final_report

Predicting Next-Day Natural Gas Price Direction with Classical ML Models (2010–2024)

Course: AMML Term Project
Date: February 2026

---

1. Executive Summary

This report presents an applied machine learning study aimed at predicting the next-day price direction of U.S. natural gas futures. Using daily observations spanning 2010 to 2024, we constructed a binary classification task: whether the next day's closing price would move Up or Not-Up. The dataset comprised 5,191 trading days with 26 features drawn from futures prices, storage inventories, drilling activity, and regional weather data. We enforced a strict chronological train/test split at January 1, 2022, yielding 4,372 training and 819 test observations. Five classical classifiers were compared: Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), Stochastic Gradient Descent (SGD), and Support Vector Machine (SVM). Each model was evaluated in baseline, grid-search-optimized, and reduced-feature configurations. Class imbalance (approximately 70% Not-Up, 30% Up) made accuracy a misleading metric; we therefore focused on F1 score for the Up class, ROC-AUC, and PR-AUC. Across all models, class weighting and hyperparameter tuning proved essential for detecting the minority Up class. The best overall model was Random Forest (optimized), which achieved a ROC-AUC of 0.760 and PR-AUC of 0.544, indicating a modest but real predictive signal. A feature-reduction ablation study showed that removing redundant regional weather variables and raw price levels preserved or slightly improved performance, suggesting that a leaner feature set generalizes better.

---

2. Introduction

2.1 Problem Statement

Predicting the short-term direction of commodity prices is a well-known challenge in financial machine learning. Natural gas markets are especially volatile due to their sensitivity to weather, storage levels, and geopolitical factors. The core task of this project was to predict whether the next day's natural gas futures price would move Up or Not-Up, framed as a binary classification

problem. This task is inherently difficult: financial returns are noisy, subject to regime shifts, and weakly stationary at best. Any predictive signal, if it exists, is expected to be modest.

2.2 Motivation

Rather than claiming to build a profitable trading system, the goal of this study was signal extraction and methodological comparison. If classical ML models can identify even a weak directional signal from publicly available fundamental and technical features, this represents useful information for strategy research and risk management. The study also serves as a practical exercise in applying machine learning to a real-world time-series classification problem, with attention to pitfalls such as data leakage, class imbalance, and look-ahead bias.

2.3 Approach Overview

We compared five classical classifiers: Logistic Regression, Decision Tree, Random Forest, SGD Classifier, and SVM. Each model was trained in a baseline configuration (default parameters) and an optimized configuration (via GridSearchCV with time-series cross-validation). We also conducted a feature-reduction ablation study, removing redundant regional weather variables and raw price levels to test whether a smaller feature set could match or improve performance. All evaluations used class-aware metrics: F1 score for the Up class, ROC-AUC, and Precision-Recall AUC.

---

3. Dataset and Target Definition

3.1 Data Sources and Size

The dataset consisted of 5,191 daily observations of U.S. natural gas market data, spanning from 2010 to 2024. The data was compiled from multiple public sources into a single analysis-ready file (`master_dataset_ml_ready_labelled.csv`). Features were drawn from four broad categories: futures prices and curve structure, natural gas storage inventories, drilling activity, and regional weather data.

3.2 Feature Groups

The full feature set comprised 26 variables organized into the following groups:

- **Momentum and returns:** `ret_1` , `ret_3` , `ret_5` , `ret_10` — rolling 1-, 3-, 5-, and 10-day returns capturing recent price momentum.
- **Futures curve:** `curve_spread` (contract_2_price minus contract_1_price), `contract_1_price` , `contract_2_price` — capturing the term structure of futures prices.

- **Supply and inventory:** `storage_bcf` (current storage level in billion cubic feet), `storage_bcf_change_7d` (7-day change in storage), `us_gas_rigs` (active drilling rigs).
- **Weather demand:** Regional heating and cooling degree days (`HDD_TX` , `HDD_PA` , `HDD_IL` , `HDD_NY` , `CDD_TX` , `CDD_PA` , `CDD_IL` , `CDD_NY`) along with aggregated totals (`HDD_total` , `CDD_total` , `net_weather`).
- **Calendar:** `year` , `month` , `day_of_year` , `day_of_week` , `quarter` .

3.3 Label Definition

The target variable `PriceUp` was defined as 1 if the next day's closing price was higher than the current day's, and 0 otherwise. The class distribution was imbalanced: approximately 70% of observations were labeled Not-Up and 30% were labeled Up. This imbalance had important consequences for model evaluation, as a naive majority-class classifier could achieve roughly 67% accuracy without detecting any Up days.

---

4. Data Cleaning, Preprocessing, and Leakage Control

4.1 Missing Values and Outliers

The dataset was pre-processed into a machine-learning-ready format prior to model training. The file name (`master_dataset_ml_ready_labelled.csv`) reflects that cleaning steps such as handling missing values and aligning time series from different sources had already been performed upstream. Engineered features like rolling returns (`ret_1` through `ret_10`) and `storage_bcf_change_7d` naturally introduce a small number of missing values at the start of the series; these rows were excluded during training.

4.2 Feature Scaling

Models that are sensitive to feature scale — specifically SVM and SGD — were trained using `StandardScaler` applied within the training pipeline. Scaling parameters were fit on the training set only and applied to the test set, avoiding information leakage. Tree-based models (Decision Tree, Random Forest) do not require scaling and were trained on unscaled features.

