HAiV

Stock movement as a Markov Process using Machine Learning predictor

Business Lab for Financial Engineering

Nguyen Minh Duc

20202026

Contents

- Trading Strategy
- Stock Movement as Markov Process
- Experiment
- Deploying on Interactive Brokers

Trading Strategy

Just buy low and sell high, simple!

How?

Trading Strategy

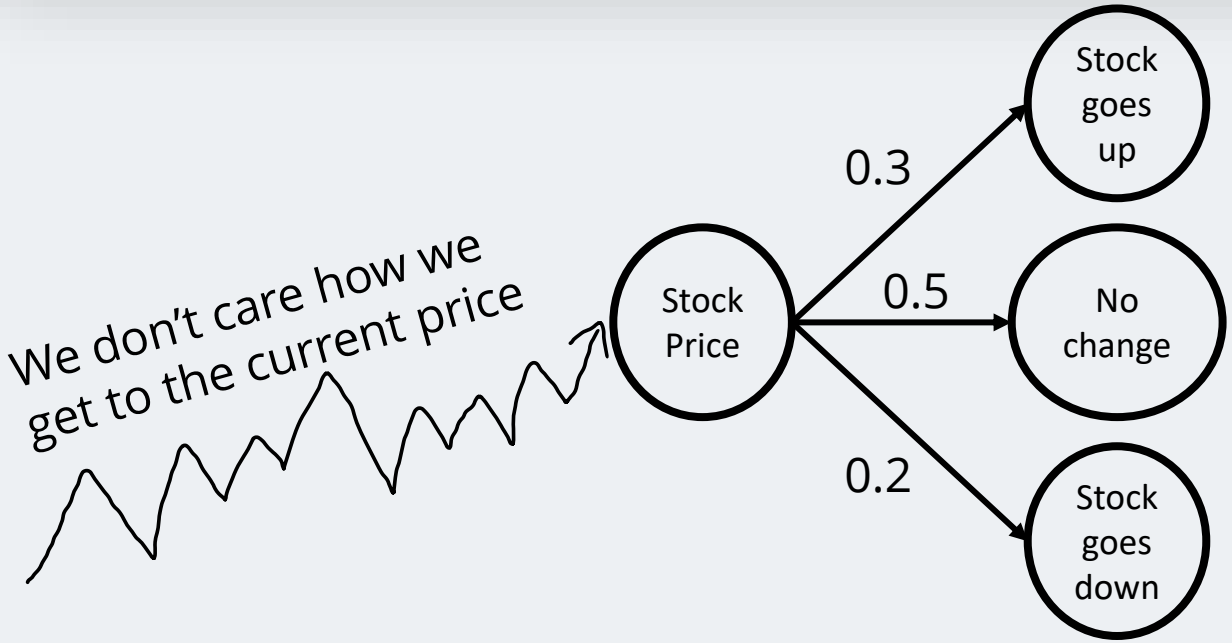
Just buy low and sell high, simple!

Then, how to predict the future?

Stock Movement as a Markov Process

1 What is a Markov process?

Given the historical states of a random system $X_1, X_2, \ldots, X_t$, the probability of moving to the next state depends only the current state, i.e., $P(X_{t+1} = x | X_1, X_2, \ldots, X_t) = P(X_{t+1} = x | X_t)$



Stock Movement as a Markov Process

2 Stock Movement as a Markov process

We consider the daily return of the stock: $\mathrm{r_t} = \frac{P_t}{P_{t-1}} - 1.$

Classify each $\mathrm{r_t}$ as

- High Increase (HI): $\mathbf{r_t} > \boldsymbol{Q_{inc}}(\mathbf{75\%})$
- Moderate Increase (MI): $\boldsymbol{Q_{inc}}(\mathbf{50\%}) \leq \mathbf{r_t} < \boldsymbol{Q_{inc}}(\mathbf{75\%})$
- Slight Increase (SI): $\boldsymbol{Q_{inc}}(\mathbf{25\%}) \leq \mathbf{r_t} < \boldsymbol{Q_{inc}}(\mathbf{50\%})$
- Neutral (Ne): $\boldsymbol{Q_{dec}}(\mathbf{25\%}) \leq \mathbf{r_t} < \boldsymbol{Q_{inc}}(\mathbf{25\%})$
- Slight Decrease (SD): $\boldsymbol{Q_{dec}}(\mathbf{25\%}) \leq \boldsymbol{r_t} < \boldsymbol{Q_{dec}}(\mathbf{25\%})$
- Moderate Decrease (MD): $\boldsymbol{Q_{dec}}(\mathbf{25\%}) \leq \boldsymbol{r_t} < \boldsymbol{Q_{dec}}(\mathbf{25\%})$
- High Decrease (HD): $\boldsymbol{r_t} < \mathbf{Q_{dec}}(\mathbf{25\%})$

Stock Movement as a Markov Process

2 Stock Movement as a Markov process

We define a hyperparameter: *lookback* – The number of prior prices to consider
forward – The number of future days to predict

The sequence of prices can be encoded as a tuple based on the daily return
Example: (HI, SD Ne, HD, MI)

Now, we can use the historical prices to estimate the probability distribution of the next price state

$$P\left(X_{t+1} = \left(x_{t+1}, x_{t+2}, \ldots, x_{t+forward}\right) \mid X_t = \left(x_{t-lookback+1}, x_{t-lookback+2}, \ldots, x_t\right)\right)$$

Stock Movement as a Markov Process

2 Stock Movement as a Markov process

How can we approximate this probability with the sample stock data?

