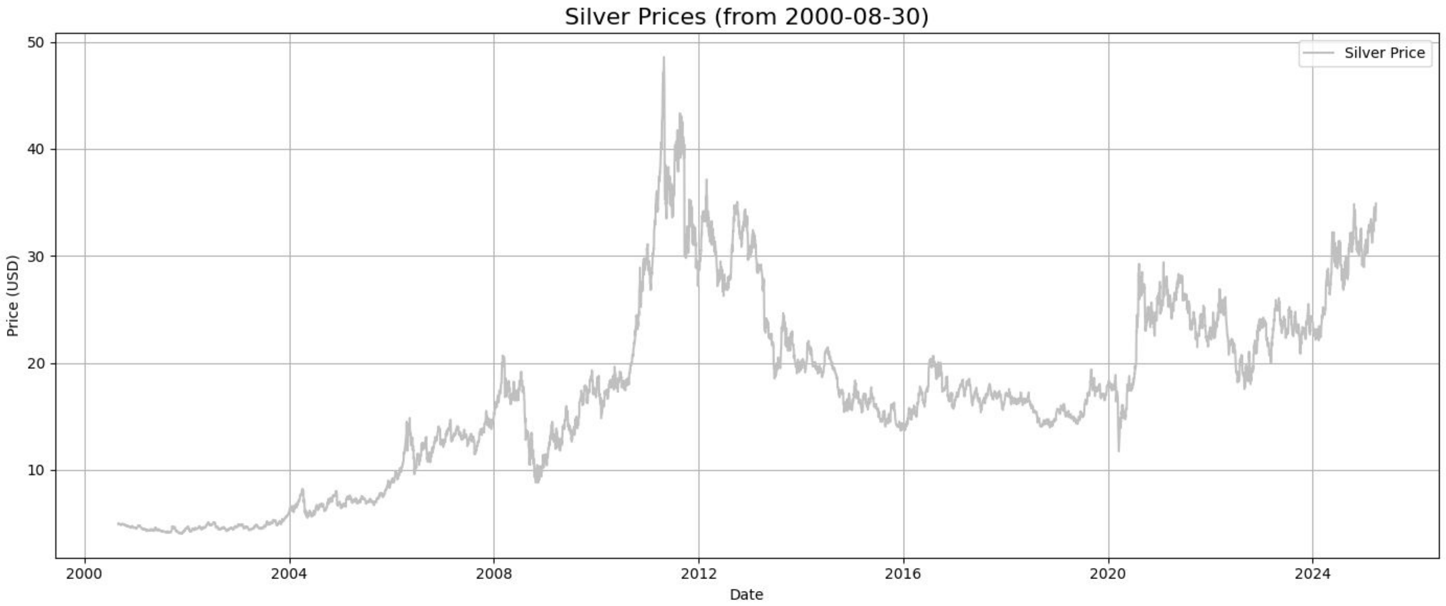
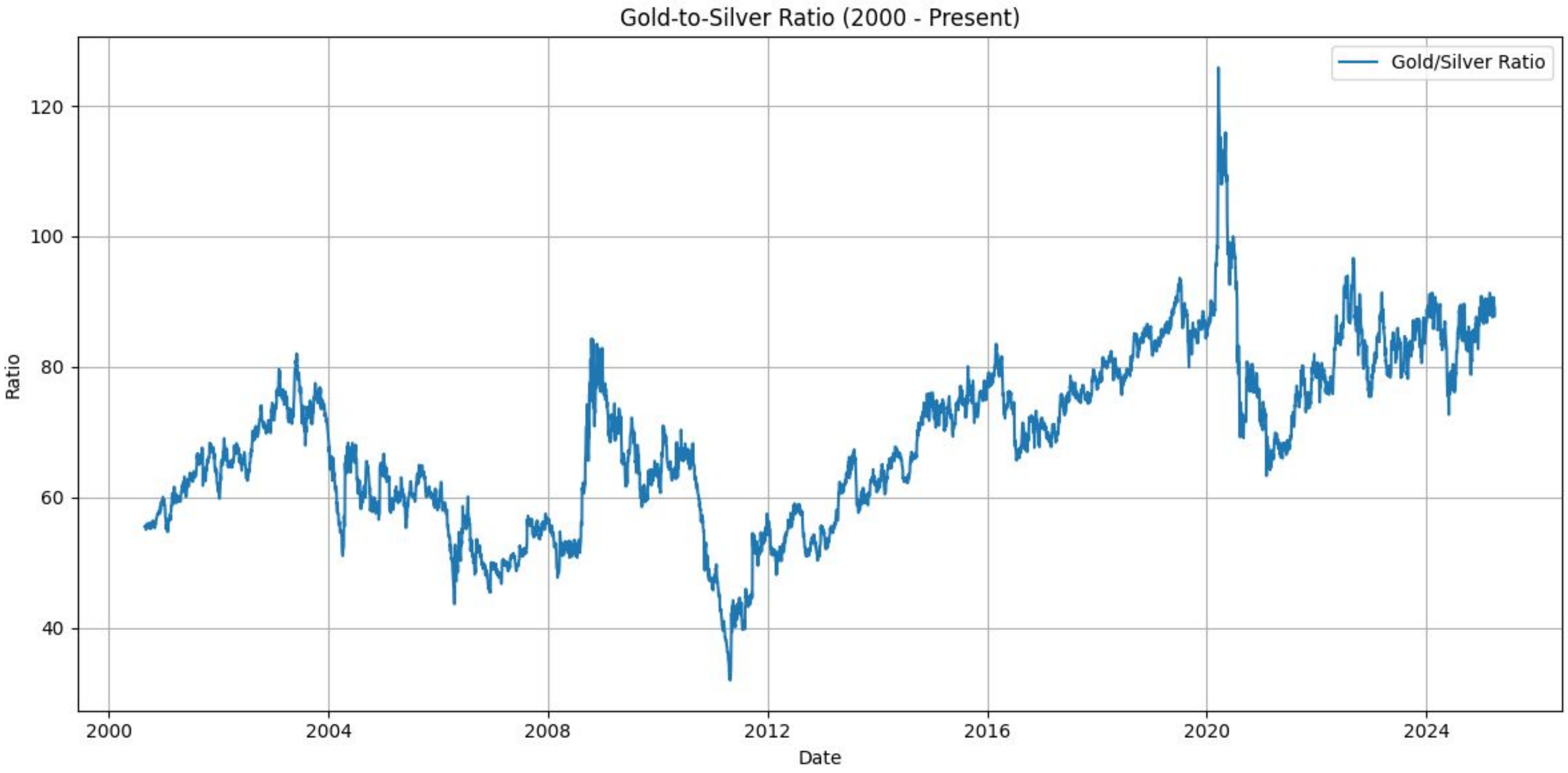
Trading on Gold and Silver

