Day Trading Simulator

Real Time Intelligent Systems: Final Project

By Maggie Wu, Sambhav Jain and Ana Ysasi

Group Members

Maggie Wu

Finance & Data Scientist

Sambhav Jain

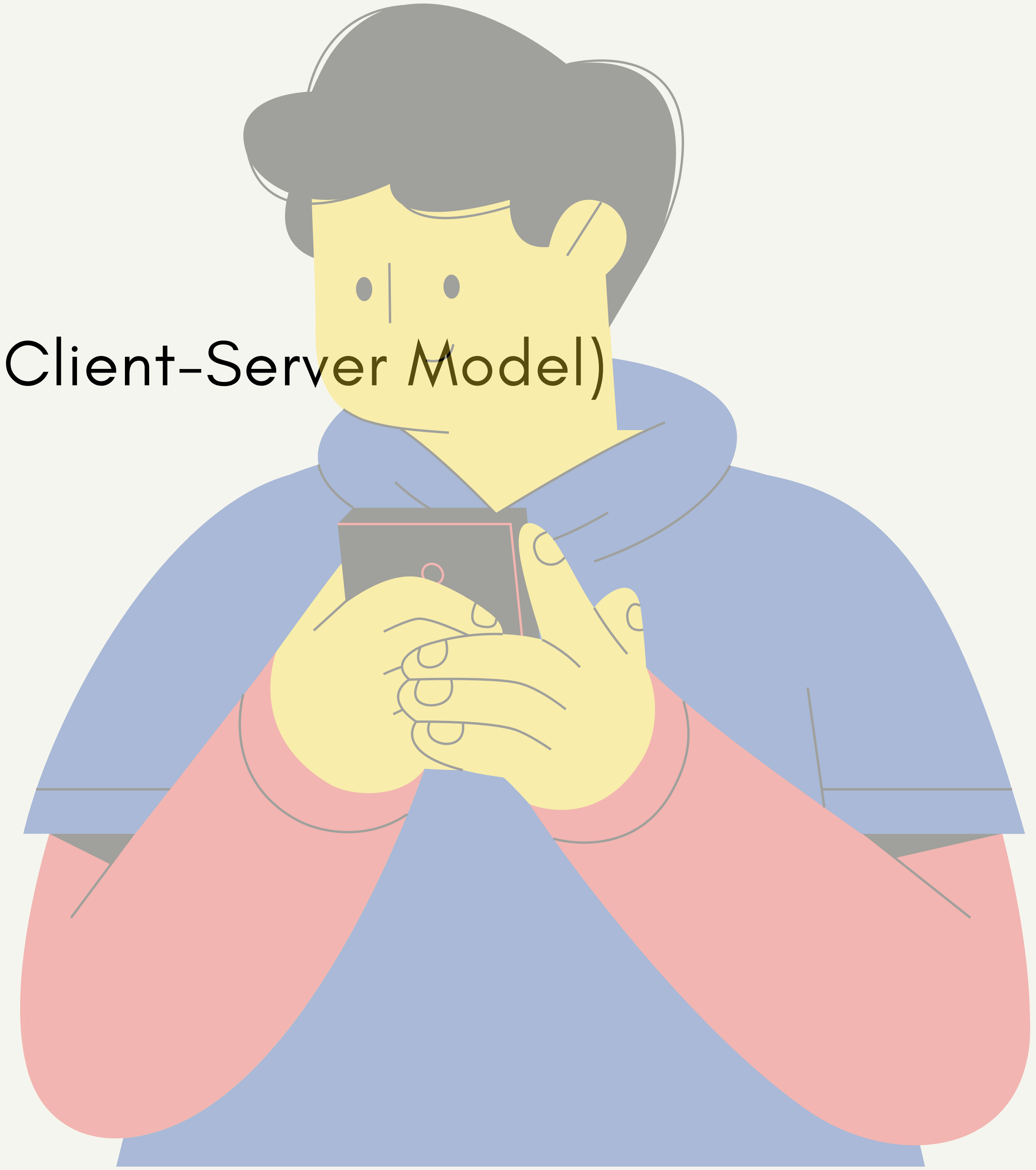
Engineer & Fantasy Sports Enthusiast

Ana Ysasi

Mathematician & Physicist

Agenda

- Background
- About Day Trading
- Real-Time System (Client-Server Model)
- Trading Strategy
- Results



