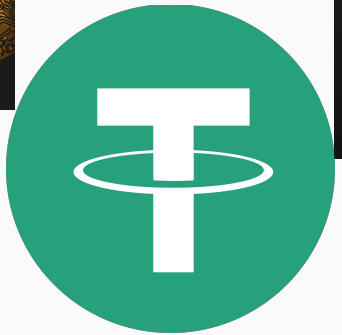
One Sigma

Ross Copeland and Jacob Marshall

Introduction

What is Cryptocurrency?

- Cryptocurrencies are virtual currencies that are able to be used/exchanged across the globe
- Some examples of cryptocurrencies you might have heard of are:
  - Bitcoin
  - Ethereum
  - Dogecoin
  - Tether



Cryptocurrency Exchanges

- Cryptocurrency, like any other currency, can be exchanged for other currencies
- Cryptocurrency exchanges, such as Binance, Coinbase, and Kraken, handle these transactions

Algorithmic Trading

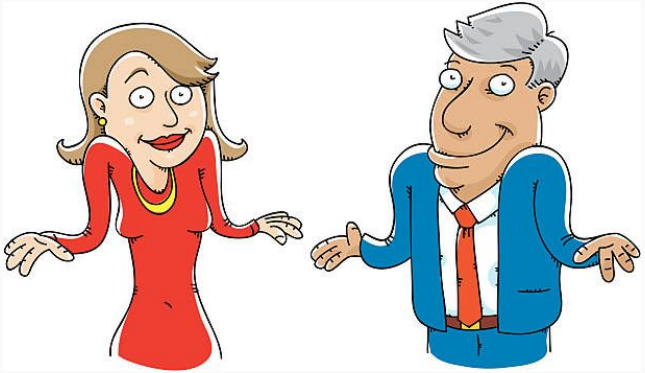
- Using Data Science + Machine Learning + Algorithms in financial markets is not a new idea
- Many firms find lucrative business via using these techniques within financial markets

Specific Problems Areas

How can you make money?

Making Money

- Who knows?
- Many of the strategies used are guarded secrets in the industry so there isn’t a clear path to success



Project Details

Project Goals

- 3 Components:
  - Trading Algorithms
  - Testing Infrastructure
  - Live-Trading Automated ‘Portfolio Manager’

Trading Algorithms

- We have focused on a few different strategies so far, which we'll get into later
  - Based upon different ML models
- Each strategy gives signals on when to enter a position, and when to exit if already entered
- ML models are good at providing signals on when to do these things, but we found better performance when we added additional logical safeguards

Testing Infrastructure

- Test the trading algorithms we've built against historical data and analyze performance
  - Historical Data that is different from the data we trained it on, (i.e. a different time period or cryptocurrency pair)
  - Provide logs/graphs on when the algorithm chose to enter/exit a position, calculate profit
  - Take into account trading fees/slippage

Live Trading Manager

- Feeds live data to algorithms
- Manages buy/sell signals from algorithms and interacts with exchange to execute trades
- Ensures portfolio stays healthy and diversified
- This is a feature we have not yet completed

Data Dive

What sort of data can we find?

- The data we found was either historical data or real time data
  - Historical data is usually guarded because of how valuable it is for backtesting
  - We obtained over 6 GB of data from the Kraken exchange
    - This data provided historical exchange rates going back up to 7 years for most major coins
    - This data was given in OHCL format (Open High Close Low) with minute by minute data points
  - Real time data enables live trading but requires more sophisticated infrastructure

Looking at the Data

- Each dataset pertains to a particular currency pair
  - Ex: BTC/USD
- Pair values give the amount of currency 2 that can be exchanged for a unit of currency 1
- Example: open, close, high, low, time, volume, and number of trades per minute for the BTC/USD pair (in 2013)