4.3 Leakage Controls

We implemented several controls to guard against data leakage:

- **Calendar features retained in full set, removed in ablation:** Calendar variables (`year` , `month` , `day_of_week` , `day_of_year` , `quarter`) were included in the full feature set but their potential to introduce weekday effects was noted. The reduced feature set retained these for consistency, but the curated set excluded them.

- **Raw price levels:** The full feature set included `contract_1_price` and `contract_2_price`. Because raw price levels can introduce look-ahead bias (models may learn to associate certain price ranges with certain periods), the reduced feature set removed these and retained only `curve_spread`, which captures relative pricing.
- **Storage update indicator:** The `storage_update` variable (indicating whether an EIA storage report was released that day) was excluded from all model feature sets and reserved for potential event-study analysis.

---

5. Experimental Design

5.1 Train/Test Split

We used a strict chronological split with January 1, 2022 as the cutoff date. All observations before this date formed the training set (4,372 samples) and all observations from 2022 onward formed the test set (819 samples). No shuffling was applied, preserving temporal order and preventing look-ahead bias.

5.2 Cross-Validation

Hyperparameter tuning was performed using `GridSearchCV` with `TimeSeriesSplit` as the cross-validation strategy. This ensured that within each fold, the training portion always preceded the validation portion in time, consistent with the temporal nature of the data. The number of folds varied between 3 and 5 depending on the model's computational cost.

5.3 Evaluation Metrics

Given the class imbalance (~70/30), accuracy alone was a misleading metric. A model predicting Not-Up for every observation would achieve approximately 67% accuracy while detecting zero Up days. We therefore evaluated models using:

- **F1 Score (Up class):** The harmonic mean of precision and recall for the minority class, balancing the trade-off between false positives and missed detections.
- **ROC-AUC:** Area under the Receiver Operating Characteristic curve, measuring overall ranking ability independent of a specific classification threshold.
- **PR-AUC:** Area under the Precision-Recall curve, which is more informative than ROC-AUC under class imbalance because it focuses on the minority class.

5.4 Class Weighting and Threshold Tuning

For all models, we included `class_weight` as a hyperparameter in the grid search, testing configurations that up-weighted the minority Up class (e.g., `{0: 1.0, 1: 2.5}` or `balanced`).

This was critical: without class weighting, most models defaulted to predicting the majority class almost exclusively, yielding near-zero recall for Up. Threshold tuning (adjusting the probability cutoff for classification) was also explored where applicable.

---

6. Models and Hyperparameter Tuning

For each model, we followed a consistent evaluation template: (a) rationale for why the model might work, (b) baseline behavior, (c) what was tuned, (d) optimized test metrics, and (e) interpretation.

6.1 Logistic Regression

Rationale. Logistic Regression provides a simple, interpretable linear baseline. If the relationship between features and price direction is approximately linear, LR should capture it with minimal overfitting.

Baseline behavior. With default parameters (C=1.0, no class weighting), LR achieved 68.0% accuracy but only 0.342 F1(Up), with recall of just 25.3%. The model strongly favored the majority class.

Tuning. GridSearchCV optimized the regularization strength C and class weights. The best configuration used strong regularization (C=0.01) with class weighting {0: 1, 1: 2.5}.

Optimized results. Accuracy dropped to 54.1% but F1(Up) improved to 0.529 with recall reaching 78.4%. ROC-AUC was 0.657 and PR-AUC was 0.481.

Table 1: Logistic Regression Results

Variant	Accuracy	F1 (Up)	Precision (Up)	Recall (Up)	ROC-AUC	PR-AUC
Baseline	0.680	0.342	0.527	0.253	0.667	0.486
Optimized	0.541	0.529	0.399	0.784	0.657	0.481

Interpretation. LR's linear decision boundary limited its ability to capture the nonlinear feature interactions present in this data. The optimized model traded accuracy for much better minority-class detection, but its overall discriminative power (ROC-AUC 0.657) was the weakest among the tuned models.

6.2 Decision Tree

Rationale. Decision Trees can capture nonlinear relationships and feature interactions through recursive splitting, and they are highly interpretable. However, they are prone to overfitting, especially on noisy financial data.

Baseline behavior. With no depth constraint and no class weighting, the baseline DT achieved 65.7% accuracy and 0.452 F1(Up), with moderate recall (43.1%).

Tuning. GridSearchCV optimized `max_depth`, `min_samples_leaf`, `min_samples_split`, and `class_weight`. The best configuration was a shallow tree: max_depth=3, min_samples_leaf=8, with class weighting {0: 1.0, 1: 2.5}.

Optimized results. F1(Up) rose to 0.631 — the highest of any model — with ROC-AUC of 0.731 and PR-AUC of 0.491.

Table 2: Decision Tree Results

Variant	Accuracy	F1 (Up)	Precision (Up)	Recall (Up)	ROC-AUC	PR-AUC
Baseline	0.657	0.452	0.475	0.431	0.599	0.392
Optimized	0.617	0.631	0.461	1.000	0.731	0.491
Reduced	0.617	0.631	0.461	1.000	0.742	0.498

Interpretation. The optimized DT achieved perfect recall (1.0), meaning it classified every test observation as Up. While this maximized F1(Up) through high recall, precision was only 46.1%, and accuracy dropped to 61.7%. The high ROC-AUC (0.731–0.742) indicates the tree's probability estimates had genuine ranking ability, but the model's operating point was aggressively biased toward the Up class. This behavior reflects the strong class weighting combined with the shallow depth constraint.