By Yichi Xue
Xin Qin
Shuyi Yang
Avi Jaiswal


Gold Prices (from 2000-08-30)
Gold Price
Price (USD)
Date
3000
2500
2000
1500
1000
500
2000
2004
2008
2012
2016
2020
2024


Silver Prices (from 2000-08-30)
Silver Price
Price (USD)
Date
50
40
30
20
10
2000
2004
2008
2012
2016
2020
2024


Gold & Silver Prices (from 2000-08-30)
Gold Price
Silver Price
Price (USD)
0
500
1000
1500
2000
2500
3000
2000
2004
2008
2012
2016
2020
2024
Date


Gold-to-Silver Ratio (2000 - Present)
Gold/Silver Ratio
120
100
80
60
40
Ratio
2000
2004
2008
2012
2016
2020
2024
Date

USC University of Southern California

Time Series: Arima

Training Set: 2010 - 2018

Testing Set: 2019

Machine learning

Train set: 2013–2018 **Test set: 2019**

Strategy: We trained an **ARIMA(3,1,2)** model on historical daily data from 2013 to 2018 and used it to generate rolling **one-business-day-ahead forecasts** for the year 2019.

Features: Gold_Silver_Ratio

Result Evaluation

Numeric Evaluation:

✅ High R^2 indicates that the model captures the underlying trend and short-term movements very well.

🔁 The use of internal differencing (d=1) likely helped stabilize the trend without oversmoothing.

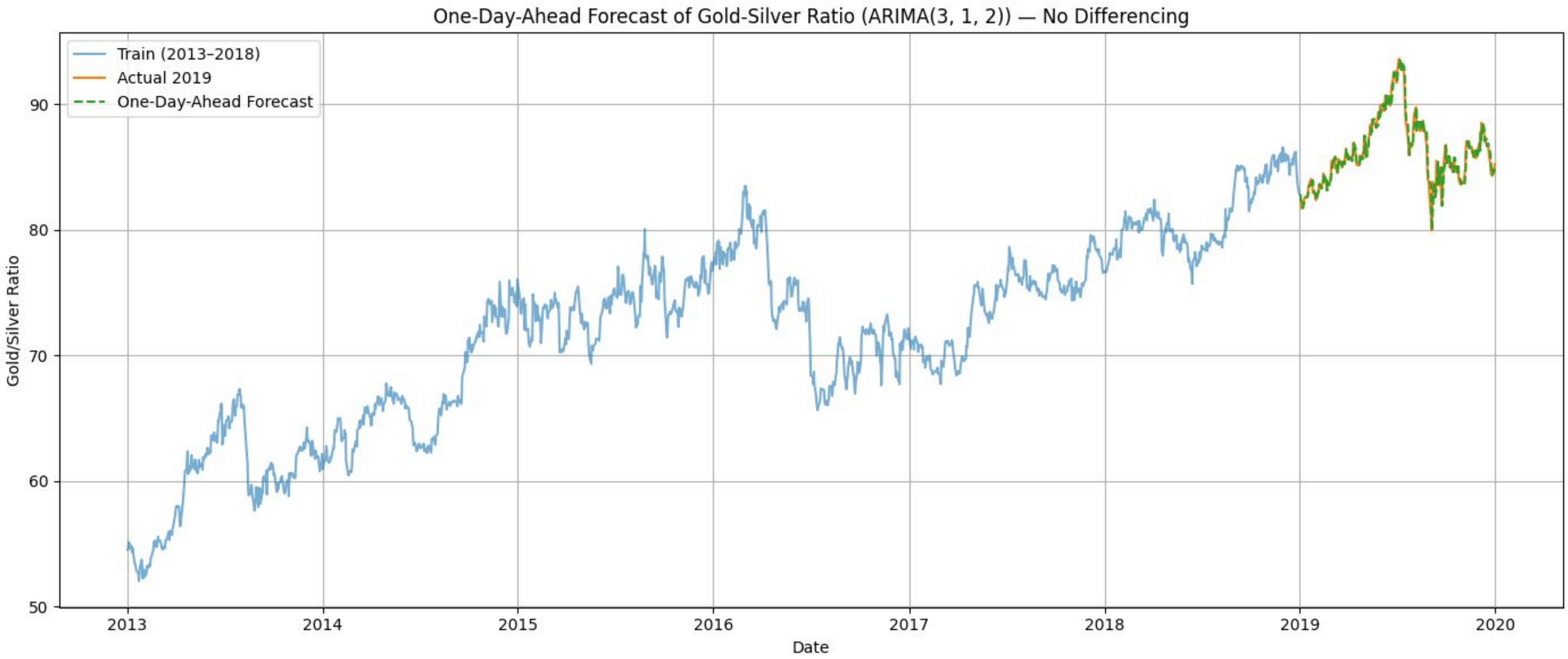
🧩 The (p=3, q=2) structure suggests moderate autoregressive and moving average components — enough to account for past values and shocks.

📉 MAE < 0.5 shows that prediction errors are low and likely within a reasonable range for trading or monitoring applications.

```
asilia@Mac Traditional Model Data % /usr/local/bin/python3 "/Users/asilia/Desktop
_ratio1.py"
🔍 Selecting best ARIMA(p,d,q) model on training data...
ARIMA(0,0,0) - AIC: 10354.62
ARIMA(0,0,1) - AIC: 8481.91
ARIMA(0,0,2) - AIC: 7115.88
ARIMA(0,0,3) - AIC: 6228.43
ARIMA(0,1,0) - AIC: 3137.91
ARIMA(0,1,1) - AIC: 3124.11
ARIMA(0,1,2) - AIC: 3124.86
ARIMA(0,1,3) - AIC: 3126.84
ARIMA(1,0,0) - AIC: 3147.83
ARIMA(1,0,1) - AIC: 3134.57
ARIMA(1,0,2) - AIC: 3135.19
ARIMA(1,0,3) - AIC: 3137.15
ARIMA(1,1,0) - AIC: 3123.41
ARIMA(1,1,1) - AIC: 3125.15
ARIMA(1,1,2) - AIC: 3126.86
ARIMA(1,1,3) - AIC: 3128.47
ARIMA(2,0,0) - AIC: 3133.85
ARIMA(2,0,1) - AIC: 3135.54
ARIMA(2,0,2) - AIC: 3137.20
ARIMA(2,0,3) - AIC: 3138.80
ARIMA(2,1,0) - AIC: 3125.05
ARIMA(2,1,1) - AIC: 3125.27
ARIMA(2,1,2) - AIC: 3124.02
ARIMA(2,1,3) - AIC: 3119.92
ARIMA(3,0,0) - AIC: 3135.41
ARIMA(3,0,1) - AIC: 3137.87
ARIMA(3,0,2) - AIC: 3139.53
ARIMA(3,0,3) - AIC: 3131.63
ARIMA(3,1,0) - AIC: 3126.69
ARIMA(3,1,1) - AIC: 3128.25
ARIMA(3,1,2) - AIC: 3119.80
ARIMA(3,1,3) - AIC: 3133.16

✅ Best ARIMA model: ARIMA(3, 1, 2) with AIC: 3119.80

🔁 Rolling one-day-ahead forecast for 2019...

📊 Evaluation on 2019 One-Day-Ahead Forecast (No Manual Differencing):
MAE:  0.4784
RMSE: 0.6673
R²:   0.9386
asilia@Mac Traditional Model Data %
```

USC

USC University of Southern California



Limitation

- **Assumes Past Patterns Will Continue**
  ARIMA models rely heavily on **historical price behavior**. In financial markets, **structural breaks** (e.g., policy shifts, geopolitical events, central bank actions) can invalidate past trends quickly.