	Time	Open	High	Low	Close	Volume	Trades
1	1381095240	122	122	122	122	0.1	1
2	1381179000	123.61	123.61	123.61	123.61	0.1	1
3	1381201080	123.91	123.91	123.9	123.9	1.9916	2
4	1381209960	124.19	124.19	124.18	124.18	2	2
5	1381311000	124.01687	124.01687	124.01687	124.01687	1	1
6	1381311060	124.01687	124.01687	123.84	123.84	1.823	2
7	1381431780	125.85	125.86	125.85	125.86	2	2
8	1381571220	127.5	127.5	127	127	4	3
9	1381690560	131.8408	131.8408	131.8408	131.8408	0.1	1
10	1381717800	134.8	134.8	134.8	134.8	0.24	1
11	1381725600	134.3523	134.3523	133.36726	133.36726	9	6
12	1381768800	134.56	134.56	134.46	134.46	1.39	2
13	1381769040	135.49	135.59	135.49	135.59	1.3757534	2
14	1381809480	135.8	135.8	135.3	135.3	4	3
15	1381856460	153	153	153	153	0.2	1
16	1381857480	133.87975	133.87975	133.87975	133.87975	0.1	1
17	1381881660	143	143	142.57101	142.57101	3.839	3
18	1381912140	142.81001	142.81001	142.66	142.66	2	2
19	1381924080	145.38	145.38	145.38	145.38	0.19	1
20	1381936080	146.14	146.14	145.14	145.14	5	4
21	1381936140	145.14	145.14	142.28538	142.28538	5	2
22	1381941360	141.23668	141.23668	141.23668	141.23668	0.2	1
23	1381963200	139.16151	139.16151	139.16151	139.16151	1	1
24	1381965060	137.63758	137.63758	137.63758	137.63758	1	1
25	1381967880	137.99	138.09	137.99	138.09	2	2
26	1381968300	137.62448	137.62448	137.52	137.52	2	2

XBTUSD_1 +

Any Problems with the Data We Found?

- The data we were given was mostly clean by virtue of **not** having:
  - Extraneous numbers
  - Blank numbers
  - False numbers
- We did have to manually verify that the data was accurate
- The data we collected was specific to the exchange we collected it from, which could cause some bias in the machine learning models

Feature Engineering

Indicators

- Indicators are constructed with economic data and are used to provide additional insights into movements of the underlying asset
- We initially used the pyti library to add many different indicators into our dataset
    - We found the library to be too slow, so we implemented all the indicators ourselves