$$P\left(X_{t+1} = \left(x_{t+1}, x_{t+2}, \dots, x_{t+forward}\right) \,\middle|\, X_t = \left(x_{t-lookback+1}, x_{t-lookback+2}, \dots, x_t\right)\right)$$

Previous approach was to use a table to count the frequency of every possible combination of price movements

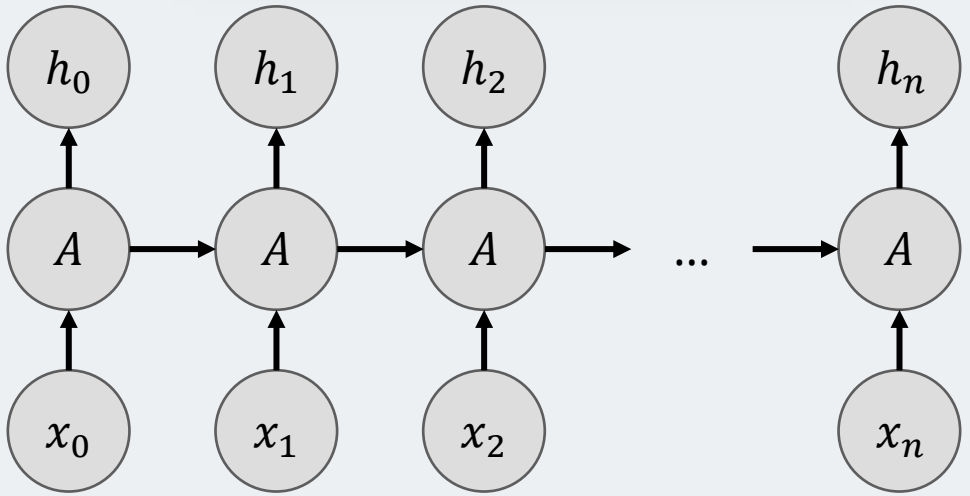
Memory complexity: $\boldsymbol{O}(\mathbf{7}^{\boldsymbol{lookback}})$
Impractical to scale the algorithm beyond 7 *lookbacks*

How to improve memory efficiency for longer sequences?

Stock Movement as a Markov Process

3 ML-based predictor

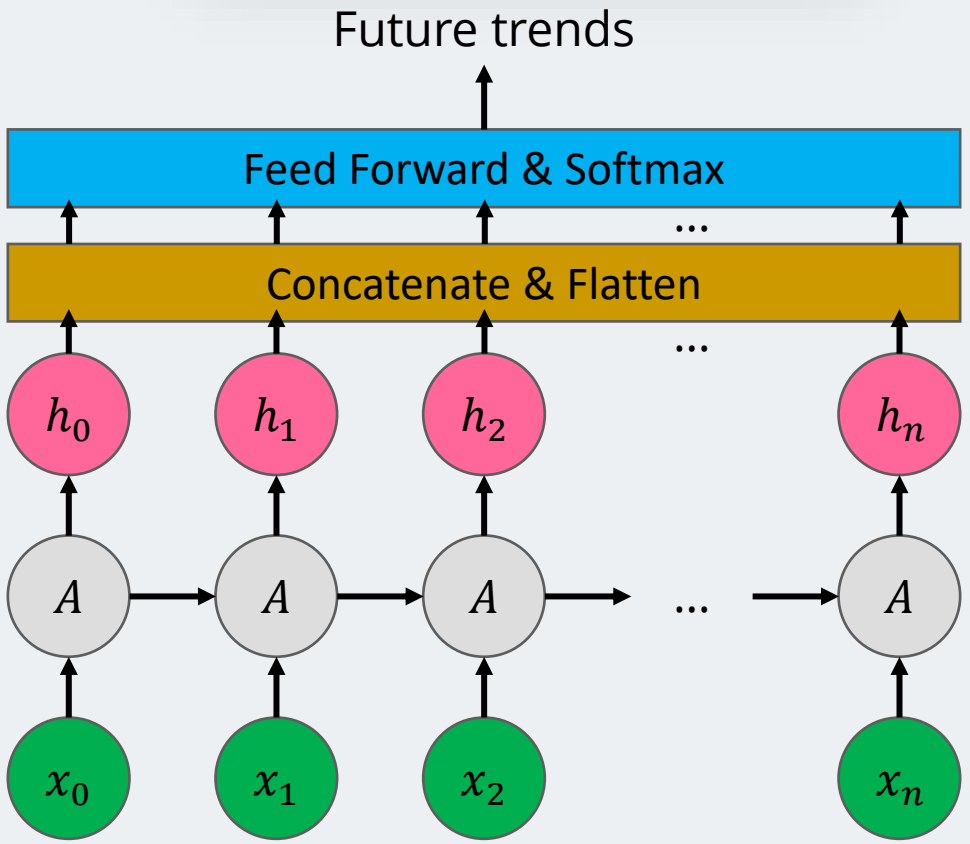
Recurrent Neural Network



Stock Movement as a Markov Process

3 ML-based predictor

Model Architecture



Stock Movement as a Markov Process

4 Use prediction in the strategy

$$P[X_{t+1} = (x_{t+1}, x_{t+2}, \dots, x_{t+forward}) \mid X_t = (x_{t-lookback+1}, x_{t-lookback+2}, \dots, x_t)]$$