- **No External Information**
  Traditional ARIMA models are **univariate** — they only use the gold-silver ratio itself. They **ignore macroeconomic indicators**, interest rates, currency fluctuations, and other market drivers.

- **Limited in Capturing Nonlinear Dynamics**
  Financial markets often show **nonlinear** behaviors (e.g., panic sell-offs or short squeezes), which linear models like ARIMA cannot capture well.

- **Short-Term Focus**
  ARIMA is best for **short-horizon forecasting**. As the forecast horizon extends, prediction accuracy typically **deteriorates rapidly**.

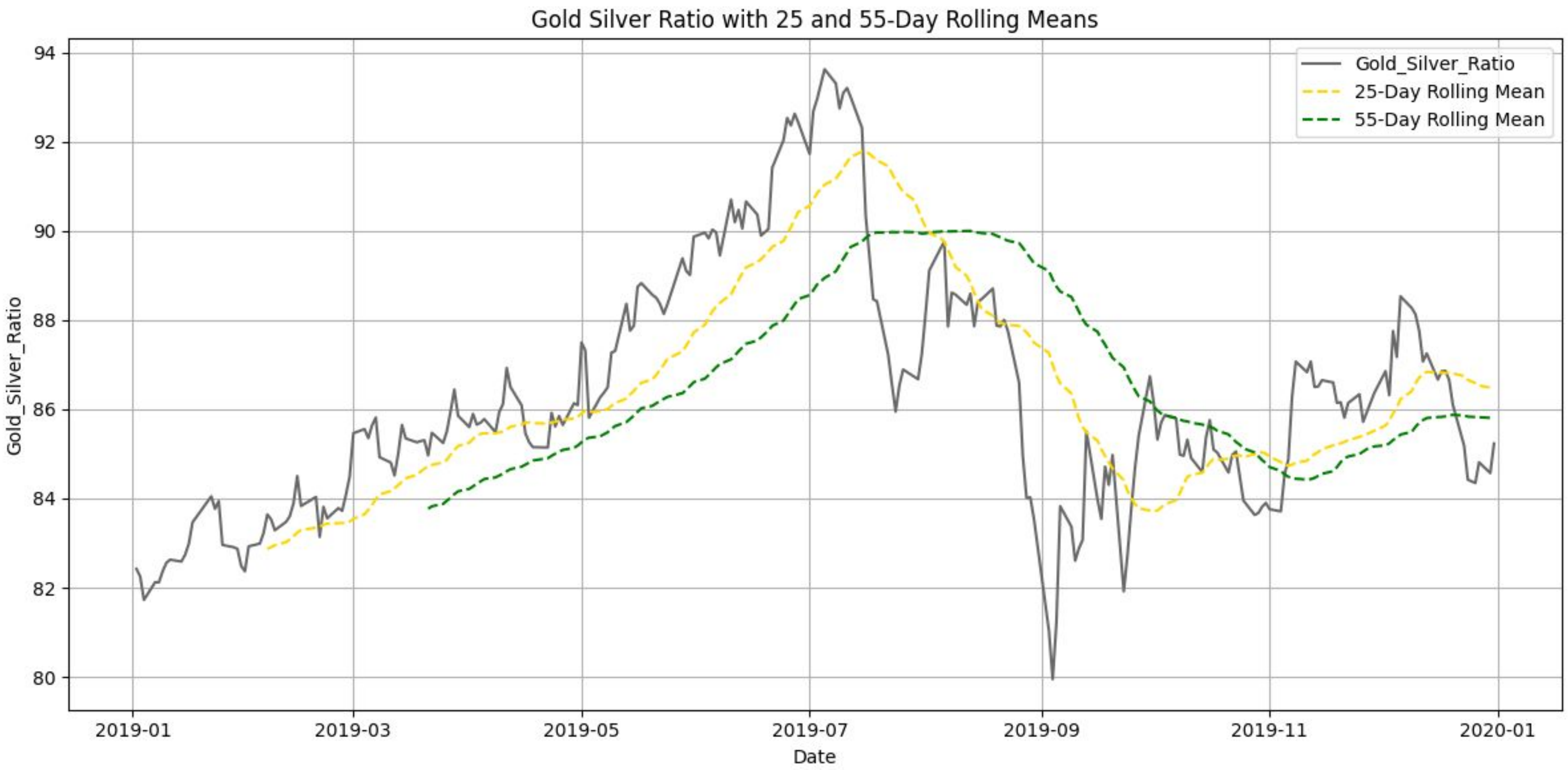
Moving average Line MA

Data: The Gold/Silver Ratio in 2019(From gold price, silver price and gold/silver ratio from Yahoo)

Strategy: Dual moving average crossover strategy. If Short MA > Long MA, it is a buy signal. If Short MA < Long MA, it is a sell signal. Hold if no crossover. Here the simple MA is used, which is the arithmetic mean of the Gold/Silver Ratio in past N days.

