Classifying Stocks to Buy for Short-Term Gains using Machine Learning and Data Mining Techniques

by

Arick Althoff

Final Report

DATA 240

1. Introduction

1.1 Background

The creation of Robinhood brought the stock market to millions of new users, but many are inexperienced, not knowledgeable, and are likely to incur losses. Some are enticed by the ability to invest for their future, some enjoy the gambling nature, and others simply enjoy the convenience of the stock market on their phone. Robinhood gives its users countless warnings on the dangers of losses, but does not release data regarding its user's overall performance on the app (Popper, 2020). Although, it is stipulated that over 75% of retail investors incur losses during their investing lifetime.

1.2 Purpose

While there are many types of investors, this project attempts to appeal to a specific type. Whether they admit to it or not, many Robinhood traders fall under the category of "swing traders". These traders seek out investments that may earn them short term gains on their investments (Mitchell, 2021). The difference between the retail traders on Robinhood and professional swing traders is that swing traders rely on a heavy background of analytics, experience, and overall stock market knowledge. This project aims to build a predictive model to give the power of data analytics to those with less knowledge or access to tools for the stock market.

1.3 Goal

The goal of this project is to build a predictive model using three types of classification models. Unlike many other predictive models that attempt to predict price increases and decreases, this model will classify a stock as a "buy" or "not buy". For a stock to be a "buy" stock, it will need to have had a modifiable percent gain over a modifiable number of days. For example, the model can be changed to predict whether a stock at a certain point in time would realize a 15% gain in 30 days. This way, users would be able to set up alerts on their phone for convenient trading. If high classification accuracy is achieved, its users will be able to detect at which points in time a stock would flash buy signals and realize those gains in a short period of time. Opposed to other methods, this would require very little operational knowledge of the stock market and the dense analytical tools that professionals utilize.

2. Literature Review

Predicting stock price movements is a thoroughly researched topic and there have been many attempts in the past. The first discussed attempt was done by Patel et al. who predicted the direction of movement of a stock and price index using four prediction models. The data consisted of two stocks and two stock indices that spanned ten years. The four models included Support Vector Machines, Artificial Neural Networks, Random Forest, and Naive Bayes with a multitude of parameters for each. The features they used were common stock indicators like moving averages, MACD, and RSI. While working with continuous data, random forest models

performed at a 83.56% accuracy. However, after converting features into “trend deterministic” data by changing values into binary -1 and 1 values, accuracy increased to a range of 86.69% to 90.19% for each model with Naive Bayes making the best predictions (Patel et al., 2014, p. 259–268).

Another attempt made by Zhang et al., utilizes a two stage method. First, they used indicators again to create a support vector regression. Then, they implemented a type of multilayered neural network called ensemble method called ensemble adaptive neuro fuzzy inference system (En-ANFIS). En-ANFIS is a technique used in predicting time series data. During feature analysis they had found that the 5 and 10 day moving averages and the 12 day exponential moving average were most important in the model. While they did not compare results against other attempts, they concluded that the two stage method had performed better than either of the stages alone (Zhang et al., 2021, p. 1237–1261).

3. Data Collection and Preparation

3.1 Data Source

Yahoo Finance is a great repository for data collection of historical prices. Specific stocks may be selected for download at any date range desired by the user. The drawback to this method is that data collection on multiple stocks may be slower than other methods. Because of this, the selection of stocks to be used for analysis should be deliberate.

For the purposes of the project, only a single stock is selected to model for and create predictions on. The reasoning behind this is attached to the uniqueness of the approach. Every single stock behaves and performs differently in the stock market, so every model should be tuned around that knowledge. For demonstration purposes, the Tesla (TSLA) stock was selected for its volatile nature, duration on the stock market, and popularity. The entirety of the report is focused around this stock, but any stock data may be used which may generate a different feature set, best performing model, and outcome than the Tesla stock.

As such, the data being modelled for demonstrative purposes has 2,851 days worth of data that ranges from 06/29/2010 to 10/22/2022. Figure 3.1 illustrates a sample of the first three days of the raw data.