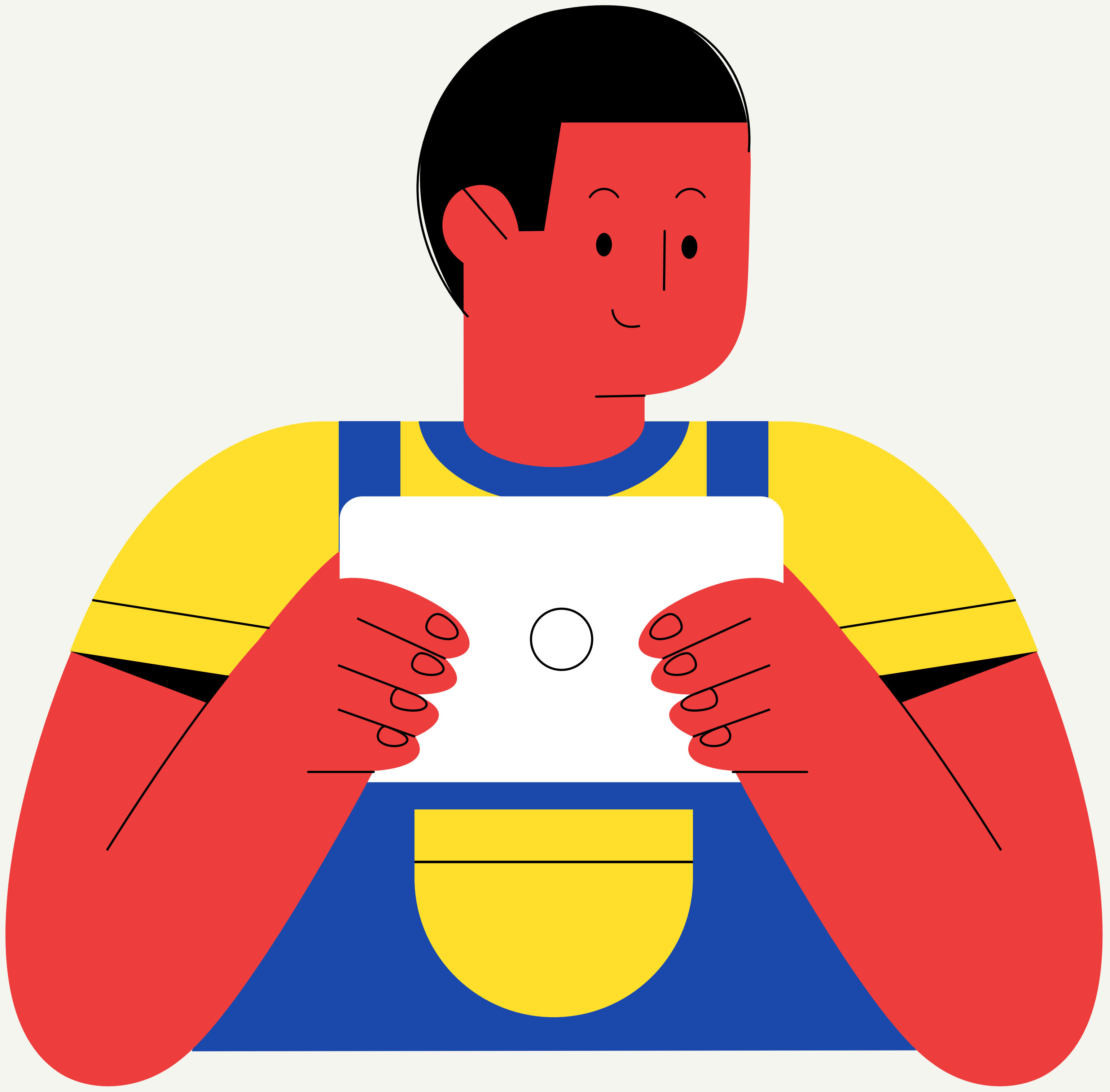
Background

Overview of the project

Relatively new to:

- real-time systems
- a successful trading strategy
- machine learning models implementation in trading

We wanted to challenge ourselves and get our hands dirty by stepping out of our comfort zones. In the last ~4 weeks we did exactly that and ended up building a simulation of a real time trading system that trades stocks by the minute. In the trailing **5 trading days**, we yielded a **1.42% HPR** with $247,500 initial investment and **beat the S&P500 by 68 bps**



Initiatives

What we want to solve

Create a trading system that involves no human input apart from the number of stocks the trader wants to trade on a given day

Hypothesis

By learning a stock's 'features' with historical price and customer behavior, every minute, we should be able to predict the right time to buy or sell a particular stock

Introduction

To Trading Strategies

Active trading is the act of buying and selling securities based on short-term movements to profit from the price movements on a short-term stock chart.