6.3 Random Forest

Rationale. Random Forest builds an ensemble of decorrelated decision trees, reducing variance and improving generalization. It handles nonlinear interactions naturally, is robust to noisy features, and provides built-in feature importance estimates.

Baseline behavior. With default parameters (100 trees, no depth limit, no class weighting), RF achieved 68.1% accuracy but only 0.339 F1(Up) with recall of 24.9%. Like other baselines, it strongly favored the majority class. Notably, even at baseline its ROC-AUC (0.753) was already the highest, suggesting strong ranking ability.

Tuning. GridSearchCV optimized `n_estimators`, `max_depth`, `min_samples_leaf`, and `class_weight`. The best configuration used 800 trees with max_depth=5, min_samples_leaf=40, and class weighting {0: 1.0, 1: 2.5}.

Optimized results. F1(Up) improved to 0.603 with recall of 80.7% and precision of 48.1%. ROC-AUC reached 0.760 — the highest of any model — and PR-AUC was 0.544.

Table 3: Random Forest Results

Variant	Accuracy	F1 (Up)	Precision (Up)	Recall (Up)	ROC-AUC	PR-AUC
Baseline	0.681	0.339	0.532	0.249	0.753	0.515
Optimized	0.651	0.603	0.481	0.807	0.760	0.544
Reduced	0.654	0.604	0.484	0.803	0.759	0.537

Interpretation. RF delivered the best overall ranking performance across all models (ROC-AUC 0.760, PR-AUC 0.544). Unlike the Decision Tree, RF maintained a more balanced precision-recall trade-off: 48% precision at 81% recall means that roughly half of its Up predictions were correct, while catching the vast majority of actual Up days. The ensemble averaging smoothed out the aggressive behavior seen in the single tree.

6.4 SGD Classifier

Rationale. The SGD Classifier trains a linear model via stochastic gradient descent, offering scalability and flexible loss functions. With elasticnet regularization, it can perform implicit feature selection by combining L1 and L2 penalties.

Baseline behavior. With default parameters (alpha=0.0001, L2 penalty, no class weighting), SGD achieved 67.3% accuracy but an extremely low F1(Up) of 0.124, with recall of just 7.1%. The model almost entirely ignored the minority class.

Tuning. GridSearchCV optimized `alpha`, `penalty` (L2 vs elasticnet), `l1_ratio`, and `class_weight`. The best configuration used alpha=0.1, elasticnet penalty with l1_ratio=0.5, and balanced class weights.

Optimized results. F1(Up) improved dramatically to 0.584, with recall of 67.3% and precision of 51.6%. ROC-AUC was 0.668 and PR-AUC was 0.469.

Table 4: SGD Classifier Results

Variant	Accuracy	F1 (Up)	Precision (Up)	Recall (Up)	ROC-AUC	PR-AUC
Baseline	0.673	0.124	0.514	0.071	0.648	0.451
Optimized	0.685	0.584	0.516	0.673	0.668	0.469
Reduced	0.685	0.584	0.516	0.673	0.668	0.469

Interpretation. SGD's performance improved substantially with class weighting and regularization tuning. The optimized model achieved the best accuracy (68.5%) among tuned models while maintaining solid F1(Up). The identical results between the optimized and reduced feature sets suggest that the elasticnet regularization was already effectively down-weighting the removed features, making the explicit reduction redundant.

6.5 Support Vector Machine (SVM)

Rationale. SVMs with nonlinear kernels (RBF) can learn complex decision boundaries that linear models cannot. The kernel trick allows SVM to implicitly map features into a higher-dimensional space where the classes may be more separable.

Baseline behavior. With default parameters (C=1.0, gamma=scale, no class weighting), SVM exhibited the most extreme majority-class bias of any model: it predicted Up for only 1 of 819 test observations, yielding F1(Up) of just 0.007 with recall near zero (0.4%).

Tuning. GridSearchCV optimized `C`, `gamma`, and `class_weight` for the RBF kernel. The best configuration used C=0.1, gamma=0.01, and class weighting {0: 1.0, 1: 2.5}.

Optimized results. F1(Up) improved to 0.578 with recall of 73.2% and precision of 47.7%. ROC-AUC reached 0.675 and PR-AUC was 0.490.

Table 5: SVM Results

Variant	Accuracy	F1 (Up)	Precision (Up)	Recall (Up)	ROC-AUC	PR-AUC
Baseline	0.673	0.007	1.000	0.004	0.614	0.445
Optimized	0.648	0.578	0.477	0.732	0.675	0.490
Reduced	0.667	0.567	0.494	0.665	0.673	0.489

Interpretation. SVM showed the most dramatic improvement from baseline to optimized of any model. The baseline SVM completely collapsed to majority-class prediction, but with appropriate class weighting and kernel tuning, it achieved competitive results. The reduced-feature SVM traded some recall for improved precision, showing a more conservative decision boundary.