Figure 3.1

	Date	Open	High	Low	Close	Adj Close	Volume
0	2010-06-29	3.800	5.000	3.508	4.778	4.778	93831500
1	2010-06-30	5.158	6.084	4.660	4.766	4.766	85935500
2	2010-07-01	5.000	5.184	4.054	4.392	4.392	41094000

3.2 Features

Stocks regularly have a small amount of features. This data collection method has only a singular important feature, price. Without generated features, a single data point would include,

stock name, date, and various prices on that date. However, a large amount of features may be generated using past prices. There will be many generated features that are meant to be reduced using feature importance methods. Some examples include, average, max, and minimum price within a 3, 5, 10, 15, and 30 day period. Other features include a static change in price between the data point, and the price from 1, 3, 5, 10, 15, and 30 days prior. Finally, classification results would also be generated using future data. In total, 23 features are being considered with more likely to be added in the future. Other features being considered are stock indicators on those days like different types of moving averages, RSI indicator, and MACD.

4. Data Preparation and Feature Generation

4.1 Data Preparation

From Figure 3.1, there are four prices for each day, volume, and date. For this attempt, modeling will be done on a single "price" feature. As such, volume and date are not being considered for this step and will be removed. Many popular stock indicators also rely on closing or high prices of the day, but to capture a stock's regular movement throughout the day, a "price average" feature will be created by averaging the high and low prices for each day. The only features that remain before the feature generation step is the average price for that day and the date for the purpose of creating time-series features.

After the feature generation step, all features will also be normalized using a min-max normalization scale of zero to one. This is an important step as the features have various scales and the modeling process may mistake the importance of one feature over another because of that scaling issue.

4.2 Feature Generation

When working with time-series data, it is important to capture each data point as a piece to a timeline instead of a single point that isn't affected by time. To do this, features should be generated to capture its contribution to that timeline. Features such as moving averages, exponential moving averages, and percent change over a certain amount of days accomplish this. Furthermore, these are already financial objects that are used to predict future stock prices.

Figure 4.1 outlines the features that are being generated and the purpose behind each one. Some feature categories have multiple similar features within them. The purpose is to find the best description of that category during the feature selection step. After the completion of this step, a total of 29 features are utilized.

Another generated column is the binary outcome of each row, whether they fulfilled the objective or not. This is created by assigning a one or zero to values if their current price increased by the inputted percent in the inputted amount of days. For demonstrative purposes, the objective being fulfilled for the results sections of this paper is a 10% increase in 21 days. A successful objective would be if an investor buys at a day's price and a 10% increase occurs 21 days in the future.

Figure 4.1 - Feature Generation Table

Features	Reasoning and Description
3, 7, 15, 25, 40 Day Moving Average	The moving average is very common in estimating future trends of a stock price.
3, 7, 15, 25 Span Exponential Moving Average	Similar to the moving average, the exponential moving average is also used in many stock price trend indicators.
Price Minimum in Past 3, 7, 15, 25 Days	The minimum price within a certain range can be used as an indicator to invest if the current price moves below this range.
Price Maximum in Past 3, 7, 15, 25 Days	Similar to the previous feature, if the current price is the highest it's been, it may be an indicator to not buy.
Price within Highest and Lowest Price in Past 3, 7, 15, 25 Days	This feature takes the current price as a percent of the lowest and highest price in a certain range. This could determine the position of the current price in comparison to the price ranges.
40 Minus 7 Day Moving Average	This feature is often used as a price predictor and is able to detect trends in stock movement.
40 Minus 7 Day E. Moving Average	Similar to the previous feature, exponential moving average may be able to detect trans in stock movement.
Momentum	The momentum of a stock is the speed at which that stock changes and it’s often used to indicate stock strength of movement. (Silver et al., 2021)
RSI	This is a momentum-like stock indicator that “measures the magnitude of recent price changes to evaluate overbought or oversold…stock.” (Fernando, 2021)
MACD	The MACD is an indicator that measures the relationship between two important moving averages and is used in many stock analyses. (Fernando, 2021)
Binary 40MA and 7MA Crossover	When a long moving average crosses over a short moving average, this is regarded as a popular indicator to buy or sell stock.
Binary 40EMA and 7EMA Crossover	Similar to the previous feature, long and short crossovers can be indicators of future price movement.

5. Feature Selection

Two methods that may be useful for feature selection are Logistic Regression’s p-value and a Random Forest Classifier importance. Both of these methods excel at calculating the

importance of a feature in a classification model. Logistic Regression may be more useful in a case where there are mostly numerical features, but the Random Forest method is hardly affected by features that are highly correlated. This may prove useful in a feature's list that was generated from a single original feature.

After testing, both methods proved to have different opinions on features, so a combined feature selection method was implemented that used the outcome of both Logistic Regression and Random Forest. This combined method used a simple scoring system for each individual technique and combined the scores generated.

