FULL LEGAL NAME	LOCATION (COUNTRY)	EMAIL ADDRESS	MARK X FOR ANY NON-CONTRIBUTING MEMBER
Marin Stoyanov	Bulgaria	azonealerts@gmx.com	
Brian Luna Patino	United Kingdom	brian1311@hotmail.co.uk	
Long Chen	China	chenlongpku@163.com	

Statement of integrity: By typing the names of all group members in the text boxes below, you confirm that the assignment submitted is original work produced by the group (excluding any non-contributing members identified with an "X" above).

Team member 1	Marin Stoyanov
Team member 2	Brian Luna Patino
Team member 3	Long Chen

Use the box below to explain any attempts to reach out to a non-contributing member. Type (N/A) if all members contributed.
Note: You may be required to provide proof of your outreach to non-contributing members upon request.

N/A

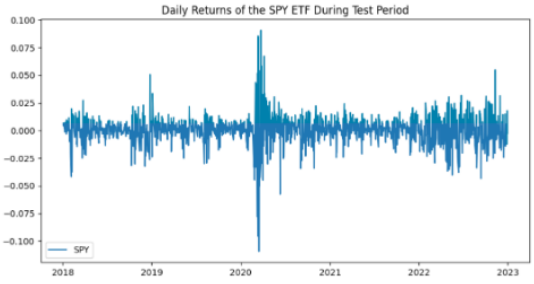
Step 1: Data Preparation & EDA

The test preriod is required to be from January 1st, 2018 to December 30th, 2022. The underlying five assets are:

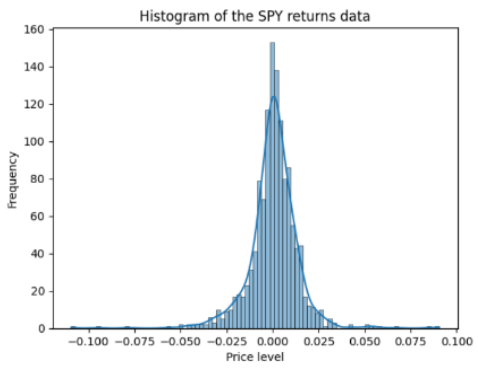
- SPY: S&P500,
- TLT: 20+ Year Treasury Bond
- SHY: 1-3 Year Treasury Bond ETF
- GLD: Gold
- DBO: Crude Oil

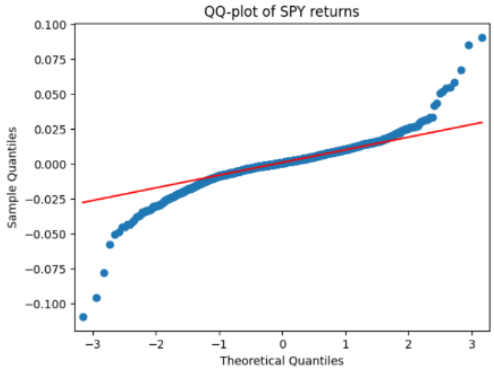
EDA is a kind of a systematic approach that is used to analyze and investigate datasets. The main goals are: to help identify patterns, relationships, and outliers that might not be immediately visible or can even be a hidden insight hard to find. EDA helps assess underlying assumptions on which statistical inference will be based and it will lead to a specific researching approach [1].

Let's see the ETFs' information one by one:

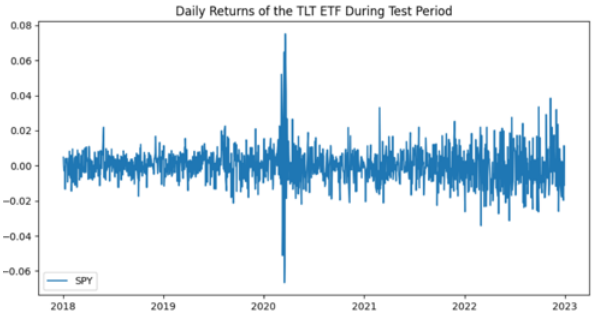



```
count    1258.000000
mean        0.000441
std         0.013575
min        -0.109424
25%        -0.005130
50%         0.000767
75%         0.007113
max         0.090603
Name: SPY, dtype: float64
```

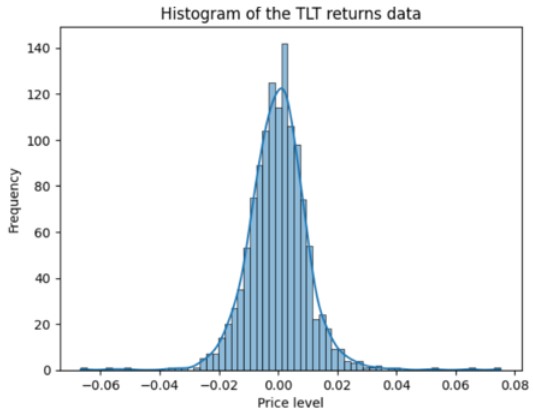


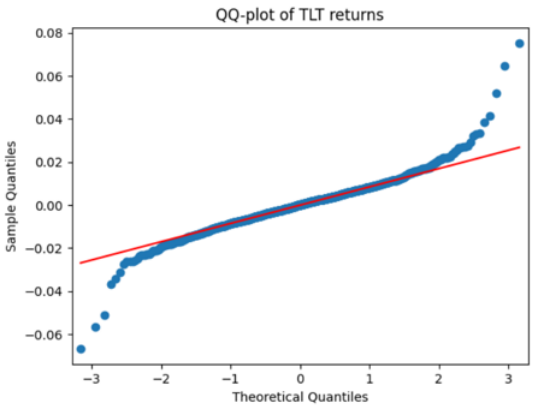


Descriptive Statistics of SPY Daily Returns

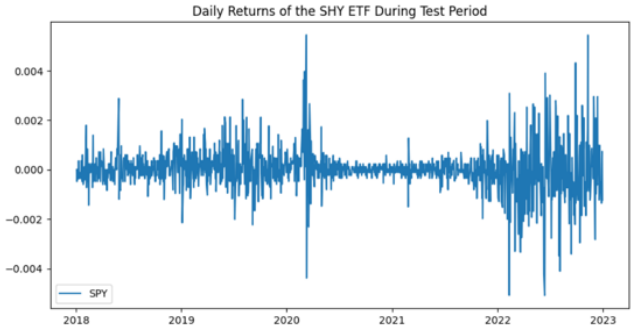



```
count    1258.000000
mean       -0.000050
std         0.010171
min        -0.066683
25%        -0.005797
50%         0.000072
75%         0.005653
max         0.075195
Name: TLT, dtype: float64
```

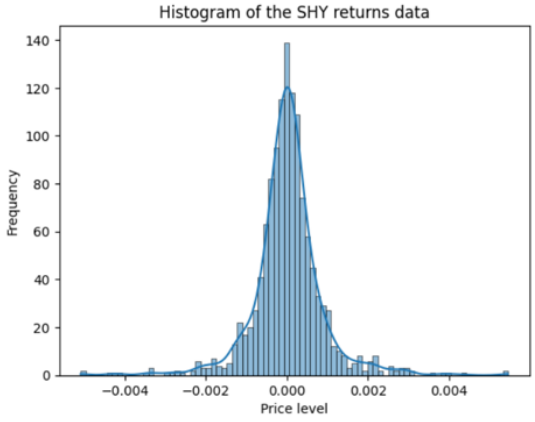


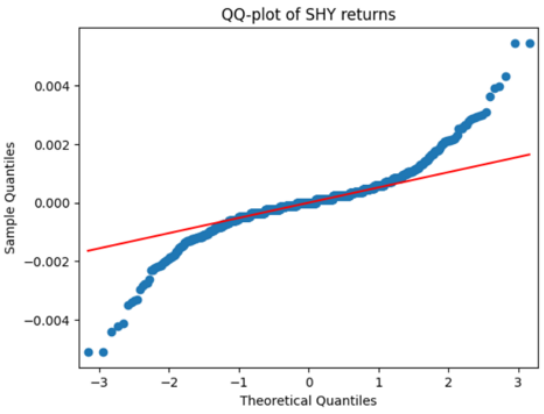


Descriptive Statistics of TLT Daily Returns

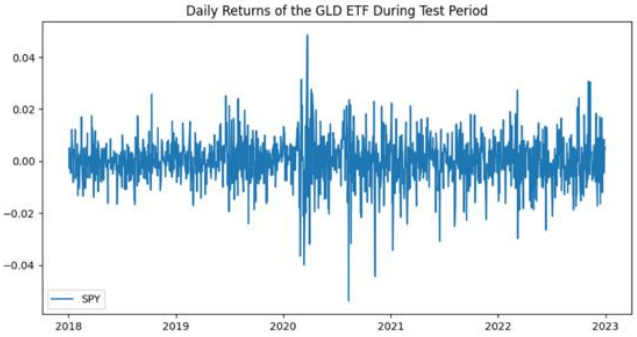



```
count    1258.000000
mean        0.000025
std         0.000883
min        -0.005088
25%        -0.000348
50%         0.000000
75%         0.000354
max         0.005452
Name: SHY, dtype: float64
```