Active traders believe that short-term movements and capturing the market trend are where the profits are made.

There are various methods used to accomplish an active trading strategy, each with appropriate market environments and risks inherent in the strategy.

The most common strategies are: Day Trading, Position Trading, Swing Trading and Scalping.

Day Trading
Highlights

Highlight 1

Method of buying and selling securities **within the same day**. Positions are closed out within the same day they are taken.

Highlight 2

Day traders try to make money by exploiting minute price movements in **individual assets**.

Highlight 3

Day trading is often characterized by **technical analysis** and requires a high degree of self-discipline and objectivity.

Highlight 4

Some good skills to have to be a successful day trader are: Knowledge and Experience in the Marketplace, sufficient capital, strategy, and discipline

Client-Server Model
- In Simulating Trading

- The model helps enable real-world like trading simulation
- The server is coded to mimic the market and send stock information over a TCP connection at client's desired pace
- The Client is then able to harness the stock information using the previously established secure socket connection
- Processing can then be applied to the harnessed information to generate signals or indications to Buy, Sell, or Hold stocks.
- This model also enables users to test multiple trading strategies on simulated trading days which allows room for errors and fine-tuning and testing of trading strategies.

A Client receives data, harnesses it, and acts on it to produce appropriate signals. It typically doesn't share resources, but it continuously requests content from a server.

Our client collects data, uses our "brain" (trading strategy) and then makes an educated decision on trading the stock of choice.

