Abstract

Predicting the price movements of crude oil is a challenge due to the vast noise and complex movements between supply-demand dynamics and market signals. In this project, we decided to use different machine learning methods to analyze the predictions and answer two questions: Are advanced methods always superior, and is "Garbage In, Garbage Out" solveable. Our results indicate that fore regression, the Random Walk model outperforms all other methods, while for classification the performance pattern differs. We also found out that advanced methods are not always superior, and "Garbage In, Garbage Out" being solvable depends on whether exploitable structure is present on data.

1 Introduction and Problem Formulation

Crude oil (WTI) serves as a structural and systematic asset class that is highly connected with the macro-economy. However, predicting its price movements remains a challenge due to the vast noise and the complex interplay between supply-demand dynamics and market signals.

This project aims to perform a comparative study of different machine learning methods applied to the prediction of crude oil. Furthermore, we aim to address two technical questions specifically in financial machine learning:

1. **Are advanced methods always superior?** We investigate the trade-off between model complexity and interpretability. Specifically, we compare "black-box" models (like Neural Networks) against robust baselines (such as the Random Walk) to see if the sacrifice in interpretability yields a statistically significant improvement in accuracy or other metrics.

2. **Is "Garbage In, Garbage Out" solveable?** Financial datasets are noisy. While traditional econometrics emphasizes feature quality, Deep Learning (e.g., LSTM/NN) theoretically excels at processing vast amounts of unstructured or noisy data. We test whether these models can distill valuable signals from a large set of raw, different-time-scale inputs where most features might be considered "garbage".

2 Data Description

Our dataset spans from **January 4, 2021, to November 24, 2025**, where the target variable is defined as the next-day return of WTI Crude Oil. This specific time window was deliberately chosen to exclude the COVID-19 pandemic period. By focusing on the post-pandemic recovery and stabilization phase, we aim to build a robust model applicable to normal market regimes.

Specifically, we constructed a feature set by merging daily market data (sourced from Yahoo Finance) with weekly fundamental reports (sourced from the US Energy Information Administration). The final dataset consists of **1,229** daily observations aggregated into **256 weekly samples**, containing:

1. Energy Complex Commodities: To capture sector-specific correlations, we included prices of energy assets, including *Brent Crude* (global benchmark), *Natural Gas*, and other related commodities.

2. Macroeconomic and Financial Indicators: Recognizing that oil is a macro-asset, we incorporated broader market signals including the *Currency* factor via the US Dollar Index (DXY) as oil is priced in dollars, and *Rates* via the US 10-Year Treasury Yield (TNX) representing the cost of capital and economic outlook. We also included *Equity* signals using the Energy Select Sector SPDR Fund (XLE) as a proxy for energy equity market sentiment, alongside *Safe Havens* such as Gold prices and the Gold/Oil ratio.

3. Technical Indicators: To capture market momentum and psychological barriers, we engineered technical features including *Relative Strength Index (RSI)*, *Moving Average Convergence Divergence (MACD)*, *Bollinger Bands*, and volatility metrics like *Average True Range (ATR)*.

4. Supply-Demand Fundamentals: Sourced directly from the Energy Information Administration (EIA), these features provide physical market context, covering *Inventory Levels* such as weekly crude oil stocks (excluding SPR) and product stocks (Gasoline, Distillates), as well as *Production Metrics* like domestic field production and refinery utilization rates across different PADD regions.

3 Methodology

To investigate the two technical question, we employ a dual-direction, multi-stage investigation framework based on our baseline model performance.

The Baseline: Simple Linear Trend

We first estimate baseline models using the original dataset without any feature engineering, which attempts to predict the next day/week's price directly using a weighted sum of all available features (Inventory, Production, RSI, etc.) from the current day/week. Performance is poor due to high dimensionality, multicollinearity, and noisy predictors. The prediction RMSE(s) are **25.54** (daily-based) and **34.59** (weekly-based). The performance is poor due to high dimensionality, multicollinearity, and noisy predictors. Furthermore, the restricted model capacity of a simple linear structure limits its ability to generalize across distinct market regimes or adapt to volatile fluctuations.

1^{st} Direction: Data Processing & Feature Engineering

We apply dimensionality reduction via Principal Component Analysis (PCA) and feature selection via Elastic Net regularization, treating different feature categories separately. PCA is applied within feature groups, including fundamental indicators, technical indicators, and EIA weekly variables. The resulting dimensionality reduction is substantial: the original 446 features are reduced to 10 principal components, achieving a 97.8% reduction while retaining a large fraction of the total variance.

In parallel, we perform feature selection using Elastic Net. We find that Lasso and Elastic Net yield nearly identical selected feature sets, but Lasso exhibits more severe convergence issues. Therefore, we focus on Elastic Net, which combines L1 and L2 regularization to improve numerical stability in the presence of correlated predictors. Feature selection is conducted separately by category, resulting in a total of 49 selected features across all groups.

2^{nd} Direction: Gradual Improving ML Methods

With the cleaned, higher-quality dataset, we investigate the possibilities of ML methods.

1^{st} Attempt: Random Walks

This model assumes the market is efficient and the best predictor for next week's price is simply this week's price. Second, to mitigate the transient volatility and high-frequency noise inherent in daily spot prices, we modify the standard martingale assumption by utilizing the **average of the trailing two weeks** as the prediction anchor, rather than relying solely on the latest closing price. Furthermore, unlike a naive random walk that operates in a vacuum, this model explicitly integrates EIA fundamental data as a *regime indicator*, acknowledging that while price movements may be stochastic, they are contextually bounded by the physical supply-demand landscape. We assume the EIA fundamentals as the regime indicator of fundamentals in the crude oil market.

