

## Executive Summary

Provided with 756 days of stock data, we were challenged to build a trading algorithm that can return the maximum amount of profit after 252 of trading. We approached the problem by first researching technical indicators and turning them into features in our data set that can potentially increase our profits. Then, we used the data and the engineered features to create both regression and classification models. A linear regression model with different parameters is built for each stock and a different classification method is also chosen for each stock to minimize error and maximize profit. Lastly, we created a trading strategy that combined the two models with rules for buying and selling.

On day 756, we sold all our stock holding and received $1,058,840 in cash, which amounts to a 5.89% profit on our initial investment. We generated a profit on 135 out of the 252 days, and our Sharpe ratio is around 0.3755.

## Problem Description

Algorithmic trading has risen in popularity over the past decades as technological advancements increasingly enable machines to trade stocks at a higher frequency, volume and profitability than what the human trader is capable of. These algorithms used for trading are essentially prediction models that incorporate machine learning techniques. Past data is used as input to train the models to predict stock trends in the near future.

For the purpose of this project, we are provided with 756 days of data on 50 stocks included in the SSE 50 index and challenged to build an algorithmic trading algorithm that maximizes the return on an initial investment of $1,000,000 that we are provided with on day 504. Beginning from day 505, we are allowed to engage in 252 days of algorithmic trading and all stocks must be sold on day 756. The final algorithm aims to achieve a high profit margin along with good common sense metrics in trading that appeals to investors’ risk-averse psychology.

## Data Description

### 3.1 Data Overview

The raw data provided consists of the daily stock data of 50 companies from days 1 to 756. For each day, each stock’s open, high, low, close prices and the stock’s trading volume are provided. In addition, 10 engineered features were also provided, which includes standard features commonly used in the technical analysis of the stock market such as moving average, n-day returns and stochastic oscillator. These features are meant to provide insight on stock price movements that can help predict the overall trend of stock prices in the short-term.

**3.2 Engineered Features**

To maximize the accuracy and outcome of our trading algorithm, we’ve also engineered our own features based on technical indicators we researched online. There are 4 major categories of technical indicators: trend indicators, momentum indicators, volatility indicators and volume indicators. As indicators from each category provide insights on different aspects of stock price movements, we selected 2-3 measures for each category to create an additional 13 features to be used in our models.

**Trend Indicators**

Trend indicators use averaged prices as baseline to measure the direction and strength of trends in periods typically ranging from 1 to 15 days. The trend indicators we engineered include Open-close Ratio, Exponential Moving Average (EMA) 5 Day and Exponential Moving Average (EMA) 15 Days.

**Momentum Indicators**

Momentum indicators use current and previous closing prices to identify the speed of price movements overtime. A divergence between predicted price and a high momentum indicator signals an expected change in future prices. The momentum indicators we engineered include Commodity Channel Index (CCI) 5 Day, Commodity Channel Index (CCI) 15 Day and Rate of Change (ROC).

**Volatility Indicators**

Volatility indicators use the highs and lows of historical prices to measure the rate of price movement overtime. These indicators typically provide useful insights on the range of buying and selling to assist traders make more informed trading decisions. The volatility indicators we engineered include Bollinger Bands and High-Low Percentage.

**Volume Indicators**

Volume indicators use averaged raw trading volumes as baseline to measure the strength of a price trend. A sizable change in volume is often positively correlated with a movement in price in the same direction. The volume indicators we engineered include Ease of Movement (EVM), Log Volume and Force Index.

## Statistical Learning Approaches

### 4.1 Regression Model

After constructing our engineered features, our team proceeded to build models to make predictions on each stock’s price movements. Our team selected multiple linear regression with least square method as our starting point.

**Response Variable**

To achieve a better outcome from our multiple linear regression, we needed to first determine our objective for prediction, which is also the dependent variable (y) in the model. After thorough discussion, our team decided to predict the percentage change of next day (Day t+1) based upon the observations from Day t.

**Features Selection**

Feature selections is a crucial process in terms of improving the performance of the model. In order to select the optimal subset of variables from 23 features in the full model, our team decided to apply the backward stepwise selection approach and used adjusted R² to compare models with different numbers of variables and eventually we choose the group of variables with highest adjusted R².

**Model Training**

After we decided the group of variables with best performance, our team used all data, which includes both training and validation data, to train the regression model before making our final predictions. To be more conservative, we took the average of the predictive percentage change of the past 5 days as the percentage change of each day since Day 505.

### 4.2 Classification

After we built an initial regression model for each stock, we decided to also try creating classification models for each stock to compare the results for different model types. The whole process of building a classification model for each stock is more complex than building the regression model, but the underlying concepts are the same.

**Response Variable**

Just like our regression model, we must first build a response variable before we proceed to building our classification model. In our classification model, we want to predict the short-term future trend of a certain stock. To narrow down our objective, we decided it was important to predict whether a stock’s price will go up in the next 5-day period.

