I began by downloading historical data from the CFTC trader commitments website. I downloaded all data between 2011-2025 of *Disaggregated Futures Only Reports*. For my dependent variable I used yahoo finance historical price data on the next contract to expiration.

I decided to begin by looking at Coffee Futures where one contract is 37,500 lbs of coffee. To get coffee data I pulled data from the KC=F “ticker” using yahoo finance using their python library. Further I aggregated all timeseries data from the 2011-2025 of *Disaggregated Futures Only Reports*, sorted by date and filtered it to only include coffee futures.

Noticing that *Disaggregated Futures Only Reports* reports weekly my next task was to align the data at time t with returns at time t+1. This meant that I had to calculate weekly returns from the yahoo finance close data. I did this by calculating pct changes for the next 5 days and merging on the CFTC data’s date. This meant that I aligned exactly the CFTC data on their reporting date with the returns from that week to the next reporting week. (to be more accurate I should shift by -1 to ensure no lookahead bias and also to calculate precise number of days between reporting dates and align with this because I assumed 5). Next, to ensure that I am not using level nonstationariry variables as my independent variables I also calculate percent changes of each feature contained in the CFTC data.

I used returns and percent changes as my dependent and independent variables because they are stationary and comparable between variables. One of the assumptions of OLS is stationarity.

Following this I cleaned my data by removing datapoints with nan’s. Then at first I tried standardizing the data (1st and second moments using standardscaler).

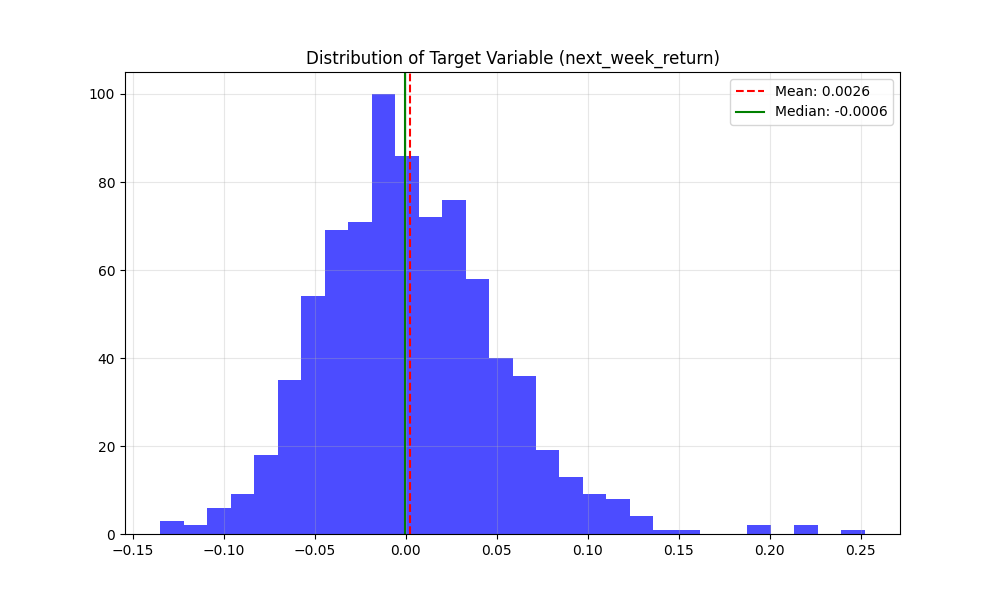
Following this data preparation I had about 130 features remaining lined up with returns for the following week. Due to the large amount of features (I wanted to narrow down the features to see which features were actually predictive) and need for a robust estimator, I ran an L1 regularized (Lasso) regression while varying the regularization parameter (alpha) with an 70-15-15 training/validation/test split on the data. I then maximized the performance on the validation data by varying the alpha parameter and finally tested my finalized model on my testing data. Didn’t want to use a nonlinear model due to the risk of overfitting to noise as I demonstrate even with a non regularized linear model in bottom figure.