---

7. Feature Reduction / Ablation Study

7.1 Hypotheses

We hypothesized that two groups of features were candidates for removal:

1. **Regional weather redundancy.** The dataset included heating and cooling degree days for four states (TX, PA, IL, NY) as well as their aggregated totals (HDD_total, CDD_total). The regional values are highly correlated with the totals, so including both may introduce multicollinearity without adding predictive value.
2. **Raw price levels.** Including `contract_1_price` and `contract_2_price` risks leakage: raw price levels encode information about the time period (prices were much higher in 2022 than 2015), and models may learn period-specific patterns that do not generalize. The curve spread (`curve_spread`) captures relative pricing without this risk.

7.2 Features Removed

The reduced feature set (20 features) was constructed by removing:

- `CDD_PA` , `CDD_IL` , `HDD_PA` , `HDD_IL` (4 redundant regional weather variables; TX and NY retained along with aggregated totals)
- `contract_1_price` , `contract_2_price` (2 raw price levels; `curve_spread` retained)

7.3 Results

Table 6: Ablation Study — Optimized vs Reduced Feature Sets

Model	F1 (Up) Full	F1 (Up) Reduced	ROC-AUC Full	ROC-AUC Reduced
SGD	0.584	0.584	0.668	0.668
SVM	0.578	0.567	0.675	0.673
DT	0.631	0.631	0.731	0.742
RF	0.603	0.604	0.760	0.759

Performance was remarkably stable across the feature reduction. SGD produced identical results with both feature sets, confirming that its elasticnet regularization was already ignoring the removed features. Decision Tree actually improved its ROC-AUC from 0.731 to 0.742 with the reduced set, suggesting the removed features introduced noise. RF and SVM showed negligible changes. These results support the hypothesis that the removed features were redundant and that a leaner model generalizes at least as well.

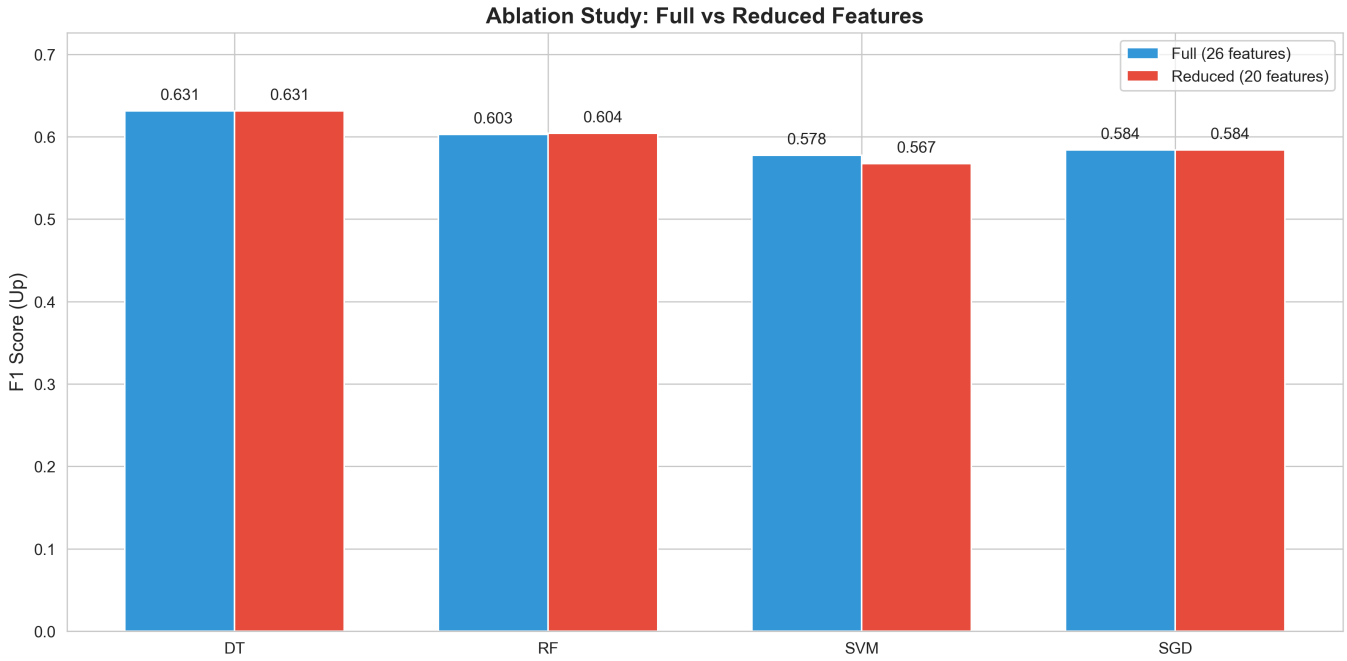

Figure 1: Comparison of model performance metrics between full and reduced feature sets.

---

8. Results Comparison

8.1 Summary Table

Table 7: Best Configuration Per Model (Test Set Performance)

Model	Variant	Accuracy	F1 (Up)	Precision (Up)	Recall (Up)	ROC-AUC	PR-AUC
LR	Optimized	0.541	0.529	0.399	0.784	0.657	0.481
SGD	Optimized	0.685	0.584	0.516	0.673	0.668	0.469
SVM	Optimized	0.648	0.578	0.477	0.732	0.675	0.490
DT	Reduced	0.617	0.631	0.461	1.000	0.742	0.498
RF	Optimized	0.651	0.603	0.481	0.807	**0.760**	**0.544**

