

# Portfolio Playground

CPSC 437/537

Chris Harshaw

Daniel Keller

Felipe Pires

December 3, 2016

# Motivation

We've **all** wanted to trade stocks

# Motivation

We've **all** wanted to trade stocks



# Motivation

We've **all** wanted to trade stocks



...but are afraid to lose large amounts of money



# Motivation

**Solution:** *Paper Trading* - simulated trading to practice buying and selling securities without actual money being involved.



# Motivation

**Solution:** *Paper Trading* - simulated trading to practice buying and selling securities without actual money being involved.



Enter **Portfolio Playground**, the premier paper-trading web application developed at Yale University!

# Outline

- 1 Main Functionality
- 2 Recommender Algorithms
- 3 Database Design
- 4 Front End Design

# Main Functionality

Portfolio Playground is a paper trading web-application with three main functionalities

- Portfolio creation and analysis
- Portfolio comparison
- Portfolio recommendation



# Main Functionality - Creation and Analysis

FRONT END PICTURE GOES HERE

Our portfolio creation supports a variety of features including

- Stocks pulled from over 37,000 US equities and mutual funds
- Large amounts of historical stock price data (19XX-2016)
- User inputs include number of shares purchased, portfolio creation date

# Main Functionality - Creation and Analysis

Suppose we have a portfolio  $P$  consisting of stocks  $P = \{s_1 \dots s_N\}$ , where  $x_i$  is the number of shares of stock  $s_i$ ,  $D_i$  is the dividends for stock  $s_i$ , and  $P_i^t$  is the price of a single share of stock  $s_i$  at time  $t$ . Then we can define,

Total Stock Return - (Weighted Percent Increase)

$$TSR = \sum_{i=1}^N x_i \left( \frac{P_i^{t_f} - P_i^{t_0} + D_i}{P_i^{t_0}} \right)$$

Diversity - (Weighted Correlation Coefficients)

$$Div = 1 - \frac{1}{Z} \sum_{i < j} x_i x_j Cor(P_i, P_j) \in [0, 1]$$

# Main Functionality - Comparison

FRONT END PICTURE GOES HERE

Our portfolio comparison supports a variety of features including

- Stock price, total stock return, and diversity comparisons
- Aesthetically pleasing visualizations

# Main Functionality - Recommendation

The most unique feature of Portfolio Playground is its state-of-the-art recommendation algorithms. This helps shape trading intuition for novice traders. The algorithms used are

- Random
- Highest Return
- Diverse Options

# Recommender Algorithms - Random

The Random algorithm recommends a random portfolio under a total budget constraint.

## Highest Return

- 1 Initialize portfolio  $P = \emptyset$ . Until budget constraints active,
  - 1  $P \leftarrow P + \text{random stock, random number of shares (under budget constraint)}$

# Recommender Algorithms - Random

The Random algorithm recommends a random portfolio under a total budget constraint.

## Highest Return

- 1 Initialize portfolio  $P = \emptyset$ . Until budget constraints active,
  - 1  $P \leftarrow P + \text{random stock, random number of shares (under budget constraint)}$

This can be used as a “control portfolio” and can also test the Efficient Market Hypothesis!

# Recommender Algorithms - Highest Return

The Highest Return algorithm recommends an optimal forecasted portfolio under budget constraints such as total portfolio budget and maximum investment per stock.

## Highest Return

- ① Fit a Vector Autoregression Model to historical stock data
- ② Forecast the stock prices  $d$  days away
- ③ Initialize portfolio  $P = \emptyset$ . Until budget constraints active,
  - ①  $P \leftarrow P + \text{stock that maximizes TSR}$

# Recommender Algorithms - Diverse Options

The Diverse Options algorithm recommends an optimal portfolio under budget constraints and *diversity* or *correlation* constraints.

## Diverse Options

- ① Fit a Vector Autoregression Model to historical stock data
- ② Forecast the stock prices  $d$  days away
- ③ Initialize portfolio  $P = \emptyset$ . Until budget constraints active,
  - ①  $A = \{s | \text{corr}(s, x) < \sigma \ \forall x \in P\}$  (options diverse from  $P$ )
  - ②  $P \leftarrow P + \text{stock from } A \text{ that maximizes TSR}$



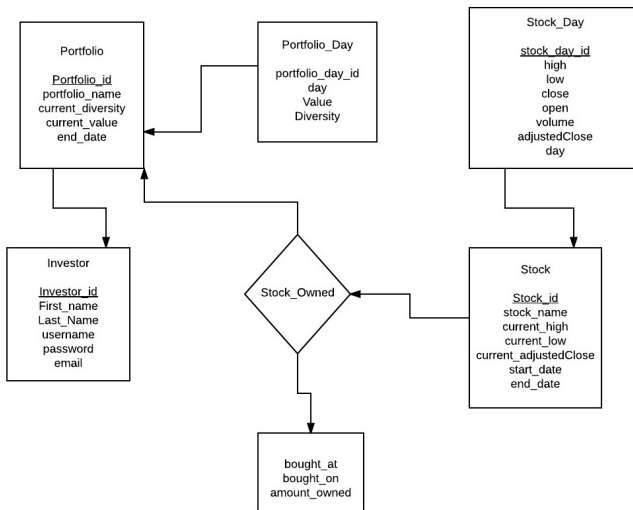
# Database Design

What need to store

- user information
- portfolio information
- historical stock information

We are using Django DB and calling Tiingo API.

# Database Design



# Front End Design

What are the design decisions?

# Front End Design

What tools did we use to achieve this?

# Questions

Questions?