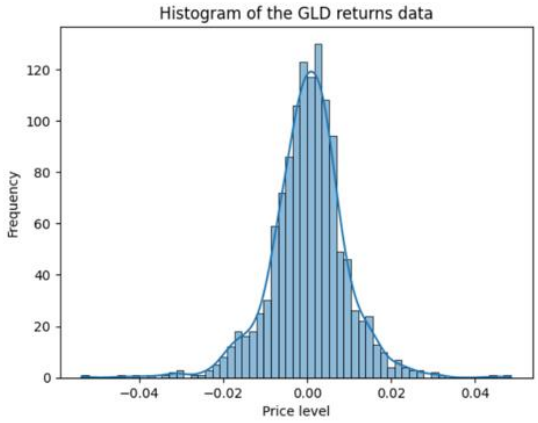


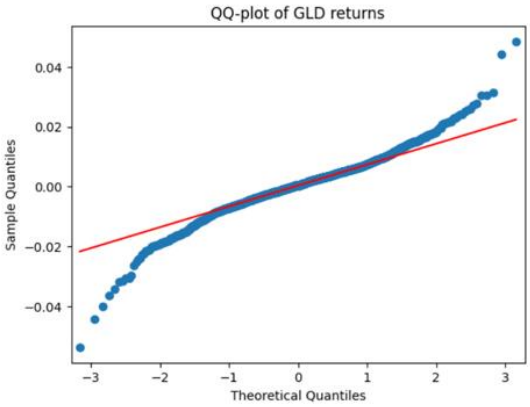


Descriptive Statistics of SHY Daily Returns

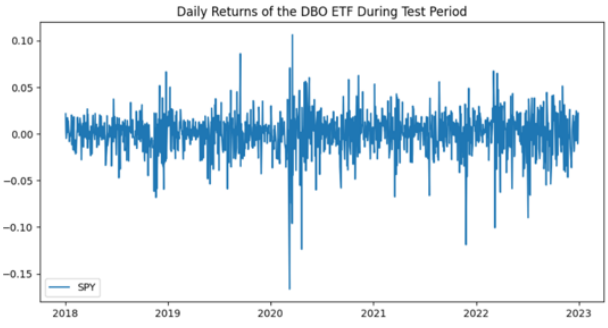



```
count    1258.000000
mean        0.000283
std         0.009044
min        -0.053694
25%        -0.004325
50%         0.000520
75%         0.005091
max         0.048530
Name: GLD, dtype: float64
```

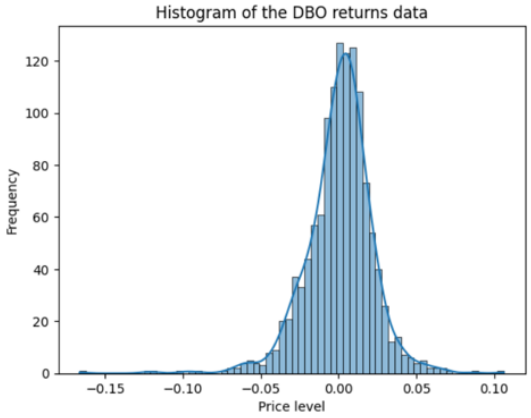


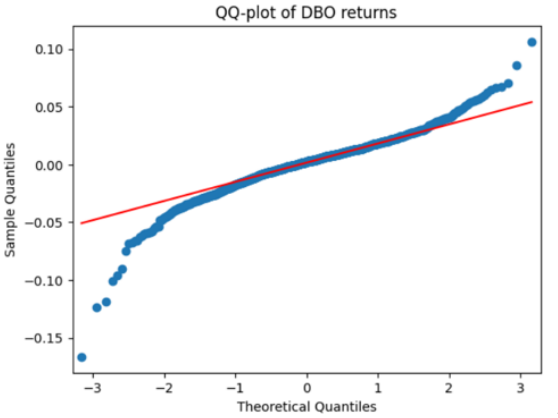


Descriptive Statistics of GLD Daily Returns



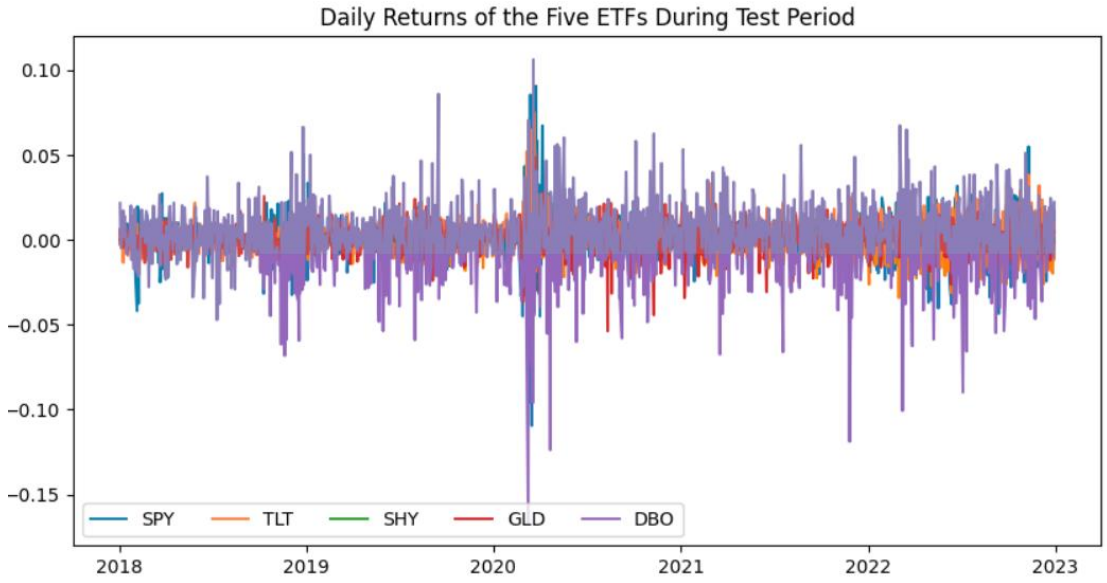

```
count    1258.000000
mean        0.000587
std         0.021621
min        -0.166453
25%        -0.009601
50%         0.002443
75%         0.012775
max         0.106227
Name: DBO, dtype: float64
```





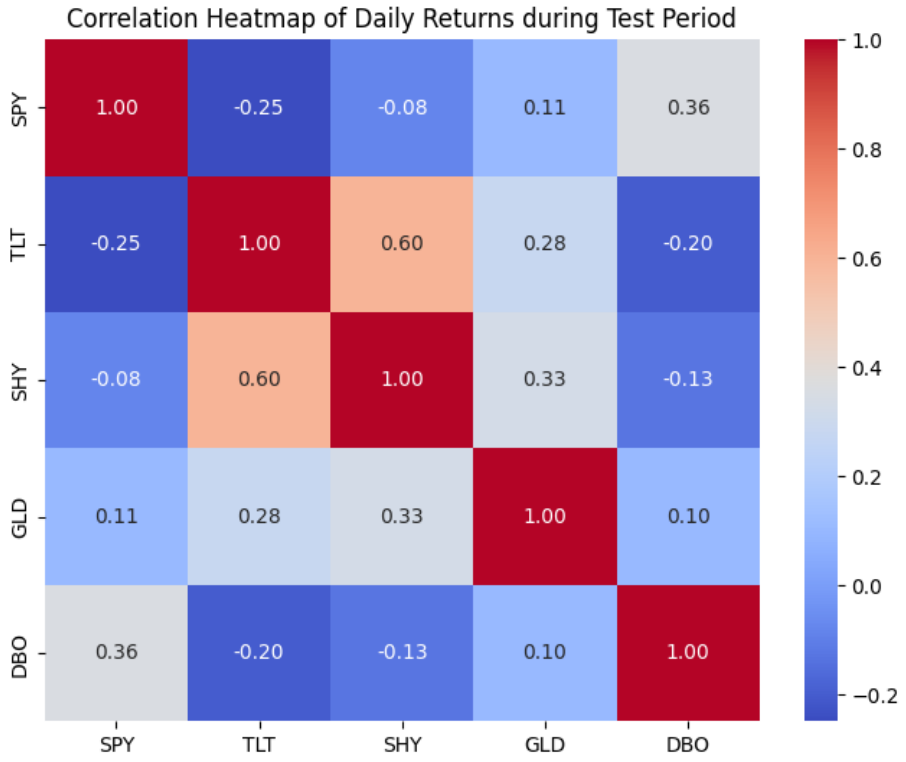
Descriptive Statistics of DBO Daily Returns

We plot all of these ETFs' returns for the period January 1st, 2018 to December 30th, 2022 in one picture:



From the timeplot during the test period, we can see that:

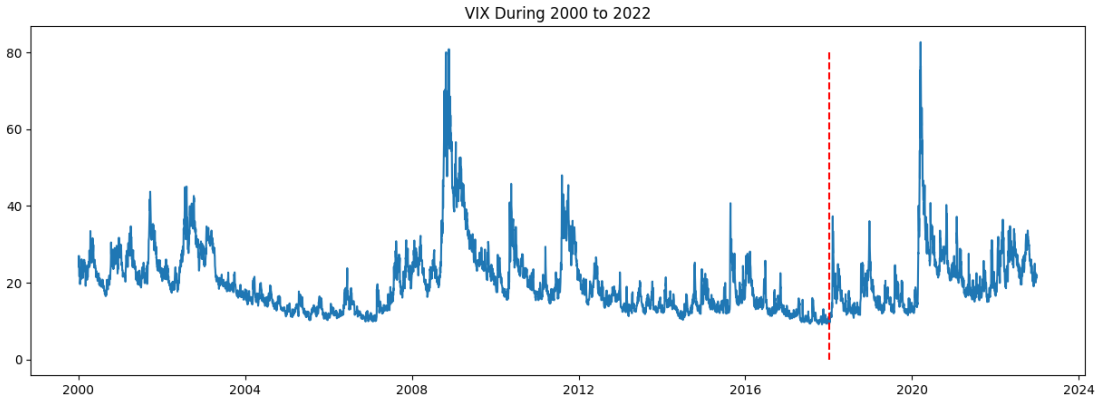
- All time series of returns seem to be stationary, showing zero-mean reverting.
- The crude oil (DBO) has a considerably higher volatility than other assets.
- There is a period in the early 2020 that shows extreme volatility, probably due to Covid-19.