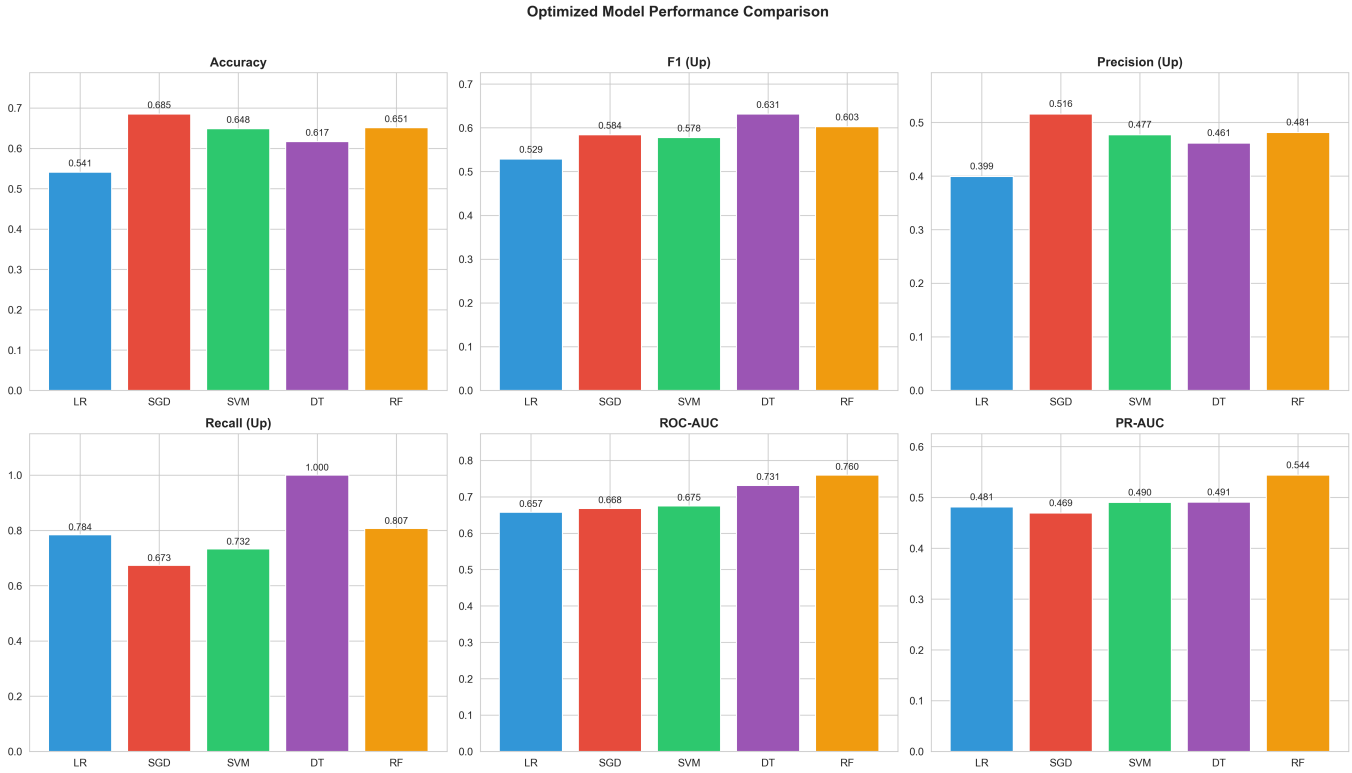

Figure 2: Bar chart comparing key metrics across all model variants.