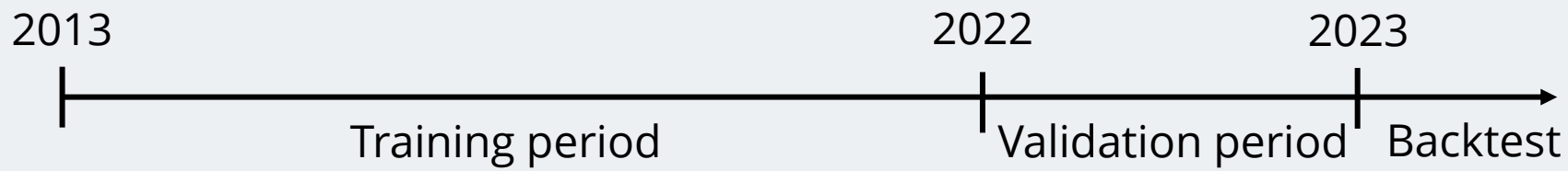
The prediction of future trend is the price state with the highest probability

$$\hat{X}_{t+1} = \text{argmax}_x \left(P[X_{t+1} = x \mid X_t = (x_{t-lookback+1}, x_{t-lookback+2}, \dots, x_t)]\right)$$

Experiment

1 Experimental setup

The historical price of stocks in S&P500 were collected
Total number of obtained stocks: 496



The stocks' prices are categorized into 7 categories

Each data point is then one-hot encoded

Experiment

2 Model hyperparameters

```
self.model = SimpleLSTM(
    lookback=lookback,
    hidden_size=16,
)
```

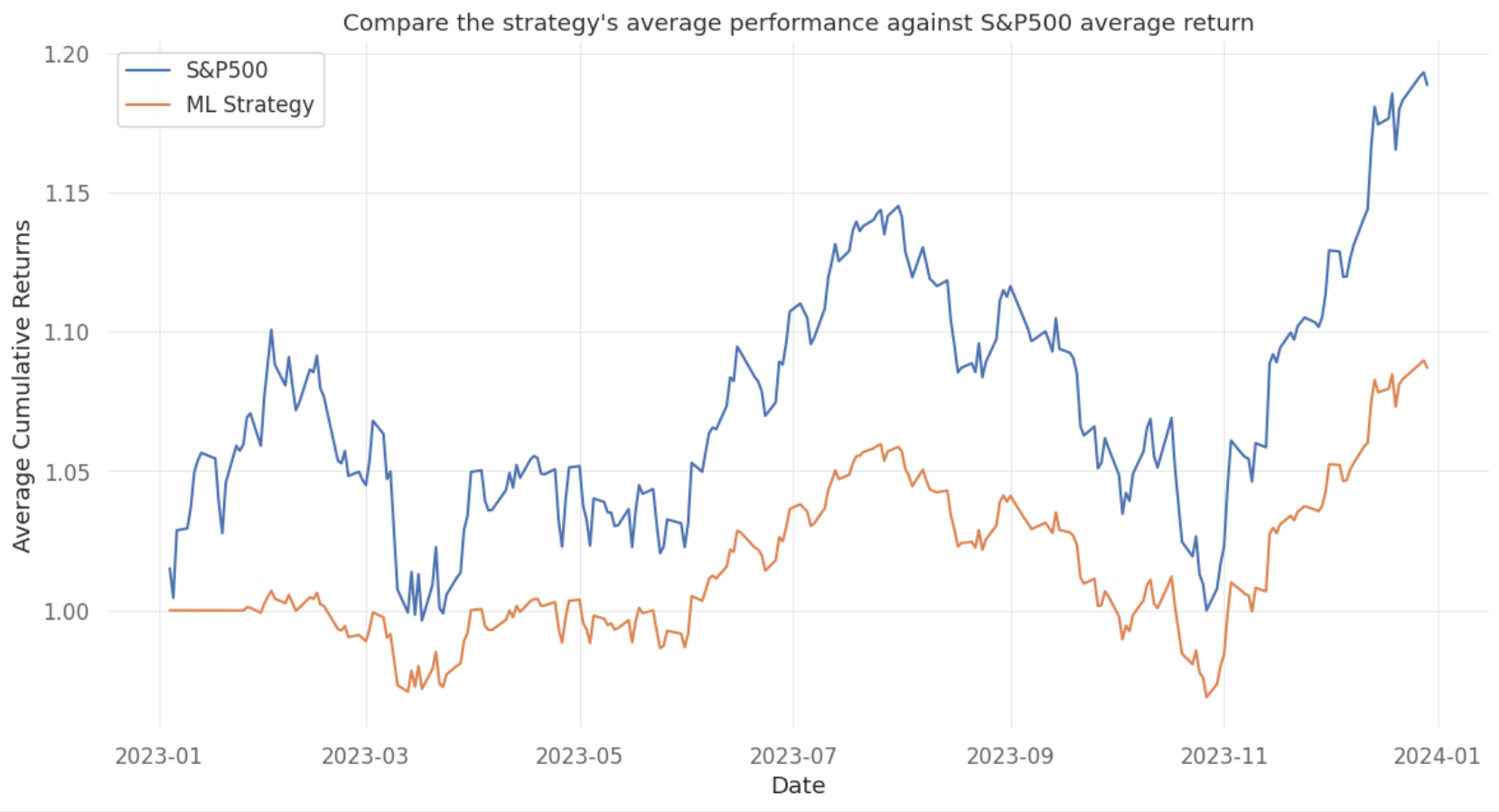
```
ml_strategy.train(
    epochs=100,
    lr=1e-2,
    batch_size=128,
)
```

$$Loss(y, \hat{y}) = -\sum_{x \in \{\text{up, side, down}\}} y_i \log(\hat{y}_i)$$

The model's parameters are optimized by *AdamW optimizer*

Experiment

3 Experimental results



S&P500 Performance

Monthly Expected Return: 0.010
Monthly Volatility: 0.045
Sharpe ratio: 0.222

Strategy Performance

Monthly Expected Return: 0.006
Monthly Volatility: 0.027
Sharpe ratio: 0.241

Can it beat the market?

