Trend-Following Strategies in Futures Markets

A Machine Learning Approach

Art Paspanthong
Divya Saini
Joe Taglic
Raghav Tibrewala
Will Vithayapalert

Outline

- Overview of Data and Strategy
- Feature Generation
- Model Review
  - Linear Regression
  - LSTM
  - Neural Network
- Portfolio Results
- Conclusion

Overview of Data and Strategy

Our Task

Problem Statement:

Replicate and improve on the basic ideas of trend following.

Datasets

Datasets of Commodities Futures

Energy	Metals	Agriculture
Crude Oil	Gold	Corn
	Silver	Wheat
	Copper	Soybean

Total Assets Considered

42 Different Assets

Filtered out by liquidity

Total Assets Traded

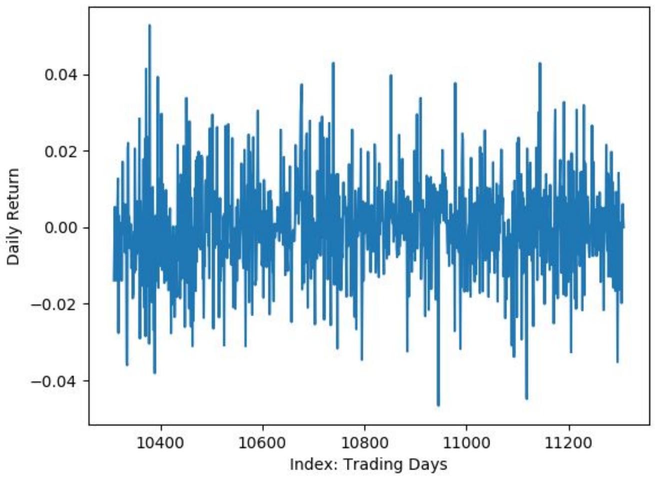
36 Different Assets

Time Frame: 1 - 6 months Expiration

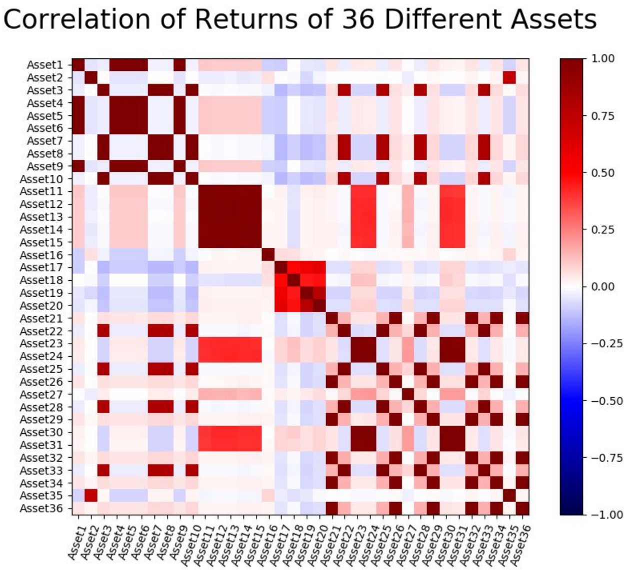
Source: Quandl

Data Exploration

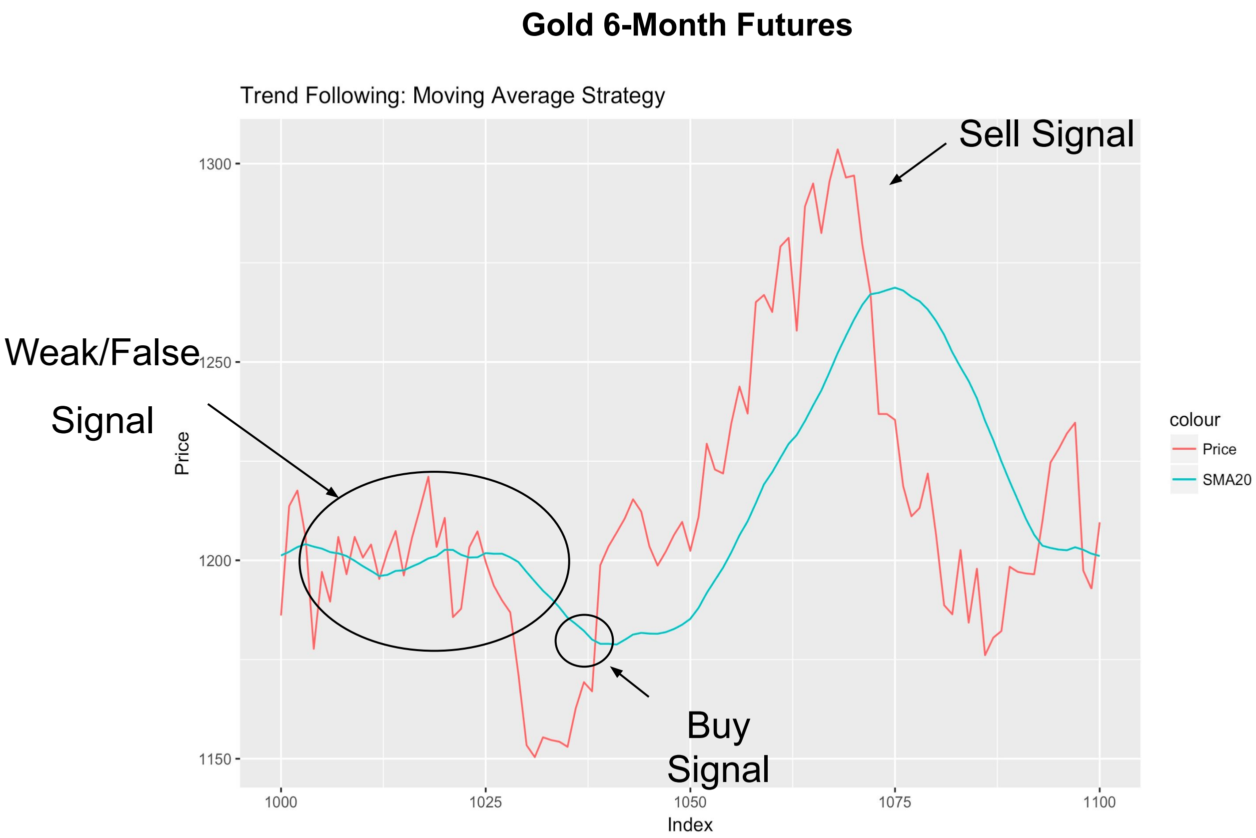
Volatility Plot



Correlation Plot



Data Exploration



Strategy

Traditional Trend Following

Not about prediction

Involves quickly detecting a trend and riding it, all while managing when to exit at the right moment

Our Approach

Use traditional trend following indicators to predict returns with machine learning techniques

Use a portfolio optimizer to weight assets using the predicted returns

Adhere to traditional investment practice with stop-loss