Figure 5.1a and 5.1b outline the outcome of each method for the Tesla dataset. In Figure 5.1a, the chart's P>|z|, or p-value, uses hypothesis testing to measure each feature's probability that a change in values would occur due to random chance. The lower this number is, the more important that feature is to the model. For Random Forest in Figure 5.1b, the information gain technique that reduces the gini index, which is a measure of impurity. When a feature is chosen to split a node, importance is measured by the amount of impurity in the remaining dataset that is reduced during that split. The higher reduction of impurity equates to a higher importance of that feature.

Different datasets and objectives may cause different features to be selected as important for each method. These were the results for the Tesla dataset, but a more predictable stock may have other features become important. Or, a longer objective or higher percent gain desires may have the same effect.

Figure 5.1a - Logistic Regression P-Values

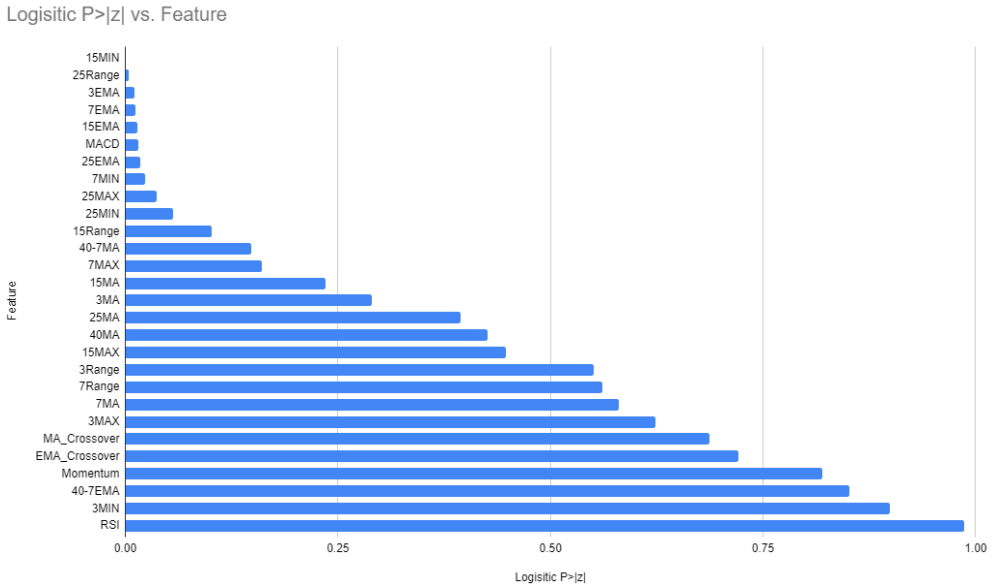
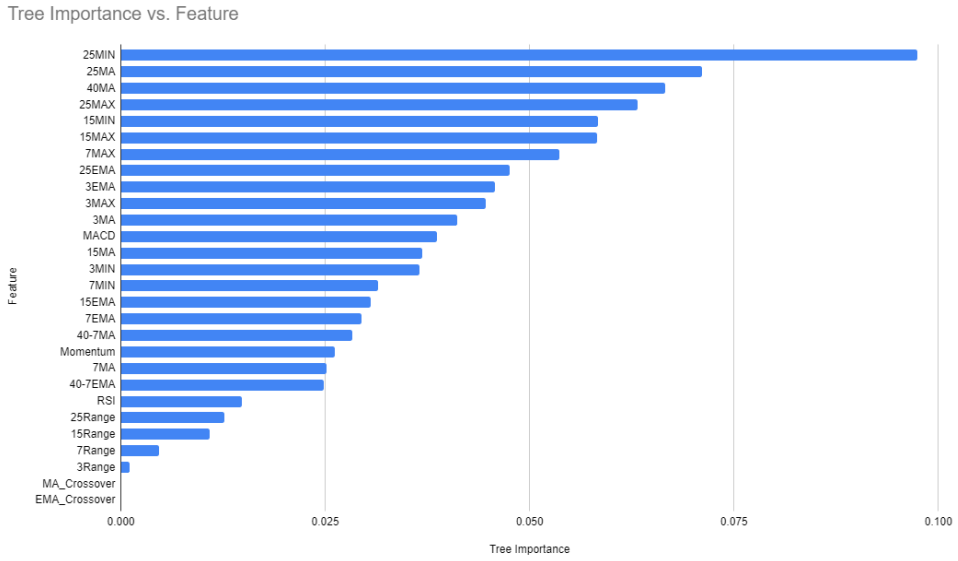