Moving Averages

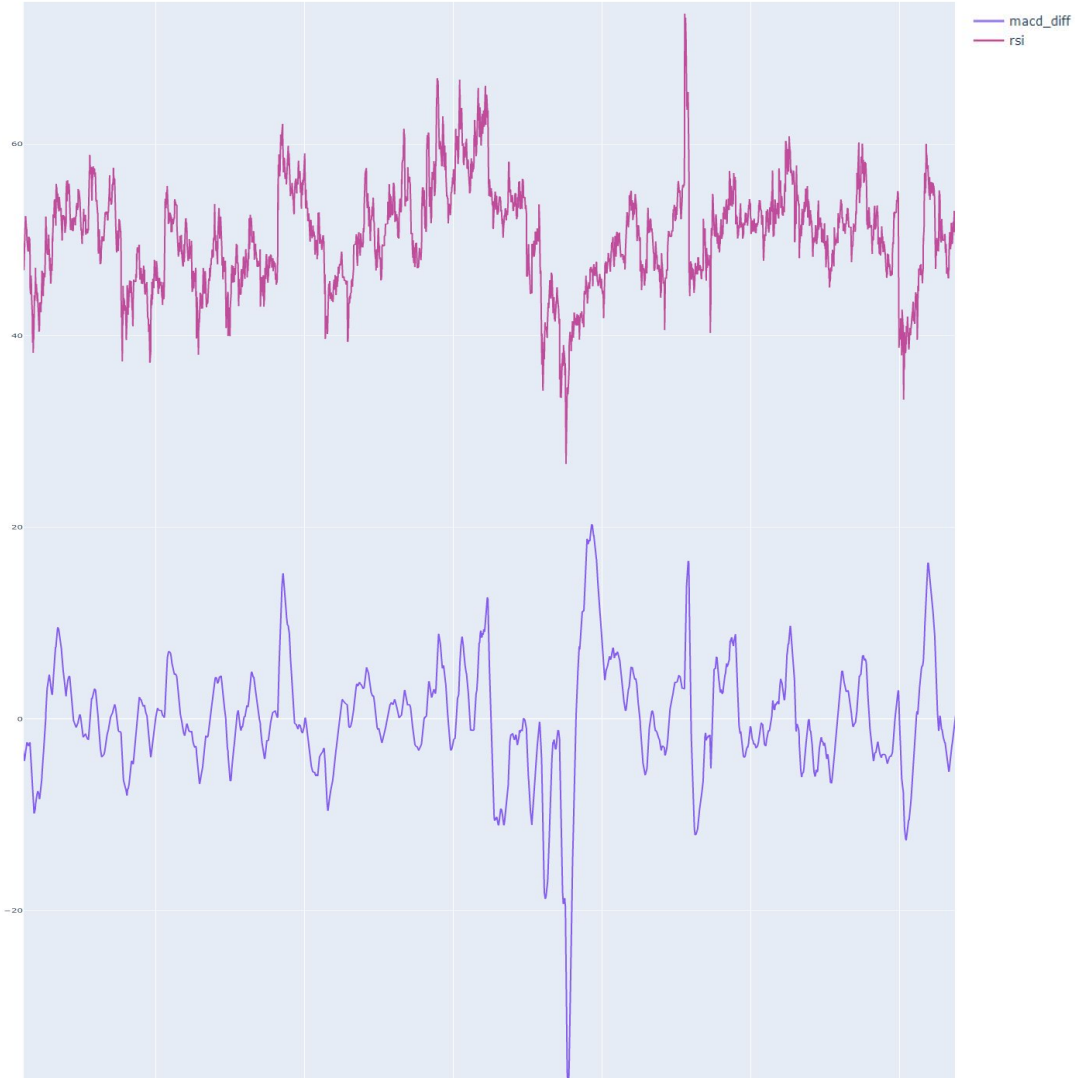
- We implemented the following moving averages as features:
    - Smoothed Moving Average
    - Simple Moving Average
    - Exponential Moving Average
  - Our implementation also allows for the moving average period to be specified by a given strategy, which allows for refinement and optimization


Candlesticks
EMA
SMA
SMMA
7400
7200

Other Indicators

- We also implemented the following other indicators and are able to add them as features to the dataset and specify their underlying parameters:
    - RSI
    - MACD
    - Stochastic Oscillator
    - Bollinger Bands
    - Variance
    - Minmax Slope


Candlesticks
boll_upper
boll_lower
400
200
300


macd_diff
rsi
60
40
20
0
−20

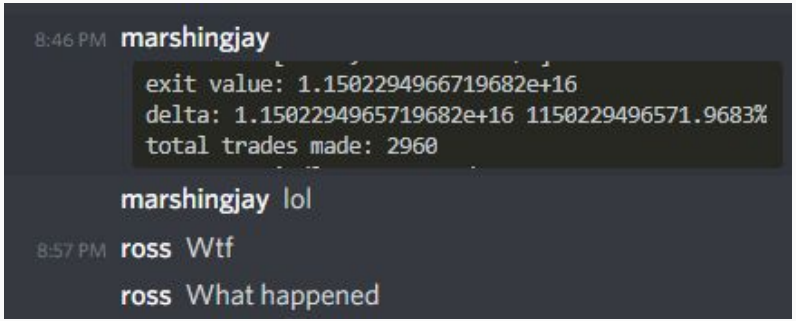
Strategies

Algorithmic Strategies

- Most of our work was on ML-centered strategies, but we did experiment with some straight-up algorithmic strategies as well
- For instance, we implemented the classic mean-reversion strategy
  - When price is x% below moving average, buy
  - When price is x% above the moving average, sell