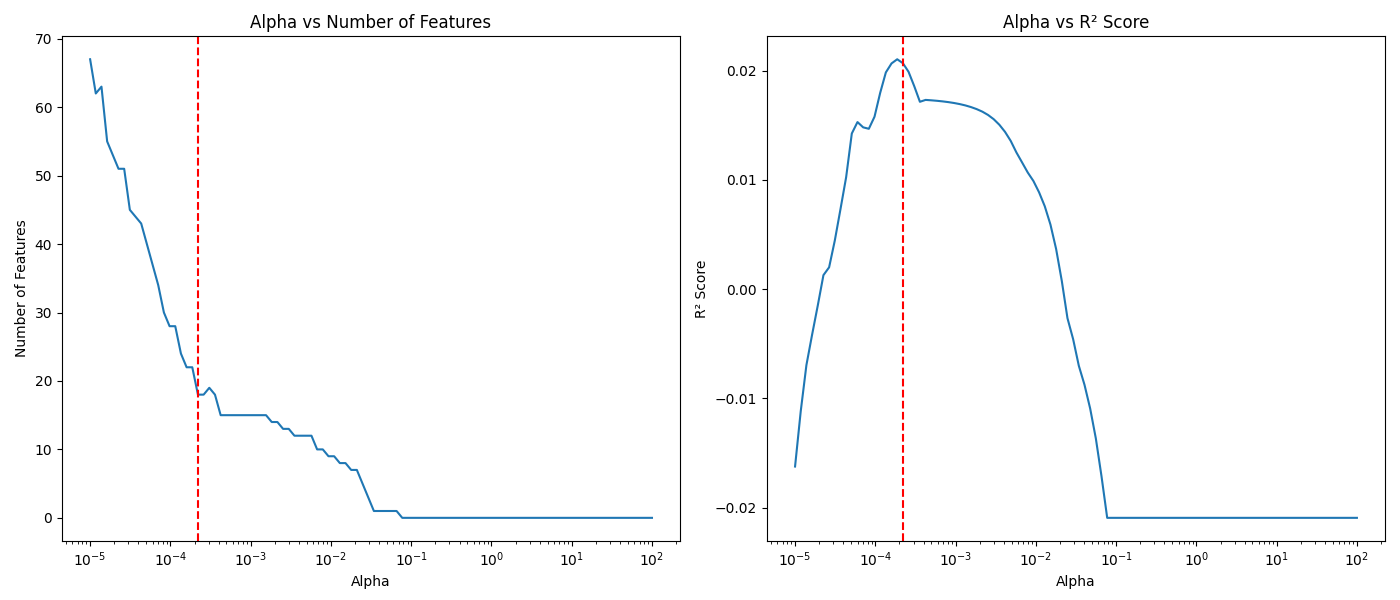
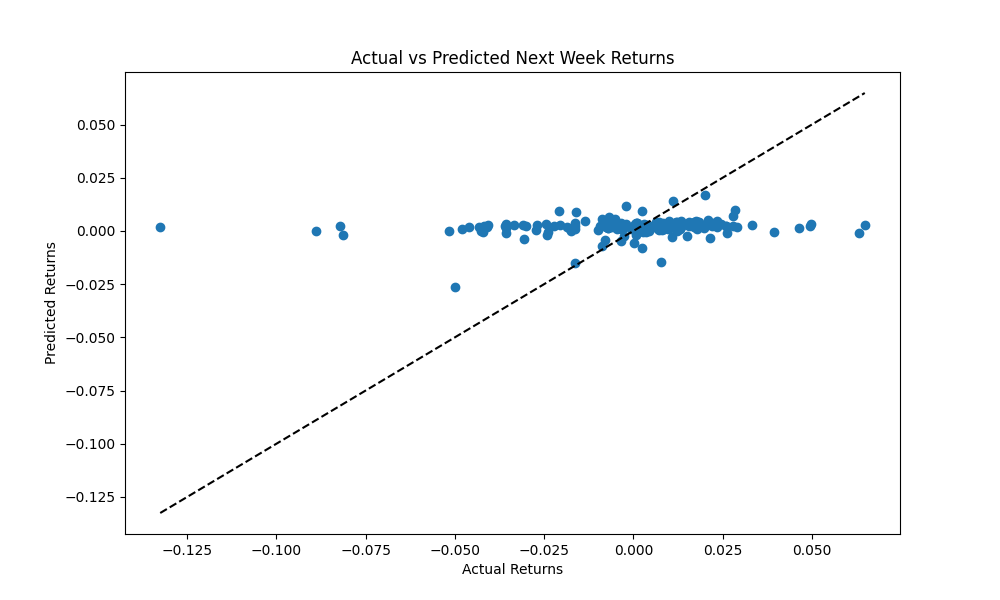
I found a really low R^2 and presence of extreme regularization to maximize performance. I believe this is partially due to the continued presence of outliers in my features.

