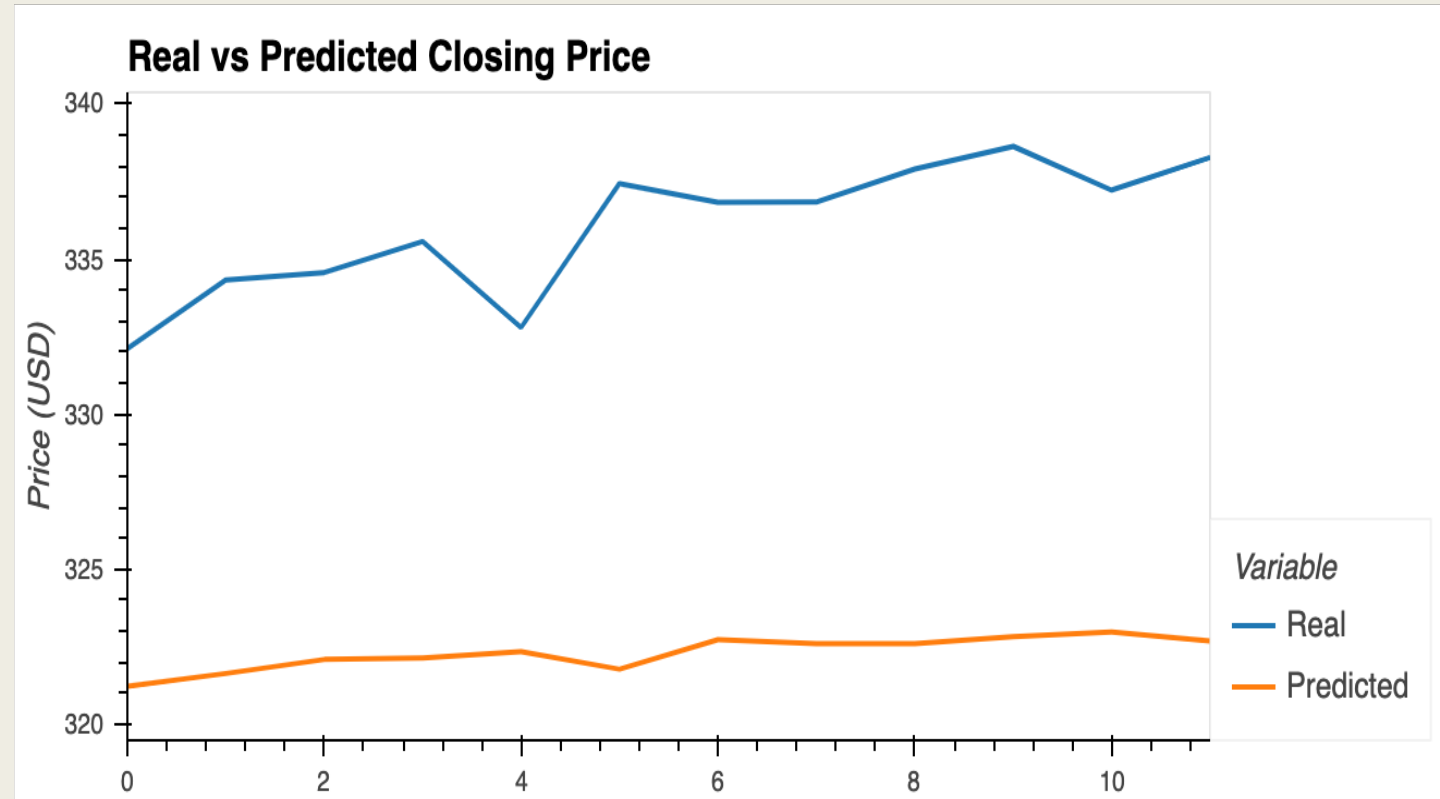
STOCK MACD SIMULATION



LSTM Predicted Price

Machine Learning Support

- Predictive Price Model
- Train/Test = 70/30 Split
- Based on Previous Price
- Prediction < \$15 > vs. Real
- Within = Positive Support
- Outside = Unsupportive



API Submit Order



API Function Call



Buy Signal (1) or Sell Signal (-1)



Send Order to Account

STOCK TRADING

ALPACA API (SPY)

```
def place_market_order(ticker, qty, side, order_type, time_in_force):  
    # Place a market order to buy qty of number of shares  
    api.submit_order(  
        symbol=ticker,  
        qty=qty,  
        side=side, #define side as either 'buy' or 'sell'  
        type=order_type, #define the type of the order: 'market', 'limit'  
        time_in_force=time_in_force, #define time in force: 'gtc'='opg'=opening  
    )
```

The screenshot displays the Alpaca Markets Paper Trading dashboard. The left sidebar contains navigation links for Overview, Paper Overview, Paper Positions, Paper Orders, OAuth Apps, Documentation, Community Slack, and Community Forum. The main content area is divided into three sections: Portfolio, Order History, and Watchlist. The Portfolio section shows a table with columns for Stock, Price, Shares, Market Value, and Total Profit. The Order History section shows a table with columns for Stock, Order, Shares, Price per share, and Status. The Watchlist section is currently empty.