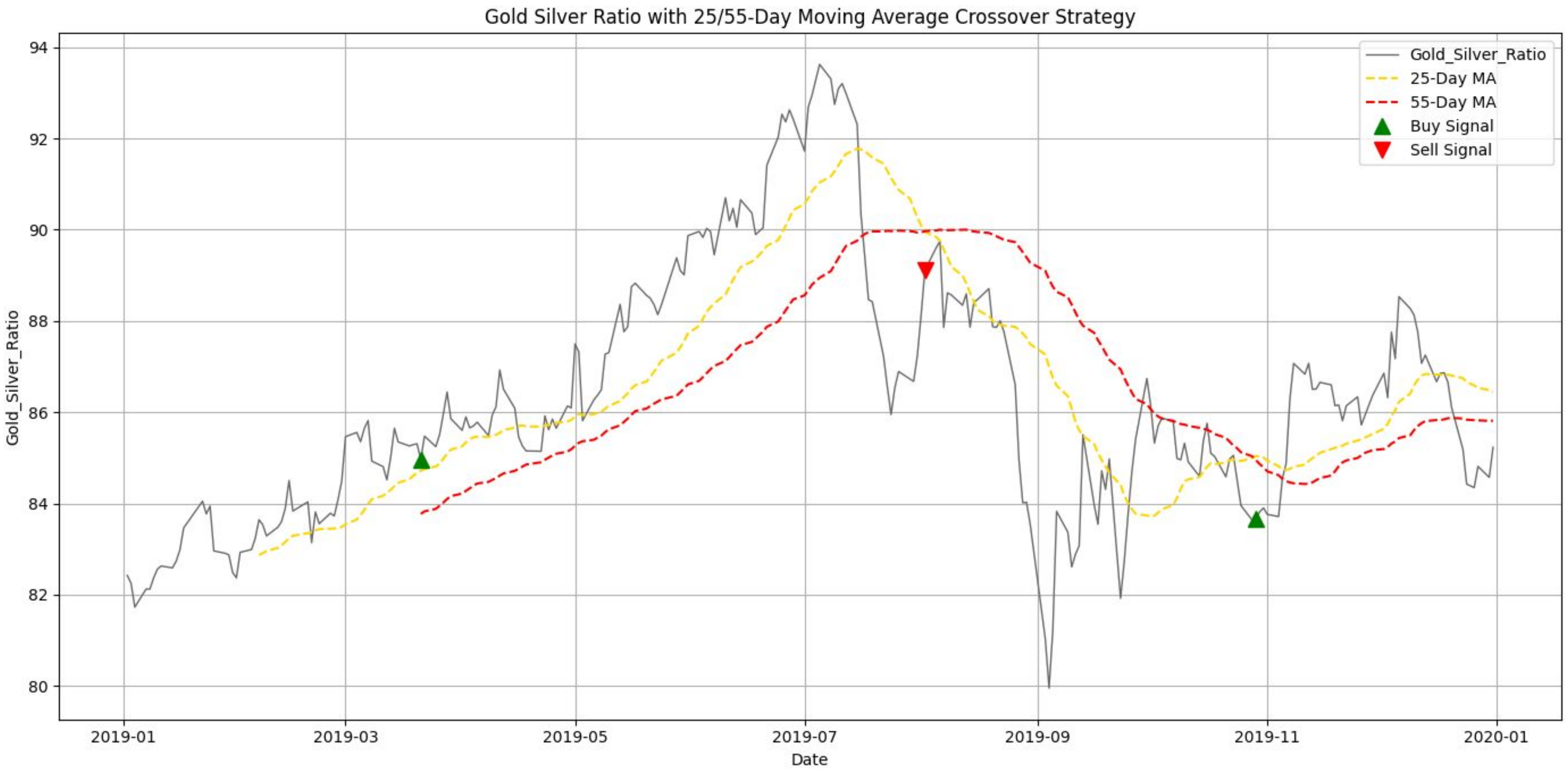
Parameter optimization: For rolling window, the short window is chosen from [5, 10, 15, 20, 25] and long window is chosen from [30, 35, 40, 45, 50, 55, 60]. The pair with highest cumulative return wins.

Two Moving average line



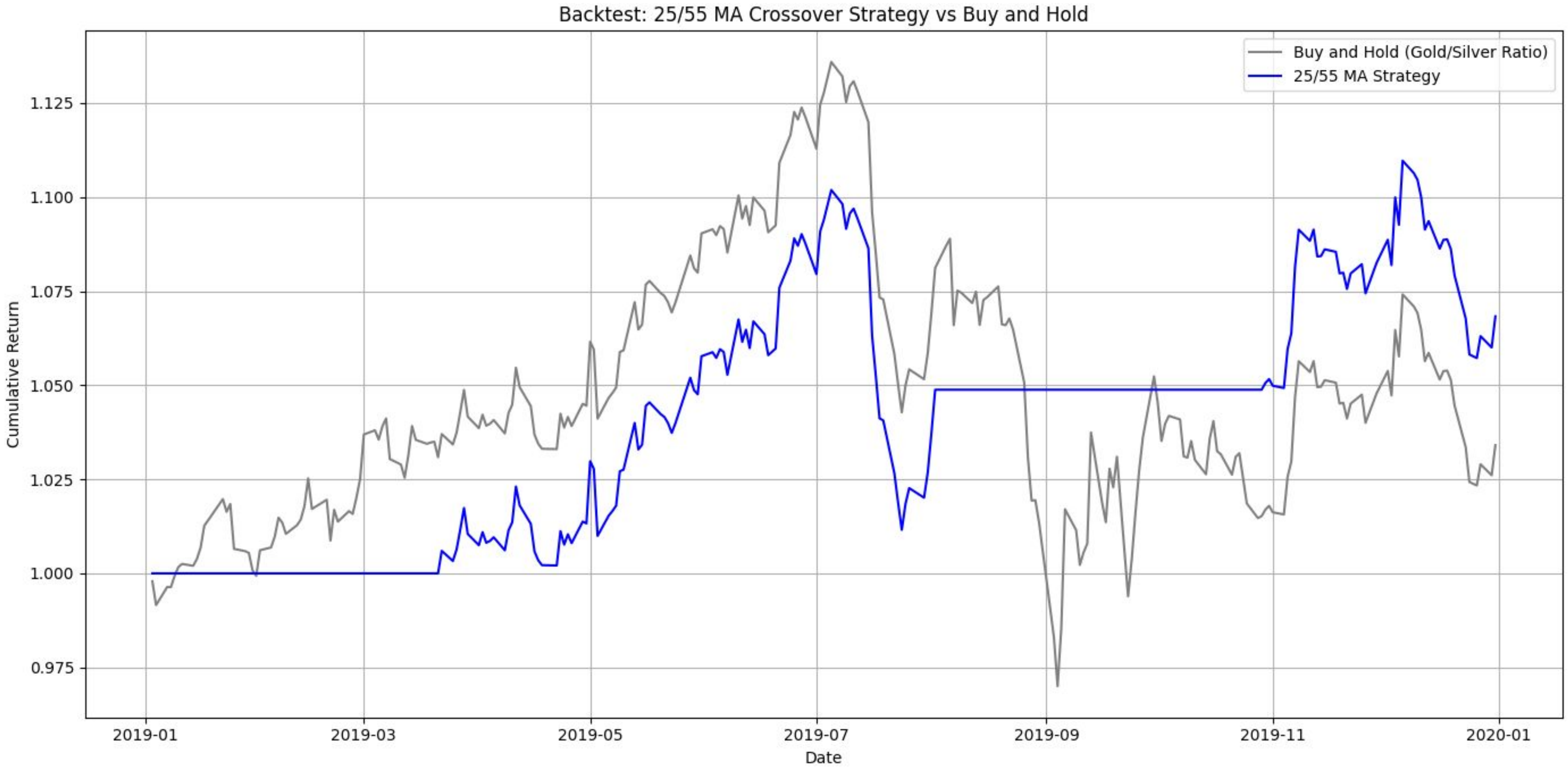
USC

Trading Signal



USC

Backtesting



Limitation

Gold-Silver Ratio Characteristics: Strong Mean Reversion, Weak Trends

- **Dual moving average strategy is trend-following and it can perform better when the assets show strong, sustained trends.**

High volatility and noise generate frequent false signals

- **The Gold/Silver Ratio can be very volatile, with sharp daily moves**
- **Short-term noise frequently triggers moving average crossovers without any real trend change.**
- **This leads to over-trading and poor returns**

Machine learning

Train set: 2013–2018   Test set: 2019

Strategy: Suppose I have 10000 dollars and use them buy gold at the beginning of the year. Use a machine learning algorithm to predict whether the price would go up or down in the next day. If the ratio is predicted to be rising, then we hold the gold. If not, we sell all the gold and buy silver. Vice versa.