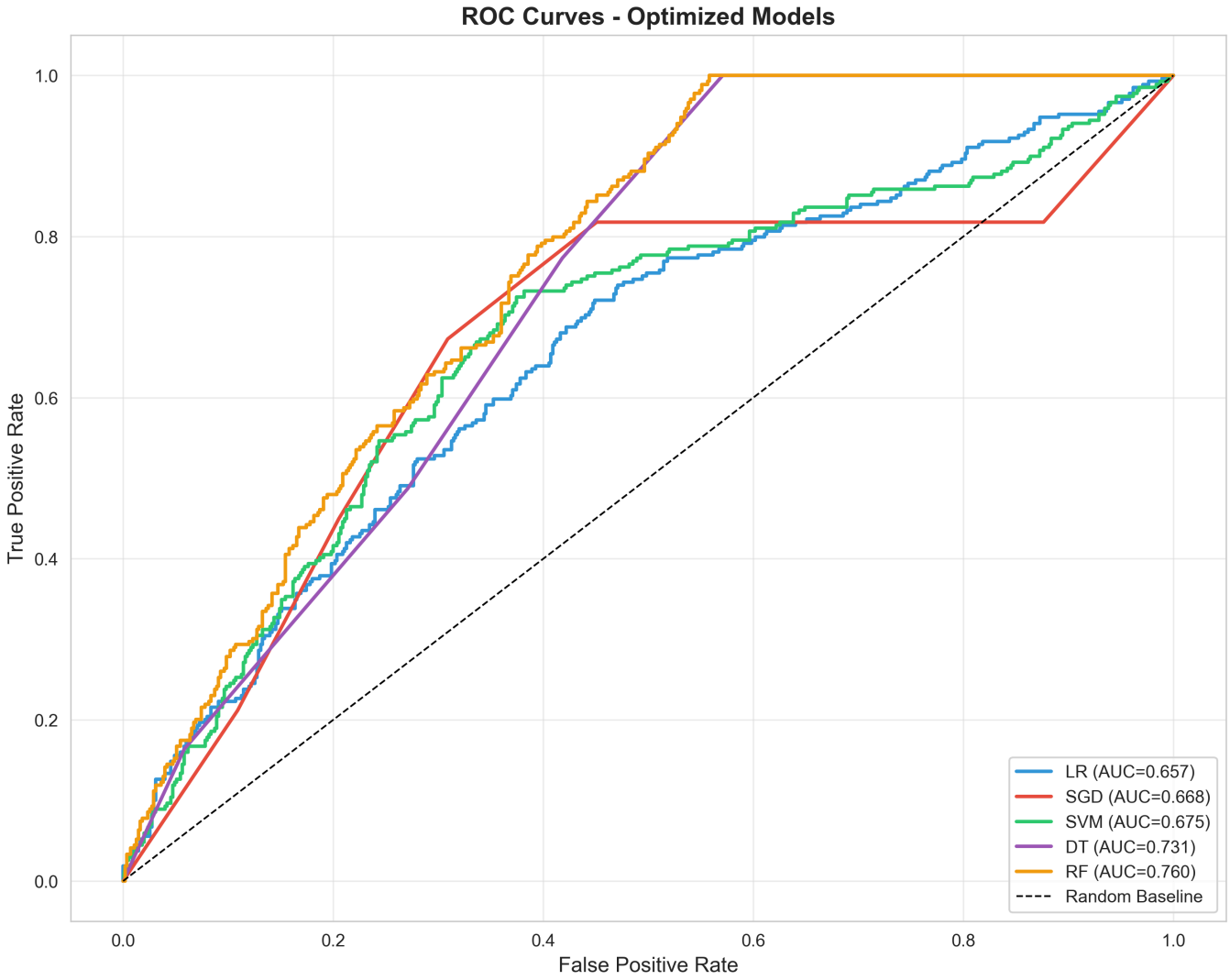

Figure 3: ROC curves for optimized model variants.

8.2 Narrative

The results reveal a clear hierarchy. Random Forest achieved the best ranking performance (ROC-AUC 0.760, PR-AUC 0.544), indicating the strongest ability to separate Up from Not-Up days across all probability thresholds. Decision Tree produced the highest F1(Up) (0.631) but did so by classifying all test observations as Up — an operating point that sacrifices specificity for sensitivity. Among the linear models, SGD performed best (F1 0.584, accuracy 68.5%), slightly outperforming SVM (F1 0.578) and substantially outperforming Logistic Regression (F1 0.529).

A consistent pattern emerged across all models: baseline configurations with no class weighting achieved ~67–68% accuracy but near-zero F1(Up), while optimized configurations with class weighting sacrificed some accuracy to dramatically improve minority-class detection. This underscores that class weighting and threshold tuning were more important than the choice of algorithm for this particular task.

The ROC-AUC values clustered into two tiers: nonlinear models (RF 0.760, DT 0.742) meaningfully outperformed linear models (SVM 0.675, SGD 0.668, LR 0.657), suggesting that the relationship between features and price direction contains nonlinear interactions that linear models cannot fully capture.

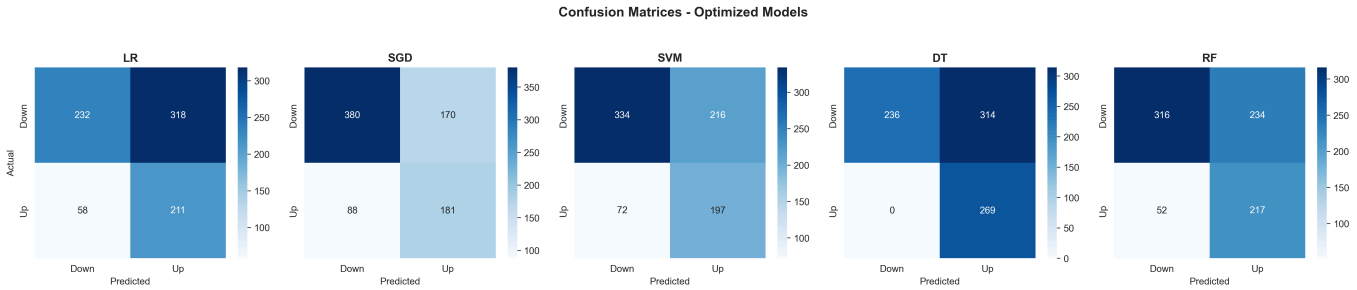

Figure 4: Confusion matrices for all model variants.

---

9. Discussion

9.1 What Mattered Most

Two factors dominated model performance more than the choice of algorithm: **class-aware tuning** and **feature engineering**. Without class weighting, every model defaulted to predicting the majority class. The simple act of up-weighting the minority class in the loss function — or using balanced class weights — unlocked meaningful predictive signal that was already present in the features but suppressed by the imbalanced optimization objective.

Feature engineering also played a critical role. The momentum features (rolling returns) and the futures curve spread were consistently among the most important predictors, as reflected in the Random Forest feature importance analysis.