Feature Generation

Trend-following Features

Continuous

Simple Moving Average **(SMA)** *with 5, 10, 15, 20, 50, 100 days lookback window*
Exponential Moving Average **(EMA)** *with 10, 12, 20, 26, 50, 100 days lookback window*
Moving Average Convergence Divergence **(MACD)** *= 12-day EMA - 26-day EMA*
Momentum Indicator *with 5, 10, 15, 20, 50, 100 days lookback window*
Day Since Cross *indicates the number of days since crossover*
Number of days ‘up’ - ‘down’ *with 5, 10, 15, 20, 50, 100 days lookback window*

Categorical

SMA Crossover indicator variables
EMA Crossover indicator variables
MACD Crossover indicator variables

(+1 = crossover with buy signal, 0 = no crossover, -1 = crossover with sell signal)

Response

Next Day Return = $[\, P_{t+1} - P_t \,] / P_t$

Model Review

Model Review

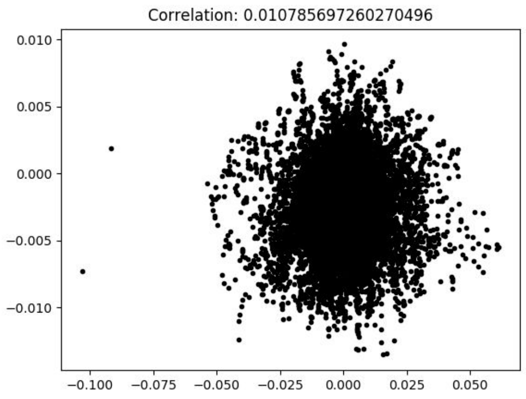
Linear Regression

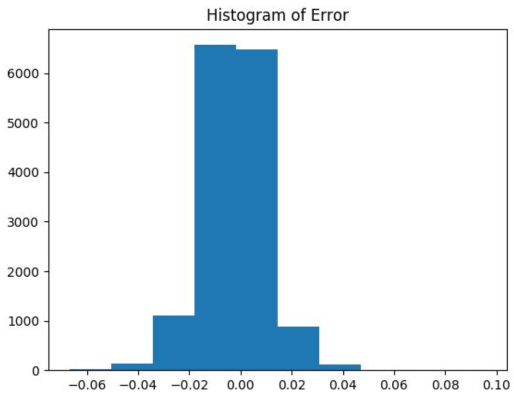
Linear Regression Model

Use a linear regression (ordinary least squares) for each asset on the available training data

Advantages:
- Simple, easy to understand
- Fits decently well to linear data

Disadvantages:
- Overfits easily
- Cannot express complex or nonlinear relationships





Plot of Predicted vs. Actual Values and Error Histogram for Linear Regression, daily return