From the correlation heatmap, we can see:

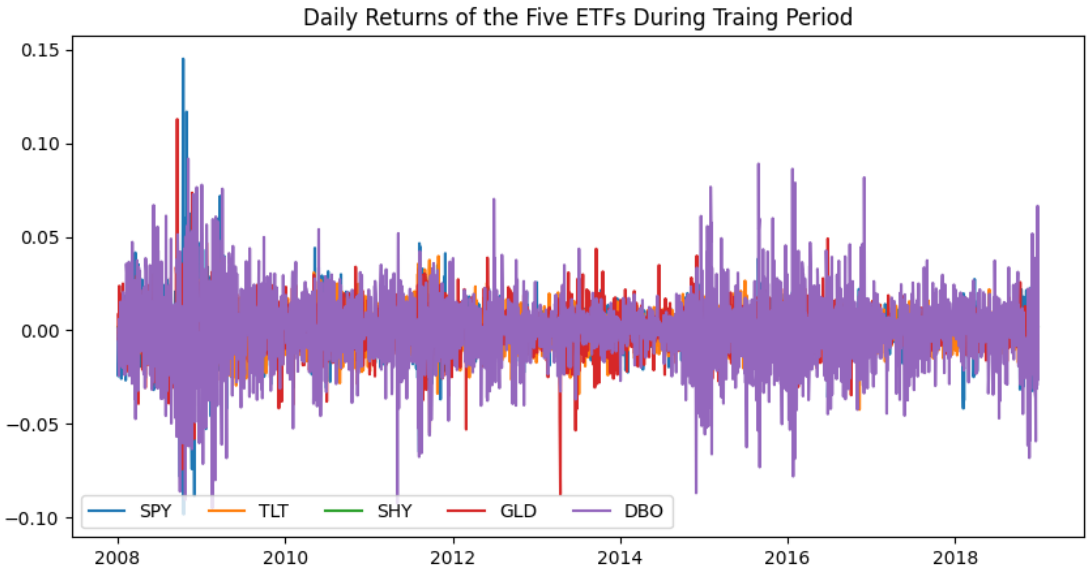
- The five assets are NOT that correlated.
- The largest correlation is 0.606 between 20+ Year Treasury Bond (TLT) and 1-2 Year Treasury Bond (SHY), which can be expected.
- Treasury Bond is negatively correlated with both S&P500 and Crude Oil.

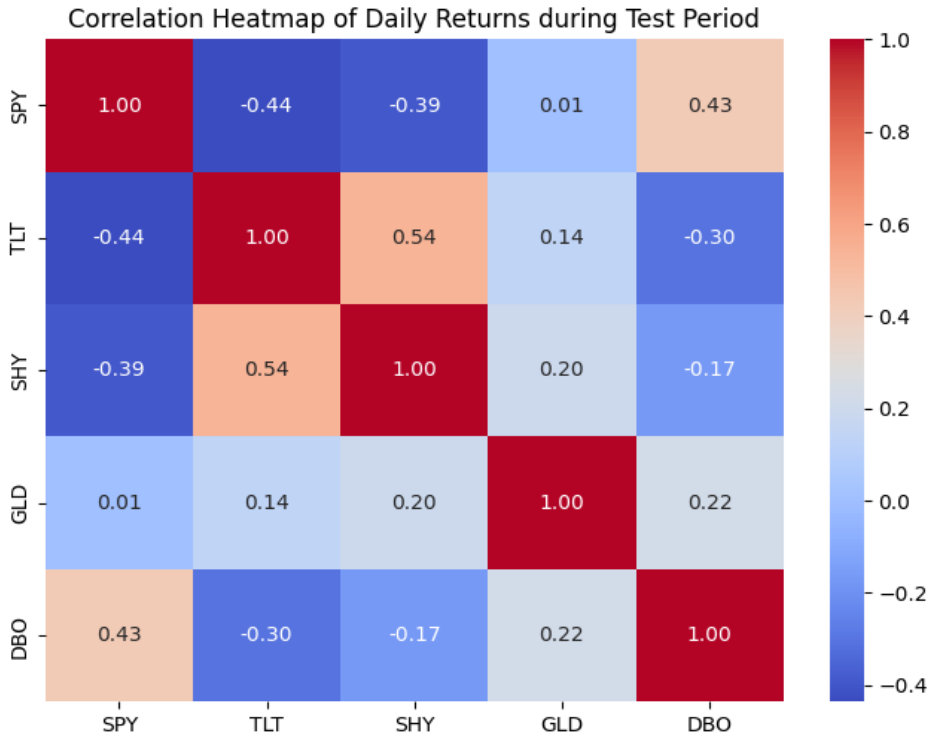
For better model performance, we want to make sure the training data has similar variance with the test data. To this end, we take a look at the VIX levels since 2000.



From the timeplot of VIX, we can see the only time that matches the level of high volatility in 2018 is around 2009. For better OOT evaluation of the trained neural networks, we opt to select **the period between 2008 and 2018 as training period**.

Again, we visualize the daily returns and the correlations of the five ETFs over the training period:





Those conclusions from visualization during the test period still hold for the training period.

Step 2:

Model Design

We have explored using features such as 5-day, 10-day, 25-day and 60-day cumulative returns in addition to 1-day return to predict the 25-day ahead return. However, we find that using a 2-layer LSTM model on only 1-day returns can already be able to predict the label pretty well. Therefore, we choose to stick with only 1-day return data as input.

We set the window size as 30, which results in 2742 samples for training and 1203 samples for testing.

We have manually tweaked the LSTM architecture in order to train the model properly. We find that the following architecture is simple yet powerful enough for this task:

Model: "sequential"

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 30, 50)	10400
lstm_1 (LSTM)	(None, 50)	20200
dense (Dense)	(None, 20)	1020
dense_1 (Dense)	(None, 10)	210
dense_2 (Dense)	(None, 1)	11

Total params: 31841 (124.38 KB)
Trainable params: 31841 (124.38 KB)
Non-trainable params: 0 (0.00 Byte)

The chosen LSTM Model Architecture

We did have a hard time training the models at first, so overfitting has become a lesser problem to worry about. After trial and error, we find these model designs helpful in order to be able to train the model:

- Use a simpler network architecture since the number of training data is quite limited in comparison to the number of trainable parameters.
- Do not rescale Y. This decision is based on the EDA result that the Y labels are in a narrow range and is zero-mean reverting. We do scale the input data though.
- Do not use dropout.
- Use mean squared error as loss, not mean absolute error.

Some of the designs are unlike those that are introduced in the lesson [2] or the article [3], yset we find it hard to train the models otherwise.

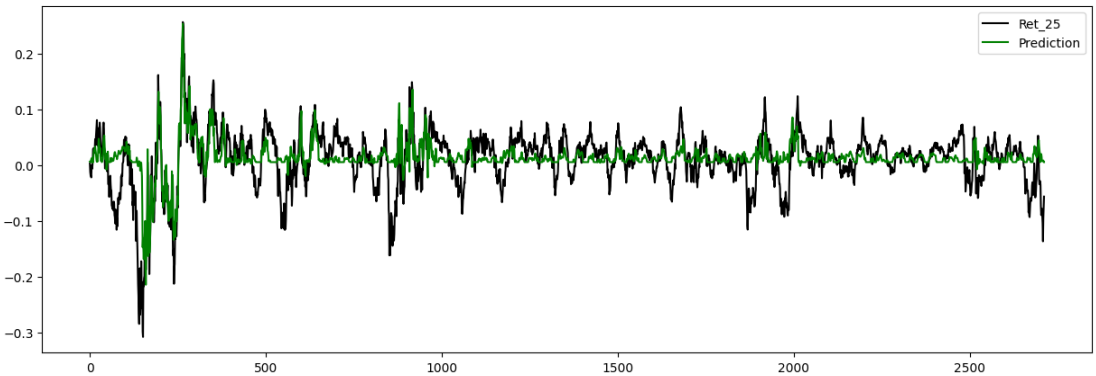
Model Training and Testing

To compare training on different ETFs, we first train each model for 100 epochs and compare their performances in terms of the following metrics:

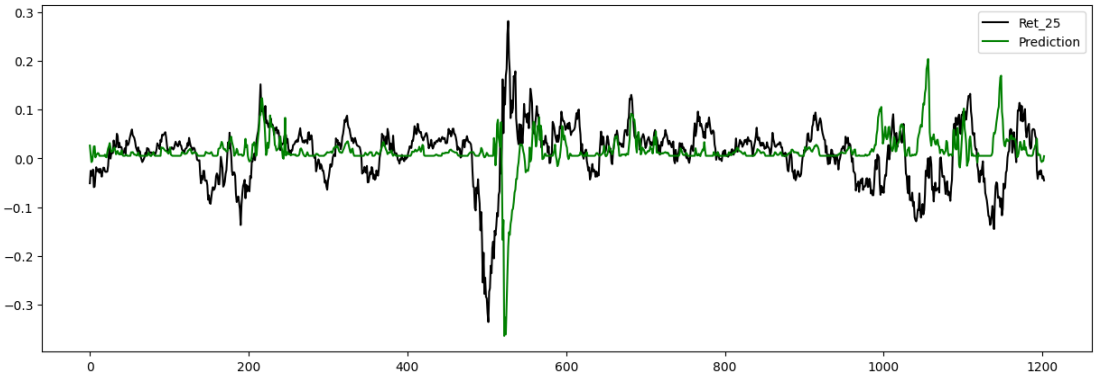
- R-squared (and R-squared OOT). We need to compare relative MSE since we do not rescale the labels.
- Sign accuracy (agreement). Though it is a regression problem, we want fewer mistakes on the direction of asset price movement as we need to take long or short position based on model prediction.

The models are trained using an Adam optimizer with a learning rate of 1e-3. Default batch size is set as 32.

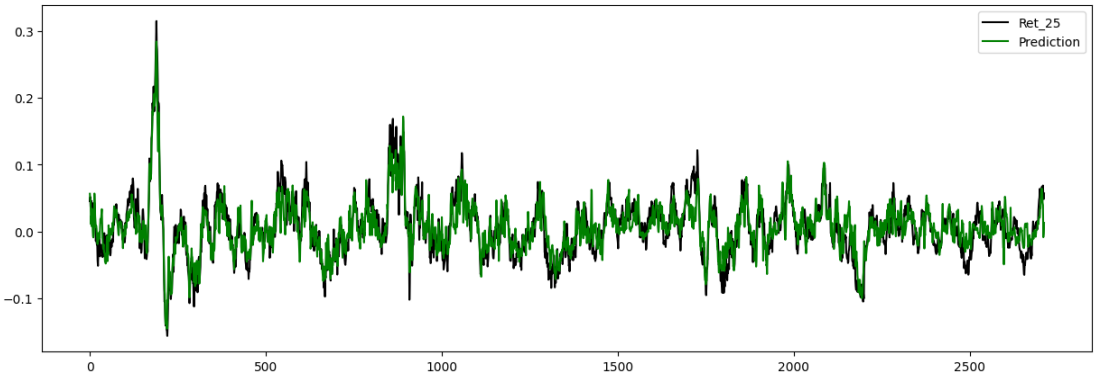
The following figures are the trained model predictions after 100 epochs for each of the five ETFs.