The strategy beats **277** stocks
→Yes, but only 56% of the time

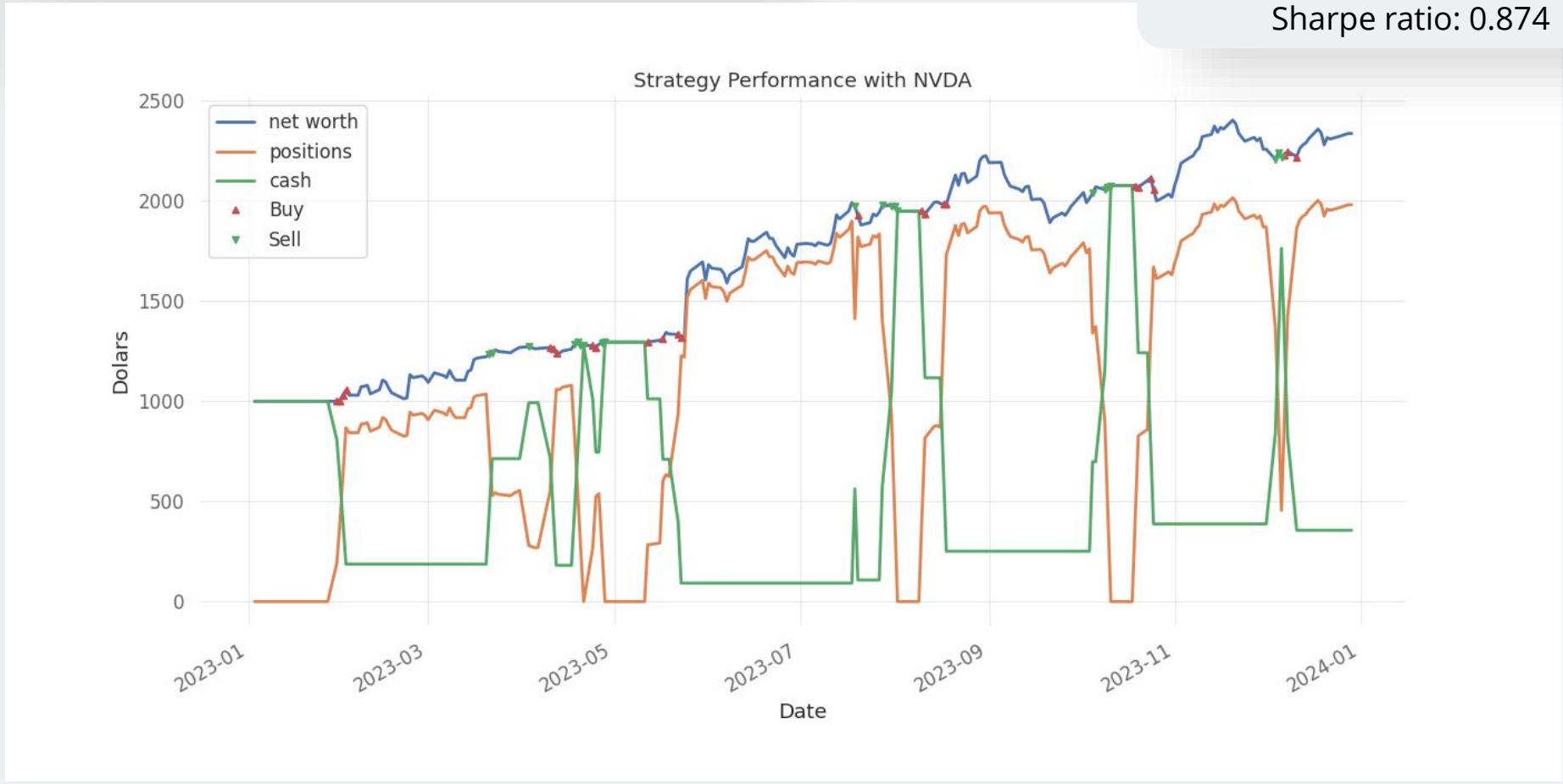
The strategy generates higher returns than 216 stocks
→Actually beat the market 43.55% of the time