Portfolio over 2017-2018

Train MSE: 2.187 E-04
Test MSE: 1.47 E-04

Linear Regression Model

Covariate Name	Beta Value	P Value
EMA 10	**4.84**	.004
EMA 20	2.45	.219
SMA 100	**0.06**	.004
EMA 100	0.03	.17
SMA 15	0.01	.817
SMA 10	-0.002	.981
SMA 20	-0.008	.866
SMA 50	-0.04	.149
SMA 5	**-0.07**	0.044
EMA 50	-0.29	.235
EMA 26	-0.45	.749
EMA 12	**-6.38**	.016

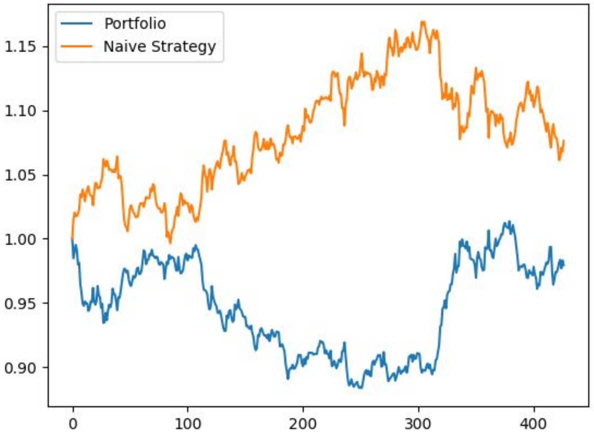
- Exponential Moving Averages are generally better predictors than simple moving averages
- Recent trends are most significant
- Change of sign between EMA 10, 12, 20 suggests importance of recent crosses
- Significance of longest-range feature shows importance of long-term trendiness

Linear Regression Model with Lasso

Lasso Regularization gets rid of some of the overfitting of a linear regression
Accomplished by automatically selecting only more important features

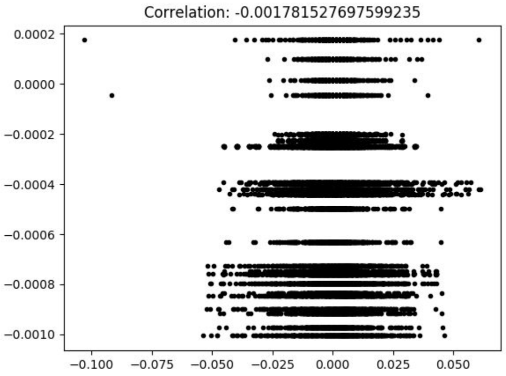
Advantages:
- Less likely to overfit
- Less prone to noise

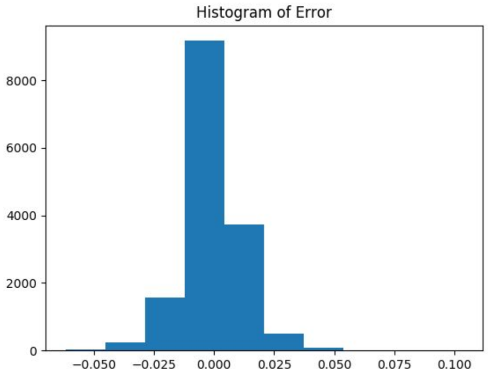
Disadvantages:
- Does not solve complexity issue
- Can lose expressiveness



Portfolio over 2017-2018

Train MSE: 2.281 E-04
Test MSE: 1.353 E-04





Predicted vs. Actual values and Error Histogram for Lasso Regression, Daily Returns

Linear Regression Model with 5-Day Returns

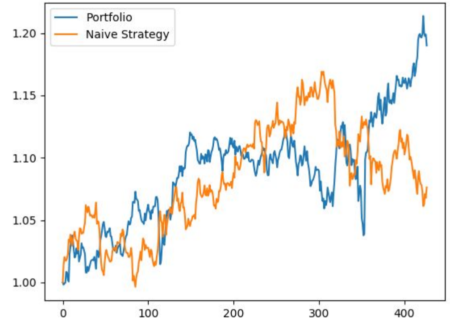
Determine whether the model can better predict returns over a longer time frame

Why is this desirable?

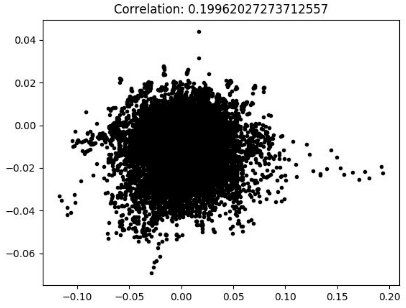
In a non-ideal trading system there are frictions:

- One-day returns are small and may be erased by transaction costs
- Might not enter the position until the next day

Can we reliably predict 5-day returns?



5-day returns are generally about 2-3x larger than 1 day returns, so a roughly 6.5x increase in mean squared error (MSE: 9.47E-04) indicates that the predictions are about equivalent to 1-day predictions.



