**Solution Design**

There are two major parts to the project.

One part is the development of the Hidden Markov Models and testing the in-sample performance of different strategies under different event states of different regimes.

So, the files used for this part are:

1. Regime\_Detection.py - The main file to be run first. Could be opened in either in Spyder or could be run

from the command prompt by issuing python <filename>

2. Output\_HMM.py - Once the Regime\_Detection.py file is run, then the next step is to run this file. This has to be

run in a different python session. Best is to use command prompt to run this file.

3. HMM\_func.R - This file need not be run. This is just to show how the HMM model was fitted in R. The R package appears to be superior to the one in python. It is more robust.

Python version used for Part1 - 2.7 (64 bit) in Anaconda

Libaries - pickle, matplotlib, six, numpy, pandas, datetime, scipy, hmmlearn, warnings, re

If any of the library is missing the library can be manually installed by issuing command pip install <package-name>

Input files necessary for running the Python files

1. HMM.csv - Hidden Markov Model for S&P 500 generated by R (This is also necessary as this is a model generated through R. Ofcourse R could be run from python to automate this. But it is just not relevant for the scope of this project)

2. GSPC - S&P 500 market data (This of course could have been avoided. But I didn’t know of the yahoo fix that could be used to download. Even the yahoo fix sometime breaks.)

3. history.csv - EDHEC hedge fund returns (This could not be automated. This can only be manually downloaded by registering at http://www.edhec-risk.com/)

Please excuse the use of csv files.

The second part is the out of sample backtesting. This uses a backtesting framework called QSTrader. This works only on Python 3.6. The complete information on installation in the anaconda framework is given in <https://www.quantstart.com/qstrader>. It says it also could be used in Linux.

My main need for a framework was the superior visualization provided and also concentrating the focus on strategy development. I explored another framework which was more simple and intuitive . This was Quantiacs. But in Quantiacs I so far couldn’t find a way to use markets they didn’t support. For example they didn’t support the symbols EEM and VXO. So I stuck with QSTrader.

Also there are some fixes that needs to be done to make it work according to our needs.

After installation, go the settings.py file. This will be located in the Qstrader library folder. In my case it was here:

C:\Users\Handygeek\Anaconda3\envs\qstrader\Lib\site-packages\qstrader

Please make the following changes:

DEFAULT = munchify({

#"CSV\_DATA\_DIR": from\_env("CSV\_DATA\_DIR", "~/data"),

#"OUTPUT\_DIR": from\_env("OUTPUT\_DIR", "~/out")

"CSV\_DATA\_DIR": from\_env("CSV\_DATA\_DIR", "C:\\Users\\Handygeek\\Documents\\Quantshub\\QSTrader\\data"),

"OUTPUT\_DIR": from\_env("OUTPUT\_DIR", "C:\\Users\\Handygeek\\Documents\\Quantshub\\QSTrader\\out")

})

TEST = munchify({

#"CSV\_DATA\_DIR": "data",

#"OUTPUT\_DIR": "out"

"CSV\_DATA\_DIR": from\_env("CSV\_DATA\_DIR", "C:\\Users\\Handygeek\\Documents\\Quantshub\\QSTrader\\data"),

"OUTPUT\_DIR": from\_env("OUTPUT\_DIR", "C:\\Users\\Handygeek\\Documents\\Quantshub\\QSTrader\\out")

Here the CSV\_DATA\_DIR should point towards the directory where you keep the CSV data files. QSTrader operates by reading csv files.

And OUTPUT\_DIR should point towards the directory where the trade and tear sheet will be created.

Then open tearsheet.py. This will be present in statistics subfolder Make the following changes:

# Plot the figure

#plt.show(block=False)

plt.show()

around lines 636 to 638 before the get\_filename function starts

Also change the savefile function as follows:

def save(self, filename=""):

filename = self.get\_filename(filename)

self.plot\_results(filename)

Now open trading\_session.py . And make the following changes towards the end:

)

# if not testing:

# self.statistics.plot\_results()

#self.statistics.plot\_results()

self.statistics.save()

return results

That’s it we are all set.

There are only three important files here of which only one should be run:

1. regime\_hmm\_backtest.py – This is the only file that needs to be run. Use Spyder from the Qstrader Environment to run this file.

One could alternate between the symbols VWO and EEM by editing lines 282 and 283 as well as lines 257 and 258.

One could change the HMM model to be used [S&P 500 model 1, S&P 500 model2 (uses trubulence), fx model (uses turbulence) by altering lines 216 to 221.

The HMM model needs to be built only once. Thereafter it can be commented out.

But in case you need to change dates for S&P 500 model 1 (It can use different date range whereas S&P 500 model 2 and currency model uses fixed data range) then the model needs to be rebuilt.

The dates for S&P 500 model 1 can be altered by changing lines 158 through 161.

To build S&P 500 model 1 , use build\_HMM\_model() and for S&P 500 model 2 use build\_HMM\_model2()

1. regime\_hmm\_strategy.py – This is where the simple moving average crossover strategy is defined
2. regime\_hmm\_risk\_manager.py - – This doesn’t use Markovian state information There are 2 more variations of this file: regime\_hmm\_risk\_manager\_with\_hmm.py (which uses Markovian information in the normal form i.e trade during normal state, abstain during event state) and regime\_hmm\_risk\_manager\_with\_hmm\_reverse.py (uses Markovian information in the reverse state)

Files used:

1. SPY.csv – Not really necessary as we are not using it for benchmarking
2. VWO.csv , EEM.csv – These need to be placed under data directory. QSTrader doesn’t download data from the web. It looks for these under the data folder pointed out earlier in this document. This could have been automated by downloading data from yahoo and manually placing it in the folder.

As far as library is concerned in addition to the libraries already discussed it uses pandas\_datareader and fix\_yahoo\_finance. Use pip install <package> under the Qstrader environment.