2^{nd} Attempt: Random Forest

Moving beyond the rigidity of linear models and the naivety of Random Walk assumptions, we implement the random forest method. For the random forest method, we performed cross-validation with iteration on max-features from 1 to 100. We do a KFold with n-splits = 5. Other grid values for RandomForestRegressor includes min-samples-leaf = 5 and n-estimators = 500. We set the random-state in this circumstance to 42. We compare the results of this cross-validation with the features selected from the previous parts.

3^{rd} Attempt: XGBoost (Boosting)

Comparably, we also implement the boosting method, specifically its variation XGBoost.XGBoost incorporates both $L1$ (Lasso) and $L2$ (Ridge) regularization terms directly, preventing the overfitting often observed in financial time-series prediction–not to mention its faster convergence and finer optimization. On the other hand, this method assumes that the driving forces of crude oil prices are inherently non-linear and dominated by complex interaction effects—for example, how a technical signal might exhibit different predictive power depending on the fundamental regime.

Furthermore, XGBoost(Boosting) effectively utilizes the full feature set constructed in the 1^{st} Direction. It addresses the high bias of previous methods by sequentially training an ensemble of decision trees, where each new tree focuses on correcting the residual errors of its predecessors. This allows the model to capture local market patterns and asymmetric responses that global linear models miss, while its built-in regularization prevents the overfitting often seen in standard decision trees.

4^{th} Attempt: Time Series Modeling (AR, MA, ARIMA)

To bridge the gap between naive baselines and complex machine learning models, we employ classical Time Series analysis. While the Random Walk hypothesis assumes that future price movements are independent of the past, we hypothesize that financial markets exhibit latent serial correlations—driven by supply chain lags and market psychology—that render price changes non-random, albeit highly complex and noisy.

A primary challenge in modeling crude oil prices is *non-stationarity*; the mean and variance of spot prices fluctuate significantly over time, making standard regression techniques prone to spurious results. Therefore, we implement the ARIMA framework as a robust linear benchmark. Specifically, the "I" portion removes the stochastic trend; the "AR" component models momentum and mean-reversion so that captures the "memory" of the market; the "MA" part models the persistence of idiosyncratic shocks by regressing on past forecast errors. By explicitly modeling these temporal dependencies, ARIMA serves as a sophisticated econometric baseline to determine if the "black-box" non-linearity of Neural Networks is truly necessary.

5^{th} Attempt: Neural Network (LSTM)

Finally, we implement a Long Short-Term Memory (LSTM) network. While traditional models assume static relationships, the crude oil market is a complex adaptive system where the dominant drivers shift over time. We utilize LSTM for four specific theoretical advantages:

1. Capturing Non-linear Combinations: Unlike linear baselines, the LSTM's multi-layer architecture captures *time-varying* non-linear interactions. It can dynamically adjust weights to reflect that different variable combinations (regimes) dominate market pricing at different times.

2. Noise Filtering (The Forget Gate): Historical data often contains "idiosyncratic events" (e.g., past one-off strikes) that generate noise but lack predictive value for the future. The LSTM's specific *forget gate* mechanism allows the model to learn which historical information is obsolete and should be discarded, preventing overfitting to past noise.

3. Reduced Lag vs. Time Series: Traditional time series models (like ARIMA) heavily rely on autoregression, often resulting in predictions that merely lag the actual trend. Due to its higher complexity and ability to synthesize leading indicators, the LSTM is expected to identify latent market momentum earlier, offering more forward-looking signals with reduced lag.

4. High Model Capacity: Our dataset comprises over 300 variables. While this dimensionality causes multicollinearity in simpler models, the LSTM possesses the high model capacity necessary to ingest this high-dimensional input directly, performing internal feature extraction without significant information loss.

Furthermore, due to the high noise level and strong *model variability* in financial prediction, we conduct the ensemble design which selects the ten best LSRM models by validation performance and ensembles them.

4 Model Validation and Evaluation Metrics

To ensure the robustness of our predictive framework and mitigate *look-ahead bias*, we strictly adhered to a chronological 80-20 train-test split without random shuffling. This setup forces the model to train exclusively on historical data and be evaluated on unseen future sequences.

Our evaluation employs a dual-perspective approach, combining traditional **Regression Metrics** (MSE, RMSE, MAPE, AIC, R^2) with **Classification Metrics** (Precision, Recall, AUC-ROC, Log Loss).

Specifically, we shall argue that relying solely on regression metrics is insufficient for financial utility. Standard regression loss functions (e.g., MSE) treat positive and negative errors symmetrically; however, in financial markets, the cost of error is fundamentally *asymmetric*. A model that predicts a small price drop when the market surges (missed opportunity) has a different payoff profile than one that predicts a surge when the market crashes (capital loss). Consequently, **Directional Forecasting** holds greater practical significance than precise point estimation. For quantitative strategies, the primary objective is often the identification of the correct trading signal (Long vs. Short) rather than the magnitude of the move.

To this end, we emphasize classification indicators not merely as accuracy checks, but as proxies for risk-adjusted performance:

- **Precision and Recall (Signal Reliability vs. Opportunity Capture)**: While Recall measures the model's aggressiveness in capturing market opportunities (profit maximization), Precision assesses the reliability of these signals. A high Recall with low Precision would imply a "spray and pray"

strategy that exposes capital to excessive false positives. We attempt to seek a balance where the model captures uptrends without generating excessive noise.