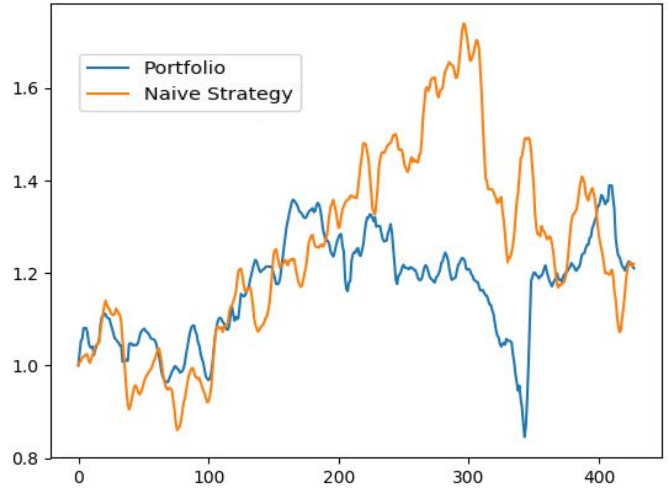
Plot of Predicted vs. Actual Values

Interestingly, the daily returns of this portfolio vs. the naive portfolio are fairly comparable (0.04 % vs. 0.02%) but the 5-day returns are notably better (0.22% vs. 0.07%).

Model Review

RNN: LSTM Model

Long-Short Term Memory (LSTM) Model



Architecture: 3-layer LSTM and one fully-connected layer with linear activation function

Regularization: Dropout, Early Stopping, Gradient Clipping

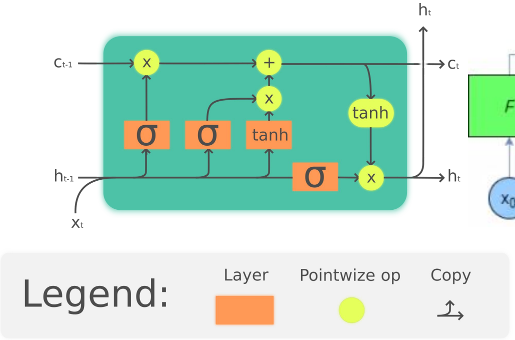
Hyperparameter Tuning: Select the best set of hyperparameters including
Batch Size: 32, 64, 128 Lookback window = 5, 10, 15 days
Optimizer: Adam

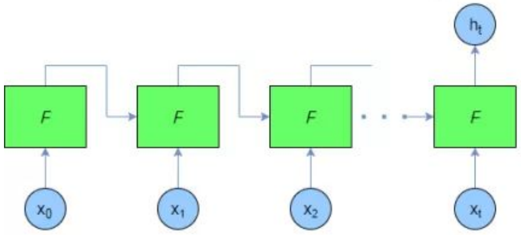
Statistical Results:

Daily Return
MSE: 0.00031
Correlation: -0.016

5-day Return
MSE: 0.0011
Correlation: -0.25

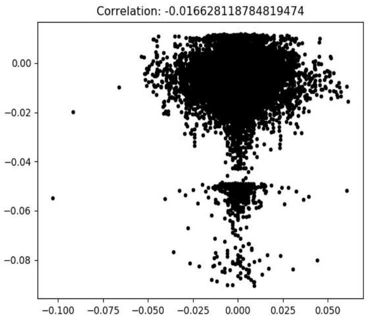
Dataset too small for LSTM to capture the "trend" and perform well