Figure 5.1b - Random Forest Importances



For the example dataset, five sets of features were used to compare their accuracy and precision in three machine learning models. P-value and importance cutoffs were selected visually where larger decline occurs in the figures above. The Random Forest importance has two sections where a large decline occurs at the feature 7MAX and 3MIN; the columns selected were split into a short and long version. Finally, a combined method that utilizes both Random Forest importance and Logistic Regression p-values were used. A scoring system was utilized to effectively average the results of both techniques. Figure 5.2 outlines the performance of each method. As well as the number of features that were chosen for each method. Each test attempted to maximize accuracy, however precision is an important metric for this case which will be discussed in the next section.

Figure 5.2 - Accuracy and Precision Results

	Combined Method	Logistic	Random Forest Short	Random Forest Long	All Columns
Logistic Precision	69.44%	62.96%	75.00%	70.27%	61.36%
Logistic Accuracy	57.89%	59.14%	58.24%	58.06%	58.96%
Random Forest Precision	90.79%	88.11%	87.40%	88.89%	89.45%
Random Forest Accuracy	90.32%	88.71%	89.43%	89.25%	88.89%
Gaussian Naive Bayes Precision	42.68%	42.73%	100.00%	42.68%	42.56%
Gaussian Naive Bayes Accuracy	42.47%	43.55%	56.27%	42.47%	42.47%
Number of Features	16	13	7	14	28

The results in Figure 5.2 show that the combined method has more success at determining successful objectives than other methods for the best performing model, Random Forest. This is common among feature reduction methods as the combined method has more input features. However, the combined method's accuracy is lower in both Gaussian Naive Bayes and the Logistic Regression classifier. Reducing the total number of features from twenty-eight to sixteen improves accuracy in the Random Forest model. While others benefitted from a more stringent reduction.

6. Accuracy Metrics and Cutoffs

There are four outcomes to the classification model; true positive, true negative, false positive, and false negative. Figure 6.1 outlines these outcomes and what they mean for the investor.

Figure 6.1 - Classification Outcomes

Outcome	Outcome Details
True Positive	The investor is told to invest and the objective is successful
False Negative	The investor misses out on an opportunity that would have met the objective
False Positive	The investor invests and fails to meet their objective.
True Negative	The investor does not invest and the objective is not met.

The true positive outcome can be regarded as the most successful outcome for the investor. This outcome can directly translate into a successful objective as well as profits. The true negative is also a successful outcome as it will discourage the investor to make an investment that would not reach their desired objective. The false negative can be referred to as a missed opportunity. The model does not detect that a successful objective would have occurred, thus the investor misses out on a single opportunity. The outcome that should be minimized is the false positive outcome. This is the outcome that can be associated with potential losses as the investor fails to meet their objective. In fact, a false positive outcome is the only outcome in which losses can occur.

First, accuracy is a great indicator on how well the model performs overall. This metric will be able to capture the number of correct classifications in the model. Second, precision is the number of true positives over the number of all predicted positives. Precision is an indicator that is especially useful when false positives are more detrimental to the application of the model than false negatives are.

Each model assigns a probability of a classification occurring and converts it into a binary classification system. If the probability of a true prediction is greater than 50%, by default, the model will classify that data point as true. Figure 5.2 uses this default value to simply find the best performing model for each feature group. However, modifying this threshold allows

for greater freedom in minimizing false positives and selecting an acceptable precision. The lower the threshold for a positive classification, the higher the number of false positives while increasing the threshold would lower false positives as well as positive predictions.

To programmatically find the optimal threshold that maximizes accuracy regardless of precision, adjust the threshold for each value in the testing data set. For each adjustment, calculate the accuracy of the set of predictions and return the highest accuracy. Now, to adjust the precision to match a user-desired risk rate, repeat this process while calculating precision as well. Finally, only return the highest accuracy that achieves the minimum precision. Figure 6.2 summarizes the outcomes of various allowed precision minimums.