Features: ratio_1d, ratio_2d, ma_diff, vol_5d

Logistic regression

```
Test accuracy (2019): 54.76%
Classification Report:
              precision    recall  f1-score   support

           0       0.57      0.32      0.41       123
           1       0.54      0.77      0.63       129

    accuracy                           0.55       252
   macro avg       0.55      0.54      0.52       252
weighted avg       0.55      0.55      0.52       252

Final portfolio value at end of 2019: $12,846.45
Total return: 28.46%
```

Precision:

Ratio of predictions come true

Recall:

The ratio of labels successfully caught

F1-score:

Harmonic average of precision and recall

Better at catching rising up

USC

Random Forest

```
Random Forest Test accuracy (2019): 54.76%
Classification Report:
              precision    recall  f1-score   support

           0       0.54      0.51      0.53       123
           1       0.56      0.58      0.57       129

    accuracy                           0.55       252
   macro avg       0.55      0.55      0.55       252
weighted avg       0.55      0.55      0.55       252

Final portfolio value at end of 2019: $13,238.00
Total return: 32.38%
```

Over-trading and missing opportunities

XGBoost

```
Model used: XGBoost
Test accuracy (2019): 55.95%
Classification Report:
              precision    recall  f1-score   support

           0       0.55      0.56      0.55       123
           1       0.57      0.56      0.56       129

    accuracy                           0.56       252
   macro avg       0.56      0.56      0.56       252
weighted avg       0.56      0.56      0.56       252

Final portfolio value at end of 2019: $13,193.86
Total return: 31.94%
```

Switch Count

	Switch Count
Logistic Regression	59
Random Forest	129
XGBoost	125

Performance Metrics

```
def compute_metrics(returns, cost_per_switch=0.001, total_switches=None):
    """
    returns: pd.Series of daily P&L returns
    cost_per_switch: transaction cost per switch (decimal, e.g. 0.001 = 10 bps)
    total_switches: if provided, override the calculated switch count
    """
    returns = returns.dropna()
    n = len(returns)

    # 1) Gross Sharpe
    ann_mean = returns.mean() * 252
    ann_vol  = returns.std() * np.sqrt(252)
    sharpe   = ann_mean / ann_vol

    # 2) Max drawdown
    cum = (1 + returns).cumprod()
    dd  = (cum / cum.cummax() - 1).min()
```

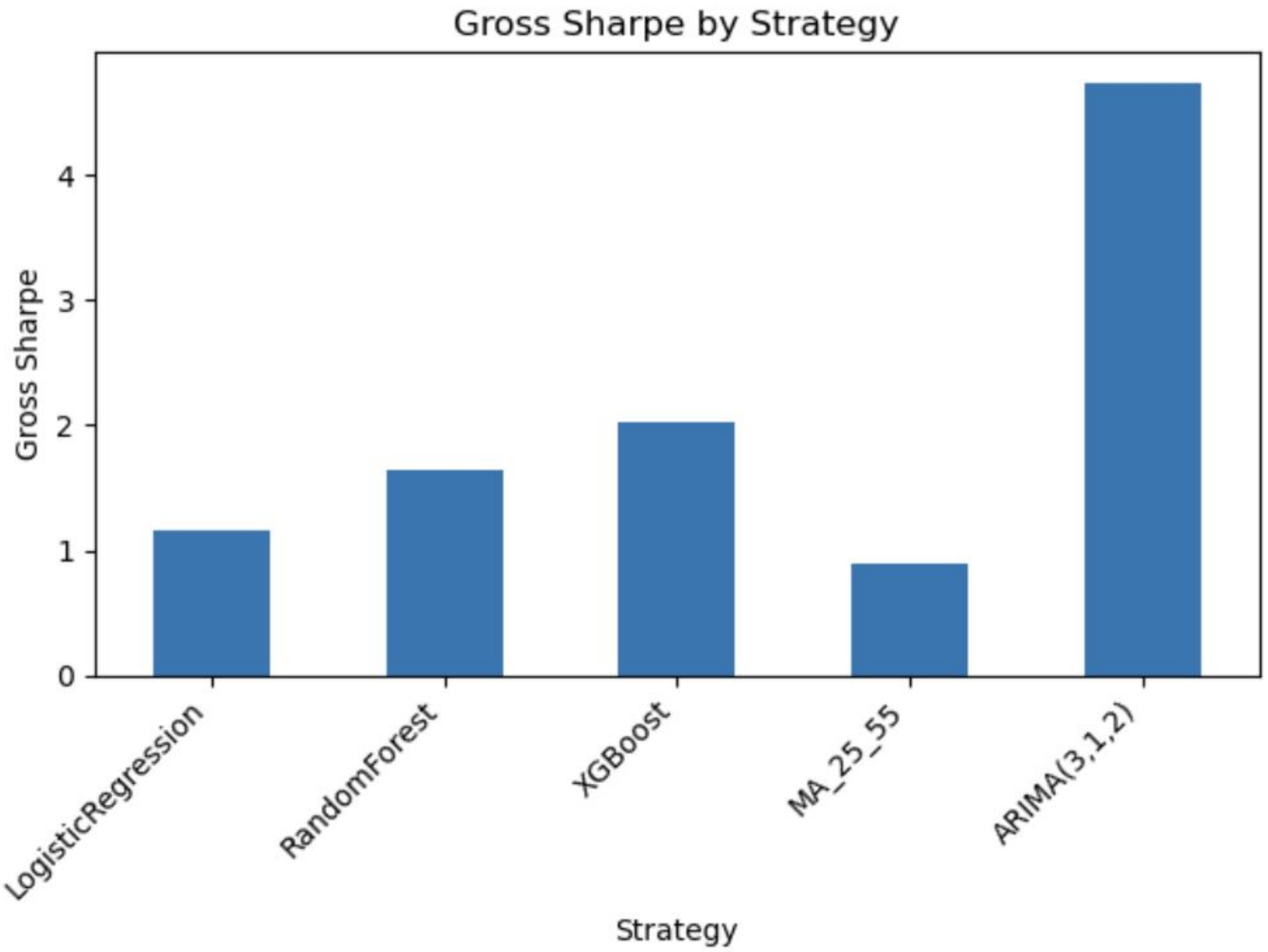
```
    # spread total cost evenly across days
    daily_cost   = (switches * cost_per_switch) / n
    net_returns  = returns - daily_cost
    net_mean     = net_returns.mean() * 252
    net_vol      = net_returns.std() * np.sqrt(252)
    net_sharpe   = net_mean / net_vol

    return sharpe, dd, net_sharpe
```