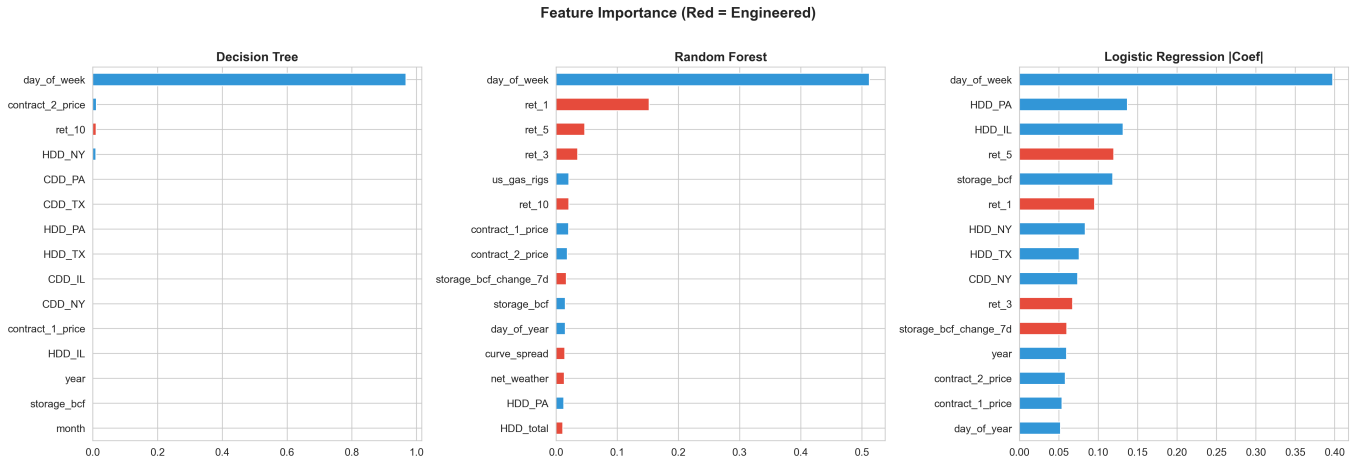

Figure 5: Feature importance from Random Forest model.

9.2 Why Nonlinear Models Performed Better

Random Forest and Decision Tree outperformed the three linear/kernel models on ROC-AUC. This suggests that the mapping from features to price direction involves interactions and threshold effects that linear decision boundaries cannot capture. For example, the relationship between storage levels and price direction likely depends on the current season (heating vs. cooling) and the prevailing price regime — exactly the kind of conditional interaction that tree-based models handle naturally through recursive splits.

9.3 Practical Interpretation

The best models achieved high recall for the Up class (80–100%) but moderate precision (46–48%). In practical terms, this means the models catch most genuine Up days but also generate a substantial number of false Up signals. If these models were used to inform trading decisions, the false positive rate would require additional filtering (e.g., confidence thresholds, position sizing rules, or combining with other signals) to be actionable. The ROC-AUC of 0.760 indicates a real but modest predictive edge — better than random but far from a reliable standalone signal.

---

10. Limitations and Future Work

Several limitations should be noted when interpreting these results.

Class imbalance. The roughly 70/30 split between Not-Up and Up observations meant that all models required explicit class-weighting adjustments. The high F1(Up) scores achieved by some models (particularly Decision Tree) partly reflect aggressive threshold settings rather than genuinely superior pattern recognition.

Non-stationarity. Financial time series are non-stationary: the statistical relationships between features and returns change over time due to shifts in market structure, regulation, and participant behavior. A model trained on 2010–2021 data may not generalize to 2022–2024 conditions, and there is no guarantee that observed patterns will persist.

Backward-looking features. All features in this study were backward-looking (lagged returns, historical storage, past weather). The models had no access to forward-looking information such as weather forecasts, scheduled maintenance, or geopolitical developments, which are major drivers of short-term price movements.

Missing information sources. The dataset did not include news sentiment, options-implied volatility, or cross-commodity signals (e.g., crude oil, electricity), all of which could provide additional predictive value.

Threshold sensitivity. The reported F1 scores depend on the classification threshold chosen. Small changes in threshold can produce meaningfully different precision-recall trade-offs, and

the optimal threshold identified on the training set may not be optimal out of sample.

Not evidence of profitability. These results should not be interpreted as evidence that a profitable trading strategy exists. Transaction costs, slippage, market impact, and the practical difficulty of acting on next-day signals were not modeled. Rigorous backtesting with realistic trading assumptions would be required before drawing any conclusions about economic value.

Future work could explore several directions: incorporating additional data sources (news, cross-commodity), testing more flexible models (gradient boosting, neural networks), implementing rolling-window retraining to adapt to regime changes, and conducting formal backtesting with transaction costs.

---

11. Conclusion

This study compared five classical machine learning models for predicting next-day natural gas price direction using fundamental and technical features. The key findings were: (1) class-aware tuning — particularly class weighting — was essential for detecting the minority Up class under 70/30 imbalance; (2) nonlinear models (Random Forest, Decision Tree) outperformed linear models (LR, SGD, SVM) on ranking metrics, suggesting feature interactions matter; and (3) a reduced feature set removing redundant regional weather variables and raw price levels matched or improved full-feature performance. The best overall model, Random Forest, achieved a ROC-AUC of 0.760 and PR-AUC of 0.544 — a modest but genuine signal. These results establish a solid baseline for further research into natural gas price prediction, while the leakage-controlled, time-series-aware methodology provides a sound framework for future feature and model exploration.