Figure 6.2 - Precision Minimum Analysis

	Minimum Precision Allowed					
	0	0.92	0.94	0.96	0.98	1
Accuracy	91.14%	86.20%	82.80%	81.90%	74.55%	60.22%
Precision	90.32%	92.57%	94.74%	96.25%	98.20%	100.00%
Occurences in Test Data						
True Positives	216	187	162	154	109	27
False Negatives	33	62	87	95	140	222
False Positives	21	15	9	6	2	0
True Negatives	288	294	300	303	307	309
% Chance of Risk	8.86%	7.43%	5.26%	3.75%	1.80%	0%

With a minimum precision of zero, or no restriction, the optimal threshold shifted from the default value to improve the accuracy of the model found in figure 5.2 by 0.82%. With this maximized accuracy, precision is set to 90.32%. However, as minimum allowed precision increases, a few things happen. First, the number of false positives or potential losses is reduced in the testing dataset, thus decreasing the percent chance of loss. Secondly, the threshold for a positive classification increases, increasing the number of false negatives and true negatives as there are more negative predictions. This means a drastic increase in missed opportunities as many true positives are converted into them. While a maximal minimum precision of one allowed for a zero percent chance for loss in the test data, a mere 10.84% of successful objectives would be found (Figure 8.2). It is up to the investor to determine the amount of risk they would like to have as more risk means more opportunity.

7. Model Performance and Comparisons

Before final model performance, a GridSearchCV function was used to determine the optimal parameters for each machine learning model. In short, up to four parameters for each model were tested against multiple numerical and categorical parameters. The best performing parameters were selected for final testing for each model, and the results for those were shown in Figure 5.2.

Out of each optimized model, Random Forest clearly outperforms Gaussian Naive Bayes and Logistic Regression in terms of accuracy. Out of all feature sets, the best performing Random Forest model achieved 90.32% accuracy, Logistic Regression achieved 59.14%, and Gaussian Naive Bayes achieved 56.27%.

Out of the three models, Naive Bayes performed the worst for a variety of reasons. Naive Bayes assumes that each feature is independent from one another. Although, this is likely untrue as every single feature in the dataset was generated from another feature. This is evidenced by the fact that the highest performing model had the least amount of features. Out of the three classifiers chosen, Naive Bayes would likely not perform well in many circumstances.

The Random Forest classifier outperformed Logistic Regression. This may be caused by a similar reason as Naive Bayes, the generated features are likely linearly related. This harms the effectiveness of the Logistic Regression. Finally, the Logistic Regression is linear in nature in that it converts predictions using a logistic function and providing a probability of a true value. This can be a very limiting factor if the set of features do not have a linear relationship with the target feature.

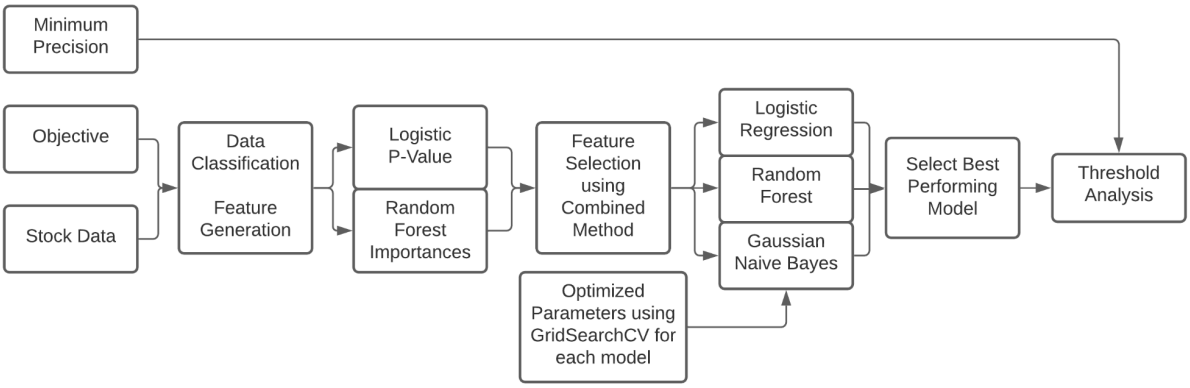
The Random Forest model was likely the best performing model for a few reasons. For most cases, the accuracy of the model increases with the size of the dataset. The more useful features and data points, the better the Random Forest Model can perform because the greater the number of data points that make it to each individual leaf node. This can increase predictive power by allowing the algorithm to select better features that increase information gain. This model is also known to be immune to outliers and noise because the rule-based system doesn't factor in the magnitude in which data is filtered into a rule. For the Tesla stock, large spikes of change could occur throughout and Random Forest is equipped to deal with those occurrences.