- **AUC-ROC (Discriminative Robustness)**: This metric evaluates the model's invariant ability to distinguish between upward and downward trends across different probability thresholds, serving as a measure of the model's fundamental signal-to-noise ratio independent of specific trading triggers.
- **Log Loss (Implicit Risk Control and Calibration)**: Crucially, we incorporate Log Loss to penalize *overconfidence*. In financial engineering, a wrong prediction made with high confidence (e.g., a "sure bet" that fails) is far more destructive than a hesitant wrong prediction. Log Loss acts as a proxy for **tail risk management**: it prevents the model from assigning extreme probabilities to uncertain events. By minimizing Log Loss, we ensure the model is well-calibrated, favoring strategies that scale down position sizes (implied by lower probability scores) during periods of uncertainty, effectively functioning as a statistical stop-loss mechanism.

5 Results & Discussion

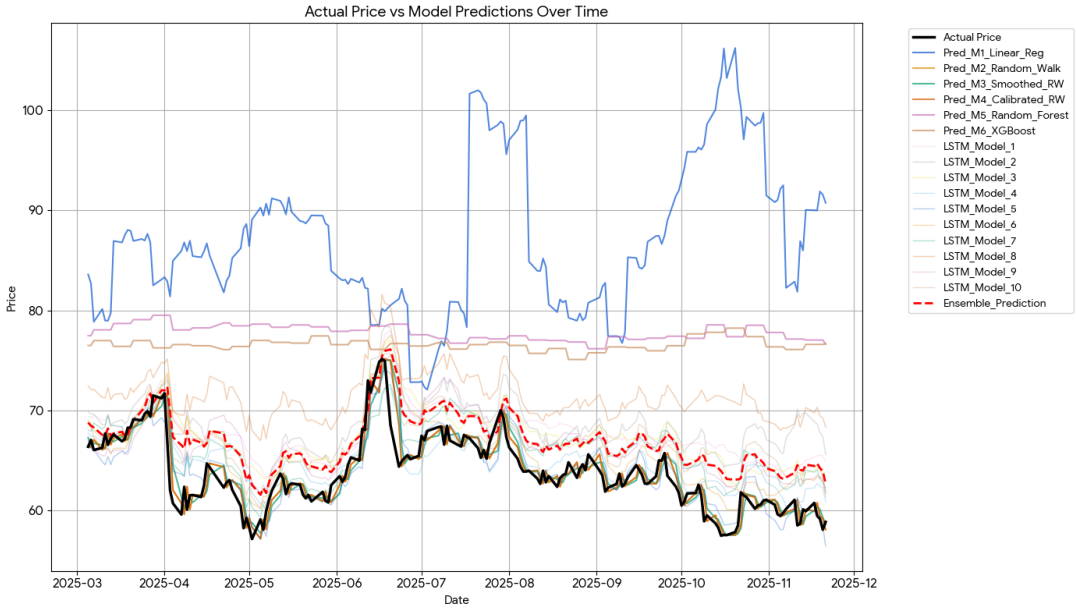

Figure 1: Comprehensive Model Comparison: Baselines vs ML

5.1 Model Performance

For regression, the Random Walk model performs best, achieving the lowest RMSE (1.34), MAPE (1.48), and a high R^2 of 0.93, reflecting strong price persistence. The Smoothed Random Walk remains competitive, while the Linear Baseline performs poorly. Classical time-series models (Naive, AR, MA, ARIMA) achieve similarly low regression errors and high R^2, confirming that simple autoregressive structure is sufficient when prices follow a near-random-walk process. In contrast, Random Forest and XGBoost exhibit much higher RMSE values (above 12) and negative R^2, indicating limited effectiveness in price-level prediction. Neural Networks achieve relatively low regression errors in some cases but show unstable performance across specifications.

For classification, the pattern reverses. The Random Walk and classical time-series models show little directional skill, with recall close to zero and AUC values around 0.50. In contrast, Random Forest and XGBoost achieve recall above 0.98 and AUC around 0.57, demonstrating a clear advantage in identifying directional movements. The Smoothed Random Walk also improves upon the raw baseline, while Neural Networks exhibit AUC values close to random guessing, indicating weak directional discrimination.

Table 1: Performance Comparison of Baseline and Machine Learning Models

	Price Prediction (Regression)				Directional Forecasting (Classification)			
Model	**RMSE**	**MAPE (%)**	**AIC**	**R^2**	**Precision**	**Recall**	**AUC-ROC**	**Log Loss**
Basics								
Linear Baseline (Raw)	25.54	33.20	2450.40	-25.08	0.50	0.91	0.49	0.97
Random Walk (Raw)	**1.34**	**1.48**	**137.60**	**0.93**	0.00	0.00	0.50	**0.69**
Smoothed RW (With Features)	1.52	1.65	198.93	0.91	**0.52**	0.53	0.54	0.77
ML Methods								
Random Forest (RF)	13.13	19.19	1207.10	-5.89	0.50	**0.99**	**0.57**	1.28
XGBoost	12.08	17.46	1295.90	-4.83	0.50	0.98	**0.57**	**1.18**
Neural Network (Single Best)	1.78	2.14	N/A	0.77	0.49	0.46	0.47	0.86
Neural Network (Ensemble)	3.65	5.16	N/A	0.01	0.47	0.42	0.45	0.80
Naive (Yesterday)	1.38	1.54	N/A	0.86	0.49	0.48	0.45	0.93
AR(7) (Tuned)	1.38	1.57	3694.59	0.86	0.51	0.48	0.45	0.93
MA(20) (Tuned)	1.73	2.00	3836.28	0.78	0.48	0.49	0.44	1.01
ARIMA(5,1,5) (Tuned)	1.37	1.53	3667.28	0.86	0.49	0.45	0.47	0.91

Regression vs. Directional Forcasting

Strong regression performance can be misleading when directional accuracy is the primary objective. Although the Random Walk achieves low RMSE due to price smoothness, it provides no directional information, with zero recall and an AUC close to 0.5, limiting its usefulness for decision-oriented tasks.