	Gross Sharpe	Max DD	Net Sharpe
LogisticRegression	1.158896	-0.148201	0.581797
RandomForest	1.641761	-0.040679	0.204154
XGBoost	2.011953	-0.031692	0.246710
MA_25_55	0.893432	-0.081988	0.854619
ARIMA(3,1,2)	4.736975	-0.018006	3.259791

```
switch_counts = {
    'LogisticRegression': 59,
    'RandomForest':       129,
    'XGBoost':            125,
    'MA_25_55':             3,
    'ARIMA(3,1,2)':       126
}
```

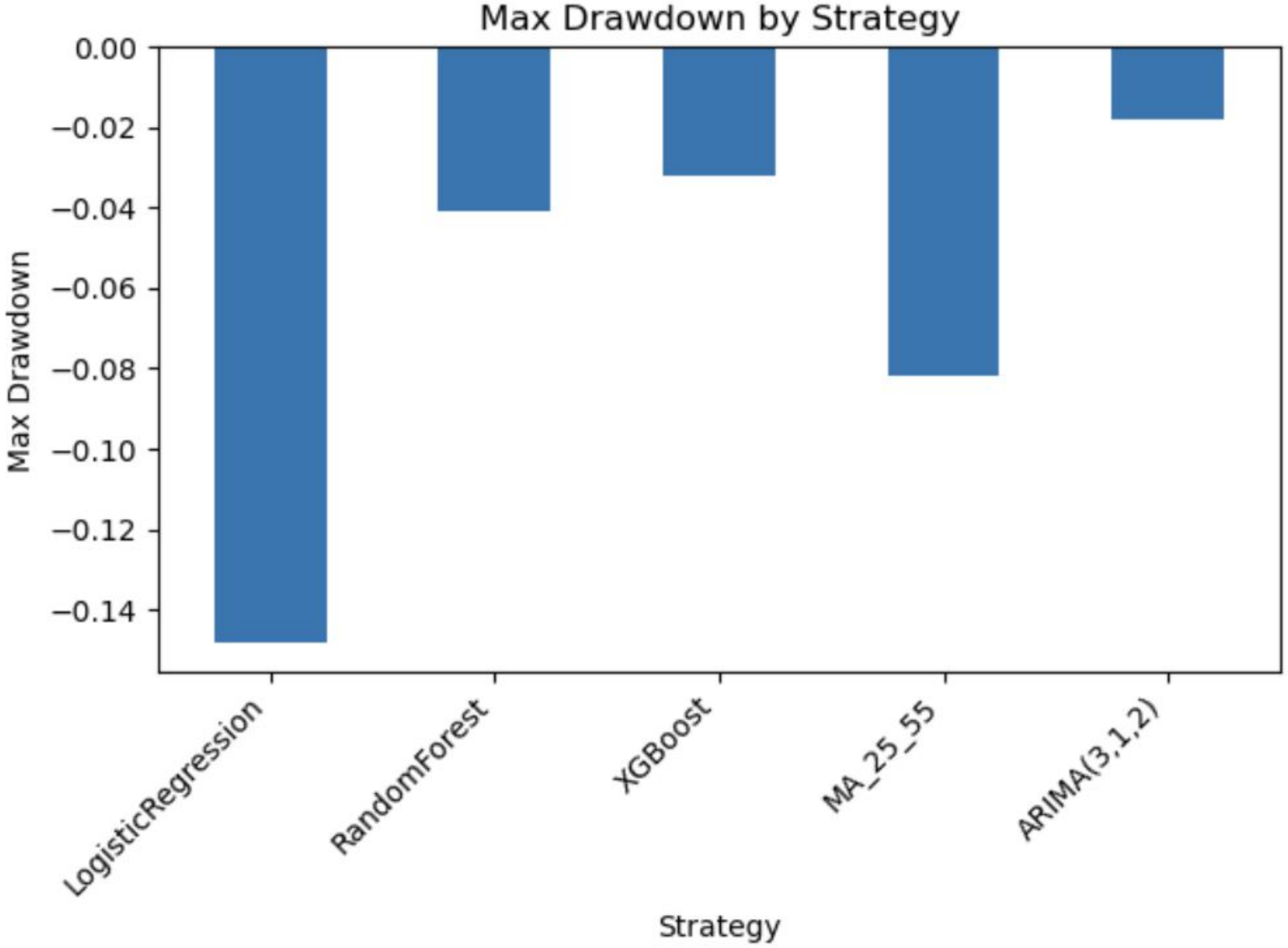
	Gross Sharpe	Max DD	Net Sharpe
LogisticRegression	1.158896	-0.148201	0.581797
RandomForest	1.641761	-0.040679	0.204154
XGBoost	2.011953	-0.031692	0.246710
MA_25_55	0.893432	-0.081988	0.854619
ARIMA(3,1,2)	4.736975	-0.018006	3.259791

