**Introduction**

This project will look at several different technical analysis indicators used on stock market data to determine their effectiveness for improving investment decision-making. These indicators would be used to give “buy” or “sell” signals in order to make an investment decision.

**Background**

Technical analysis indicators are the result of formulas derived from the price movement of equities. These can be very complex or as simple as a moving average. Many traders swear by them as a means of knowing when to buy or sell. Others feel the opposite, that they are of no greater use than astrology or tea leaves would be for making investment decisions.

**Problem Statement**

Through thorough analysis of market data and technical indicators, my goal is to identify indicators that appear to be of potential use. Once these features have been pared down the goal is to build a logistic regression model to deliver buy, sell, or hold signals.

**Scope**

I’ll be looking at both long-term and short-term data. Long-term data in this context will be defined as approximately 20 years of daily market data for the symbols in question. Short-term data will be approximately one month of data collected every five minutes that the stock market is open. I’m looking to gather this for approximately 100 different stock tickers for analysis, which will be determined by their average daily trading volume.

**Preliminary Requirement**

The primary requirement needed for this project will be stock market data. I primarily intend to acquire this from AlphaVantage’s free API. In a previous course, I built a Python API wrapper for AlphaVantage and will be using that to automatically collect market data for this project.

**Technical Approach**

**Analysis**

Once the data set has been built containing all of the stock symbols intended for testing and the full suit of technical indicators intended to be tested, the next step will be to build a function that tests various parameters and combinations of these indicators to find combinations that perform well on out-of-sample data after being optimized on in-sample data.

**Requirement Development**

The time series gathered will be analyzed into three different clusters based on whether they are stationary. I will use three tests to determine if a time series is stationary or not: The Dickey-Fuller Test, the Kwiatkowski–Phillips–Schmidt–Shin Test, and Hurst Exponent. If all three of tests indicate that a series is either stationary or trending, they will be categorized as such. If a mix of results is received, they will be filed into a third cluster. This filtering is important as the use of technical indicators highly depends on whether we believe the price of a stock will be trending or stationary.

Once each time series has been categorized, they will be tested on various technical analysis indicators. The current plan is to examine the following indicators:

* Bollinger Bands
* Moving Average Convergence-Divergence
* Simple Moving Average
* Exponential Moving Average
* Moving Standard Deviation
* Moving Variance
* Williams Index
* Commodity Channel Index
* Money Flow Index
* Relative Strength Index
* Stochastic Oscillator

Multiple parameters will be tested for each of these indicators and success will be determined on their ability to generate profitable returns across the entire sector of time series.

**Model Deployment**

Several models will be built based on the indicators that are found to show potential for each sector of time series. These models will utilize logistic regression to classify each entry as a buy, hold, or sell signal.

**Testing and Evaluation**

Due to the focus of these models being to generate profit, classic classification evaluation metrics may not be ideal. We may have a signal that is only correct 49% of the time. Normally, that wouldn’t be ideal. However, if the 49% of the times it provides an accurate signal are significantly more profitable than the 51% of the time it doesn’t, it could be a valuable model. For this reason, models will be evaluated on their ability to outperform a benchmark in terms of financial return.

**Expected Results**

In the time I have spent viewing technical analysis indicators, I haven’t seen many that generate profitable returns over a long-term. They may show promise in a small sample of data, but the performance doesn’t often continue beyond a sample they were optimized for. My expected results are for that trend to continue and for none of the indicators examined to show successful long-term performance when analyzed across a basket of similar stocks.

**Management** **Approach**

**Project Plan**

As we are currently in the second week of the project, my plan for this week is to begin to accumulate all the data that will be needed for this project. This will include gathering the raw data from the AlphaVantage API as well as the technical indicator data. I have examined a few Python libraries that look promising for generating the technical indicator data but haven’t decided on which one(s) I will be continuing with yet. AlphaVantage also provides this data through their API but they have a rather strict rate limit and I don’t want to use up my API calls on data that could be generated elsewhere.

Next week my focus will be on beginning to analyze each of the times series I have collected and begin to categorize them as stationary, non-stationary, or undetermined. Long and short-term series will also be handled individually, resulting in a total of six clusters of time series to work with.

Moving forward from there, I will begin to create performance benchmarks. For time series determined to be non-stationary, these will be the percentage return on investment achieved if one had bought the stock in the earliest data point and held onto it until the most recent data point. For time series determined to be stationary and ones that are undetermined, this will be evaluated using randomness to determine whether to buy or sell a stock. This benchmark could change moving forward as I would like to generate a similar number of trades to the number generated by the models built. The returns of approximately 1,000 simulations of the random buying and selling will be averaged down into a benchmark for comparison.

I will then begin to analyze the previously mentioned technical analysis indicators with various parameters. I anticipate this being a time-intensive process as far as computer processing goes and am considering two different paths to move through this. I will initially run a few of these tests on my personal computer and see how long the process takes. If the anticipated time for all the tests is reasonable, I’ll continue to do that but design my code in a way that it can run parallel across the multiple cores of my processers. If this time doesn’t seem to be reasonable, I will investigate using Amazon Web Services to run all the tests in parallel, which should take very little time to run but would take more time to set up and incur a fee.

Once the analysis of each individual time series has been completed, I will re-aggregate the results by their original clusters and observe how each indicator performed across all time series within the cluster. Those whose performance exceeded the benchmark will be included in future modeling. The rest will be discarded. These models will be classification models designed to deliver *buy*, *hold,* or *sell¸* signals based on the information received from the technical analysis indicators. I will test various classification models to identify the best performing ones and may choose to use an ensemble model if I find it does a better job of generalizing.

For the sake of organization and planning, I will be using a Git repository to organize and version-control my code, data, and other work. In addition, I will be using the boards feature of the GitKraken software to keep track of what tasks need to be done, detail the finer points of how I intend to accomplish each task, and to provide myself personal due dates on when I want to have each objective completed by.

**Project Risk**

One of the biggest risks with this project will be scope creep. I’m already trying to handle 100 different stock tickers worth of data. I’m not entirely sure what the computing time on each of these will be. Every set of indicators and the parameters I choose to test in these indicators will be increasing the computation time necessary. While I’ve already set myself up for a lot of computing, I need to ensure that I don’t make decisions that increase the time needed to an impractical level.

Two additional risks I am aware of relate to the usage of time series data. The indicators I am generating are going to be for the current time period. In a real-world trading environment, these indicators wouldn’t be available until the time period has passed. Consequentially, these indicators need to all be time shifted one period forward in the future to simulate the way they would be used in the real world. Leaving them in their current time period could lead to results that aren’t repeatable in the real-world. Also, if training models using cross-validation, I need to be sure that the selection of test data is being done in sequential pieces, not a random selection. If data were to be selected randomly, it would be easy for all the surrounding training data to train the model very well on a piece of training data. This would lead to an overfit model that won’t generalize well across the entire time series and be of no practical use.

**References**

AlphaVantage. (2017). API Documentation | Alpha Vantage. Retrieved September 6, 2019, from Alphavantage.co website: https://www.alphavantage.co/documentation/

Investopedia. (2019, June 25). Using Technical Indicators to Develop Trading Strategies. Retrieved September 5, 2019, from Investopedia website: https://www.investopedia.com/articles/trading/11/indicators-and-strategies-explained.asp

‌