Limitations of R^2 as Justification

R^2 is not a suitable metric in this context, as tree-based models are optimized for predictive accuracy rather than variance explanation. Consequently, low or negative R^2 mainly reflects poor price-level fit and does not necessarily imply weak predictive usefulness.

Why near-flat ML price forecasts?

As shown in Figure 1, Random Forest and XGBoost generate near-flat price forecasts. By minimizing squared error, they estimate conditional expectations, which collapse toward a stable mean in a low signal-to-noise environment. This is a mismatch between regression-based forecasting and financial objectives.

Trade-off: Is ML worth it?

If a model requires extensive hyperparameter tuning, high computational cost, or operates on noisy data where signal quality is limited, switching from a robust baseline to an advanced machine learning method may not be worthwhile. In such cases, increased complexity does not necessarily translate into meaningful gains. Therefore, the choice between baseline and advanced models should not be viewed as a simple dichotomy of simplicity versus complexity, but as a balance between prediction accuracy, interpretability, computational efficiency, and data conditions. This trade-off is inherently dataset-dependent and should be evaluated in the context of the specific application.

Is "Garbage In, Garbage Out" Solvable?

In the context of financial markets, noisy data is a structural outcome of competition, as exploitable signals are quickly arbitraged away. Increasing model complexity cannot recover nonexistent signals. In other words, GIGO is mitigated not by using more complex models, but by asking models to predict the right object using methods that are appropriate for that object.

Are Advanced Methods Always Superior?

Advanced methods are not universally superior. While machine learning models underperform in price-level regression and lack clear economic interpretability, they outperform the Random Walk in classification tasks by extracting weak directional signals from noisy data. **Ultimately, the choice between baseline and advanced models should not be viewed as a simple dichotomy of simplicity versus complexity, but as a balance between prediction accuracy, interpretability, computational efficiency, and data conditions. This trade-off is inherently dataset-dependent and should be evaluated in the context of the specific application.**