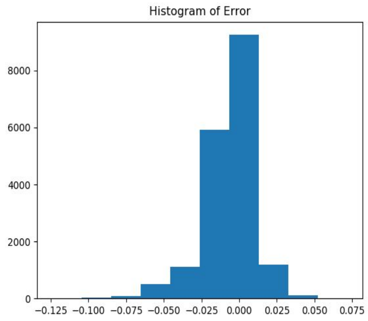


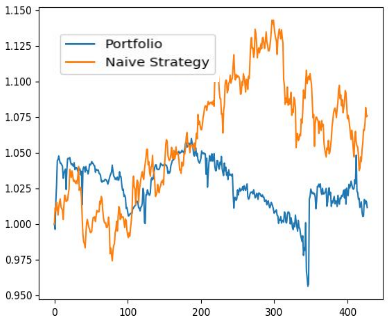


Statistics and Visualizations

Next Day's Return

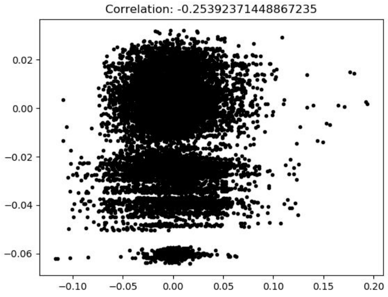


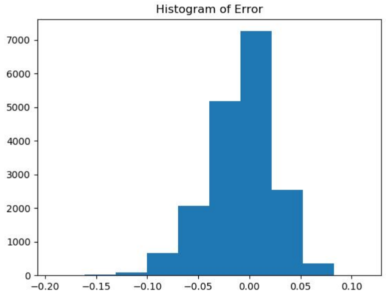


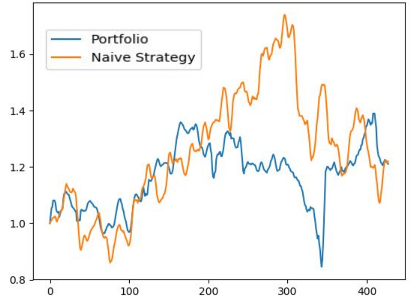


Prediction on further returns is less accurate and more variant

Next 5-Day's Return





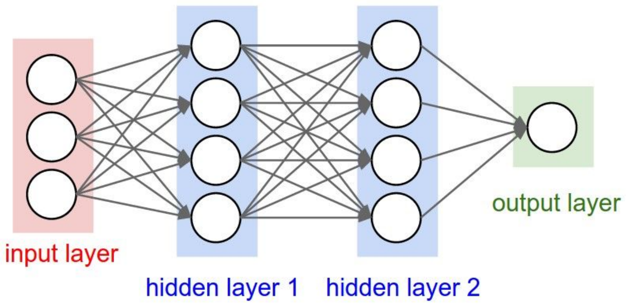


Model Review

Neural Net

Neural Net Model

Architecture: One input layer with 26 input units, two hidden layers with RELU activation functions and one output layer. The neural network is fully connected.



Features Used: A total of 26 input features, given below:

➔ Normalized Simple Moving Average *with 5, 10, 15, 20, 50, 100 days lookback window*
➔ Normalized Exponential Moving Average *with 10, 12, 20, 26, 50, 100 days lookback window*
➔ Normalized Moving Average Convergence Divergence
➔ Crossover data

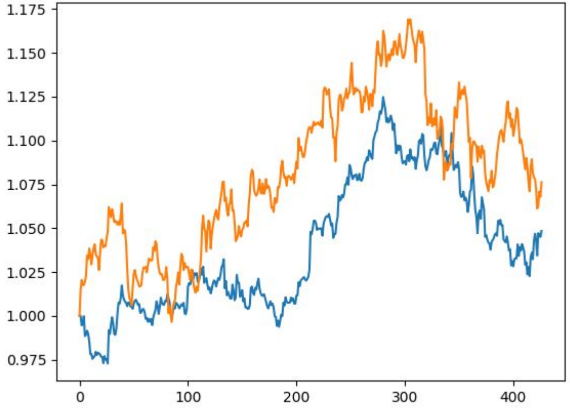
Stopping Criterion: The validation dataset is used for stopping. When the loss difference over the validation set decreases below the convergence error, we stop the training.

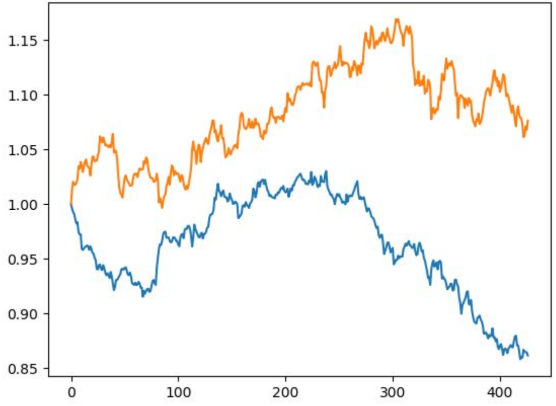
Hyperparameters:

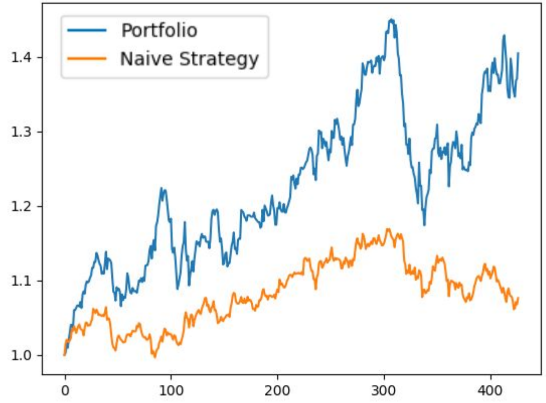
- Learning Rate: 1e-3
- Convergence Error: 1e-6
- Number of units in hidden layers: 50 and 20 respectively
- Optimizer: Adam

Difficulty: The neural net model uses random initialization of the parameters and uses an iterative process (gradient descent) to find the minima. The stochastic nature of the model gives us different results on running the model.