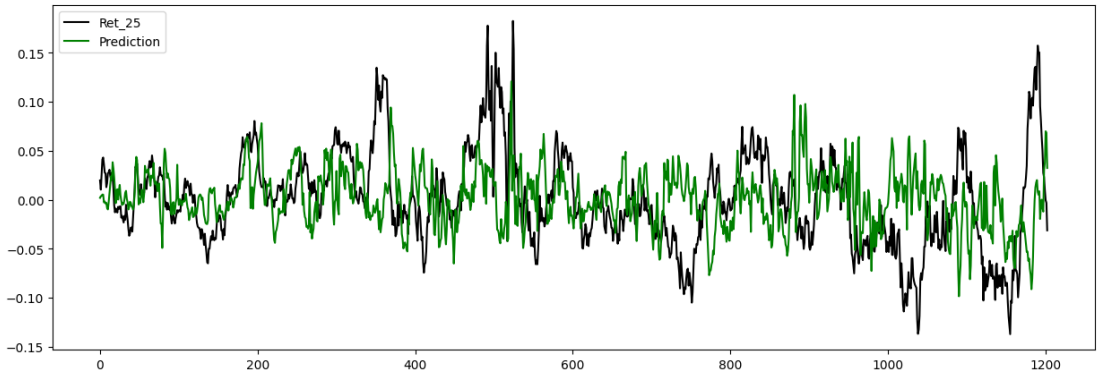
SPY Model Predictions on Training Set (100 Epochs)



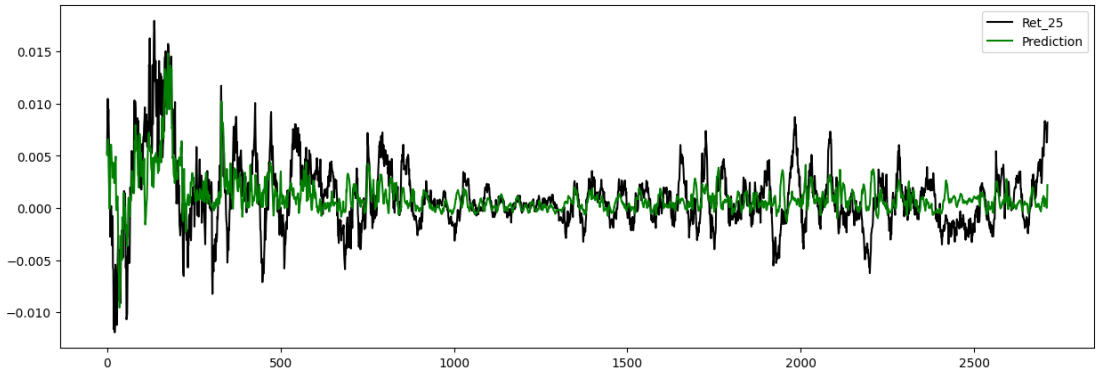
SPY Model Predictions on Test Set (100 Epochs)



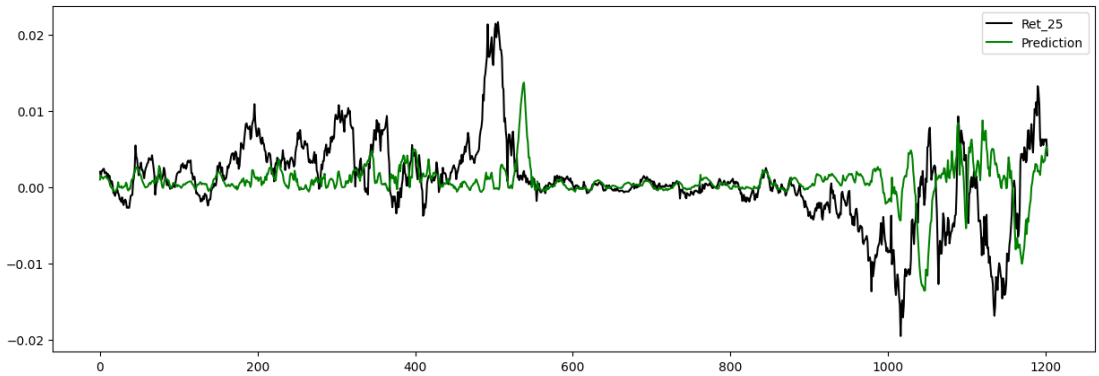
TLT Model Predictions on Training Set (100 Epochs)



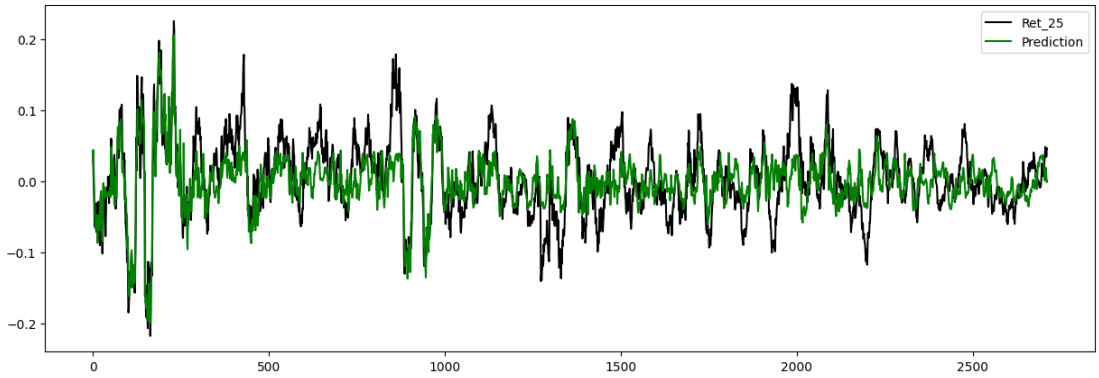
TLT Model Predictions on Test Set (100 Epochs)



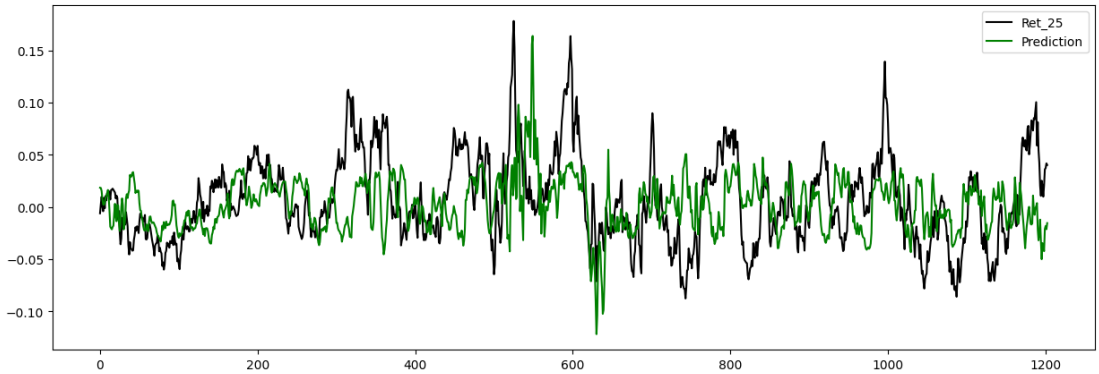
SHY Model Predictions on Test Set (100 Epochs)



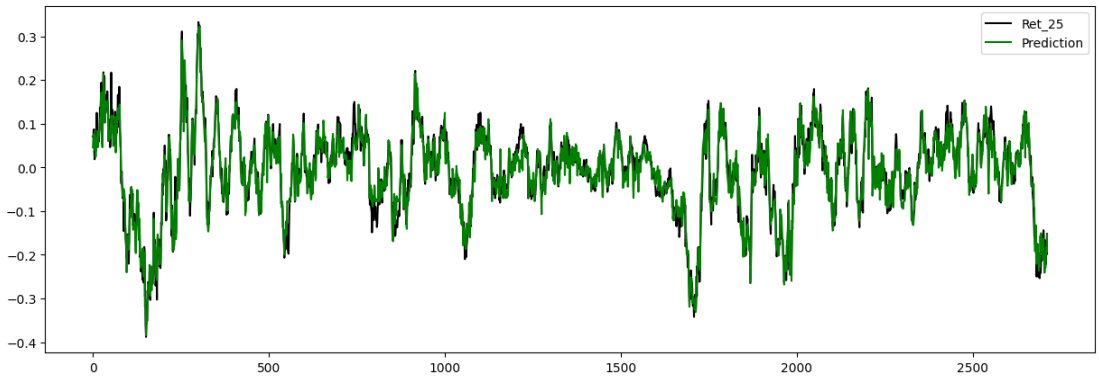
SHY Model Predictions on Test Set (100 Epochs)



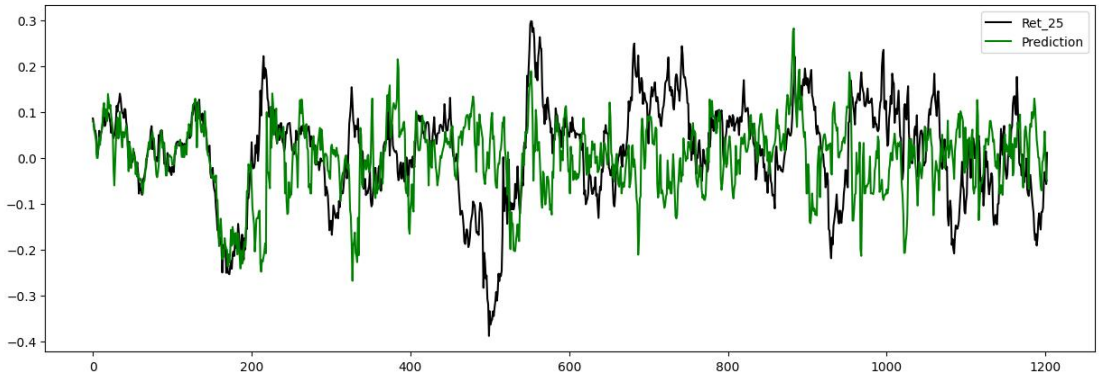
GLD Model Predictions on Test Set (100 Epochs)



GLD Model Predictions on Test Set (100 Epochs)



DBO Model Predictions on Test Set (100 Epochs)



DBO Model Predictions on Test Set (100 Epochs)

For SPY, increasing the batch_size from 32 to 64 helps. And for SHY, since its TSE is so small, we change the learning rate from 1e-3 to 1e-4 to facilitate training. For others, batch_size 64 and learning rate 1e-3 are chosen throughout. The measured metrics are summarized in the following table.