Appendix: Code

https://github.com/haoruizhang2001/INDENG-242A-Group-Project/tree/main

Appendix: Tables and Figures

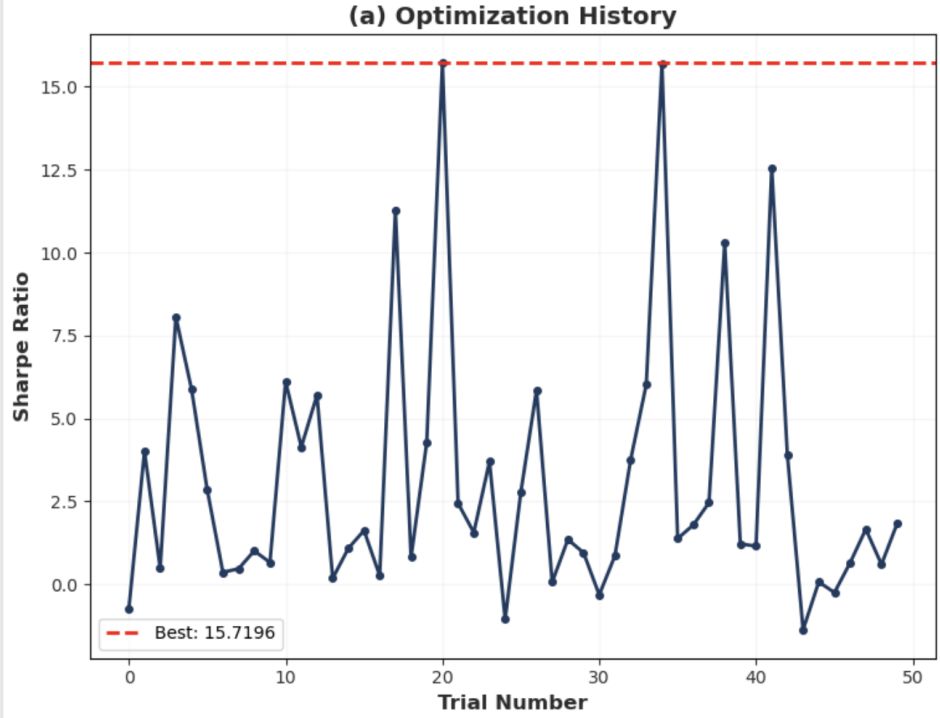

Figure 2: XGBoost Optimization History