Portfolio

Stock	Price	Shares	Market Value	Total Profit
SPY	+1.15% \$343.39	2	\$686.78	+\$2.46

Order History

Stock	Order	Shares	Price per share	Status
SPY	Market BUY 08/22/2020 02:21 PM	1	\$342.11	Filled
SPY	Market BUY 08/22/2020 01:42 PM	1	\$342.21	Filled

Watchlist

Your watchlist is currently empty. Click **Edit** to

CRYPTO TRADING

BINANCE API + KRAKEN API

```
order = client.create_order( #create_test_order
    symbol='BNBBTC',
    side=Client.SIDE_BUY,
    type=Client.ORDER_TYPE_MARKET,
    quantity=1)
```

0.00071197

47

21:18:36

0.00071154

74

21:18:37

0.00071153

150

21:18:36

0.00071147

56

0.03984232

Open Orders(0)

Order History

Trade History

Funds

☐ Hide Other Pairs

1 Day ▾

YYYY-MM-DD - YYYY-MM-DD

Search

Date	Pair	Type	Side	Average	Price	Executed	Amount	Total	Trig
08-24 23:15:04	XRP/ETH	Market	Buy	0.00071297	Market	20	20	0.01425940	—
08-24 23:14:42	XRP/ETH	Market	Sell	0.00071147	Market	15	15	0.01067205	—
08-24 23:00:09	XRP/ETH	Market	Sell	0.00071265	Market	15	15	0.01068975	—



Price(ETH)	Amount(XRP)	Time		Price(ETH)	Amount(XRP)	Total
0.00073206	51	12:31:32		0.00073232	454	0.33247328
0.00073206	43	12:31:30		0.00073230	4,113	3.01194990
0.00073206	51	12:31:30		0.00073227	1,752	1.28293704
0.00073400	69	12:29:01		0.00073225	59	0.04320275
0.00073290	46	12:28:48				
0.00073260	100	12:28:06		0.00073238 ↓ \$0.294322		More 
0.00073417	793	12:27:48		0.00073067	14	0.01022938
0.00073412	3,625	12:27:48		0.00073022	8,226	6.00678972
0.00073402	658	12:27:36		0.00073021	4,114	3.00408394
0.00073402	5	12:27:35		0.00072980	5,077	3.70519460
0.00073384	17	12:27:27				
0.00073333	87	12:27:25				
0.00073314	1,487	12:26:49				
0.00073283	25	12:26:49				
0.00073301	4,000	12:26:48				

[Open Orders\(0\)](#)
[Order History](#)
[Trade History](#)
[Funds](#)
☐ Hide Other Pairs

1 Week ▼ YYYY-MM-DD - YYYY-MM-DD

Search

Date	Pair	Type	Side	Average	Price	Executed
08-24 23:15:04	XRP/ETH	Market	Buy	0.00071297	Market	20
08-24 23:14:42	XRP/ETH	Market	Sell	0.00071147	Market	15
08-24 23:00:09	XRP/ETH	Market	Sell	0.00071265	Market	15

Amount

Total

Buy XRP

Assets

Deposit

Withdraw

XRP Available: 42.272138

ETH Available: 0.007081

main.ipynb ● binance.ipynb

```
[57]: import os
import binance
import datetime
from binance.client import Client
from dotenv import load_dotenv
```

```
[58]: load_dotenv()
      api_key = os.environ.get('api')
      api_secret = os.environ.get('secret')
```

```
[59]: client = Client(api_key,api_secret)
```

```
[60]: trades = client.get_historical_trades(symbol='BNBBTC')
```

```
[61]: #print(client.get_account())
```

```
[84]: #printing my balance on binance for Trx,
      print(client.get_asset_balance(asset='XRP'))
```

```
{'asset': 'XRP', 'free': '52.29213800', 'locked': '0.00000000'}
```

```
[63]: # get latest price from Binance API
btc_price = client.get_symbol_ticker(symbol="BTCUSDT")
# print full output (dictionary)
print(btc_price)
```

```
[96]: info = client.get_symbol_info('XRPETH')  
print(info)  
  
{'symbol': 'XRPETH', 'status': 'TRADING', 'baseAsset': 'XRP', 'ba  
ssetPrecision': 8, 'baseCommissionPrecision': 8, 'quoteCommis  
OSS_LIMIT', 'TAKE_PROFIT_LIMIT'], 'icebergAllowed': True, 'ocoAll
```

TURN THE ALGORITHM ON CONTINUOUSLY



CALL MAIN.PY FUNCTION
INSIDE
ACTIVATE_TRADING.PY



RUN ACTIVATE_TRADING.PY



INFINITE WHILE LOOP