Gross Sharpe by Strategy Graph



Max Drawdown by Strategy Graph

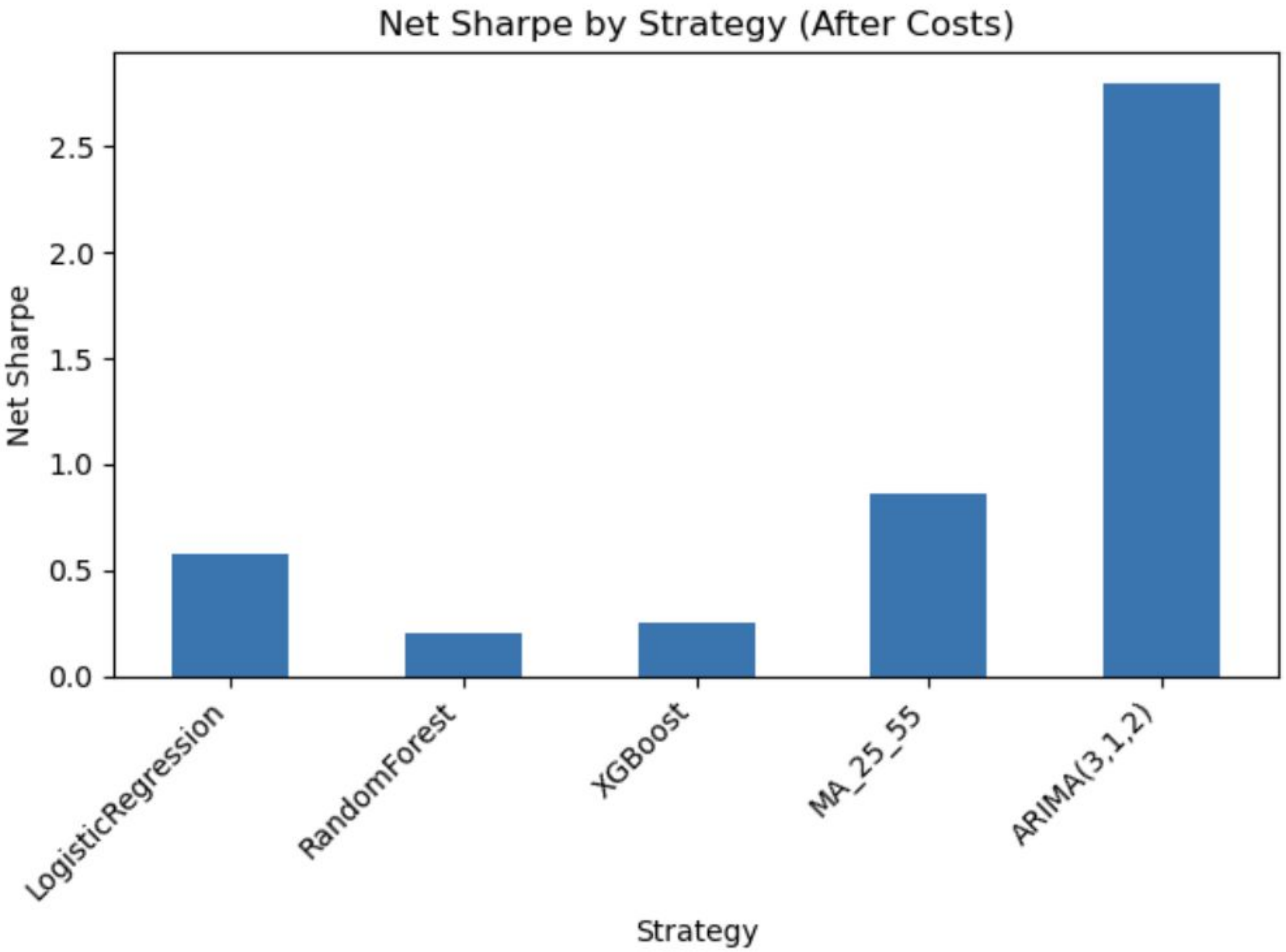
	Gross Sharpe	Max DD	Net Sharpe
LogisticRegression	1.158896	-0.148201	0.581797
RandomForest	1.641761	-0.040679	0.204154
XGBoost	2.011953	-0.031692	0.246710
MA_25_55	0.893432	-0.081988	0.854619
ARIMA(3,1,2)	4.736975	-0.018006	3.259791



USC

Net Sharpe by Strategy Graph

	Gross Sharpe	Max DD	Net Sharpe
LogisticRegression	1.158896	-0.148201	0.581797
RandomForest	1.641761	-0.040679	0.204154
XGBoost	2.011953	-0.031692	0.246710
MA_25_55	0.893432	-0.081988	0.854619
ARIMA(3,1,2)	4.736975	-0.018006	3.259791