Experiment

4

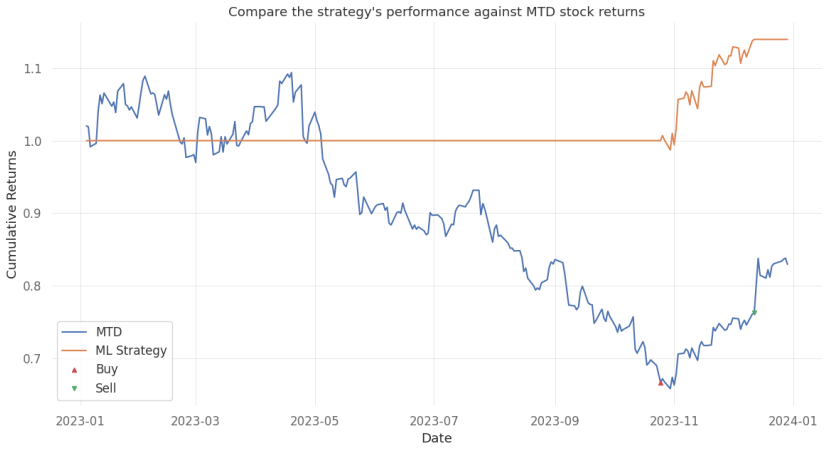
Case study - NVDA

Monthly Expected Return: 0.082
Monthly Volatility: 0.094
Sharpe ratio: 0.874



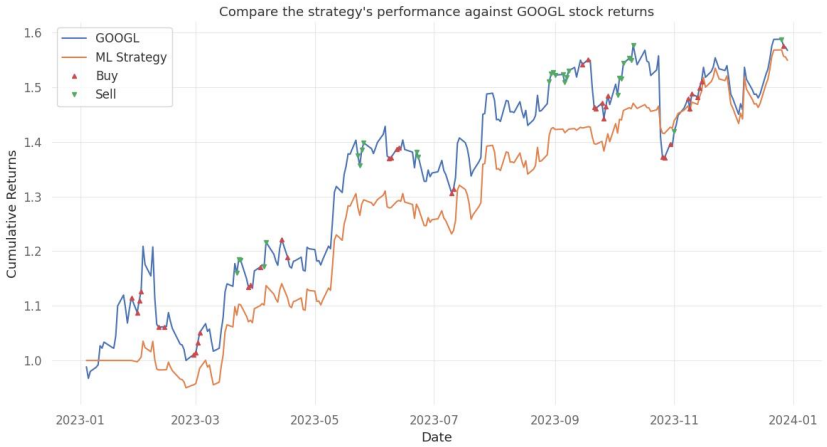
Experiment

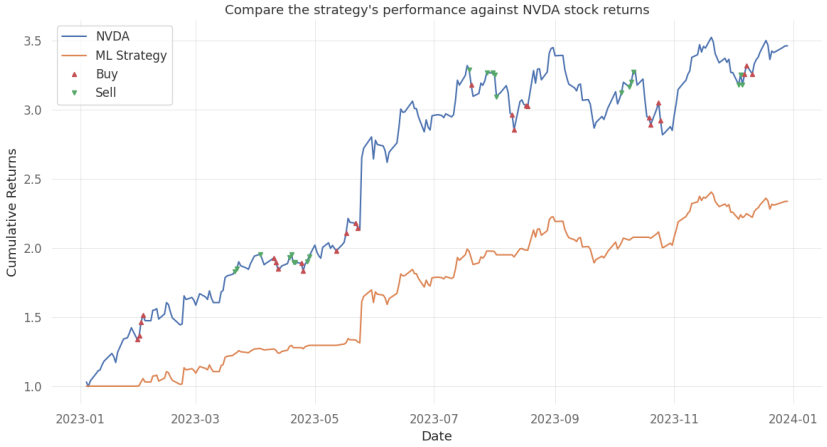
4 Case study – Bearish Markets



Experiment

4 Case study – Bullish Markets





Deploying on Interactive Brokers

1 Implementation Details

```python
def main(client_id: int, duration: str, freq: int, ticker: str) -> None:
    """
    Main function to train and deploy the models
    """
    print(f"Conecting to IB {client_id + 1}...")
    ib = IB()
    ib.connect('127.0.0.1', 7497, clientId=client_id + 1)
    print(f"Connected to IB {client_id + 1}!")
    train_model(ib, duration, freq, ticker)
    deploy_model(ib, duration, freq, ticker)


if __name__ == "__main__":
    print("=" * 30, "  Deploying models  ", "=" * 30)

    duration = "1 M"
    freq = 3
    fns = [main for _ in range(len(TICKERS))]
    fns_arg = [(idx, duration, freq, ticker) for idx, ticker in enumerate(TICKERS)]
    run_parallel(fns, fns_arg)

    print("=" * 30, "Finish the deployment", "=" * 30)
```

Create separate process for each stock

Deploying on Interactive Brokers

1 Implementation Details

```
class MLStrategy(MarkovStrategy):
    def deploy(self,
            while True:
                now = datetime.datetime.now()
                bars = ib.reqHistoricalData(stock,
                                            endDateTime = '',
                                            durationStr = '1 D',
                                            barSizeSetting = f"{freq} mins",
                                            whatToShow = 'MIDPOINT',
                                            useRTH = True,
                                            formatDate = 2)
                financial_data = util.df(bars)

                price = financial_data["close"]
                daily_return = price.pct_change()
                daily_return.iloc[0] = 1

                states = self.define_states(daily_return)

                cur_state = np.array(list(map(lambda x: STATE_TO_NUM[x], states[-self.lookback:])))
                cur_state = torch.tensor(one_hot(cur_state - STATE_TO_NUM["High Decrease"], len(STATE_TO_NUM)),
                                            dtype=torch.float32).unsqueeze(0)
                signal_pred = self.model(cur_state).argmax(dim=1).item() - 1

                prices.append(price.iloc[-1])
```