LISTENS TO THE SERVER

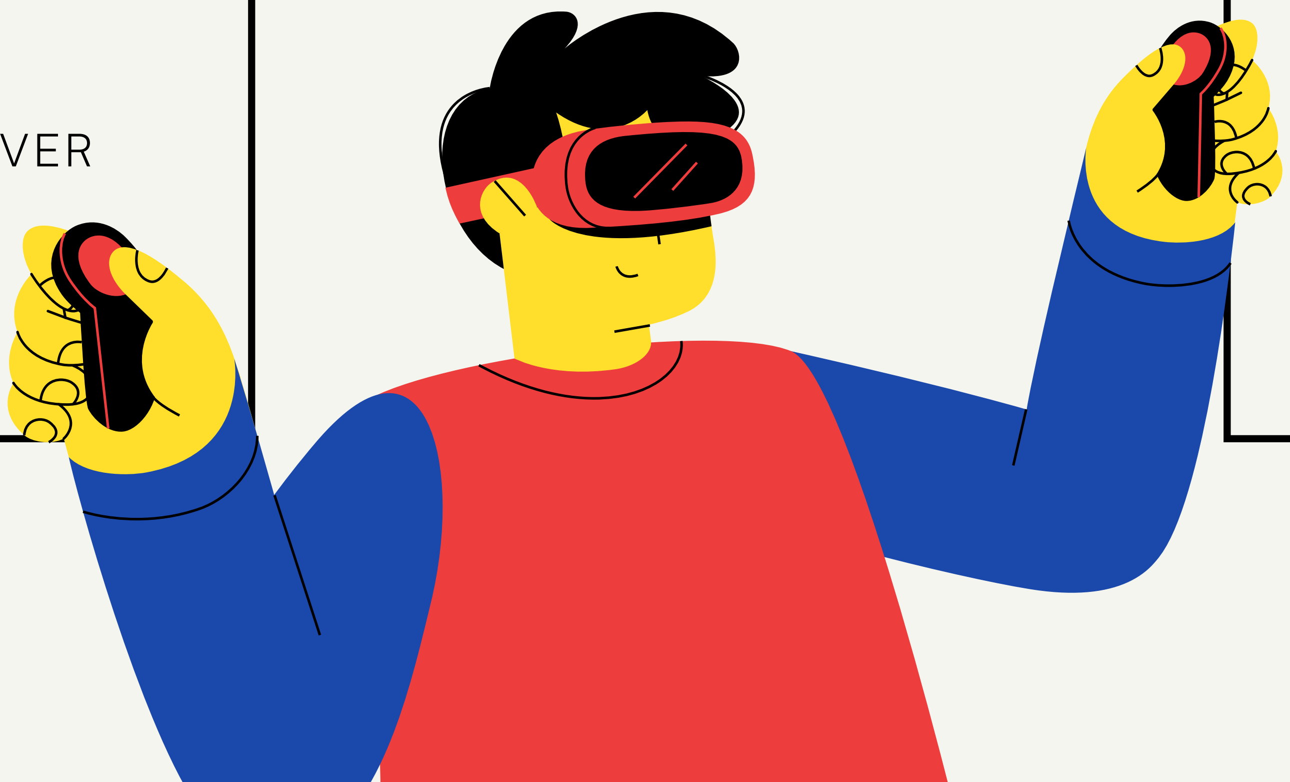
Client

A Server runs one or more programs that share resources with and distribute work and information among clients. It could be a database or web server.

In our trading system, the server sends out market data every second - for a number of stocks that is pre-determined.

SENDS STREAM TO CLIENT

Server



How Our System Works

Getting features

Using the base level information for every stock, every minute, we generate advanced features for each stock and append them to their respective lists

Fitting the model

No actions are taken for the first 40 periods (minutes) - the model accepts information and takes 40 periods to have enough data to successfully make accurate trading decisions

Actions

A BUY signal results in buying 10 shares of the stock, a HOLD action does nothing and a SELL action sells the current position of the stock - BUY & SELL actions are printed to the terminal

Features for the model

How to decide

We created a file that calculates a lot of indicators used to trade stocks.
But in order to fit it to the logistic regression we need only the most important features for this model.

We calculated the importance coefficient for the logistic regression model for all the stocks that can be trade (S&P 500) for an specific day and average the results over all of them.

We then selected the features with the highest scores.
The final selected features are: the awesome oscillator, the daily logarithm return, change, volatility, 10 minutes moving average and HBand/LBand indicators.

Features

An overview of some of the predictors we generated using the standard information from price updates

Change

It is the between the current value and the previous value and can be a negative or a positive value

Awesome Oscillator

It is an indicator used to measure market momentum by calculating the difference between a 34 and 5 period moving average

Moving Average

It smooths out price data by creating a constantly updated average price over a pre-determined period of time

Volatility

It is a measure of how prices or returns are scattered over time for a particular stock - helps identify areas of high potential

HBand Indicator

Returns 1 if a stock is higher than 2 standard deviations away from its 20 period moving average

LBand Indicator

Returns 1 if a stock is lower than 2 standard deviations away from its 20 period moving average

Trading Strategy : 1

Logistic Regression

Why

Since the stock market price movement is random, and only a BUY or SELL direction is ultimately what we care about for the purposes of our trading strategy, we chose binomial classification algorithm

How

With traditional financial factors/indicators (volume, MACD, RSI) as model features, our logistic regression model will generate the whether a stock goes up in price or down. We've also tried GNB, SVM, Random Forest, and other ensemble models, but after testing, a simple logistic regression beats all the others