ETF	**Train TSE**	**Train R-squared**	**Test R-squared**	**Train Sign Agreement**	**Test Sign Agreement**
SPY	0.00256	0.3549	-0.5685	0.7091	0.6534
TLT	0.00199	0.7553	-0.2762	0.7998	0.5478
SHY	0.00001	0.3035	-0.2063	0.6585	0.6010
GLD	0.00291	0.4764	-0.1404	0.6969	0.5844
DBO	0.00944	0.9396	-0.3570	0.9174	0.5428

R-squared and Sign Agreement of Each ETF Model after 100 Epochs

Note that for SPY and SHY, after 100 epochs, R-squared is still less than 40% of the TSE. It suggests that models for SPY and SHY are underfitting while those for DBO and TLT are overfitting.

Then we try to control overfitting by early-stopping: we monitor the performances every 10 epochs, and the training is stopped if the MSE is less than 60% of the TSE.

The measured metrics are summarized in the following table.

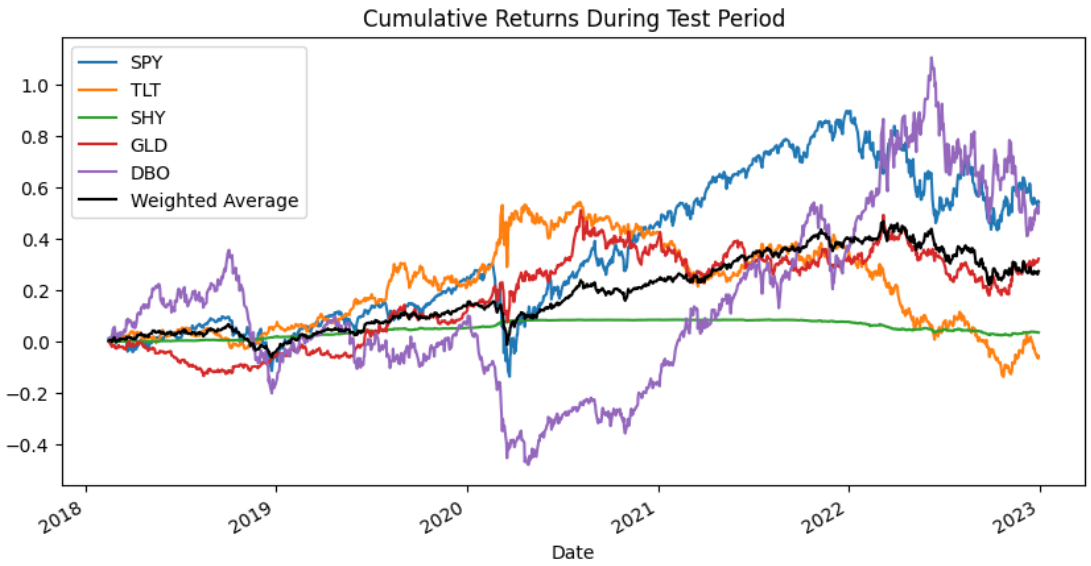
ETF	Epochs	Train R-squared	Test R-squared	Train Sign Agreement	Test Sign Agreement
SPY	100				
TLT	70	0.4548	-0.0293	0.6829	0.5743
SHY	100				
GLD	80	0.3342	-0.0909	0.6217	0.5686
DBO	60	0.5660	-0.0232	0.7134	0.6242

Trained Epochs, R-squared and Sign Agreement of Each ETF Model after Early Stopping

Notice that for both TLT and DBO, the model performances on the test set are much better in terms of both OOT R-squared and Sign Agreement.

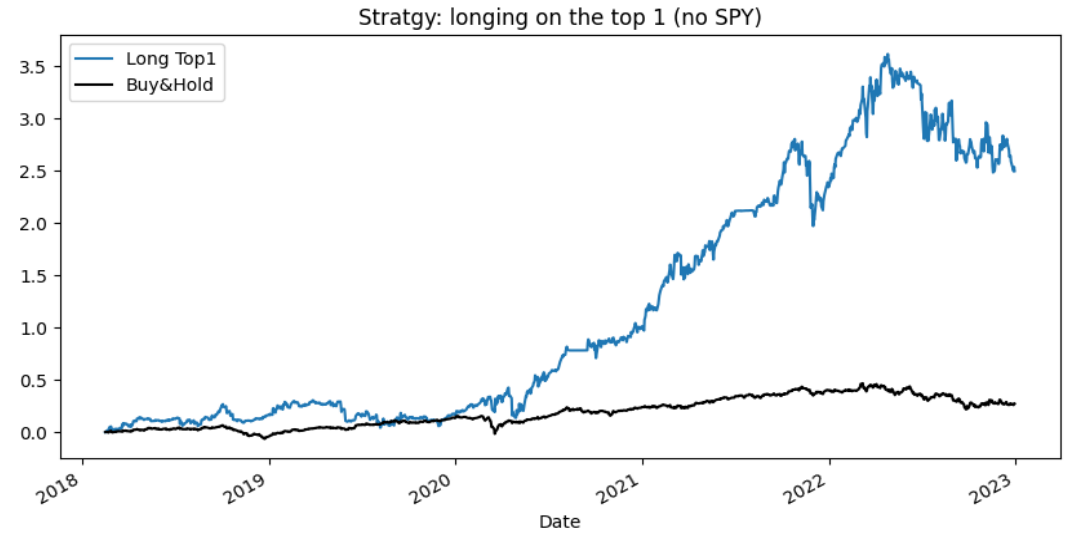
Trading Stratigies

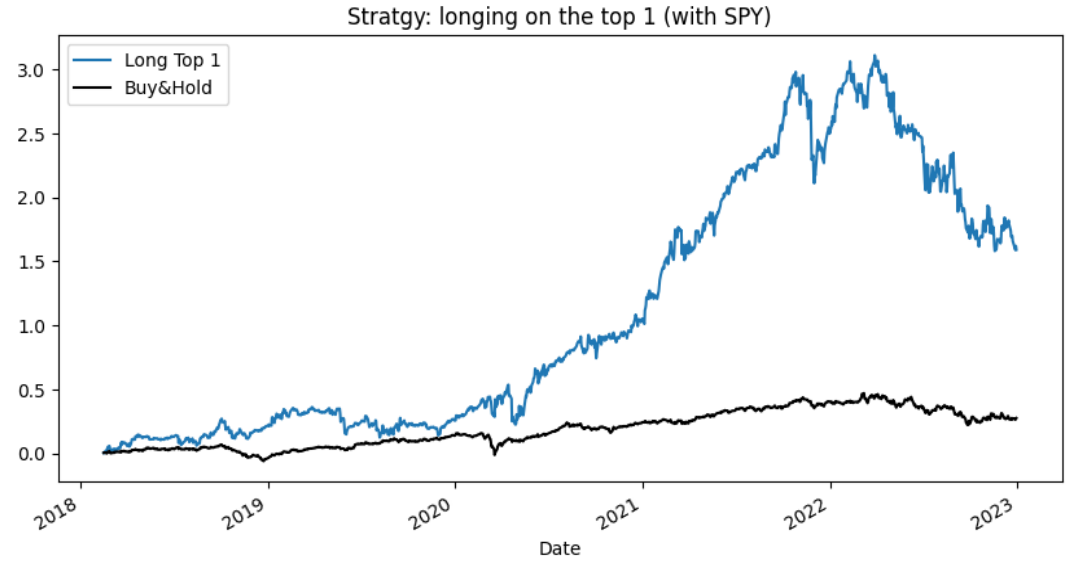
For buy&hold strategy of the weighted average of the five ETFs:

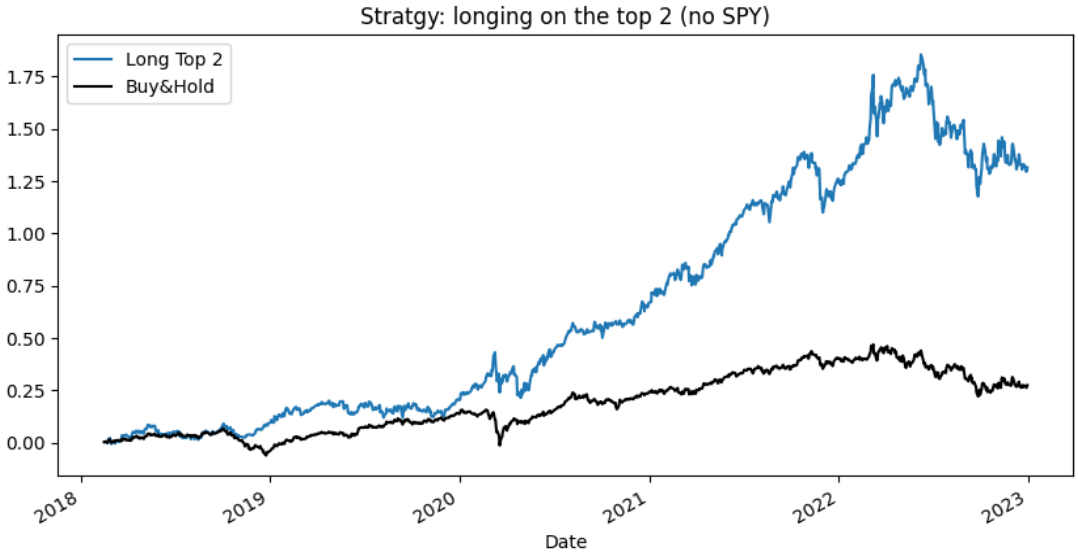


The plots start at 2018-02-15, which is the 31st data point in the test set. The final cumulative return is only 27.4%.