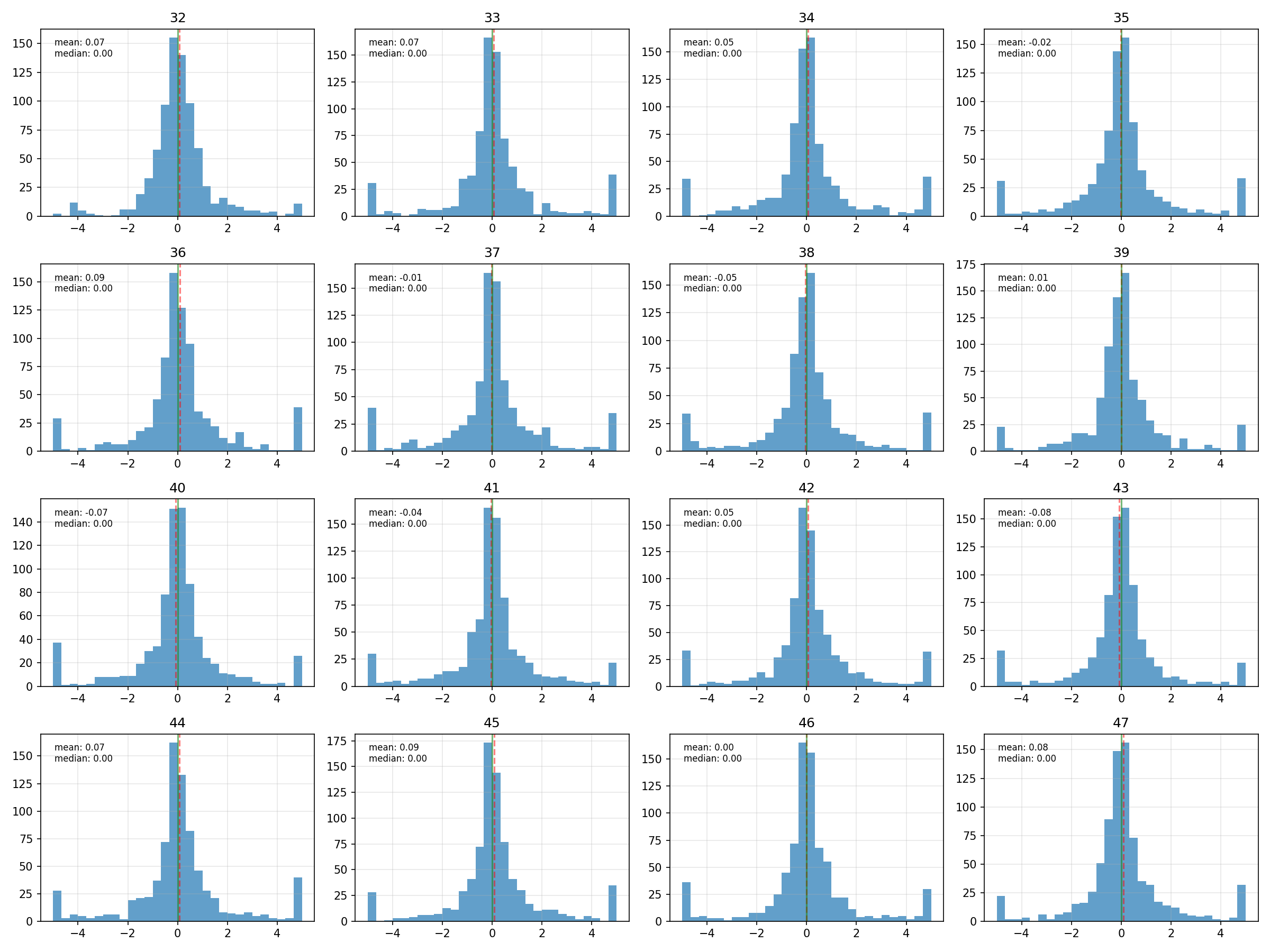
Then I repeated a robust scaler but I found that even after scaling there were many outliers past 3 standard deviations. I then repeated with winsorize my data to be between -5,5 (also tried +-4std,+-3std,+=2std) because the presence of really strong outliers. However I found that the presence of outliers caused the amount of data being binned at the endpoints be more then the central bins. Finally I repeated once more with my last attempt being the box cox transform (actually used Yeo Johnson because of the presence of negative datapoints) which allowed for the control of higher moments and controlled for many of the outliers. I reran the lasso with this.