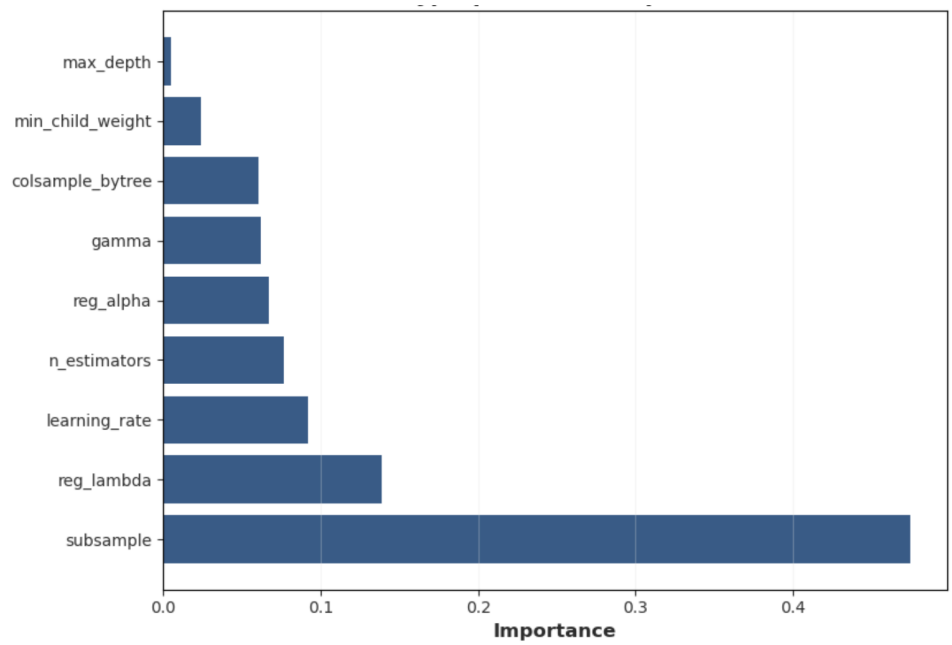

Figure 3: XGBoost Hyperparameter Importance

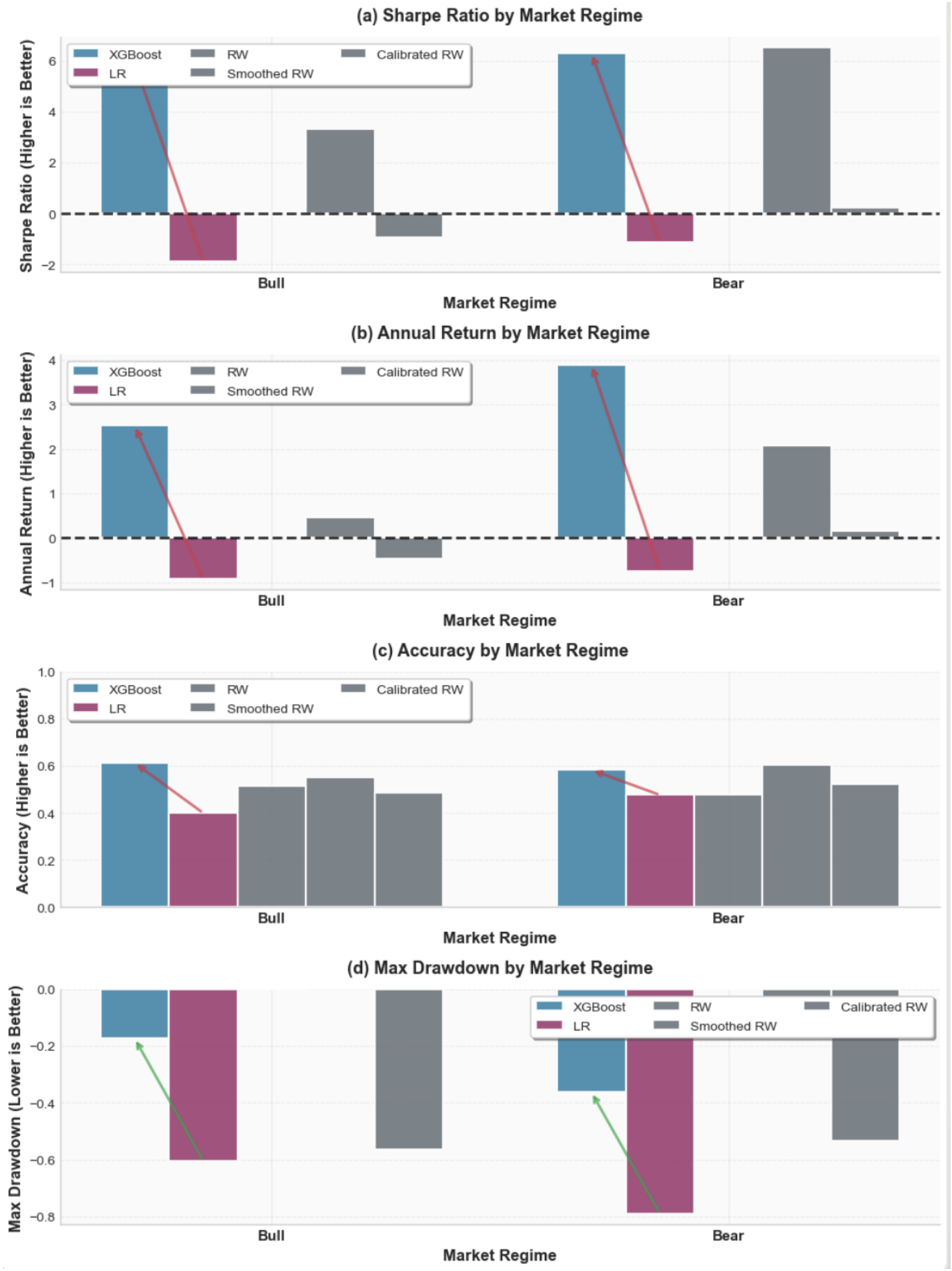

Figure 4: XGBoost vs Baseline Models by Bear And Bull Market

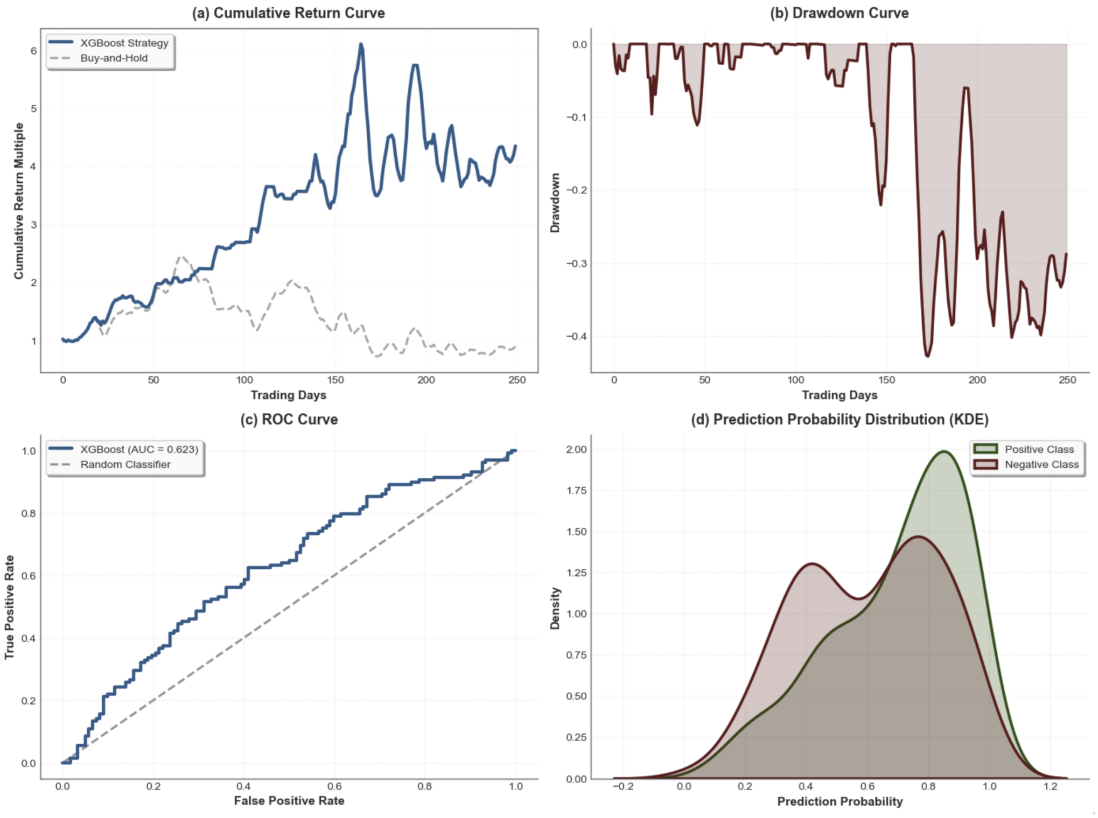