Trading Strategy : 2

Oscillator & Bands Ensemble

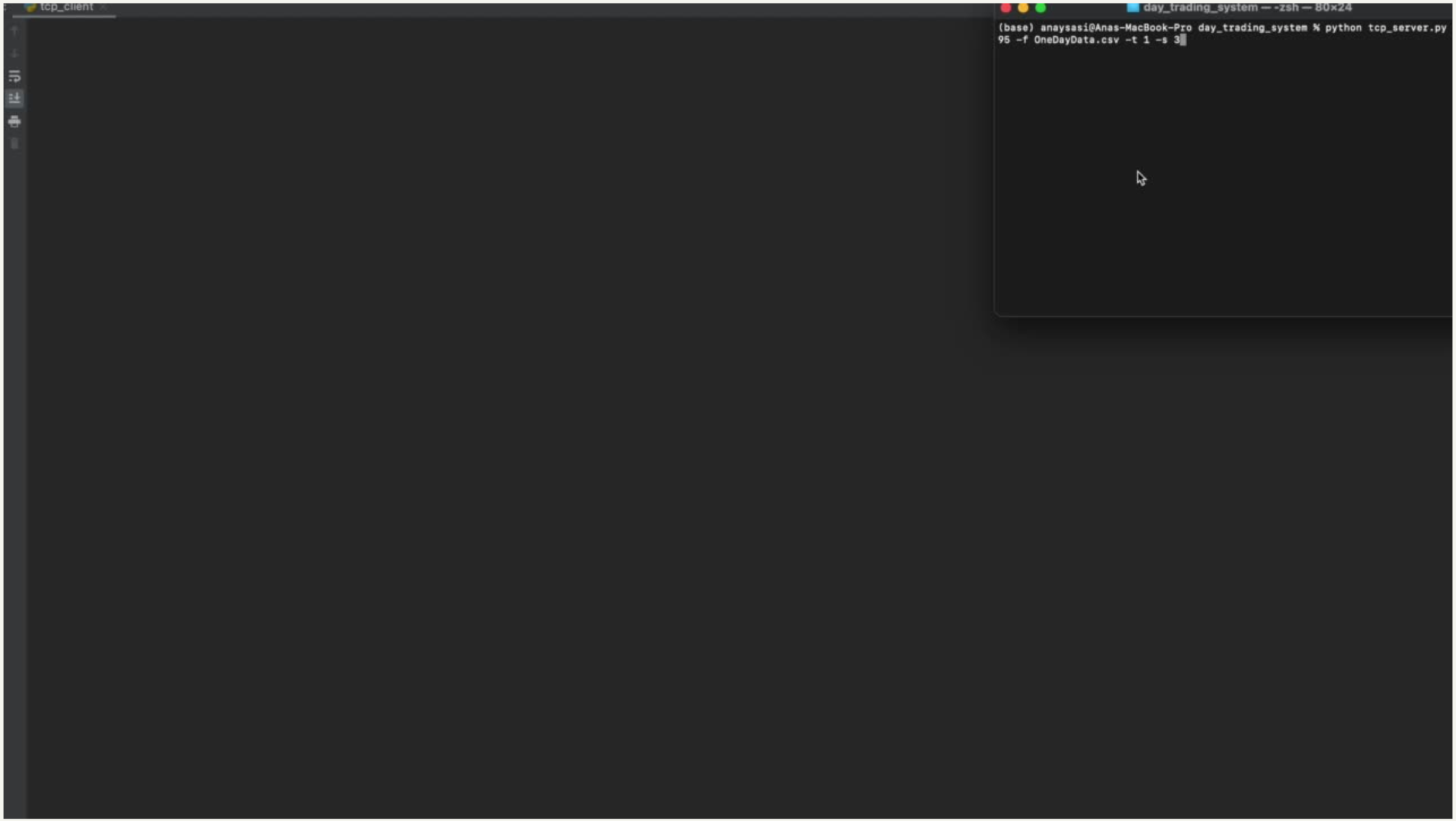
Why

Strategy 1 relied on fitting a new Logistic Regression model and using it to give us a BUY or SELL action every minute. If the user wanted to trade on a high number of stocks simultaneously, the model fitting was to slow to keep up with the stream data.

How

Using the Awesome Oscillator and Band Indicators in conjunction with one another, our model generated a BUY signal anytime the price of the stock returned a negative value on the Oscillator OR had a signal using the LBand Indicator and on the flip-side, generated a SELL signal anytime the Osciallator was positive OR had a signal using the HBand Indicator

Sample Output



Results

We ran the ensemble strategy on 100 random stocks from the S&P500 averaged a return of $702 per trading day with a total return of $3,510 on an initial investment of $247,500 which **nets out to a 1.42% return in a time when the S&P500 returned 0.74% - a 68 bps beat**

We plan on implementing this strategy to real life, real-time data.

```
/local/bin/python3.9 /Users/anaysasi/Documents/GitHub/day_trading_system/run_script.py
 downloading the data for day 2021-03-18
nade: 637.9568481445312
 downloading the data for day 2021-03-17
nade: 671.8923282623291
 downloading the data for day 2021-03-16
nade: 692.6617240905762
 downloading the data for day 2021-03-15
nade: 703.7693405151367
 downloading the data for day 2021-03-14
nade: 725.818042755127
 downloading the data for day 2021-03-13
nade: 740.254955291748
 downloading the data for day 2021-03-12
nade: 805.6529808044434

ess finished with exit code 0
```

SCAN ME

Thanks for listening!

If you would like to take a look at the code to try it yourself, give us feedback, or improve upon it, check out our GitHub!

Thank You

Maggie Wu
Sambhav Jain
Ana Ysasi

DAY TRADING SIMULATOR

Real Time Intelligent Systems

FINAL PROJECT

University of Chicago
Winter 2021