For example, we have these three plots which run the same neural network code but gave us different results:







Neural Net Model

Comparison with Linear Regression

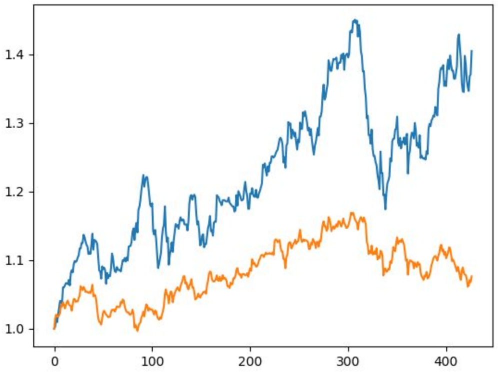
- The linear regression model gives the same result every time but the neural network without activations (which performs simple gradient descent) does not give the same result every time.

- The neural network has been given validation data and the model stops training based on the validation loss, however, the linear regression can theoretically overfit the past data.

- Sometimes the neural network (without activations) performs better than linear regression but sometimes worse. The MSE for linear regression and neural network (without activations) is almost equal.

- However, with RELU activations, it performs worse because of the nature of the data and the nature of RELU.

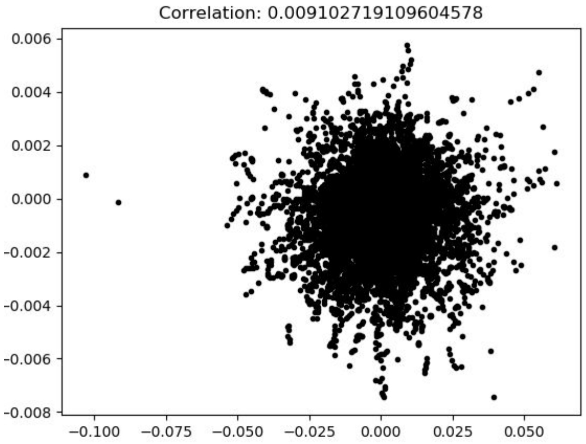
LR



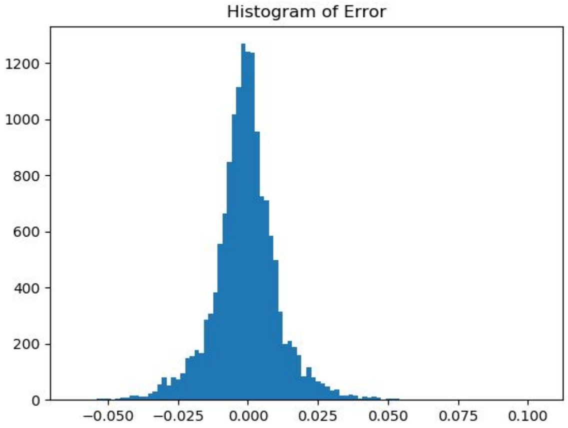
NN

Statistics and Visualizations

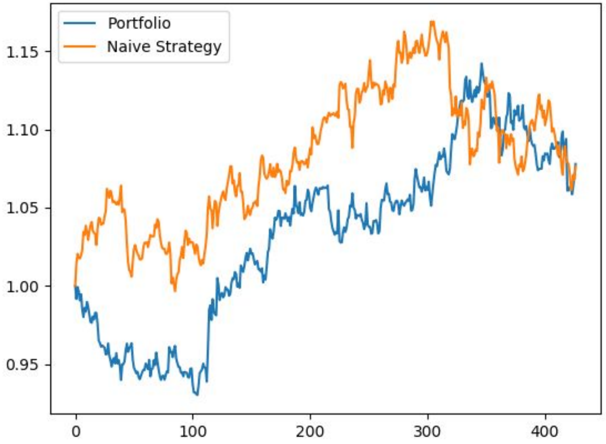
For comparison with the other models, we have used one of neural network models that we trained. The statistics and visualizations for that model have been shown below.



Correlations between all results and all predictions



Histogram of errors



Neural network model with portfolio optimization

Results Summary

Models	MSE on train set	MSE on test set
Linear Regression	**2.187 E-04**	**1.47 E-04**
Lasso Model	**2.281 E-04**	**1.353 E-04**
Neural Net	**2.329 E-04**	**1.358 E-04**
LSTM	**1.34 E-03**	**3.05 E-03**

Portfolio Results

Portfolio Optimization

- Calculate expected returns using model of choice
- Calculate covariance matrix using historical data
- Determine desired variance based on a benchmark
  - Usually an equal-weight long-only portfolio

Objective: max expected portfolio return after transaction costs
Constraints:

Does not leverage portfolio
Expected variance below desired variance

Current Holdings
Expected Returns
Observed Covariance
Desired Variance
Transaction Costs

Trades
New Holdings

Note: Covariance Shrinkage

- Transformed the sample covariance matrix using LeDoit-Wolf Shrinkage
- *Mathematically:* pulls the most extreme coefficients towards more central values, thereby systematically reducing estimation error
- *Intuitively:* not betting the ranch on noisy coefficients that are too extreme

