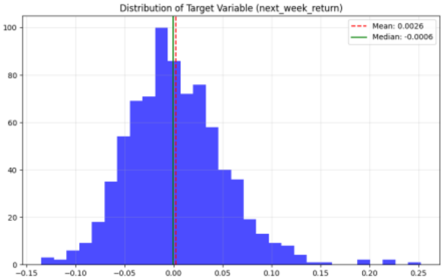
15.437 Optional Project
Alexander Podrez

I began by collecting historical data from the CFTC's *Disaggregated Futures Only Reports*, downloading weekly data from 2011 through 2025. For return data, I used historical prices from Yahoo Finance, specifically targeting the next contract to expiration.

To start the analysis, I focused on Coffee Futures, where each contract represents 37,500 lbs of coffee. I retrieved price data using Yahoo Finance's Python API for the KC=F ticker. Simultaneously, I aggregated and cleaned the CFTC reports across all years, filtered to only include records relevant to coffee.

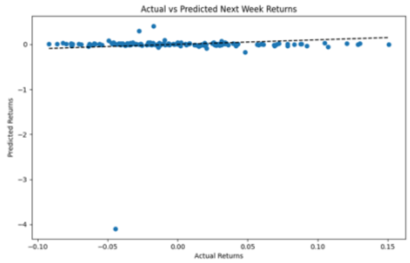
Since the CFTC data is reported weekly, my next task was to align each report at time t with returns over the subsequent week (t+1). To compute weekly returns, I calculated the percent change in closing prices over the following 5 trading days. These were then merged with the CFTC data on their reporting dates. While I initially assumed a consistent 5-day interval between reports, a more accurate implementation would shift the returns by one period and dynamically calculate the number of days between report dates to avoid look-ahead bias.



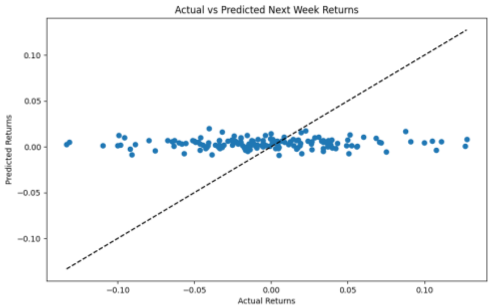
To address potential non-stationarity in the independent variables, I also computed percent changes for each CFTC feature. This meant that the inputs were in comparable, stationary form which is an important assumption for linear models.

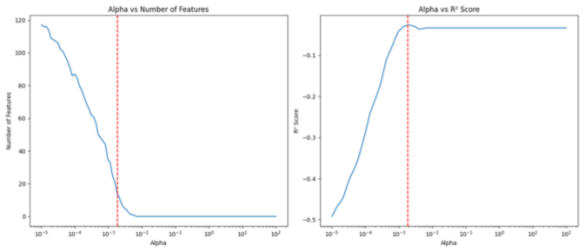
After removing observations with missing values, I initially standardized the data (zero mean, unit variance) using StandardScaler. This left me with approximately 130 features aligned with forward weekly returns. To manage the high-dimensional feature space and reduce the risk of overfitting, I decided to use L1-regularized Lasso regression, using a 70-15-15 train-validation-test split. I tuned the regularization parameter alpha to optimize performance on the validation set, then evaluated the final model on the test set. I decided against using nonlinear models to prevent overfitting, which I observed even with an unregularized linear model (as shown below) getting an out of sample R^2 of -46.

15.437 Optional Project
Alexander Podrez

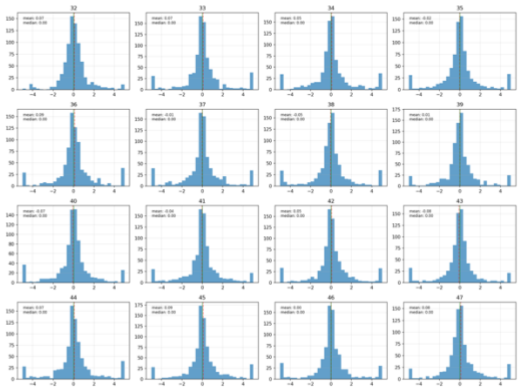


Initial results showed low explanatory power, with R^2 values between -0.05 and 0.02 (shown below). In many cases, the optimal model relied on heavy regularization, with about 5-10 active coefficients which tells me that these are noisy features that have poor signal-to-noise ratios.

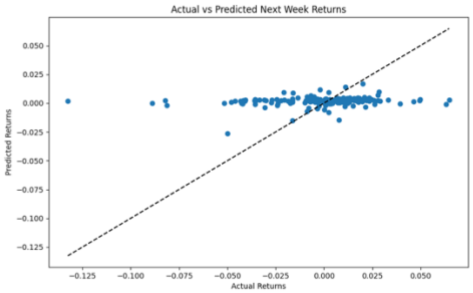


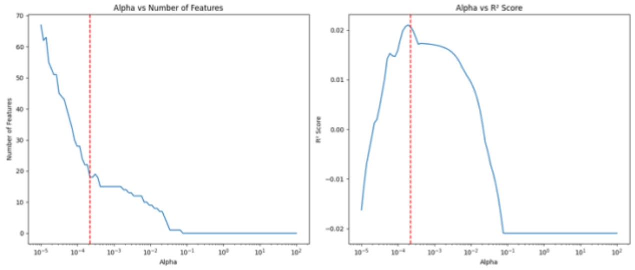


To address potential outliers, I experimented with several transformations. First, I applied a RobustScaler, but many features still exceeded ±3 standard deviations. I then tried winsorizing the features to various bounds (e.g., ±2, ±3, ±4, and ±5 standard deviations), but this often led to excessive binning at the endpoints (shown below). Finally, I applied the Yeo-Johnson transformation (a generalization of Box-Cox that accommodates negative values), which better normalized the data and mitigated the impact of extreme outliers. Despite this, the Lasso regression continued to fail in providing major improvements in predictive power.



I extended the analysis to two other commodities: gold (a highly storable asset) and live cattle (a less storable one) to see if storability could affect predictability. For gold, the model performed worse than a constant predictor, with the best R^2 < 0. For live cattle, performance was slightly better, with the highest R^2 being about 0.02 (shown below). The most predictive variable was Traders_M_Money_Short_All, but even its coefficient was small (approximately -0.00755).





In summary, despite extensive preprocessing, feature transformations, and attempt at more robust modeling techniques, I found little evidence of predictive power in the weekly disaggregated CFTC positioning data for short-term returns across the commodities analyzed.

References

Zhang, Yang and Laws, Jason, Investor Sentiment and Forecasting Ability: Evidence from COT Reports in Precious Metal Futures Markets (September 16, 2013). Available at SSRN: https://ssrn.com/abstract=2382299 or http://dx.doi.org/10.2139/ssrn.2382299

OpenAI. *ChatGPT (April 16, 2025)*. Available at: https://chat.openai.com