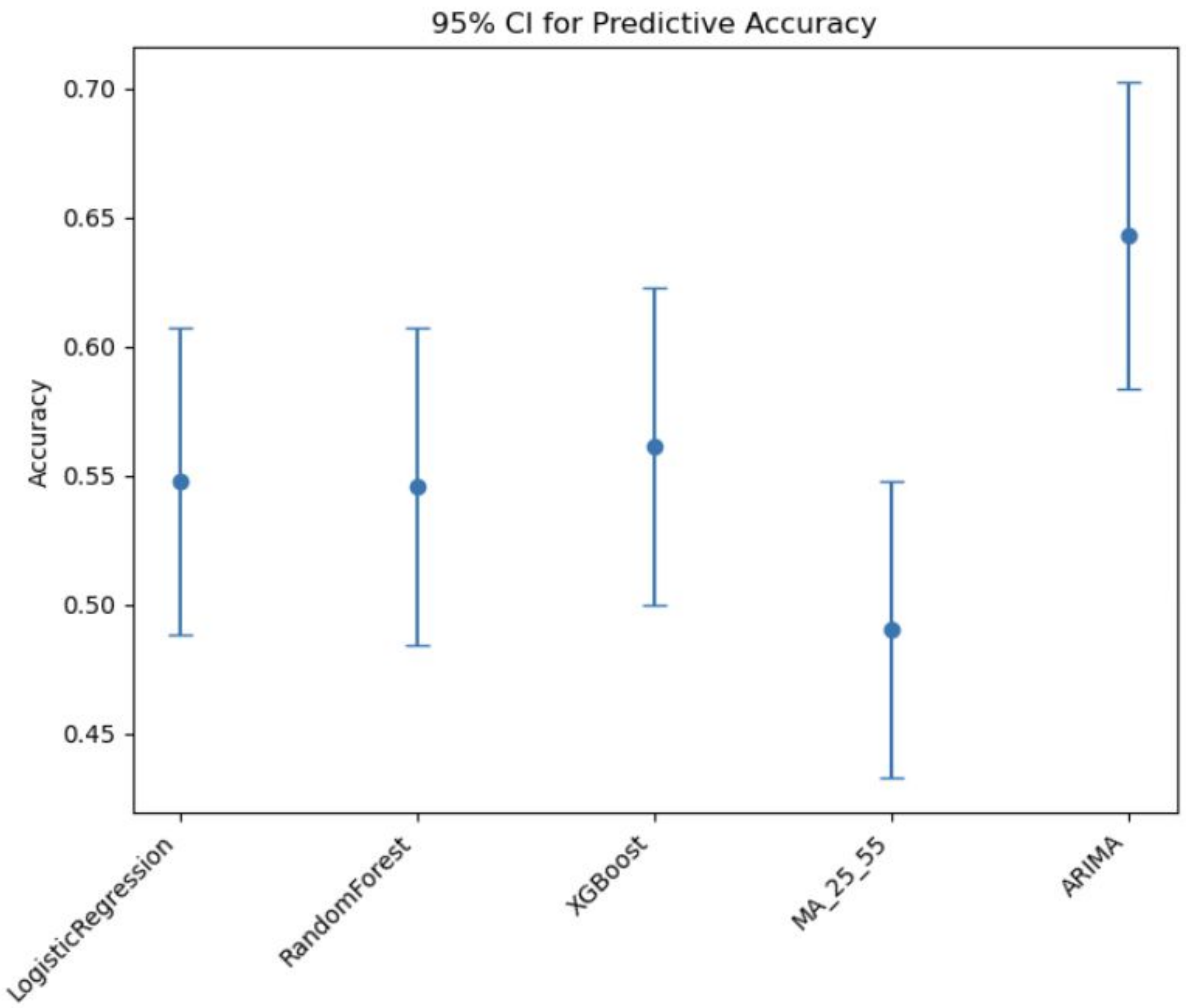


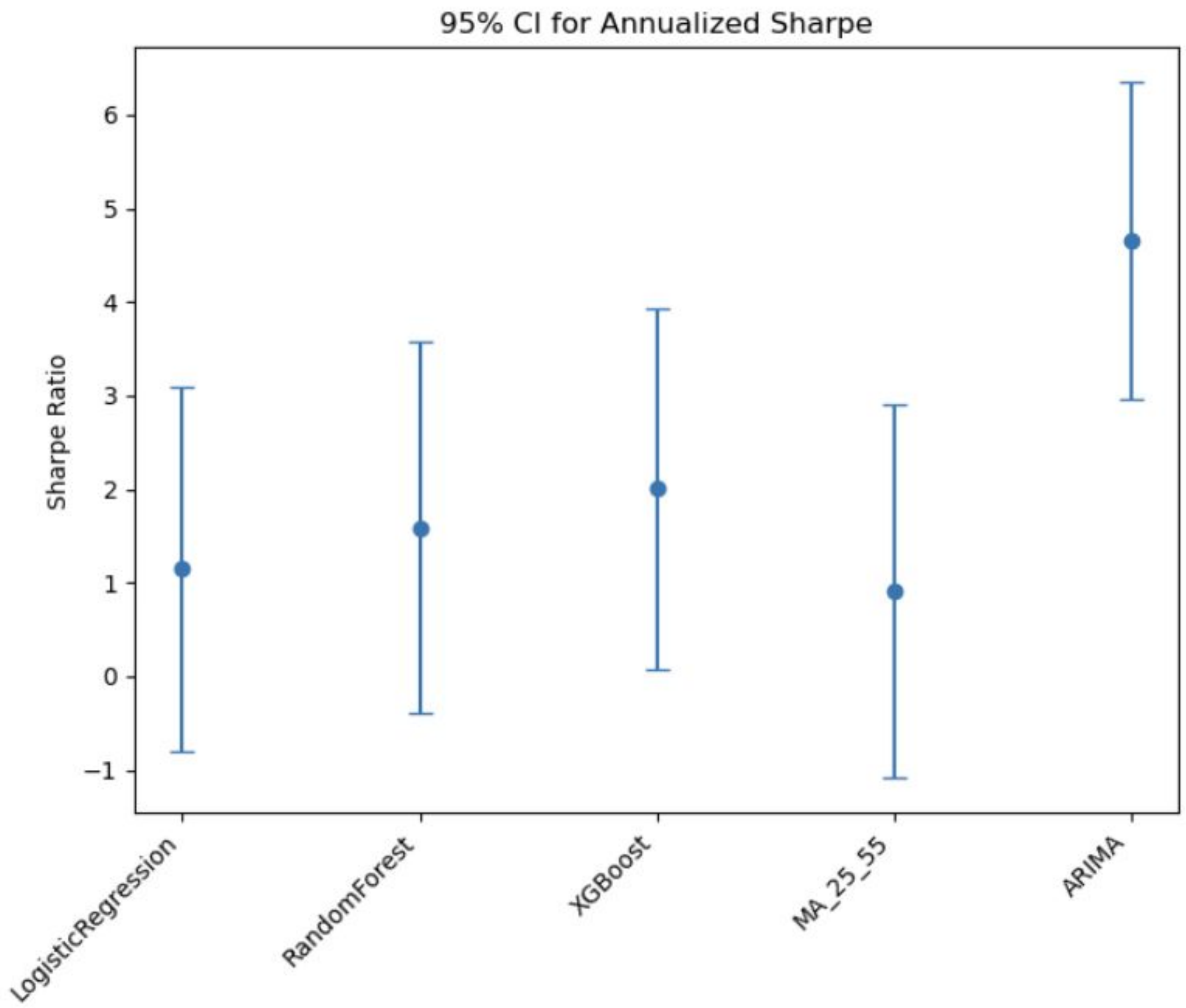
Robustness

Bootstrap 95% Confidence Intervals:

Strategy	Acc 2.5%	Acc 97.5%	Sharpe 2.5%	Sharpe 97.5%
LogisticRegression	0.484127	0.611111	-0.816609	3.238200
RandomForest	0.491964	0.607143	-0.339538	3.394605
XGBoost	0.496032	0.615079	0.084126	3.866194
MA_25_55	0.432540	0.555556	-1.166603	2.764800
ARIMA	0.583333	0.698413	2.949326	6.299024

Confidence Intervals Graphs





Statistical Significance

```
=== Friedman Test ===
                     Chi² statistic   p-value
Friedman Test              3.613977  0.460761


=== Wilcoxon Pairwise Comparisons vs ARIMA ===
                         Wilcoxon W   p-value
Comparison
ARIMA vs Logistic            3675.0  0.136591
ARIMA vs RandomForest        4272.0  0.328816
ARIMA vs XGBoost             3676.0  0.105357
ARIMA vs MA_25_55            3187.0  0.114098
```

- **Friedman Null**: “All 5 strategies have the same distribution of daily returns.”

Interpretation: With p >> 0.5, null hypothesis cannot be rejected.

- **Wilcoxon Pairwise Null:** “The paired daily returns of ARIMA and the other strategy have the same distribution.

Interpretation: With all values of p > 0.5, the null hypothesis cannot be rejected in any case.

Diebold-Mariano Table

	Model 1	Model 2	DM stat	p-value
0	Logistic	RandomForest	6.026673	1.673689e-09
1	Logistic	XGBoost	6.771107	1.278000e-11
2	Logistic	MA_25_55	5.071848	3.939713e-07
3	Logistic	ARIMA(3,1,2)	8.407509	0.000000e+00
4	RandomForest	XGBoost	-1.739399	8.196464e-02
5	RandomForest	MA_25_55	-0.165247	8.687496e-01
6	RandomForest	ARIMA(3,1,2)	-0.714991	4.746144e-01
7	XGBoost	MA_25_55	1.185203	2.359372e-01
8	XGBoost	ARIMA(3,1,2)	0.805123	4.207485e-01
9	MA_25_55	ARIMA(3,1,2)	-0.492183	6.225900e-01

Aside from Logistic Regression (inferior), none of the other four strategies show a statistically significant edge over each other in day-to-day profitability at the 5% level.

Thank You!