Short-Term Trading Strategy based on Volatility and Technical Indicators – Status Report

Osama Iqbal
iqbal.osama@icloud.com

Abstract

The project aims to establish a trading system in which the user creates a portfolio of stocks by entering a ticker or by uploading data through a file. A signal filter is then generated using Technical Indicators such as SMA, ADX, RSI, Bollinger Bands etc. These indicators are then used in a short term trading strategy in combination. For example, ADX-MACD-RSI-BB strategy, where the indicators from the signal filter are used in combination.

Keywords: *Technical Analysis, Trading System, RSI, Bollinger Bands, MACD, ADX*

1. Introduction

The aim of the project is to create a Trading System that consists of a Short Term strategy. For performing short term trading, we use Technical Indicators in order to generate signal filters, upon which strategies could be devised. In essence, the step by step procedure to achieve this is as follows. **Points in bold indicate completion**:

1. **Get ticker/s from user as an entry, or from a file source. Only stocks are permitted to be used. (Scope change, only allow the user to enter tickers rather than custom data, as much exception handling is required in terms of erroneous data)**
2. **Generate a portfolio based on the tickers that have been entered, and gather the data (EoD) from a source**
3. Plot the outlier stocks based on volatility
4. Generate a signal filter based on SMA, ADX, RSI, BB, MACD
5. Devise a strategy using ADX-MACH-RSI-BB
6. Calculate and plot Key Performance Indicators such as Drawdown, Win-to-Loss Ratio, Gain-to-Pain Ratio and other factors such as Number of trades etc.

2. Main Program Entry and Stock Data Gathering

The main program, or **'main.py'** is the main entry to the entire system. This consists of the '__main__' function that drives the logic. Essentially, the main program performs the following steps:

1. Allow the user to enter the stock combinations
2. Fetch the historical data (EoD) for the combinations
3. Identify the volatile stocks (work in progress)
4. Run strategy and plot results (work in progress)

At this point in time, the main program consists of only prompting the user to use a mode to enter the data, and gathering of historical data

```
"""
Short-Term Trading Strategy based on Volatility and Technical Indicators
"""
# Imports of custom modules
from stock_entry import prompt_for_stock_entry
from data_fetcher import fetch_stock_hist_data


# The main entry loop into the program
if __name__ == '__main__':
    # Prompt the user for the mode of entry for stock data and get list of data
    tickers = prompt_for_stock_entry()
    # Fetch historical data from datasource
    data = fetch_historical_data_for_stocks(tickers)
```

3. Gathering ticker data from user

The function prompt_for_stock_entry() within the module stock_entry, consist of a prompt indicating the user to enter data based on stocks that are available on SnP500 for now. If the user has a file of data, he or she can press the ‘F’ key, which will then take the data from the file or the ‘T’ key that will allow the user to enter tickers instead. After the user has entered the tickers, he must decide upon the number of stocks that need to be used in his/her portfolio for every ticker entered. This is prompted to the user.

4. Fetching of Historical Data

Historical Data is fetched from Yahoo Finance, using a module called as Fix Yahoo Finance, since Yahoo discontinued their API service in 2017. This means that this additional third-party module must be included in order to fetch the data. The data is fetched in the form of a DataFrame which is then loaded into a dictionary. This dictionaries’ keys are nothing but the ticker symbols. I do this instead of fetching it in a panel since it’s easier for me to use conventional python than pandas’ iteritems() etc. Plus, visualizing a dictionary is easier in PyCharm than a Panel.