---

12. References

1. U.S. Energy Information Administration (EIA). Weekly Natural Gas Storage Report. https://www.eia.gov/naturalgas/storage/
2. U.S. Energy Information Administration (EIA). Natural Gas Futures Prices. https://www.eia.gov/naturalgas/data.php
3. Baker Hughes. North America Rig Count. https://bakerhughes.com/rig-count
4. NOAA National Centers for Environmental Information. Heating and Cooling Degree Day Data. https://www.ncei.noaa.gov/
5. Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

Appendix

A. Hyperparameter Grids

Table A1: Tuned Hyperparameters by Model

Model	Parameter	Values Searched	Best
LR	C	Grid including 0.01–100	0.01
LR	class_weight	None, {0:1, 1:2.5}	{0:1, 1:2.5}
SGD	alpha	Grid including 0.0001–0.1	0.1
SGD	penalty	L2, elasticnet	elasticnet
SGD	l1_ratio	0.5	0.5
SGD	class_weight	None, balanced	balanced
SVM	C	Grid including 0.1–10	0.1
SVM	gamma	Grid including 0.01–scale	0.01
SVM	class_weight	None, {0:1, 1:2.5}	{0:1, 1:2.5}
DT	max_depth	None, 3, 5, ...	3
DT	min_samples_leaf	1, 8, ...	8
DT	class_weight	None, {0:1, 1:2.5}	{0:1, 1:2.5}
RF	n_estimators	100, 200, 400, 800	800
RF	max_depth	None, 3, 5, 10	5
RF	min_samples_leaf	1, 10, 20, 40	40
RF	class_weight	None, {0:1, 1:2.5}	{0:1, 1:2.5}

B. Confusion Matrices

See Figure 4 in the main text, or the full confusion matrix visualization at `model_outputs/confusion_matrices.png`.

C. ROC and PR Curves

See Figure 3 in the main text, or the full ROC curve visualization at `model_outputs/roc_curves.png`.

D. Class Distribution

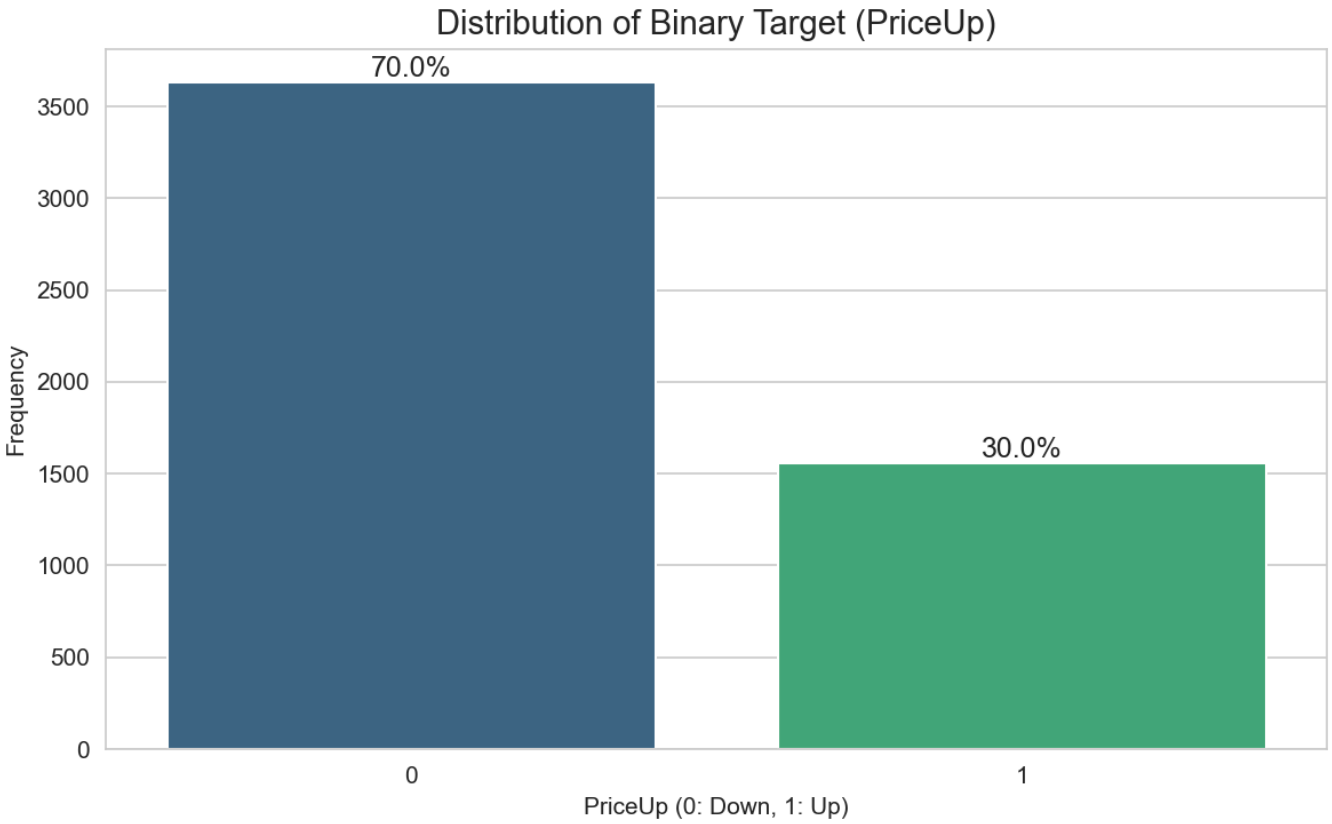

Figure A1: Distribution of the PriceUp target variable in the full dataset.