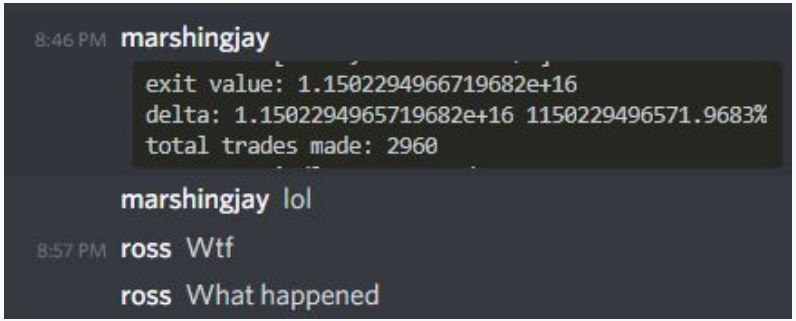
Mean-Reversion Algorithm

- We then used a genetic algorithm to optimize the algorithm parameters (% below to buy and % above to sell)
- We were met with some very positive results when testing the strategy on Bitcoin historical data



Mean-Reversion Algorithm

- We had run into the classic pitfall… overfitting!
- On any dataset other than Bitcoin (which we used to perform parameter optimization), the algorithm performed quite terribly



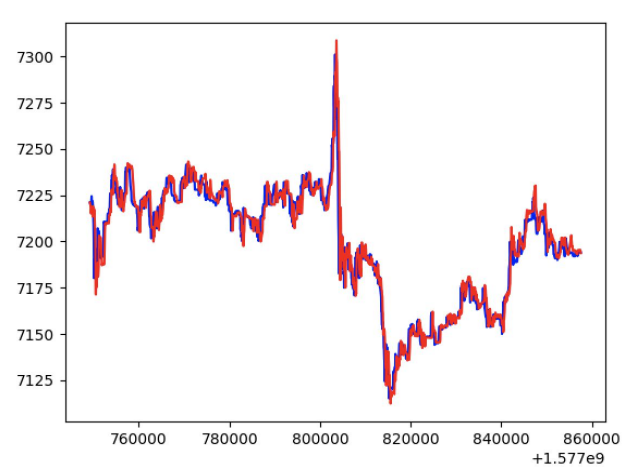
Machine Learning Strategies

- We found many different problems within this space that relate to topics we discussed in class
  - Regression
  - Classification
  - Anomaly Detection
- We will mainly focus on the Regression and Classification.
- Regression was used for price prediction
- Classification was used to classify optimal times to buy
- (The following graphs we will show depict time vs. the price of Bitcoin in USD)

Price Prediction

- Can either predict the exact price or the motion of the price (up or down)
- We decide to try to predict the exact price to provide more accurate insight into price movements
- Implementation
  - Regression
    - Linear Regression
    - Random Forest
    - XGBoost
    - Neural Network