Baseline Strategy

Traditional Trend Following

Enter a position based on **crossovers**[1] and/or **breakouts**[2]

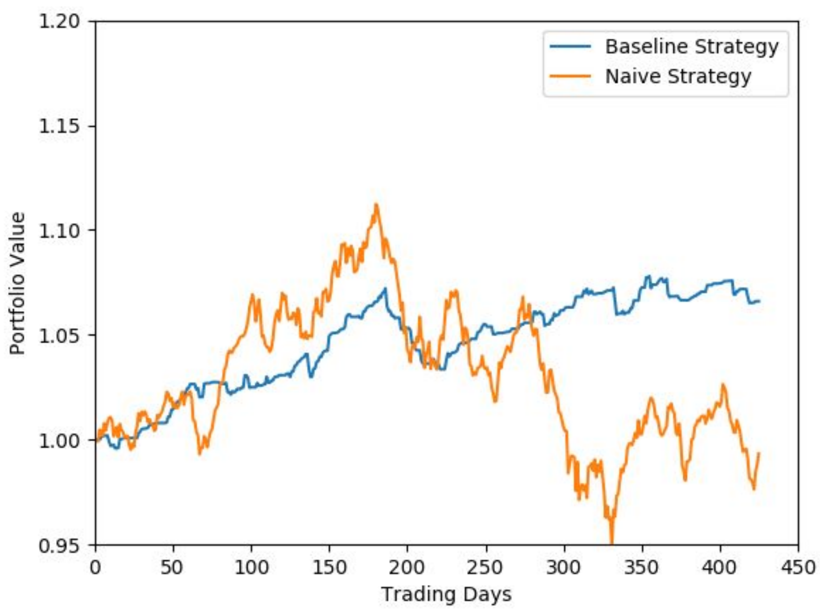
(1) Crossover: Crosses between moving averages and actual price of the asset

(2) Breakout: When the price of the asset breaks out from the range defined by support and resistance line

Our Baseline Strategy

- Equal allocation to each assets
- Trading using crossovers
- Utilizing both non-moving stops and trailing stops

Backtesting Result

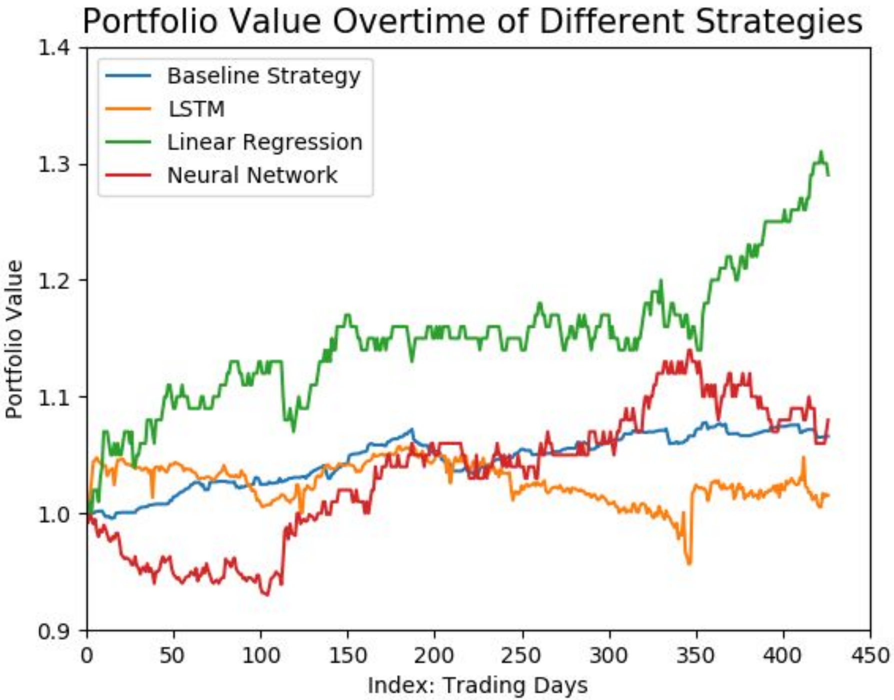


Sharpe Ratio: 1.494

Annualized Profit: 3.815%

Maximum Drawdown: -0.4134%

Best Strategy



Strategies	Sharpe Ratio	Annual Return	Max Drawdown
Baseline Strategy	1.494	3.815%	-0.4134%
Linear Regression	1.436	15.66%	0%
Neural Network	0.4777	5.129%	-7.000%
LSTM	0.1434	1.351%	-4.360%