Now we draw the backtest result of having long positions in only the top picks (based on predicted 25-day ahead returns). We consider either Top 2 or Top 1. Additionally, as the model hardly trains on SPY data, we opt to drop SPY from the pool since its predictions on SPY won't be reliable.







Longing on top 1 is the clear winner. The test result seems wonderful. The highest cumulative return is 350% during the test period, during which the highest cumulative return of an individual ETF is

DBO (Crude Oil) with 100% return. The results may largely be attributed to accurately predicting the future return of DBO, which experienced a surge since March, 2020 to mid-2022.

If the predictions are accurate enough, longing on Top 2 only averages their cumulative returns. That is perhaps why Top 2 option is not as good as Top 1 option.

Step 3: Multi-head LSTM for ETFs

Model Design

Since we are only using one feature for each ETF, we can simply stack them into one input for the model. We just change the last layer to Dense(5) to predict five real numbers at the same time. The other model designs are kept the same:

Model: "sequential"

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 30, 50)	11200
lstm_1 (LSTM)	(None, 50)	20200
dense (Dense)	(None, 20)	1020
dense_1 (Dense)	(None, 10)	210
dense_2 (Dense)	(None, 5)	55

Total params: 32685 (127.68 KB)
Trainable params: 32685 (127.68 KB)
Non-trainable params: 0 (0.00 Byte)

Network Architecture of the Multi-head LSTM

Model Training and Testing

We compare the performance of models trained after 100, 50, and 20 epochs, and the results are summarized in the following table.

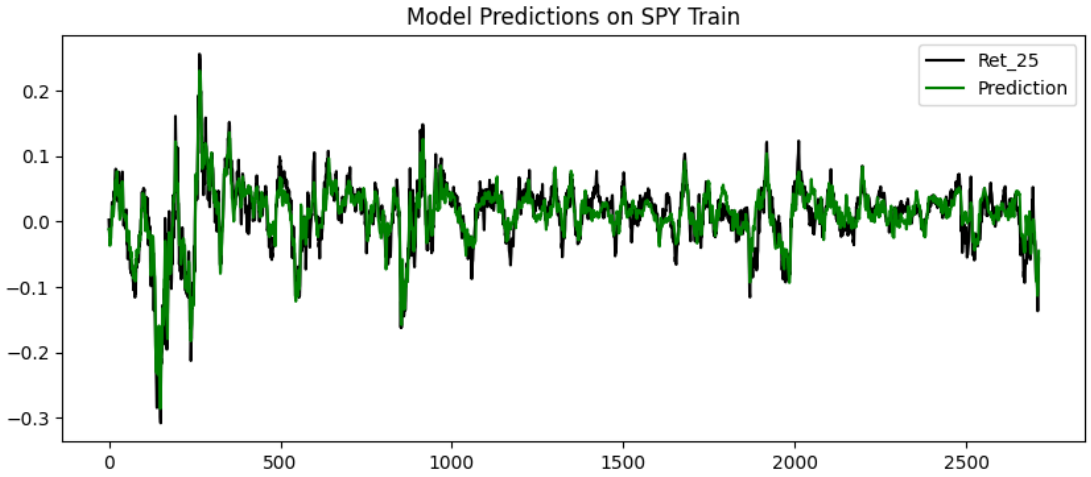
ETF	Train TSE	Train 100 Epochs	Train 50 Epochs	Train 20 Epochs	Test 100 Epochs	Test 50 Epochs	Test 20 Epochs
SPY	0.00256	0.8292	0.6065	0.5096	-0.226	-0.1300	-0.0730
TLT	0.00199	0.6524	0.5453	0.3123	-0.1216	-0.2657	-0.0048
SHY	0.00001	0.3319	0.3037	-1.2508	-0.0355	-0.0953	-0.4333
GLD	0.00291	0.8841	0.5376	0.2578	-0.8318	-0.5546	-0.2335
DBO	0.00944	0.9614	0.8752	0.5338	-0.0392	-0.1878	-0.0349

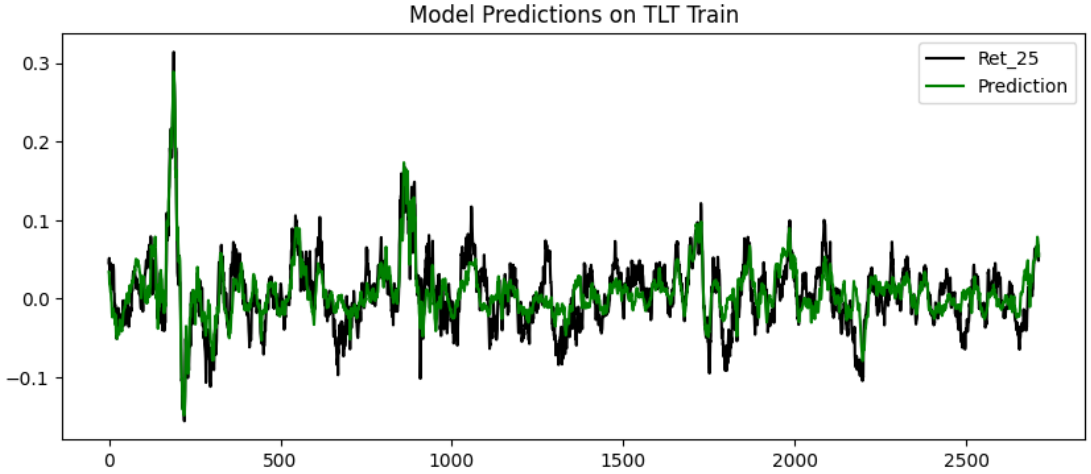
R-squared of the Multi-head LSTM Trained after Different Epochs

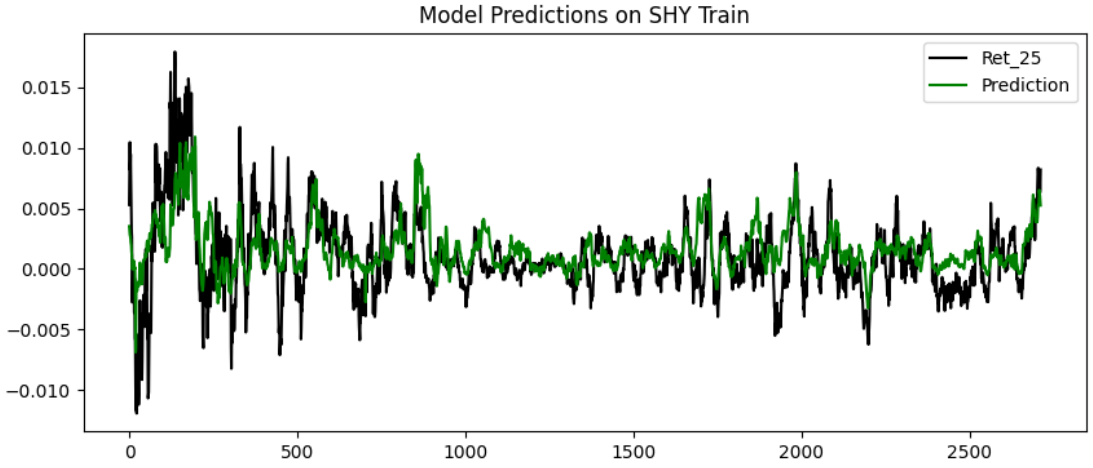
ETF	Train TSE	Train 100 Epochs	Train 50 Epochs	Train 20 Epochs	Test 100 Epochs	Test 50 Epochs	Test 20 Epochs
SPY	0.00256	0.8680	0.7729	0.7349	0.6858	0.6367	0.6492
TLT	0.00199	0.7430	0.6914	0.5852	0.5320	0.4830	0.4838
SHY	0.00001	0.6681	0.6243	0.5258	0.6259	0.5785	0.4946
GLD	0.00291	0.8838	0.7127	0.5815	0.6010	0.4938	0.4830
DBO	0.00944	0.9333	0.8706	0.6976	0.5827	0.5428	0.4572

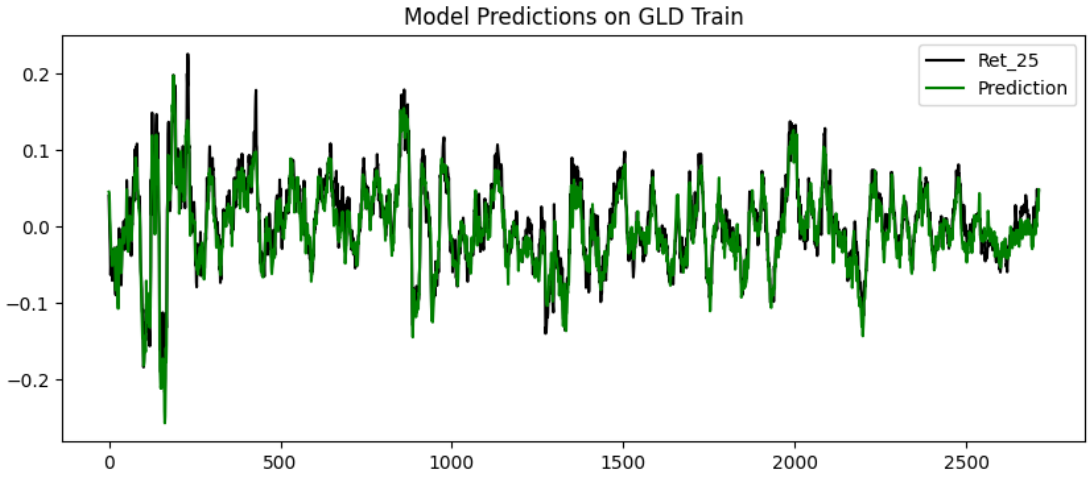
Sign Agreement of the Multi-head LSTM Trained after Different Epochs