8. Results and Important Features

Before analyzing results, Figure 8.1 summarizes the unique approach for this classification problem. The inputs for the model are the minimum precision desired, the objective the user wishes to accomplish, and historical stock data. The highest precision of one maximizes safety but also maximizes the amount of missed opportunities. While having no minimum precision will select the highest performing threshold that maximizes accuracy. The classifications for the data are based on the stock data that is inputted as well as the objective the user desires to achieve. Each input passes through a unique feature selection as different stocks may have different features classified as important. However, each input uses the combined method for feature selection using a unique point system of both techniques. Finally, the best

performing model is selected to find the most accurate threshold for the minimum precision desired.

Figure 8.1 - Outline



For the Tesla stock, Random Forest would be selected by the program as the best performing model. And, depending on the precision desired, the probability threshold is adjusted to maximize accuracy while maintaining the amount of risk desired. For the Tesla testing dataset, there were a total of 558 days with predictions and out of those, 249 of those days satisfied the objective while 309 did not. With precision unrestricted, there are 38.71% of days that satisfy the condition and are detected by the model. This may translate to around that portion the program may be able to detect on the actual stock and predict an investment opportunity. If risk is minimized, only 4.84% of opportunity exists which is only one successful prediction in around twenty days.

With a maximum amount of risk, there is a 8.86% percent of risk associated with any prediction made. This can be calculated by dividing the number of times an objective isn't met by that number plus the number of true positives. Of the number of times that a positive prediction is made, there is a 8.86% chance that the prediction is false and the objective isn't met. To minimize this risk to zero, at around 89.16% of the opportunities are missed. However, to minimize the amount of missed opportunities, there needs to be no minimal precision floor.

Figure 8.2 - Additional Results

	Minimum Precision					
	0	0.92	0.94	0.96	0.98	1
% of Missed Opportunity	13.25%	24.90%	34.94%	38.15%	56.22%	89.16%
% Chance of Risk	8.86%	7.43%	5.26%	3.75%	1.80%	0.00%
Opportunity Available	38.71%	33.51%	29.03%	27.60%	19.53%	4.84%

Some of the most important features detected in the Random Forest technique for importance were the minimum price in 25 days, 25 day moving average, 40 day moving average, and the maximum price in 25 days.

The minimum and maximum price in 25 days could be very important in the model even though they aren't popular indicators in studies on stock behavior. The reason they may be good indicators is because in relation to the current price, these may indicate an opening to invest or not invest. For example, if the current price is equal to the 25 day minimum, it could be a heavy indicator to invest. Moving averages in stock predictions are used every day, so it's no surprise that the 25 day and 40 day moving averages are both great indicators for the Random Forest model.

The most underperforming were the crossover detectors, the percent range, RSI, and even Momentum. Crossover detectors could be a victim to the test objective, which was a 10% increase in 21 days. Moving average crossovers could take months to realize their potential. The reason that feature selection is repeated for every unique input of stock and objective is for that reason. RSI and Momentum also underperformed as they were likely not implemented in a way that is used in stock prediction. Here, only a flat number was calculated and inputted, but in finance they are often compared to another value or used to detect trends in their history.

In conclusion, the Random Forest model was decent at detecting a 10% increase in 21 days for the Tesla stock. Even with minimal risk detected in the model, there were still opportunities detected through the 558 days of the testing dataset. This classification model may not be able to detect how to maximize profits, but it can help the consumer invest safer or for shorter periods. In the end, this tool could be useful for some applications, but more thorough testing should be done before a real implementation.

References

Fernando, J. (2019a). *Moving Average Convergence Divergence – MACD Definition*. Investopedia. https://www.investopedia.com/terms/m/macd.asp

Fernando, J. (2019b). *Relative Strength Index – RSI*. Investopedia. https://www.investopedia.com/terms/r/rsi.asp

Mitchell, C. (2021, February 18). *Swing Trading Definition and Tactics*. Investopedia. https://www.investopedia.com/terms/s/swingtrading.asp

Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert systems with applications*, *42*(1), 259-268.

Popper, N. (2020, July 8). Robinhood Has Lured Young Traders, Sometimes With Devastating Results. *The New York Times*. https://www.nytimes.com/2020/07/08/technology/robinhood-risky-trading.html

Silver, C. (2021, January). *Momentum Indicates Stock Price Strength*. Investopedia. https://www.investopedia.com/articles/technical/081501.asp

Zhang, J., Li, L., & Chen, W. (2021). Predicting stock price using two-stage machine learning techniques. *Computational Economics*, *57*(4), 1237-1261.