Trading Prediction Analysis Platform - Executive Summary

Evaluation Scenarios

Baseline Strategies (4 strategies - no predictions)

1. **Immediate Sale**: Liquidates entire inventory on day 1. Floor performance benchmark.
2. **Equal Batches**: Sells inventory/N portions every 30 days, where N = 365/30 ≈ 12 batches. Pure calendar-based liquidation.
3. **Price Threshold**: Sells 25% when price exceeds 30-day moving average by 5%. Tests simple price momentum.
4. **Moving Average**: Sells 25% when price crosses above 30-day moving average. Functionally identical to Price Threshold.

Prediction-Based Strategies (5 strategies - enhanced with forecasts)

1. **Consensus**: Computes bullish_pct (fraction of paths with positive 14-day returns) and trend acceleration (comparing mean daily changes in days 1-7 vs 8-14). Classifies market regime as strong uptrend, confident bullish, decelerating, high uncertainty, or bearish. Varies batch size from 10% (uncertain) to 60% (strong uptrend) based on signal classification.

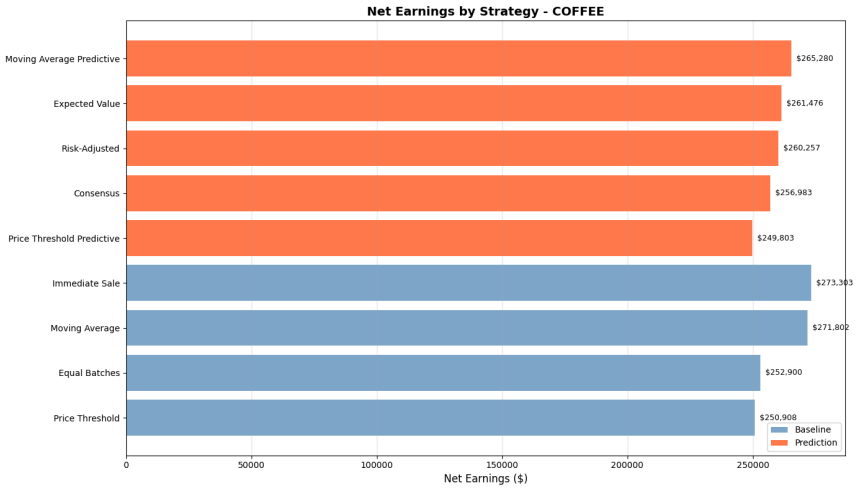
2. **Expected Value**: For each forecast horizon h=1..14, computes time-normalized returns (predicted_price[h] - current) / (current × h). Finds optimal horizon h* with maximum normalized return. If return exceeds threshold, sells batch sized by return-to-volatility ratio.

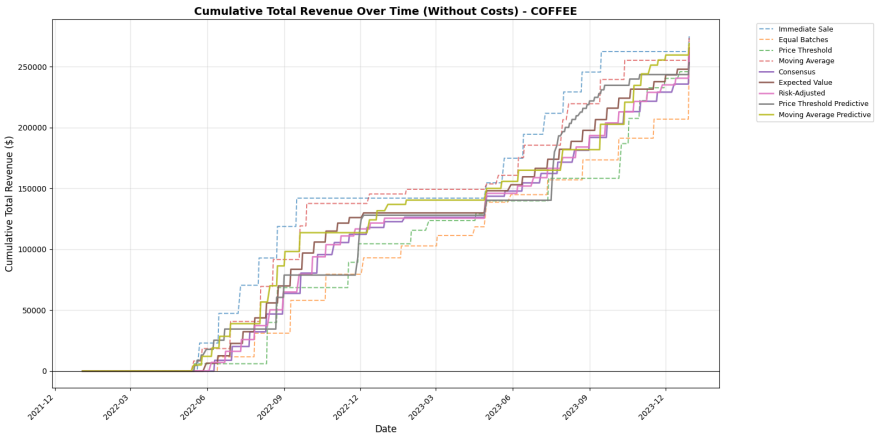
3. **Risk-Adjusted**: Calculates Sharpe ratio across all predicted price paths (mean return / std of returns at 14-day horizon). Sells larger batches when Sharpe exceeds thresholds (>1.5, >1.0, >0.8), otherwise holds.

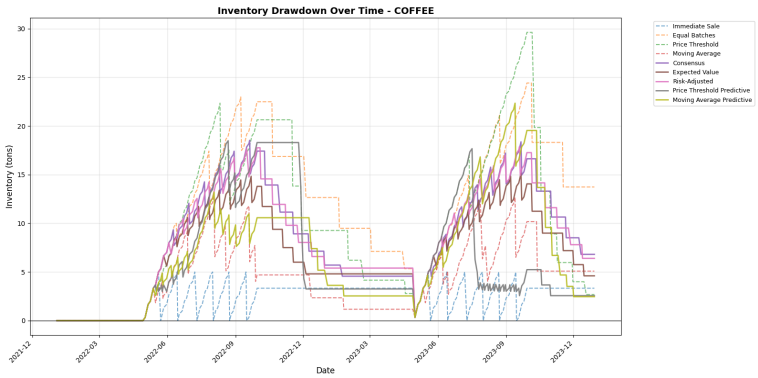
4. **Price Threshold Predictive**: Baseline trigger is price > 1.05 × 30d_avg. Enhancement: analyzes prediction trajectory slope. If strongly rising (>70% days

positive, endpoint > current), HOLDS for further gains. If peak passed (endpoint < current), sells 30% aggressively. Otherwise baseline 25%.