Figure 5: XGBoost Model Performance Analysis

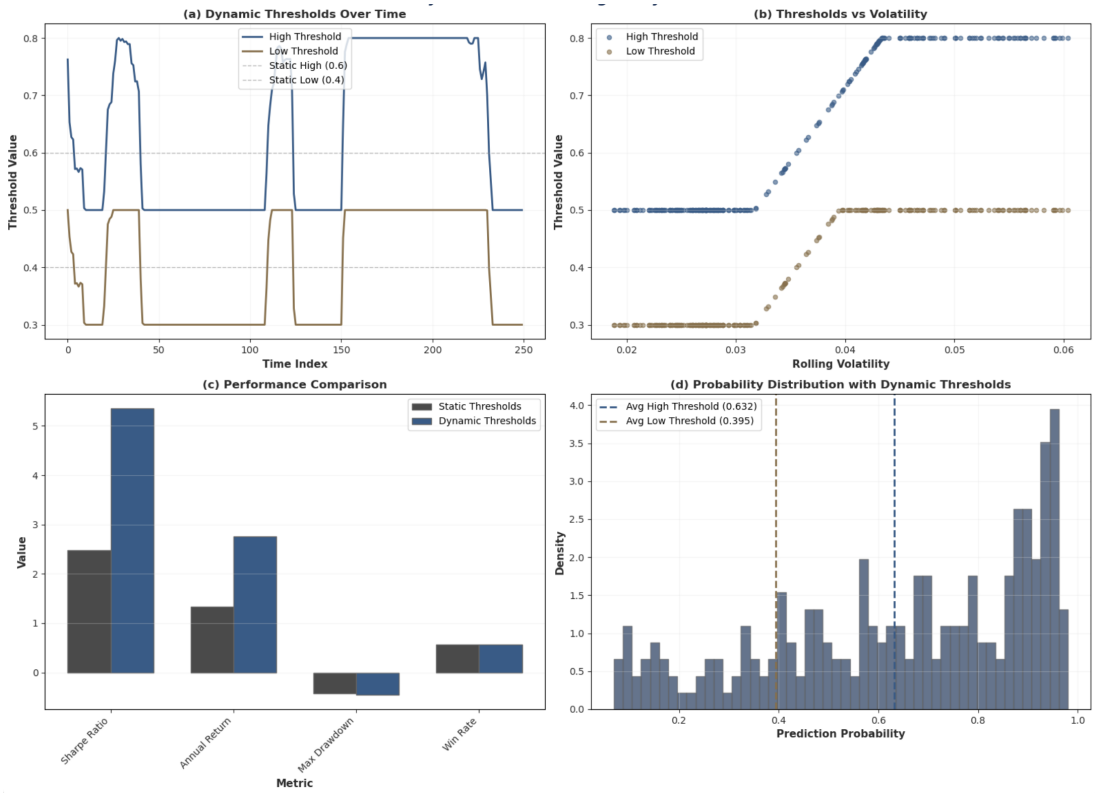

Figure 6: XGBoost Dynamic Thresholding Analysis

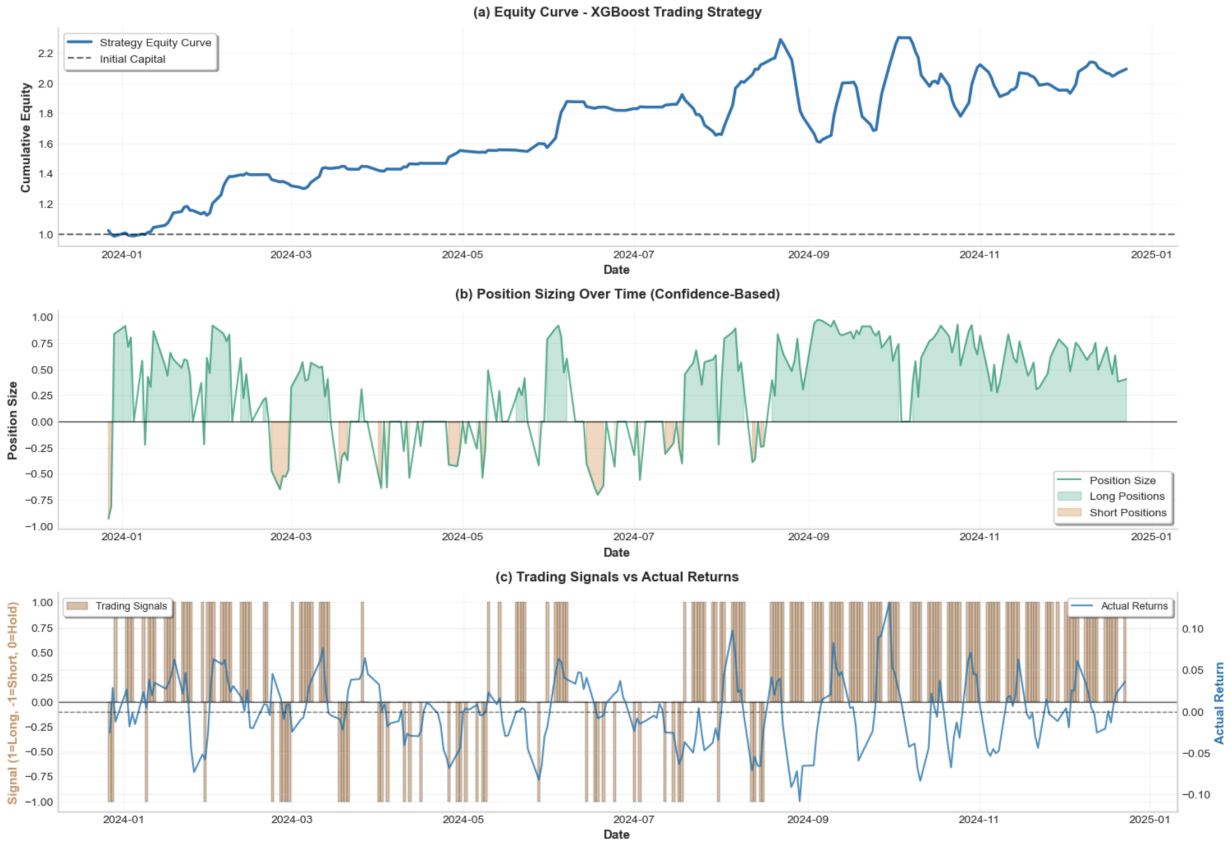