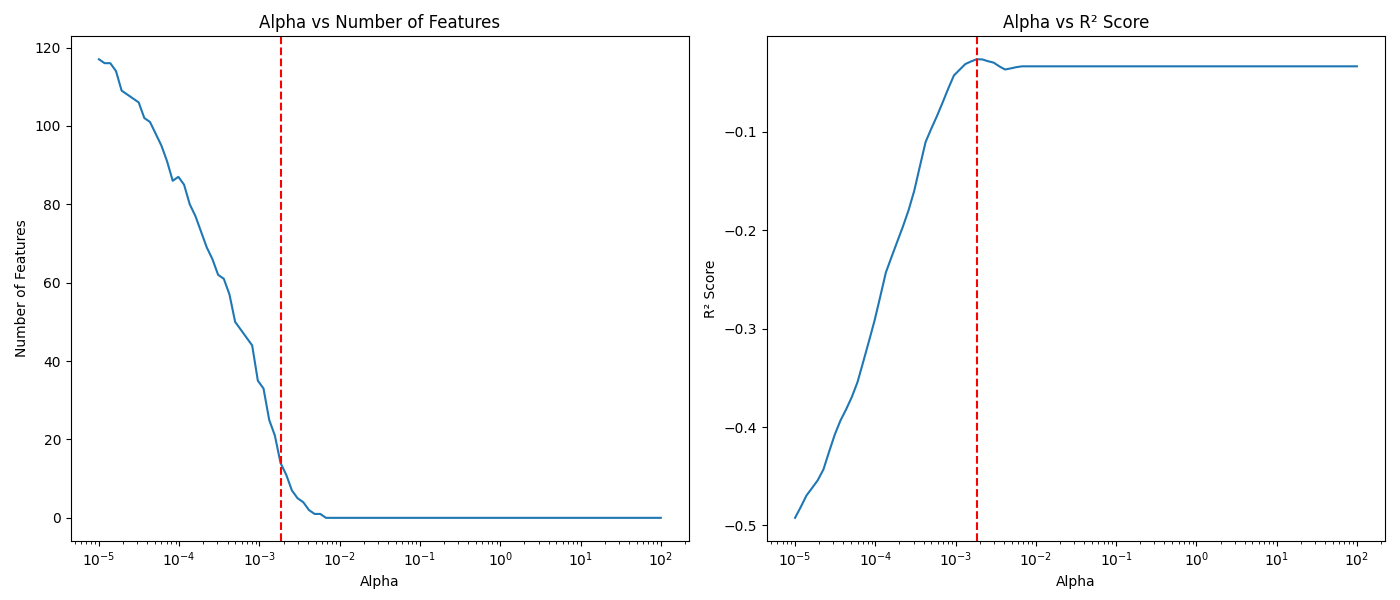
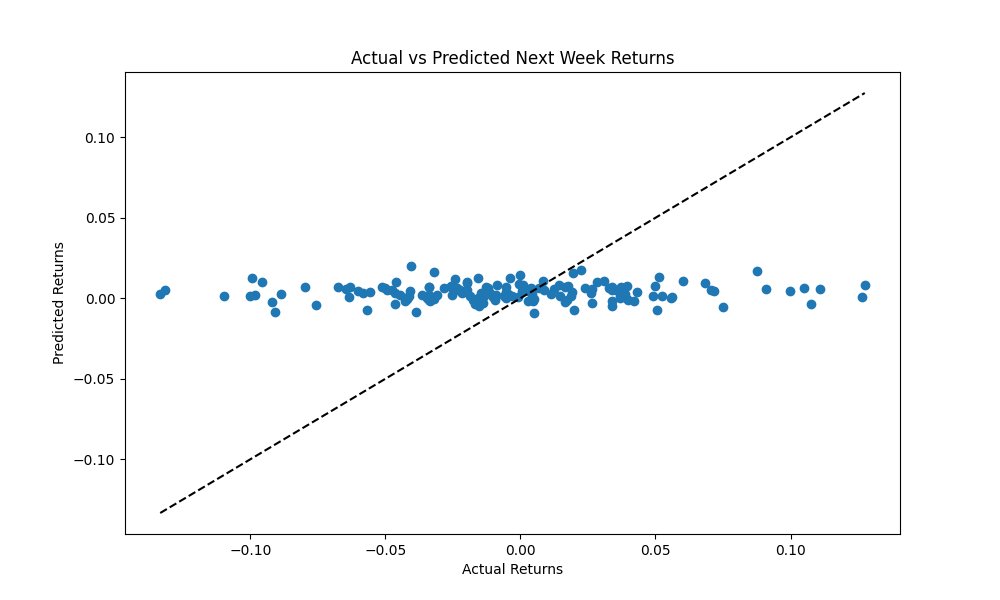
In all scenarios I found an R^2 between -.05 and 0.02 where the optimal solution was an extremely regularized one with maybe one or two points being predicted correctly which made the R^2 better than 0.