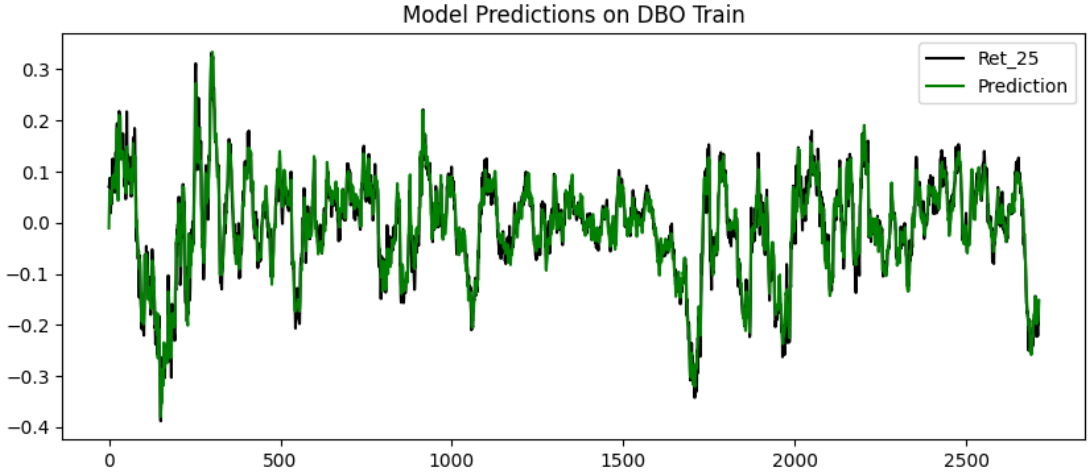
Even the multi-head model seems to be overfitting the training data, it still has the best accuracy of "sign agreement." Therefore, we still choose this model to evaluate trading strategies in the next section. It predictions on the training and test set for each individual ETF are shown below:

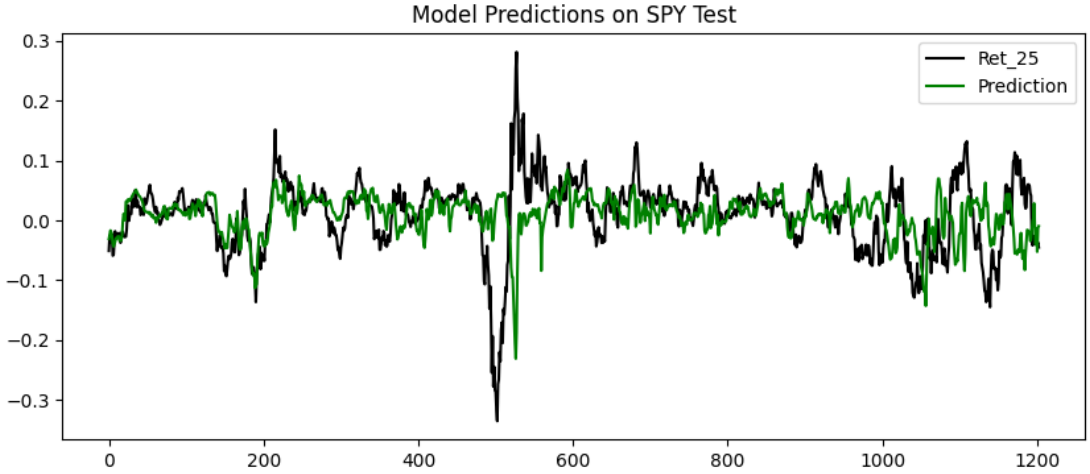

Model Predictions on SPY Train
Ret_25
Prediction
0.2
0.1
0.0
−0.1
−0.2
−0.3
0
500
1000
1500
2000
2500

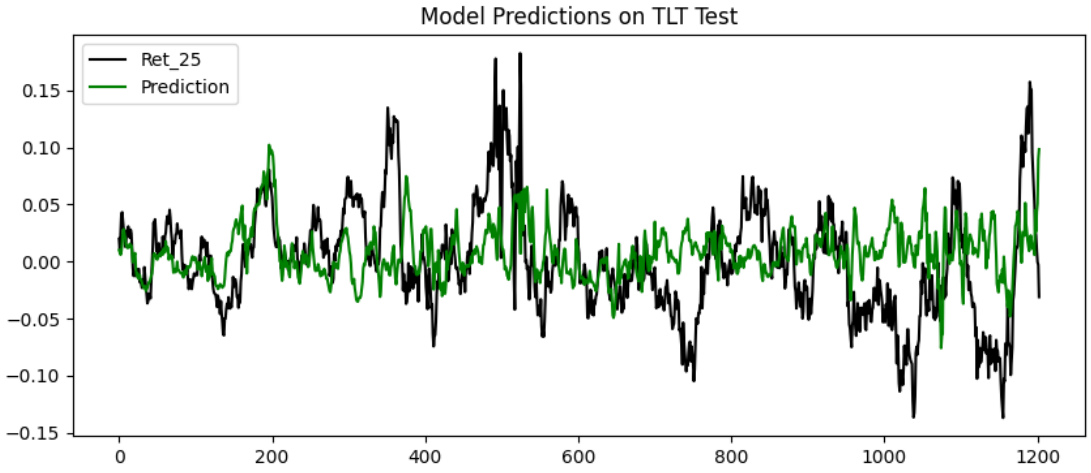

Model Predictions on TLT Train
Ret_25
Prediction
0.3
0.2
0.1
0.0
−0.1
0
500
1000
1500
2000
2500

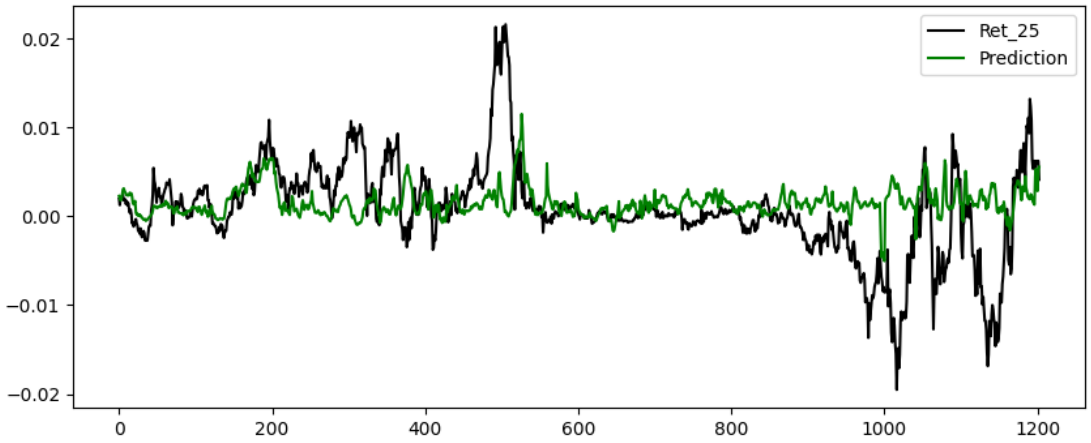

Model Predictions on SHY Train
Ret_25
Prediction
0.015
0.010
0.005
0.000
−0.005
−0.010
0
500
1000
1500
2000
2500

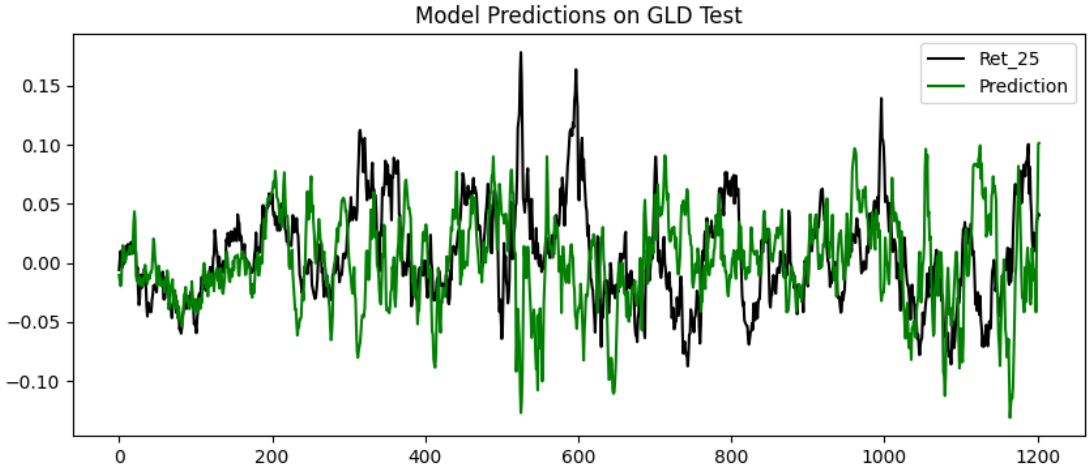

Model Predictions on GLD Train
Ret_25
Prediction
0.2
0.1
0.0
−0.1
−0.2
0
500
1000
1500
2000
2500

