MScFE 642 DEEP LEARNING FOR FINANCE

Group Work Project # 2

See grading rubric here.

Scenario

Most institutional investors' portfolio allocations have exposure to multiple asset classes and/or geographical areas. This makes a lot of sense, since this way investors can exploit variations in rewards for risk across different asset classes throughout market regimes. Strategic Asset Allocation (SAA) and Tactical Asset Allocation (TAA) are different approaches to multi-asset investment strategies. You can read more about these in this short article from Vanguard.

In this GWP, you will work on producing a Deep Learning model that tries to predict short-term market trends across different asset classes. Then, you will test the out-of-sample performance of different asset allocation strategies based on the predictions of the model.

Tasks

Step 1

First, you need to gather data on the different asset classes that we are going to consider. To simplify, in this case we will work with 5 different ETFs, each a major representative of their asset class. Here you have their symbols, so that you can work with the ***yfinance*** library to download the data:

- Equity → SPDR S&P500 ETF. Symbol: "*SPY*"
- Fixed Income → iShares 20+ Year Treasury Bond ETF. Symbol: "*TLT*"
- Cash-like → iShares 1-3 Year Treasury Bond ETF. Symbol: "*SHY*"
- Precious metals (Gold) SPDR Gold Shares. Symbol: "*GLD*"
- Crude Oil → Invesco DB Oil Fund. Symbol: "*DBO*"

***Note**: Groups of 2 students are free to work using only 3 out of the 5 different ETFs (free choice). This will apply throughout the whole task.*

a. First, use the ***yfinance*** library to download the data. You need to **get daily prices** for each of the 5 ETFs. You will use data in the period from January 1st, 2018 to December 30th, 2022 for your **test sample**. We are leaving to your discretion the definition of validation and training sample periods.

b. Once you have downloaded the data for each individual ETF, **perform some exploratory data analysis**. Specifically, use visualization tools to describe the different dynamics and characteristics of the time-series (summary statistics, seasonality, stationarity, …). Try keeping this brief and focus on the main characteristics you consider worth it of highlighting. You can focus on prices and/or daily returns. The exploratory data analysis should address both individual distributions AND joint distributions, including correlations/covariances.

Step 2

In Step 2, you will create a Deep Learning model using recurrent neural networks to predict the 25-day ahead return of each of the 5 different asset classes.

a. For each of the 5 different asset classes, create a Deep Learning model that uses LSTM network architecture (you may combine LSTM layers with regular dense layers) to **predict the 25-day ahead return of each ETF**. Use as inputs the past information of the time series (feel free to include whatever lags in returns you consider appropriate). You are free to select the model architecture you like. You can even play around with different architectures and choose the one that performs better.

b. **Train** each of the 5 models. Comment on the in-sample predictive performance of the models, as well as the comparative results across asset classes.

c. **Test** each of the 5 models **out of sample**. Comment on the out-of-sample predictive performance of the models, as well as the comparative results across asset classes.

d. Create a trading strategy that uses the out-of-sample predictions of the 5 different models. You can choose the strategy that you want. You could, for example, decide to go long on the 2 (or 3) asset classes the model predicts will do best and go short on the 2 (or 3) asset classes the model expects to do

worse. **The strategy needs to be rebalanced every 25 days and use information from the predictions of at least 4 of the 5 asset classes.**

e. **Backtest** the previous strategy in the test period. Comment on the results. Compare the results versus the Buy-and-hold return of an Equally weighted portfolio of the 5 asset classes.

Step 3

So far, our models have always had 1 output unit. You will now implement a Deep Learning model with multi-output. If you need more information on multi-output regression, feel free to research online.

a. Create a **multi-output model** that blends together all of the information from the 5 models of Step 2. Specifically, create a model architecture that uses all the inputs from the 5 different previous models together to predict the 25-day ahead return from the 5 different ETFs. In other words, the model will have 5 outputs (one for each 25-day ahead return of an ETF).

b. **Train and test** the previous model. Comment on the in- and out-of-sample predictive performance of the model and compare them with the ones from Step 2.

c. **Create a trading strategy that uses the out-of-sample predictions of the model.** Same instructions as in Step 2 part d) apply.

d. **Backtest** the previous strategy in the test period. Comment on the results. Compare the results versus the (i) Buy-and-hold return of an Equally weighted portfolio of the 5 asset classes, and (ii) the return from the trading strategy of Step 2 part e).

Groups of 2: You will develop a multi-output model, but it will only contain the output from 3 rather than 5 different ETFs.

Step 4

Now, as a group, discuss the different results obtained. Specifically, what are the predictability implications of the multi-output versus the single-output architectures? What about in terms of backtesting performance? Is the same information being captured by both model architectures?

Step 5

As a group, please submit all files that reproduce the assignment. This will include the Python file for all the steps as well as work together on a report that contains the answers to all the questions above (except for the code) in a clear and organized manner (think about how you would present this to your boss).

Submission Requirements and Format

One team member submits the following on behalf of the entire group:

- **1 PDF document** containing ONLY the answers to the questions, EXCLUDING code.
    - Use the available Report Template and fill out the required information in the first page.
- **1 zipped folder that** contains:
    - Jupyter notebook that is executable
    - html of both the Jupyter notebook containing all code, results, and graphs

***Use Google Colab or GitHub to collaborate** in completing the executable Python program.*

NOTE: *The PDF must be uploaded **separately** from the zipped folder that includes any other types of files. This allows Turnitin to generate a similarity report.*

Rubric

Your instructor will evaluate your group submission for GWP1 using the following rubric:

Quantitative Analysis (Open-Ended Questions) 40 Points	Technical and Non-Technical Reports 30 Points	Writing and Formatting 20 Points
The group is able to apply results, formulas, and their knowledge of theory to real-life finance scenarios by doing the following: ● Providing all the necessary information to support their arguments. ● Presenting arguments that reflect group discussion and research. ● Using authoritative references to support a position and provide updated information. ● Concluding with practical takeaways for more insightful financial decision-making.	Technical reports contain 3 parts: 1) **code** for each question (be sure to explicitly state the question number), 2) the corresponding output of that **code**, and 3) **interpretations** and/or **recommended courses of action** that reasonably follow from those results. **Note**: Technical reports will include the technicalities of models, such as names, methods of estimation, parameter values, etc., and exclude generalities about the work done. It should NOT include names of Python code that were used.	A submission that looks professional should: ● Include the axes, labels, and scales in graphs. ● Be free of significant grammatical errors or typos. ● Be an organized, well-structured, and easy-to-read document. ● Include proper citations and a bibliography in MLA format.
	Non-technical reports contain 3 parts: 1) clear explanation of results; 2) the recommended course of action that follows; and 3) the identification of factors that impact each portfolio. **Note**: AVOID all references to model names, algorithms, and unnecessary details. Instead, focus on the investment decision.	