Figure 7: XGBoost Trading Strategy Performance Analysis

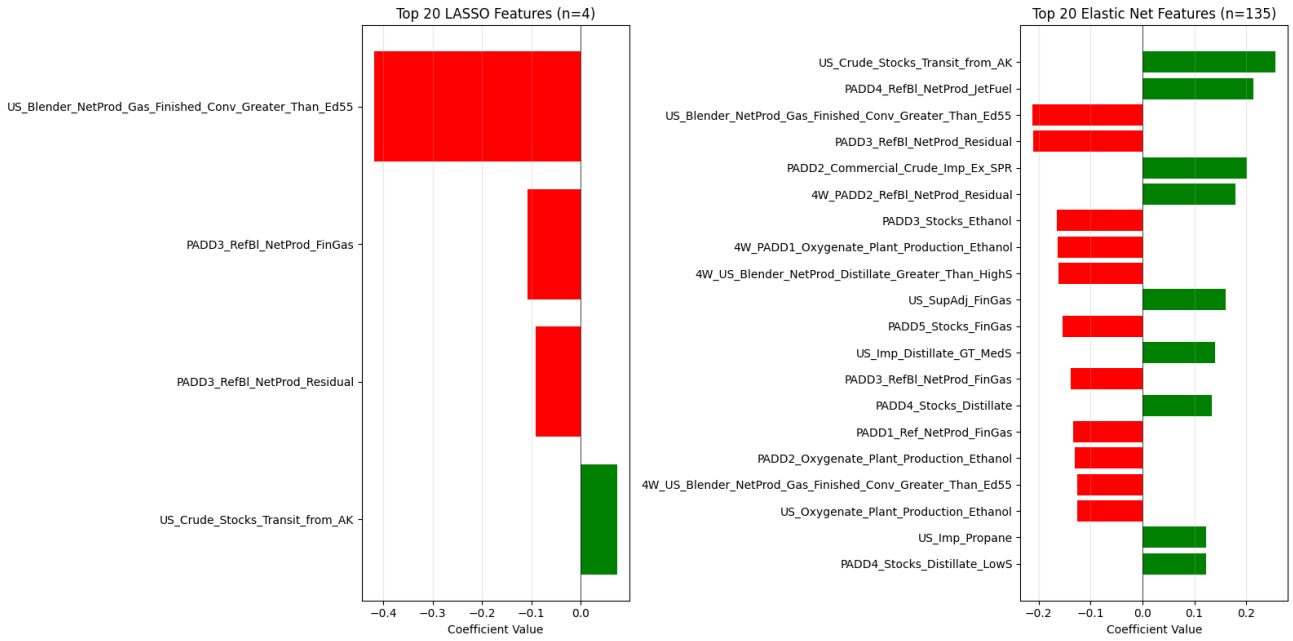

Figure 8: PCA Results on Weekly Variables

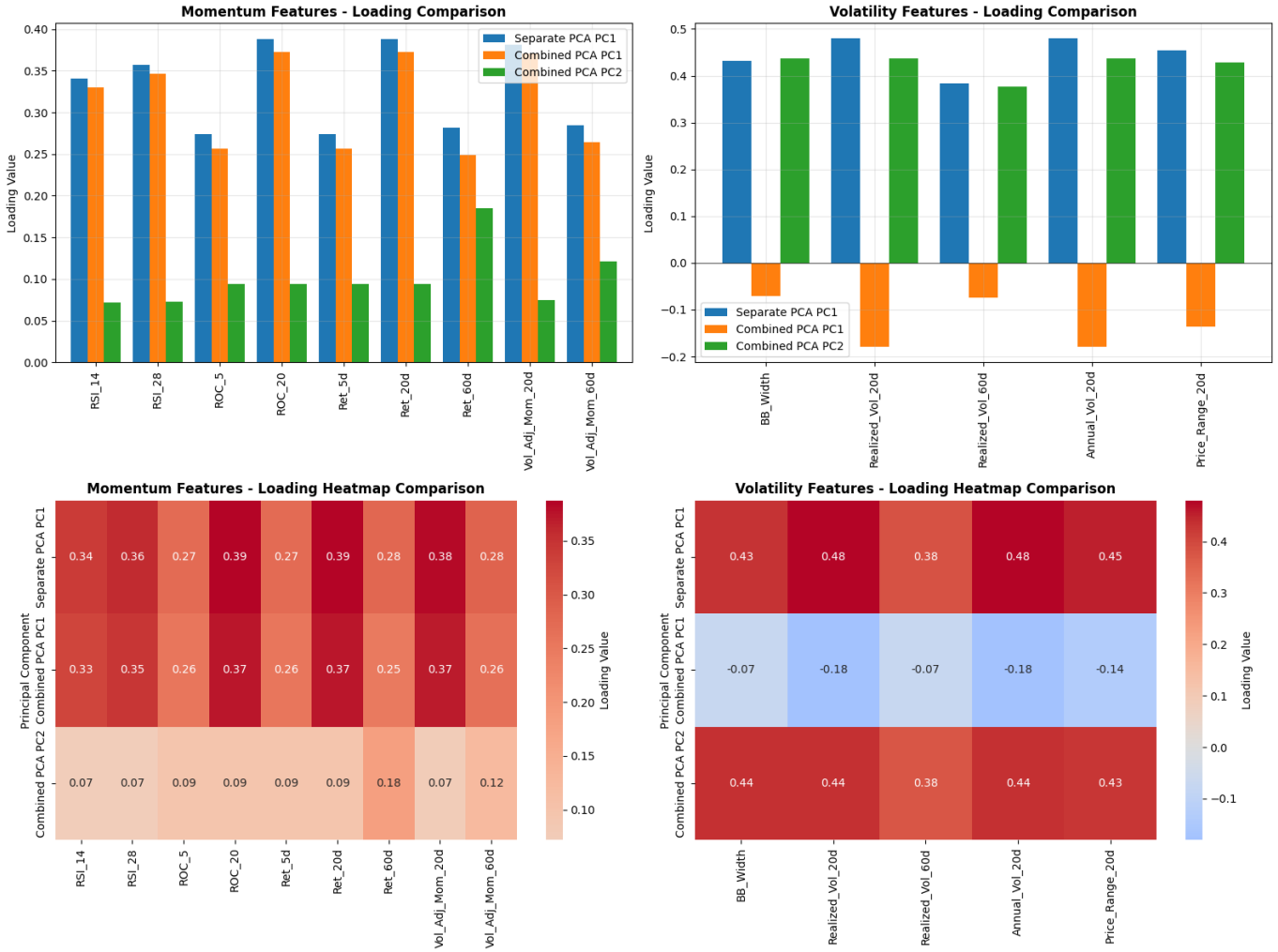

Figure 9: Feature Engineering on Daily Variables