December 10, 2022
Ai Hua Li
Saiful Islam
Ziyi Huang

Buy Low, Sell High

Motivation

In this day in age, people are always trying to figure out how to invest their money and achieve a high return on investment. One method of investing people usually turn to is day-trading. Day traders in the stock market sometimes buy and sell a stock within the same trading day with the goal of receiving profit from the trade by capitalizing off short-term changes in price.

Day traders have various methods of identifying when to buy and sell a stock as they explore different factors that they think may influence or indicate trends for the price of a stock. We want to explore whether time could be one of these factors. Specifically, what would be the best hour intervals to buy (start of the interval) and sell (end of the interval) shares of an SPY stock within the same trading day, based on the changes in the stock price in recent history.

By solving this problem, we can identify another strategy that day traders can potentially use to plan their trades and have greater chances of success in receiving profit. Hence, we built classification models that predict whether or not an hourly interval would yield a profit, if one were to buy a share of an SPY stock at the beginning of the interval and sell that share at the end of the interval. The model could then be used for each hour interval in a trading day to see if there is profit in that interval.

Data

We access the data by calling a Polygon API that allows us to have access to 2 years of historical data for stocks (all we needed was 360 days of information for the SPY stock). We choose this API because it allows us to specify the parameters we want, such as stock symbol ticker, intraday time interval, start time, and end time, so that it can give us the standard stock intraday dataset we needed. The data we used is historical intraday stock data in hour intervals in the last 360 days. There are six attributes in this dataset: Volume, Open, High, Low, Close, and Datetime.

Below, you will find the data dictionary used in our model.

Column	Data Type	Description	Example
Volume	int	The amount of shares or contracts	115539

		traded in an asset or security over an interval	
Open	float	The price at which a security first trades upon the opening of an exchange on a trading interval	3899
High	float	The highest intraday price of a stock in the most recent (or current) trading interval	1219.60
Low	float	The lowest intraday price of a stock in the most recent (or current) trading interval	1205.00
Close	float	The price at which a security trades upon the closing of an exchange on a trading interval	1211.45
Datetime	date	Current Datetime in yyyy-mm-dd-hh	2022-08-25 10:00:00
Hour	int	Current hour (10 am to 16 pm)	10
Profit	float	Calculated by Close - Open	0.4500
Total Profit in Last N Days	float	Total profit of last N days for that hour interval	6.8776
Profit Days in Last N Days	int	Number of days profit is made within that hour interval for last N days	15
Is_Profit	int	If profit is greater than or equal to zero,	1

		it's 1; if profit is less than zero, it's 0	

One limitation in using this API was that it did not provide extensive stock information that could potentially introduce more features; very basic info was provided by the API that we had to get creative with.

Models

What did you build? Why?

Since our problem of identifying whether or not an interval yields profit mirrors a classification problem, it was appropriate for us to use classification models. Hence, we experimented with three classification models to see which performs best: Logistic Regression, Decision Tree, and Random Forest.

Initially, we didn't like the accuracy of our Logistic Regression model so we did some model debugging by working with different kinds of models. Also, we added more data later as we saw the model performed better when we introduced more hourly intervals into the dataset.

In order to train our models, we split our data with a 80-20 split for training and testing (20% of the data was dedicated towards testing), as this split made sense.

Features

We had four features in our model: Open, Hour, Total Profit in Last N Days, and Profit Days in Last N Days, where each row represents an hourly interval from the past 360 days. We utilized MinMaxScaler from scikit-learn to standardize the features.

Open

Open represents the opening price of the SPY stock at an hourly interval. This feature was already given by the API.

Hour

Hour represents the hourly interval of the row from a given day, which ranges from 9 to 16. 9 indicates the hourly interval from 9AM to 10AM, 10 indicates the hourly interval from 10AM to 11AM, 16 indicates the hourly interval from 4PM to 5PM, etc. This feature was computed from the Timestamp column provided by the API.

December 10, 2022
Ai Hua Li
Saiful Islam
Ziyi Huang

Total Profit in Last N Days

Total Profit in Last N Days represents the profit made within that hourly interval for the past N days (takes data from N rows), and is one of our important features. The value of N we used was 30, as we saw it worked best. This feature was computed from the Open, Close, and Timestamp columns provided by the API.

Profit Days in Last N Days

Profit Days in Last N Days represents the number of days there were profit within that hourly interval for the past N days (takes data from N rows), and is one of our important features. The value of N we used was 30, as we saw it worked best. This feature was computed from the Open, Close, and Timestamp columns provided by the API.

Label

Our label was Is_Profit which indicates if there was a profit for the hourly interval (row).

Evaluation

Before reading this section, please note that the numbers here may vary as the model pulls in the latest data when it is run.

Out of the three models, our Random Forest classifier worked best with a testing accuracy of 0.65, while our Logistic Regression and Decision Tree models provided testing accuracies of roughly 0.52. When testing our models with the training data (to see if there was overfitting), the accuracies were low for the Logistic Regression and Decision Tree models (0.56 and 0.63 respectively), which is not favorable. The Random Forest classifier provided a better accuracy of 0.86 when tested with the training data. We also evaluated all our models with a cross fold validation of 5 folds. Overall, our models were not overfitted to the training data, and our Random Forest classifier worked best.

It should be noted that our Random Forest classifier took more training time than the other two classifiers. More specifically, it required 0.15 seconds while the other two models required less than half a second. Hence, our Random Forest classifier required 30x more time to train than the other two models, so the better accuracies from this classifier came with a cost.

We don’t feel confident about the performance of our Random Forest classifier model despite it being our best-performing model. It only has an accuracy of 0.65 with the test data, so more work can be put into the future by reevaluating our features and possibly introducing new ones.

December 10, 2022
Ai Hua Li
Saiful Islam
Ziyi Huang

Future Work

In terms of work for the future, we can improve the model further. For one, we used hourly intervals as it was easy to work with. However, we could have experimented with different lengths for intervals to determine the optimal length. For instance, maybe buying in and getting out in shorter intervals of 5 minutes can yield higher returns.

Also, being that we are not financial experts and have very limited knowledge on the stock market, there are potential features out there that we are not well-acquainted with which we can explore in the future. To mention a few, some potential features are AUM, P/E ratio, and 30-Day yield, which are common stock market terms. However, the dataset we used made it almost impossible to explore these potential features. As mentioned earlier, very basic info was provided by the API, which we had to get creative with. Sourcing different datasets would be a great next step to take to find more relevant features.

Hence, one interesting question we uncovered while working on our model was if having new features like the ones mentioned would benefit our model.

In the future, we can also compare our current model with a Naive Bayes Classifier model.