I then performed the same excersize on a more and less storable commodity to see if perhaps this would improve namely gold and live cattle. With gold I found that the best R^2 was below 0 meaning that due to the presence of outliers the model was predicting worse than just taking the average return. With cattle I found an R^2 slightly above 0.02 with the most significant loading being on Traders\_M\_Money\_Short\_All variable but its loading was still small being -0.00755.

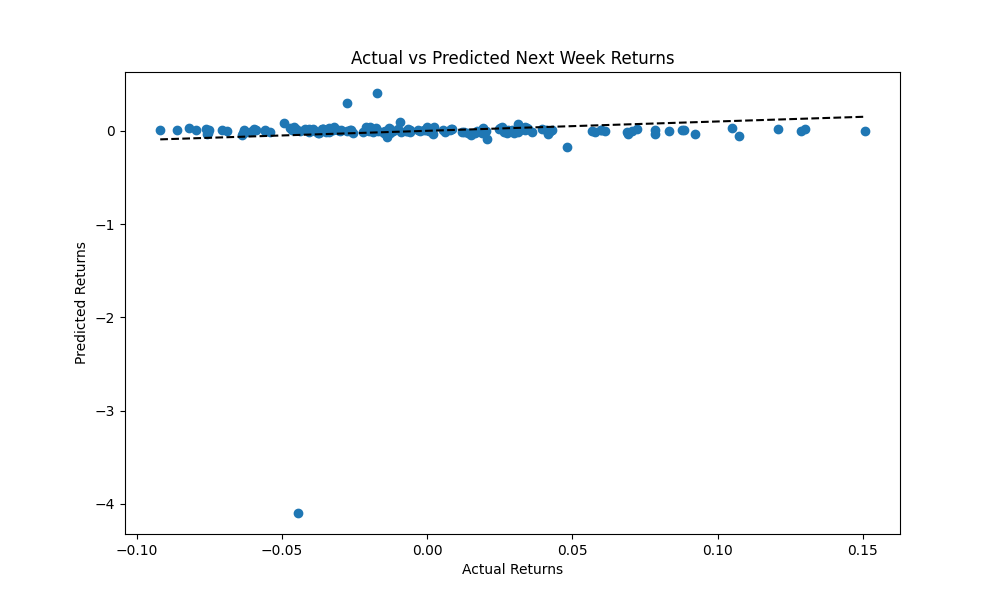




Live cattle

Notice outliers for coffee

Lasso for coffee



Linear overfits and sensitive to outliers