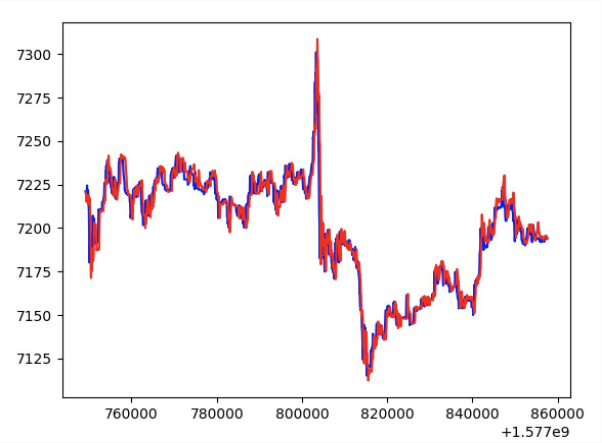
Price Prediction Results



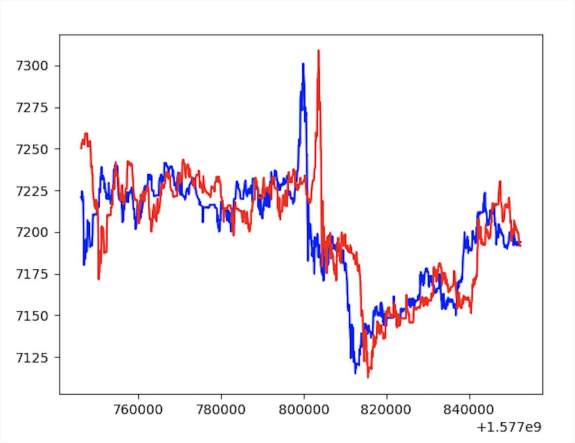
Attention we have solved price prediction. We are billionaires