Get the real time data and let the model predict the movement

Deploying on Interactive Brokers

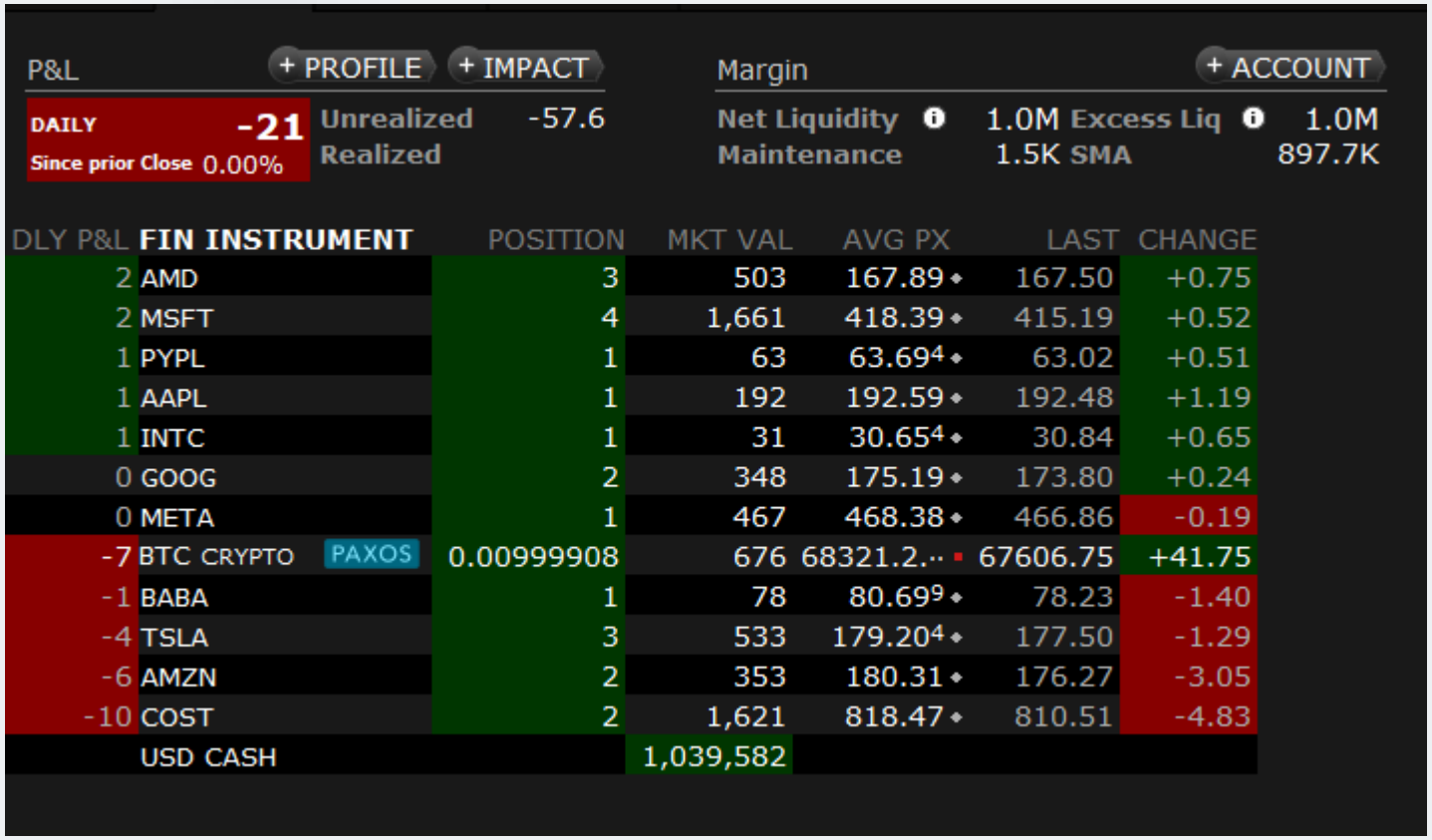
1 Implementation Details

```
if cur_cash - num_share_per_trade * price.iloc[-1] >= -limit_borrow and \
signal_pred == 1 and \
np.abs(price.iloc[-1] - last_trade) > commission:
    position += num_share_per_trade
    cur_cash -= num_share_per_trade * price.iloc[-1]
    cur_cash -= commission
    signal = 1
    last_trade = price.iloc[-1]
    print(f"{now} Buy", file=f)
    order = MarketOrder("Buy", num_share_per_trade)
    ib.placeOrder(stock, order)

elif position - num_share_per_trade >= -limit_num_shorts and \
signal_pred == -1 and \
np.abs(price.iloc[-1] - last_trade) > commission:
    position -= num_share_per_trade
    cur_cash += num_share_per_trade * price.iloc[-1]
    cur_cash -= commission
    signal = -1
    last_trade = price.iloc[-1]
    print(f"{now} Sell", file=f)
    order = MarketOrder("Sell", num_share_per_trade)
    ib.placeOrder(stock, order)
else:
    signal = 0
    print(f"{now} No position", file=f)

# Saving the actions
signals.append(signal)
cash.append(cur_cash)
positions.append(position)
```

Submit market orders according to the model's prediction

Deploying on Interactive Brokers

2 Current Portfolio



P&L + PROFILE + IMPACT

DAILY -21 Unrealized -57.6
Since prior Close 0.00% Realized

Margin + ACCOUNT

Net Liquidity 1.0M Excess Liq 1.0M
Maintenance 1.5K SMA 897.7K

DLY P&L	FIN INSTRUMENT	POSITION	MKT VAL	AVG PX	LAST	CHANGE
2	AMD	3	503	167.89	167.50	+0.75
2	MSFT	4	1,661	418.39	415.19	+0.52
1	PYPL	1	63	63.69⁴	63.02	+0.51
1	AAPL	1	192	192.59	192.48	+1.19
1	INTC	1	31	30.65⁴	30.84	+0.65
0	GOOG	2	348	175.19	173.80	+0.24
0	META	1	467	468.38	466.86	-0.19
-7	BTC CRYPTO PAXOS	0.00999908	676	68321.2...	67606.75	+41.75
-1	BABA	1	78	80.69⁹	78.23	-1.40
-4	TSLA	3	533	179.20⁴	177.50	-1.29
-6	AMZN	2	353	180.31	176.27	-3.05
-10	COST	2	1,621	818.47	810.51	-4.83
	USD CASH		1,039,582			