We built a new feature that calculates the highest percentage change of a certain stock in the next 5 days. We used the highest closing price in the next 5 days divided by the current day’s closing price and then subtracted it by 1. Then, we set a threshold to classify each record based on this new feature. If the percentage change is above the threshold, we labeled the record as 1; otherwise, we labeled it as 0. We ran a logistic regression model on each stock and tried different percentage values for the threshold, from 0% to 5% with a 0.5% interval. Finally, we decided to use 2% as our threshold since it has the lowest average overall false rate among all the stocks.

**Model and Features Selection**

Initially, we chose KNN as our classification model as it returned the lowest average overall alse rate among all the stocks. The results returned on the test data was also satisfactory. However, we later realized that each stock might have different patterns, and that some could be linear and others are non-linear. A single model, which in this case is KNN, is probably not the perfect classifier for each of the 50 stocks. Since there are many readily available off-the-shelf classification models, we decided to try to identify the best classifier for each individual stock.

We had 7 classification model candidates: logistic regression, KNN, SVM, decision tree, random forest, neural net and naive bayes. We fed in the data with all 23 features to test every model candidate, and then the model with the lowest overall false rate is chosen for each stock.

After we had chosen a model for each stock, we started working on feature selection. Similar to model selection, we performed backward stepwise feature selection for each model and found the best number of features on each stock’s level instead of an overall level.

**Model Training**

Unlike our regression model, where we only trained the model once, our classification models were trained repeatedly. We wanted to be able to capture recent trends in each stock, therefore instead of using all past data, we chose to use only data up to 100 days before the predicted 20-day window to train the model. After each 20-day window, we trained a new model and performed the same process again for another 20 days. For example, we trained the model on Day 400 ~ Day 499, to predict prices for Day 505 ~ Day 524. And then, we trained the model again on Day 420 ~ Day 519 to predict prices for Day 525 ~ Day 544. Note that we did not use data from Day 500 ~ Day 504 or Day 520 ~ Day 524 to train the model as the classification label we built is based on the next 5-day data.

### 4.3 Combining Regression & Classification Models

As we are allowed to trade from Day 505 ~ Day 756, we need to test and optimize our trading rules based on our test data. We decided to combine our regression model and classification model, because we believe by ensembling these two weak models, we can potentially gain more precise predictions. As a result, our trading rules will be based on the combined results of the two models. This means, we will purchase more stocks when both the regression model and classification model predict a positive outlook, and we will sell our stocks when either model predicts a negative outlook. In this project, we considered Day 400 ~ Day 499 as our test data (the reason why we are not using data from Day 500 ~ Day 504 is explained in 4.2), and we finalized the parameter of our trading rules on the test data.

## Final Trading Strategy

Now that the predictions for each of the stock prices were obtained from both the linear regression model and classification model based on the given and newly engineered features, the team had to decide how exactly trading decisions will be made from Day 505 to Day 756.

Given the unpredictable nature of the stock market and the two models that are not yet proven, the team decided to take an approach that involves both conservative and aggressive strategies.

**Conservative Approach**

1. Divide the initial capital of $1M equally among 50 stocks ($20,000 each)
   * Prevents the possibility of few stocks’ poor performances leading to a huge drop in the overall portfolio value
2. Purchase stocks only when **both** linear and classification models agree to purchase.
   * Makes sure we have enough confidence that the purchased stock price will increase
   * Further details on the purchase threshold will be discussed below
3. Sell stocks when even one of the models indicates negative growth.
   * The team thought it would be safer to pay the transaction fee than the potential drop in stock prices as predicted by our model
   * Further details on the sale threshold will be discussed below
4. Create additional engineered features as described above.

**Aggressive Approach**

1. Purchase as many stocks as possible within the initially assigned limit of $20,000 when the decision to buy the stock is made.
   * Making as much profit as possible when expecting a growth
2. Sell all stocks in possession when the decision to sell the particular stock is made.
   * Liquidating everything on hand in order to prevent loss when expecting a loss

The unique characteristic of our trading strategy is that it uses the output of both models. During the initial model building process, the team concluded that we would not be comfortable relying only on a single model. As a result, we decided to implement both models that will work together as verification of each other’s results and increase the accuracy of the prediction.

Then, as mentioned above, the team needed to convert the outputs of the two models to the decision to purchase, sell, or do nothing by setting a threshold.

**Purchase Threshold**

|  |  |  |
| --- | --- | --- |
| Classification label predicts an increase in stock price | **AND** | Linear regression predicts growth of at least 5.5% in stock price |

Note that the threshold of 5.5% is not just a random number but it is the threshold that gave us the highest return rate on our test data of 5.89%.