jk lol

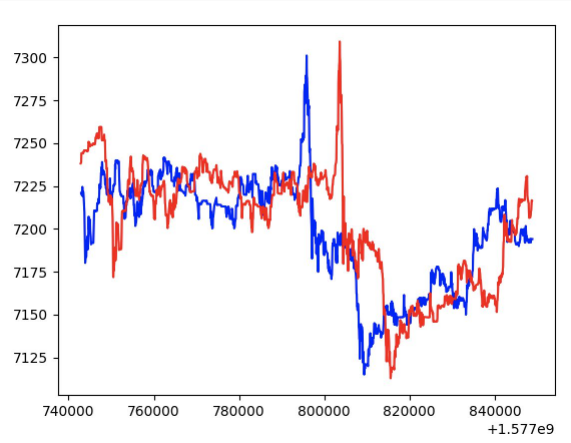
Price of Bitcoin



5 min



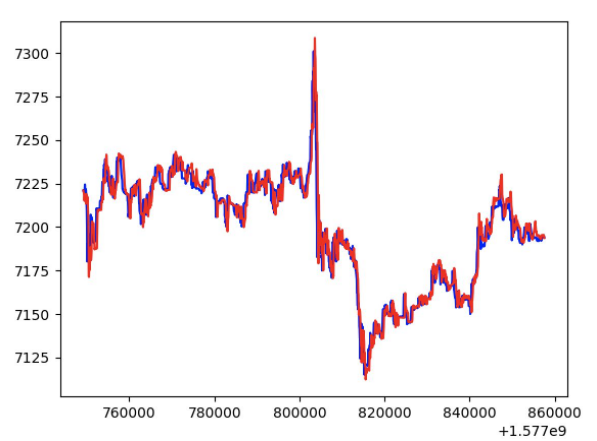
50 min

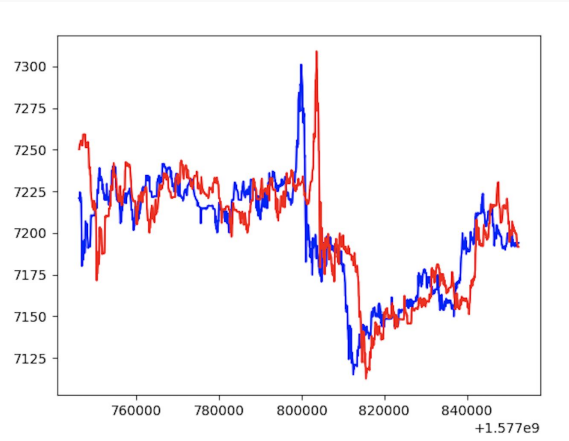


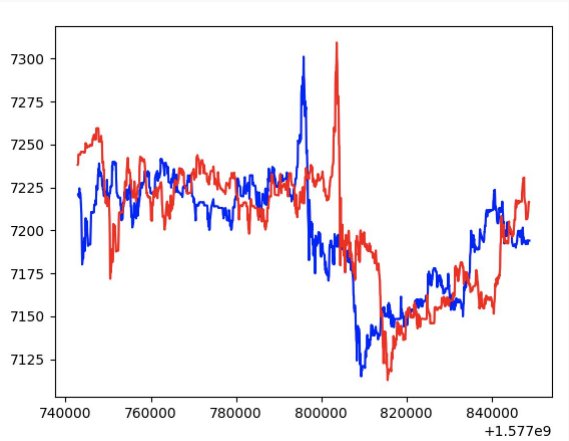
100 min

Red is the predicted and blue is the actual price

Models used







For this, we tested many different models, from linear regression to LSTM Neural Networks, and even using multiple models together.

All had this obstacle

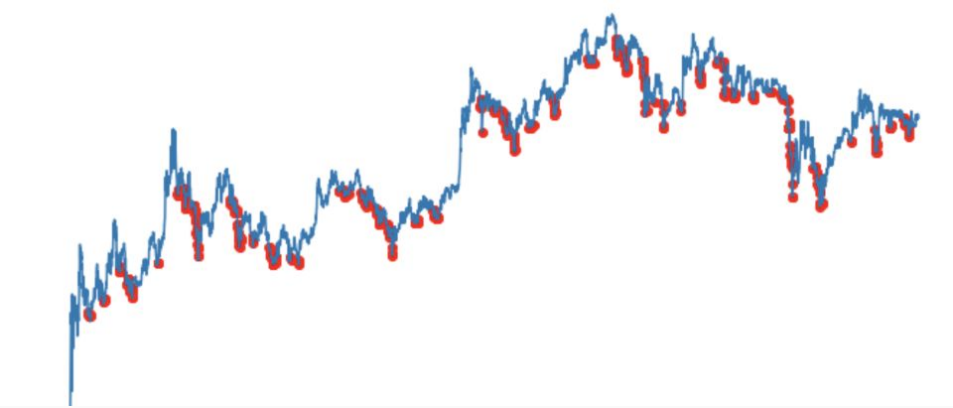
Why Price Prediction Doesn’t Work

- Future prices are somewhat related to the current price
  - So predicting a price similar to the current price usually results in high accuracy
  - Optimizing our model for accuracy results in the model just prices close to the current price, which while being most accurate, does not provide profitable insights
- So you can’t discern much from this.

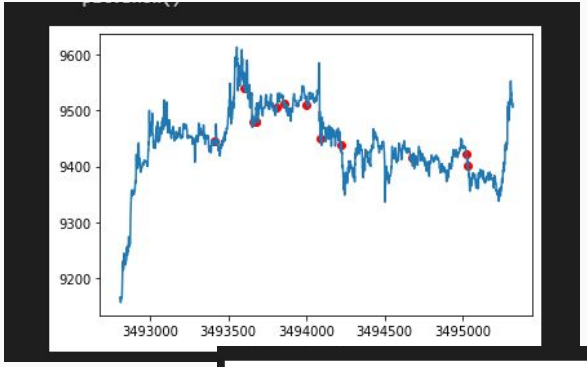
Entry Point Classification

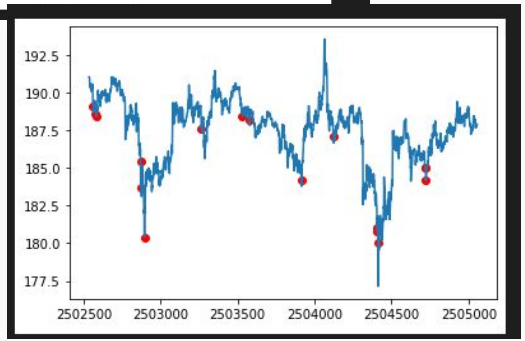
- The bulk of our work was focused on using classification to identify good potential entry points
- First, we needed to identify where these points existed within our training data
    - We began by using numpy to identify local minima, as these points will tend to correlate with opportune times to buy
    - Later, in an attempt to improve our model, we wrote an algorithm to calculate the most optimal points to enter a position and used this to generate our training data