Current Performance is not great

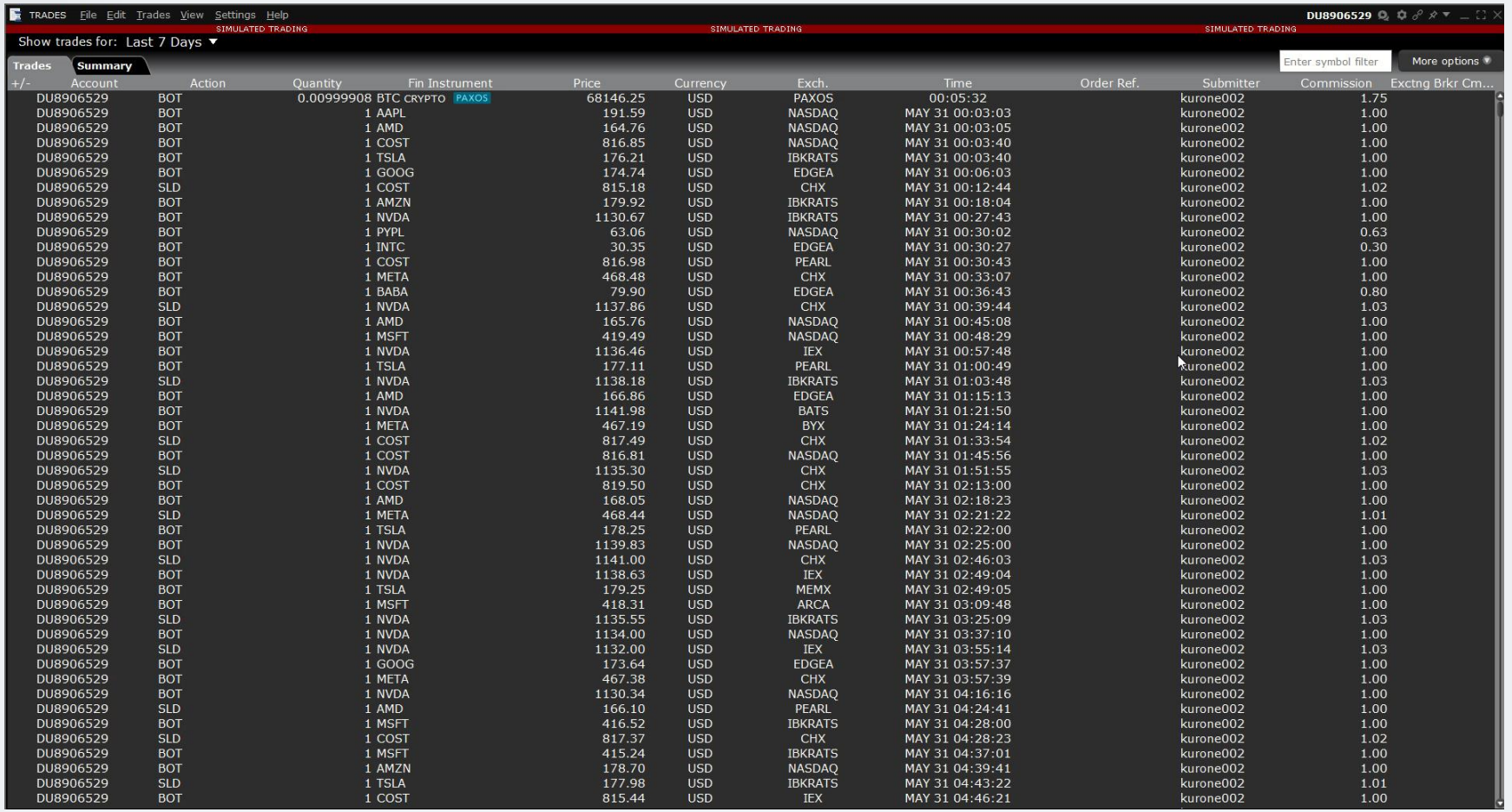
Might be due to trading in high frequency