**Sale Threshold**

|  |  |  |
| --- | --- | --- |
| Classification label predicts NO increase in stock price | **OR** | Linear regression predicts growth of 0% or less in stock price |

For a better understanding of the trading strategy, please refer to the table below that shows how the trading decision is made.

|  |  |  |  |
| --- | --- | --- | --- |
| **Day** | **LR Prediction (Expected % Change)** | **Classification Prediction**  **(1 : Stock price will go up**  **2: Stock price will not go up)** | **Trading**  **Decision** |
| 600 | + 8.3% | 0 | Nothing |
| 601 | + 3.1%  (Positive but not as high as the threshold) | 1 | Nothing |
| 602 | + 5.7% | 1 | Buy |
| 603 | - 2.3% | 1 | Sell |
| 604 | + 7.5% | 0 | Sell |
| 605 | - 1.0% | 0 | Sell |
| : | : | : | : |

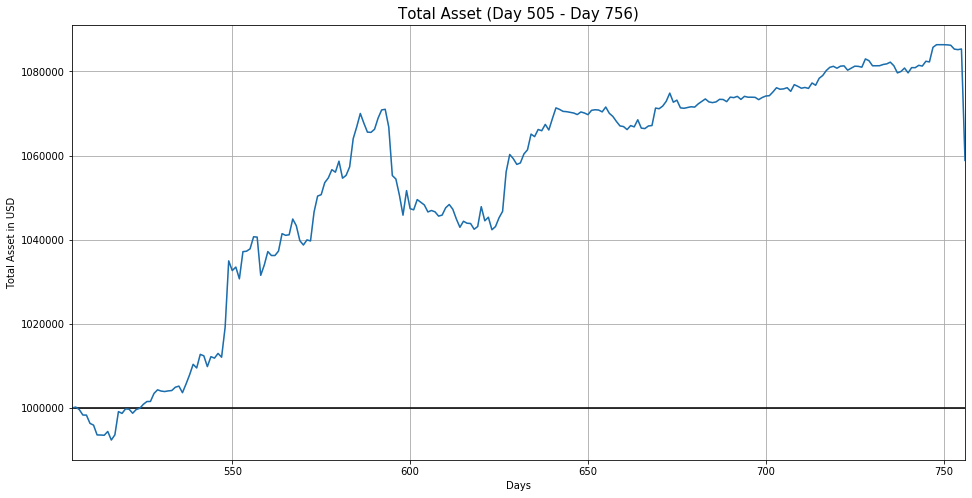
## 

## Results

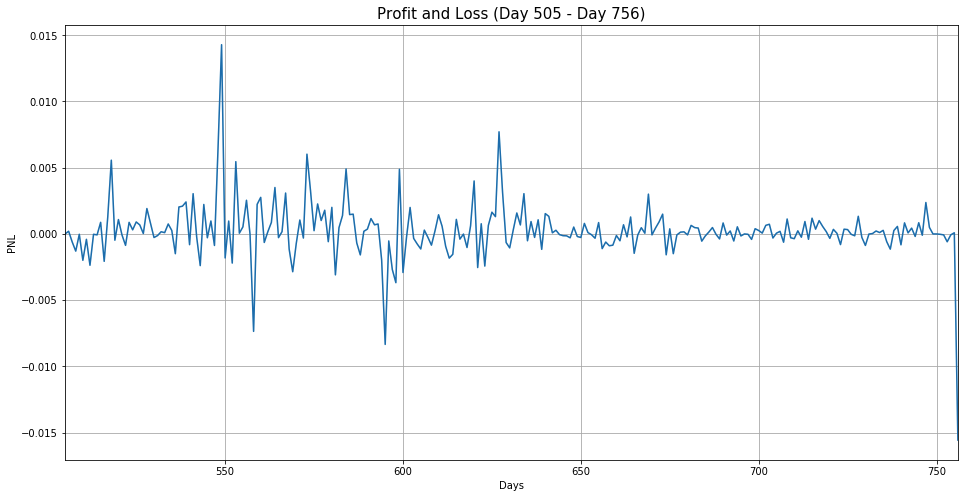
On day 756, all of the stocks in possession were sold to calculate the amount of cash gained over the trading periods. As a result,

* **$1,058,840** was the ending balance of the trading
  + 5.89% Profit
* Sharpe Ratio: **0.3755**
* Number of Days a profit was generated: **135 days**

The graphs below also demonstrate the performances during the trading period. The first graph shows how the total asset has changed over the days. Despite the dip around Day 600, the graph shows the overall gain throughout the period.

****

The second graph shows the profit and loss for each of the days. Although the fluctuations are inevitable due to the unpredictability of the stock market, it also shows that our model still has room for improvement.

****

**References**

https://www.visualcapitalist.com/12-types-technical-indicators-stocks/

https://blog.quantinsti.com/build-technical-indicators-in-python/

https://towardsdatascience.com/in-12-minutes-stocks-analysis-with-pandas-and-scikit-learn-a8d8a7b50ee7

https://medium.com/analytics-vidhya/stock-trend-prediction-with-technical-indicators-feature-engineering-and-python-code-1fa54d5806ba