Progression of Classification (Beginning)



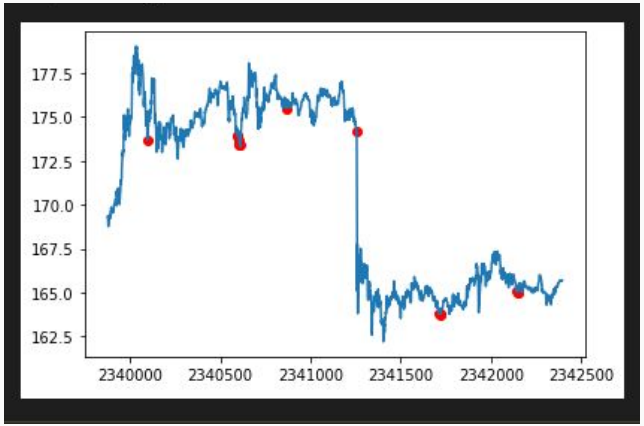
Simple Decision Tree





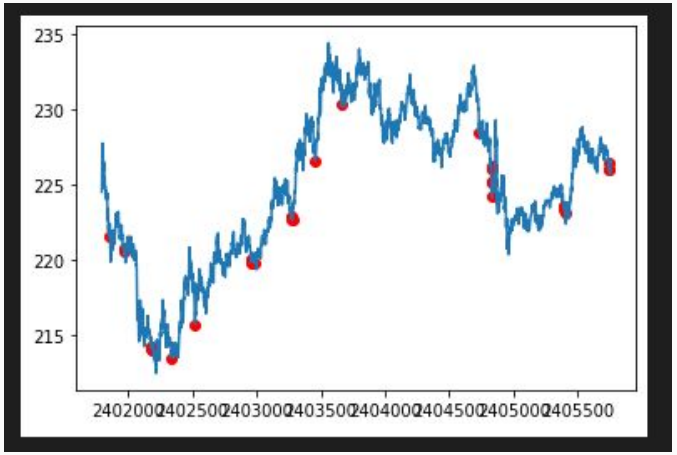
Random Forest

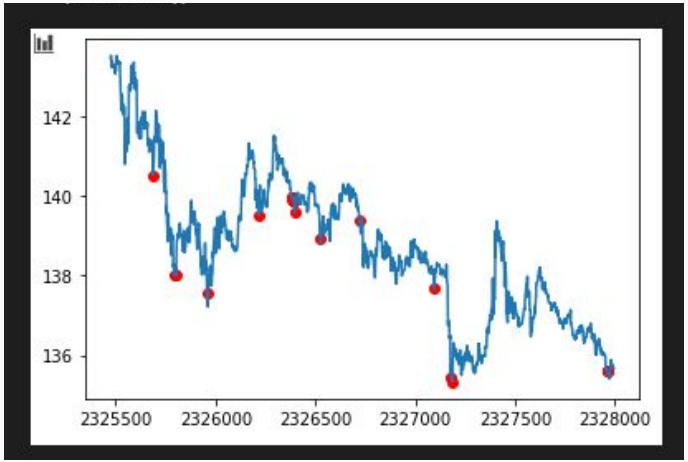
Progression of Classification (Middle)



XGBoost Classifier

Progression of Classification (End)





Model using multiple models:

- Random Forest
- XGBoost
- Linear Regression

Future Work

- We plan on continuing to work on the project
- Up until now we have mainly been researching and implementing different strategies with the skills we learned in this Data Science class
- Going forward, we can start fleshing out more profitable strategies, a real time trading system, and a robust portfolio manager

Questions?