Commission Fees

Different frequencies between training and deploying

Deploying on Interactive Brokers

3 Orders History



+/-	Account	Action	Quantity	Fin Instrument	Price	Currency	Exch.	Time	Order Ref.	Submitter	Commission	Exctng Brkr Cm...
	DU8906529	BOT	0.00999908	BTC CRYPTO PAXOS	68146.25	USD	PAXOS	00:05:32		kurone002	1.75	
	DU8906529	BOT	1	AAPL	191.59	USD	NASDAQ	MAY 31 00:03:03		kurone002	1.00	
	DU8906529	BOT	1	AMD	164.76	USD	NASDAQ	MAY 31 00:03:05		kurone002	1.00	
	DU8906529	BOT	1	COST	816.85	USD	NASDAQ	MAY 31 00:03:40		kurone002	1.00	
	DU8906529	BOT	1	TSLA	176.21	USD	IBKRATS	MAY 31 00:03:40		kurone002	1.00	
	DU8906529	BOT	1	GOOG	174.74	USD	EDGEA	MAY 31 00:06:03		kurone002	1.00	
	DU8906529	SLD	1	COST	815.18	USD	CHX	MAY 31 00:12:44		kurone002	1.02	
	DU8906529	BOT	1	AMZN	179.92	USD	IBKRATS	MAY 31 00:18:04		kurone002	1.00	
	DU8906529	BOT	1	NVDA	1130.67	USD	IBKRATS	MAY 31 00:27:43		kurone002	1.00	
	DU8906529	BOT	1	PYPL	63.06	USD	NASDAQ	MAY 31 00:30:02		kurone002	0.63	
	DU8906529	BOT	1	INTC	30.35	USD	EDGEA	MAY 31 00:30:27		kurone002	0.30	
	DU8906529	BOT	1	COST	816.98	USD	PEARL	MAY 31 00:30:43		kurone002	1.00	
	DU8906529	BOT	1	META	468.48	USD	CHX	MAY 31 00:33:07		kurone002	1.00	
	DU8906529	BOT	1	BABA	79.90	USD	EDGEA	MAY 31 00:36:43		kurone002	0.80	
	DU8906529	SLD	1	NVDA	1137.86	USD	CHX	MAY 31 00:39:44		kurone002	1.03	
	DU8906529	BOT	1	AMD	165.76	USD	NASDAQ	MAY 31 00:45:08		kurone002	1.00	
	DU8906529	BOT	1	MSFT	419.49	USD	NASDAQ	MAY 31 00:48:29		kurone002	1.00	
	DU8906529	BOT	1	NVDA	1136.46	USD	IEX	MAY 31 00:57:48		kurone002	1.00	
	DU8906529	BOT	1	TSLA	177.11	USD	PEARL	MAY 31 01:00:49		kurone002	1.00	
	DU8906529	SLD	1	NVDA	1138.18	USD	IBKRATS	MAY 31 01:03:48		kurone002	1.03	
	DU8906529	BOT	1	AMD	166.86	USD	EDGEA	MAY 31 01:15:13		kurone002	1.00	
	DU8906529	BOT	1	NVDA	1141.98	USD	BATS	MAY 31 01:21:50		kurone002	1.00	
	DU8906529	BOT	1	META	467.19	USD	BYX	MAY 31 01:24:14		kurone002	1.00	
	DU8906529	SLD	1	COST	817.49	USD	CHX	MAY 31 01:33:54		kurone002	1.02	
	DU8906529	BOT	1	COST	816.81	USD	NASDAQ	MAY 31 01:45:56		kurone002	1.00	
	DU8906529	SLD	1	NVDA	1135.30	USD	CHX	MAY 31 01:51:55		kurone002	1.03	
	DU8906529	BOT	1	COST	819.50	USD	CHX	MAY 31 02:13:00		kurone002	1.00	
	DU8906529	BOT	1	AMD	168.05	USD	NASDAQ	MAY 31 02:18:23		kurone002	1.00	
	DU8906529	SLD	1	META	468.44	USD	NASDAQ	MAY 31 02:21:22		kurone002	1.01	
	DU8906529	BOT	1	TSLA	178.25	USD	PEARL	MAY 31 02:22:00		kurone002	1.00	
	DU8906529	BOT	1	NVDA	1139.83	USD	NASDAQ	MAY 31 02:25:00		kurone002	1.00	
	DU8906529	SLD	1	NVDA	1141.00	USD	CHX	MAY 31 02:46:03		kurone002	1.03	
	DU8906529	BOT	1	NVDA	1138.63	USD	IEX	MAY 31 02:49:04		kurone002	1.00	
	DU8906529	BOT	1	TSLA	179.25	USD	MEMX	MAY 31 02:49:05		kurone002	1.00	
	DU8906529	BOT	1	MSFT	418.31	USD	ARCA	MAY 31 03:09:48		kurone002	1.00	
	DU8906529	SLD	1	NVDA	1135.55	USD	IBKRATS	MAY 31 03:25:09		kurone002	1.03	
	DU8906529	BOT	1	NVDA	1134.00	USD	NASDAQ	MAY 31 03:37:10		kurone002	1.00	
	DU8906529	SLD	1	NVDA	1132.00	USD	IEX	MAY 31 03:55:14		kurone002	1.03	
	DU8906529	BOT	1	GOOG	173.64	USD	EDGEA	MAY 31 03:57:37		kurone002	1.00	
	DU8906529	BOT	1	META	467.38	USD	CHX	MAY 31 03:57:39		kurone002	1.00	
	DU8906529	BOT	1	NVDA	1130.34	USD	NASDAQ	MAY 31 04:16:16		kurone002	1.00	
	DU8906529	SLD	1	AMD	166.10	USD	PEARL	MAY 31 04:24:41		kurone002	1.00	
	DU8906529	BOT	1	MSFT	416.52	USD	IBKRATS	MAY 31 04:28:00		kurone002	1.00	
	DU8906529	SLD	1	COST	817.37	USD	CHX	MAY 31 04:28:23		kurone002	1.02	
	DU8906529	BOT	1	MSFT	415.24	USD	IBKRATS	MAY 31 04:37:01		kurone002	1.00	
	DU8906529	BOT	1	AMZN	178.70	USD	NASDAQ	MAY 31 04:39:41		kurone002	1.00	
	DU8906529	SLD	1	TSLA	177.98	USD	IBKRATS	MAY 31 04:43:22		kurone002	1.01	
	DU8906529	BOT	1	COST	815.44	USD	IEX	MAY 31 04:46:21		kurone002	1.00	

Thank you