5. **Moving Average Predictive**: Baseline trigger is price > 30d_avg. Enhancement: compares days 1-7 vs 8-14 prediction slopes to detect deceleration. Holds if strongly rising, sells 40% if peak passed, 30% if decelerating, else 25% baseline.







Statistical Testing & Metrics

Hypothesis Testing

First, identify the best-performing baseline strategy (highest net earnings among Immediate Sale, Equal Batches, Price Threshold, Moving Average). Then run three separate comparisons: Consensus vs best baseline, Expected Value vs best baseline, Risk-Adjusted vs best baseline.

Paired t-test: Each strategy produces a time series of daily portfolio values over 365 days. For each day, compute the change from previous day (e.g., day 50: $102k → day 51: $103k = +$1k). For each of the three prediction strategies, align its daily changes with the best baseline's daily changes on the same calendar days. The t-test asks: "Are Consensus's daily changes systematically larger than the best baseline's daily changes?" Repeat for Expected Value vs best baseline and Risk-Adjusted vs best baseline. Reports three separate p-values.

Effect Size (Cohen's d): For each prediction strategy vs best baseline pair, compute 365 paired differences (prediction_day_change - baseline_day_change for each day). Cohen's d = mean(365 differences) / std(365 differences). Produces three Cohen's d values, one for each prediction strategy vs best baseline comparison. Quantifies how many standard deviations better each prediction strategy performs.

Bootstrap CI: For each strategy, you have a list of 365 daily changes like [+$100, -$50, +$200, +$75, ...]. To bootstrap: randomly pick one value from this list (say -$50), write it down, then PUT IT BACK in the list so it can be picked again. Pick another value (might be -$50 again, or something else), write it down, put it back. Repeat 365 times. Now sum these 365 randomly-selected values to get one possible terminal portfolio value. The "with replacement" means after picking a value, you return it to the list so it can be picked

multiple times. Repeat this entire process 1000 times to build a distribution of 1000 possible terminal values. Report [2.5%, 97.5%] percentiles as confidence interval.

Prediction Quality Assessment

Random Forest Regression: This validates whether our prediction features actually contained useful information. At each trading decision point, we extracted characteristics from the predictions: expected_return (median 14-day forecast return), pct_positive_days (fraction of forecast days trending up), bullish_pct (fraction of paths showing gains), volatility, trend_acceleration. After 14 days pass, we measure what the price actually did: actual_return = (price_14days_later - price_at_decision) / price_at_decision. The Random Forest asks: "Given that we saw expected_return=3% and bullish_pct=75% in our predictions, could we predict that actual_return would be 2.5%?" This is NOT calculating optimal decisions—it's checking whether prediction features like "high expected return" or "strong consensus" actually correlated with real price movements. High R^2 means predictions contained genuine signal about future prices. Low R^2 means prediction features had no relationship to actual outcomes.

Feature Importance: GINI importance ranks which prediction characteristics (expected_return, pct_positive_days, volatility, etc.) best predicted actual outcomes. If expected_return has high importance, it means predictions showing high expected returns tended to precede actual high returns. If volatility has low importance, it means prediction uncertainty didn't help forecast actual movements.

Core Performance Metrics

Net Earnings: Total revenue from all sales minus transaction costs (percentage of sale value) minus daily storage costs (accrued on held inventory). Computed independently for each of the 9 strategies (4 baselines + 5 prediction-based).

Trading Patterns: Number of sales executed, time span from first to last sale, average days between consecutive sales. Measured per strategy.

Risk-Adjusted Returns: Sharpe ratio = annualized_return / annualized_volatility. Quantifies return per unit of risk. Computed per strategy.

Sensitivity Testing

For the three main prediction strategies (Consensus, Expected Value, Risk-Adjusted), systematically varies parameters: consensus_threshold (0.5-0.8), min_return (0.01-0.05), Sharpe threshold (0.8-1.5), plus cost assumptions (transaction 0.5-2.0%, storage ±50%). Re-runs full backtest for each parameter combination. Measures earnings degradation relative to baseline to identify robust vs fragile strategies.

Output Deliverables

Per-Commodity Statistics: CSV with [t_statistic, p_value, cohens_d, CI_bounds, earnings_difference] for each prediction vs baseline comparison. Bootstrap summaries with terminal value distributions.

Cross-Commodity Dashboard: 4-panel chart showing (1) earnings comparison, (2) absolute $ advantage, (3) relative % advantage, (4) R^2 model quality across commodities.

Decision Criterion: Strategy approved for deployment if ($p < 0.05$) AND ($|\text{Cohen's d}| > 0.2$) AND (earnings_advantage > 0)—i.e., statistically significant, economically meaningful, and profitable.