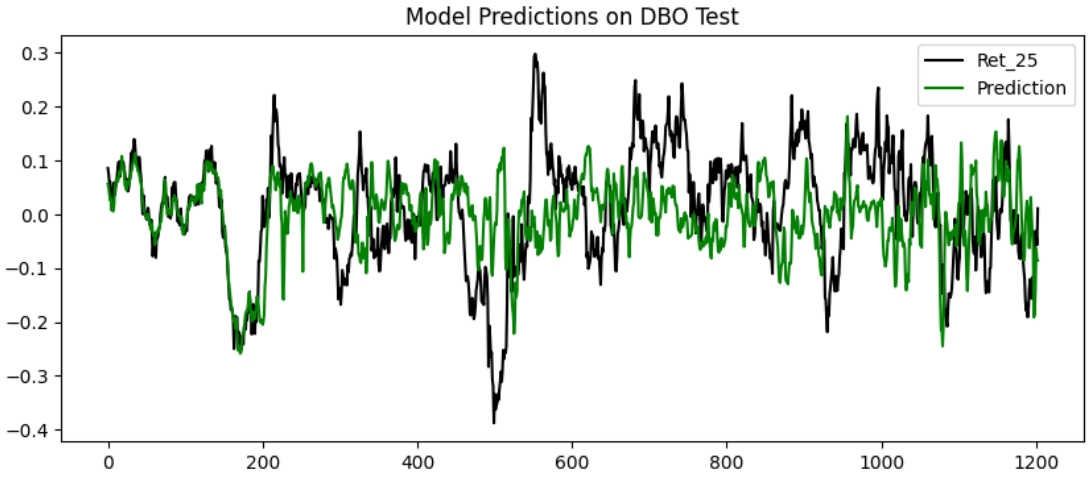

Model Predictions on DBO Train
Ret_25
Prediction
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
−0.4
0
500
1000
1500
2000
2500


Model Predictions on SPY Test
Ret_25
Prediction
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
0
200
400
600
800
1000
1200


Model Predictions on TLT Test
Ret_25
Prediction
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
0
200
400
600
800
1000
1200

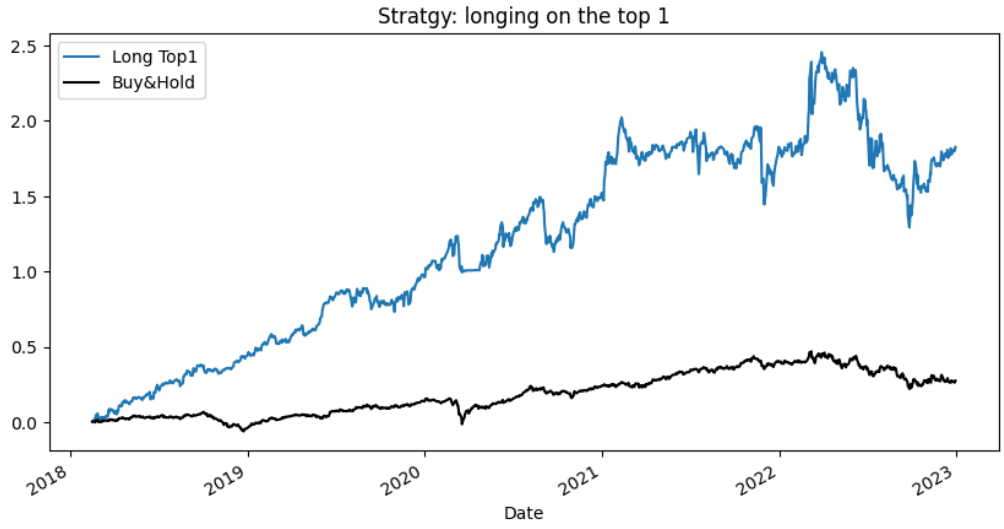

Model Predictions on SHY Test
Ret_25
Prediction
0.02
0.01
0.00
−0.01
−0.02
0
200
400
600
800
1000
1200

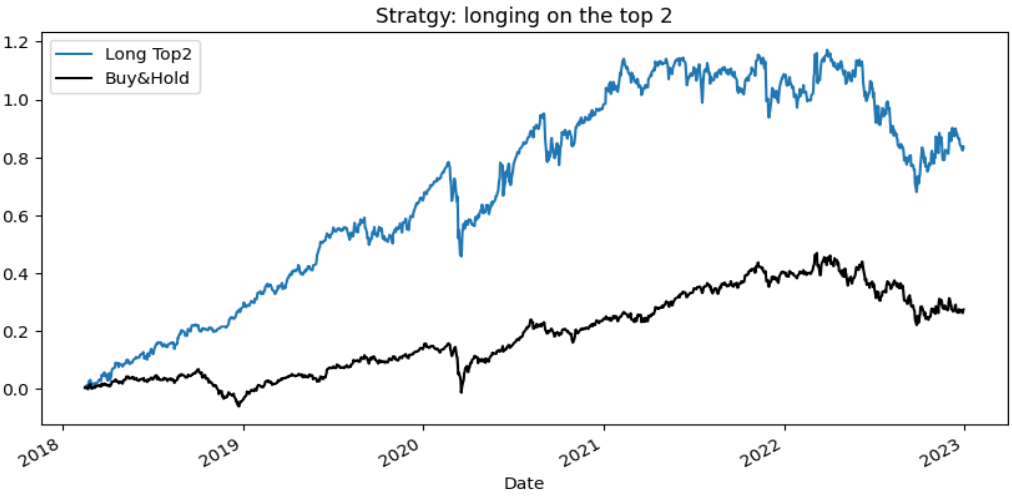

Model Predictions on GLD Test
Ret_25
Prediction
0.15
0.10
0.05
0.00
−0.05
−0.10
0
200
400
600
800
1000
1200

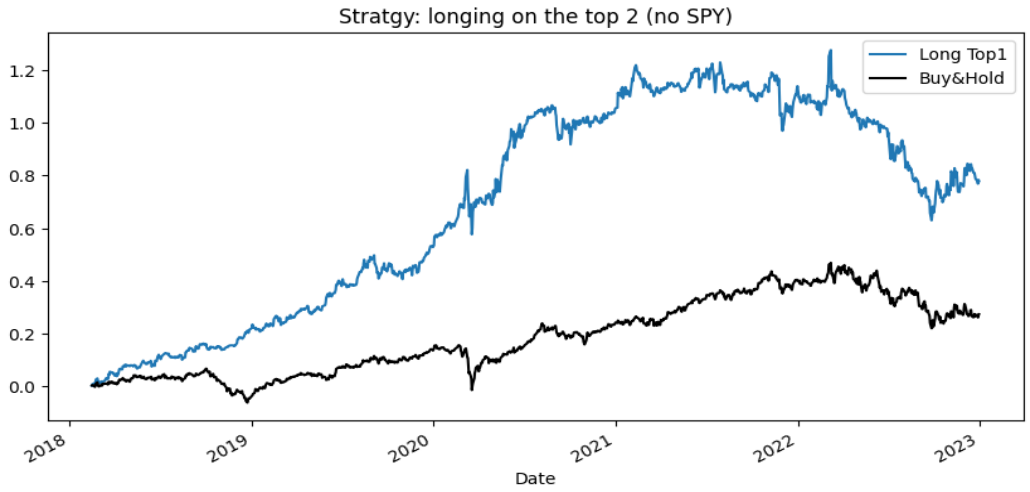


Trading Strategies

We tested four strategies: long Top 1 or Top 2 assets based on predictions of the multi-head LSTM, w/o SPY for comparison with individual ones.







Again, choosing only Top 1 based on predicted 25-day ahead return is better than choosing Top 2 interns of highest overall cumulative return at the end point.

The results are less desirable than the counterpart based on predictions of five individually trained models.

Step 4: Discussion

We can do some PCA analysis:

```
##  PCA
pca  =  PCA(n_components=5)
pca.fit(test_df.values)
print(pca.components_.T)
#  Cumulative  share  of  the  variation  of  the  data  predicted  the  factors
print(np.cumsum(pca.explained_variance_ratio_))
```

```
[[-0.31781552 -0.88387045 -0.28021374 -0.19806697  0.00400698]
 [ 0.12743865  0.31715028 -0.7120577  -0.61140986  0.04823801]
 [ 0.00627608  0.01180722 -0.04639316 -0.01732948 -0.99868342]
 [-0.04613728  0.04880346 -0.64143257  0.76404889  0.01682634]
 [-0.93839472  0.34009931  0.02942833 -0.05363243 -0.00231271]]
[0.60893867 0.79456694 0.93213207 0.99943491 1.        ]
```

PCA on the test data

From PCA, we can see that the first three principal components are able to explain 93%+ of the variation among the daily returns of the five assets. It suggests that there is some comovement that can be modelled together using a single LSTM model.

This is an advantage as opposed to using five LSTM modes to model individual ETF, which may explain why it is much easier to train the single multi-head LSTM. This is especially true for SPY. For whatever reason, it is very hard to train the LSTM model for SPY individually in our experiments, yet the multi-head LSTM has no problem in learning its patterns.

The second advantage of using a multi-head architecture is a fewer number of parameters to train due to shared parameters, which is especially useful when training data is limited. This resembles the convolution kernels in CNN.

However, a potential disadvantage is that the predictions of the multi-head model on individual ETF are intertwined: one may affect another in an undesirable way. The evaluation of the top N long-only trading strategies suggest that training separately may yield a better result. The models for each ETF can be tuned separately using different hyperparameters, which gives more flexibility to control the bias-variance trade-off.

Also we find that the multi-head LSTM model has a very bad performance on the test set of GLD, as opposed to the individual one. It is perhaps due to that the daily returns of GLD have a smaller TSE and a narrower range compared to other assets. This problem may be mitigated if we have scaled

the labels to the same range.

If time permits, we would like to explore these directions:

- Evaluating the trading strategies on other periods.
- Choose different rebalance starting date or interval when evaluating trading strategies.
- Considering short positions when designing trading strategies.
- Calculate more metrics when evaluating and comparing trading strategies such as Sharpe ratio or maximum retreat, etc.
- Scale the labels of each ETF before training the multi-head LSTM.
- Explore the network architectures and hyperparameters more.

Reference

1. Exploratory Data Analysis. In: The Concise Encyclopedia of Statistics. Springer, New York, NY. https://doi.org/10.1007/978-0-387-32833-1_136
2. Lecture note for Derivative Pricing, Module 5 Lesson 2. World Quant University, 2023.
3. Machine Learning to Predict Stock Prices, https://towardsdatascience.com/predicting-stock-prices-using-a-keras-lstm-model-4225457f0233