Trading Costs and Stop Loss

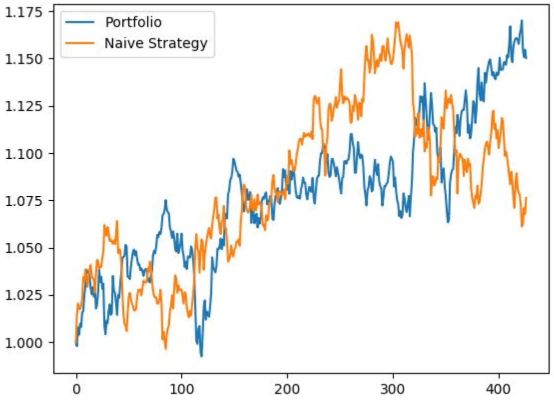
Trading costs represented as a constant percentage of our trade size

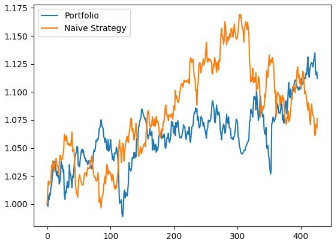
25 bp (0.25%) as a rough estimate (equities trade at **15 bp**[1])

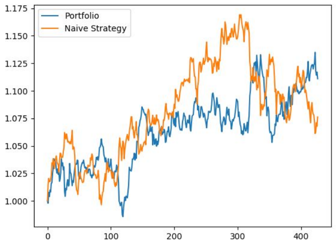
For each position, if the asset has a loss greater than *x*% since opening that position, close the position

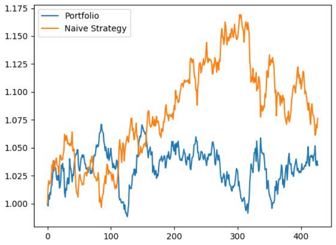
Stop losses at 15%, 10%, and 5% loss shown below, all seem to have a negative impact on portfolio.

The losses we experience are typically not gradual, but sudden, and thus the stop loss is ineffective







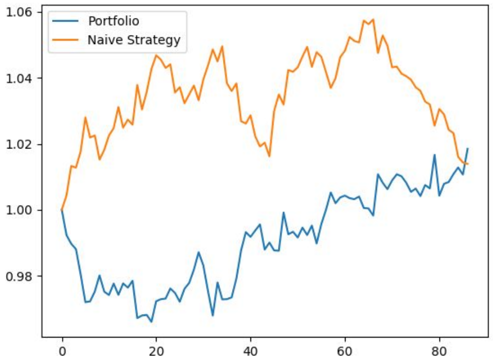


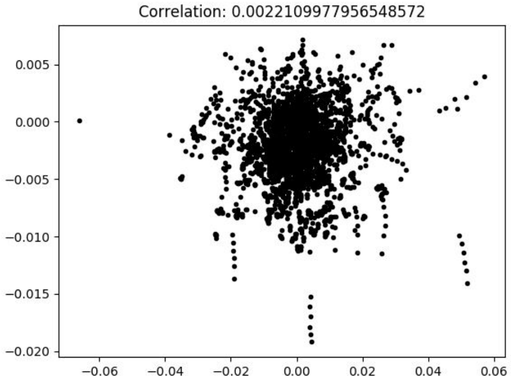
[1]http://www.integrity-research.com/equity-commission-rates-remain-steady/

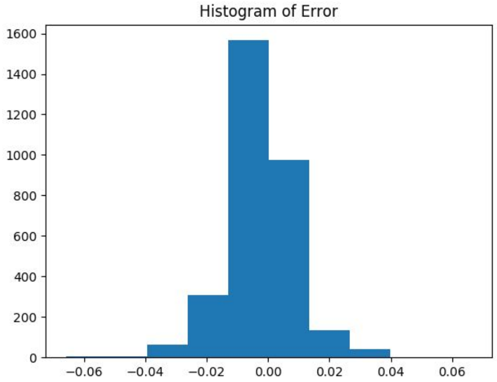
Final Results

Look at 2019 data as a small secondary test set, for a selected strategy

Use Linear Regression and account for trading costs, without using a stop loss







The results are not exactly encouraging:

- The portfolio spends almost all year in the red compared to the simple allocation strategy
- Where before the errors were more balanced, they are now distinctly more often negative
- The regression has several large mistakes that cost it significantly

Conclusion

Conclusions

- Simple Trend Following has the largest Sharpe ratio
- Learned models did not capture the logic behind trend following
- Accuracy doesn't necessarily improve performance
- Stop loss did not improve performance of modeled strategies

Future Work

- ➔ Data Collection: Larger Datasets
  - ◆ To better facilitate deep learning models
  - ◆ Seek a larger universe of assets
- ➔ Features: Introducing new features
  - ◆ Adding fundamental features (P/E, ROE, ROA, etc.)
  - ◆ Adding more technical indicators
- ➔ Model:
  - ◆ Develop the models for more accurate predictions
  - ◆ Better tuning parameters: Random Search and Bayesian Optimization
- ➔ Backtesting: Using test sets with tail events
  - ◆ Our current test set (Year 2017 - 2018) is quite ‘typical’

Questions?