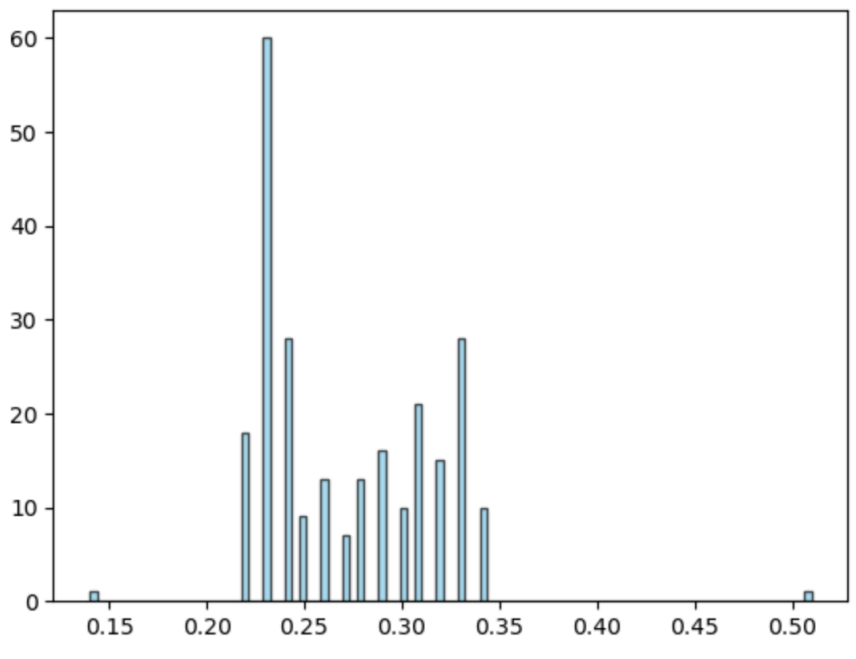
Project Report: Predictive Modeling with Market Dynamics

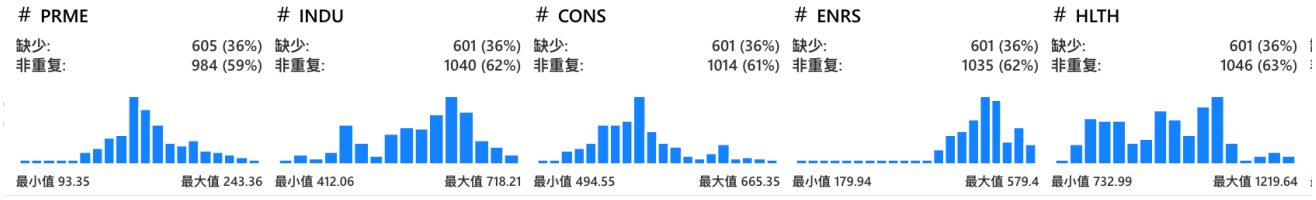
▼ 0. Data Preprocessing

- For all 12 features, there are no values in non-trading days, but LL100 is has no missing values. So I assume LL100 is traded in all dates provided.

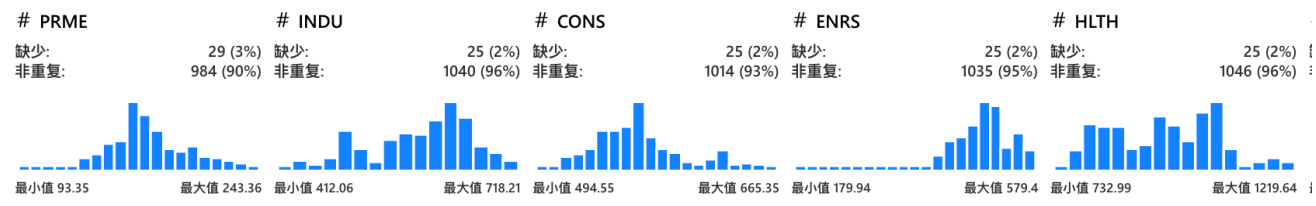
Although I notice that the weekend data follows approximately a uniform distribution:



The original feature missing values account for 35%-36%. Therefore, for the prediction model, I decided to use only trading days from `2016-01-01` to `2020-04-30` and filter out the missing values for feature input in weekends.



- After filter out non-trading days, there're 0%-3% missing values for features, and then use forward fill for filling missing data.



For the strategy implementation part, I add a random value from the distribution to the Friday's data to simulate the LL100 weekend prediction value.

- All feature values are missing(except SP500) between `2020-05-01` and `2020-07-22`. Therefore this period is removed in future analysis to avoid introducing inaccuracies in feature inputs.

▼ 1. Non-Stationarity Analysis

▼ **Stationarity Testing**

- The Augmented Dickey-Fuller (ADF) test was applied to all features and the target variable LL100.
  - Null Hypothesis (H0): The series has a unit root and is non-stationary.
- Result:
  - LL100 is non-stationary, while all other 12 features are stationary.

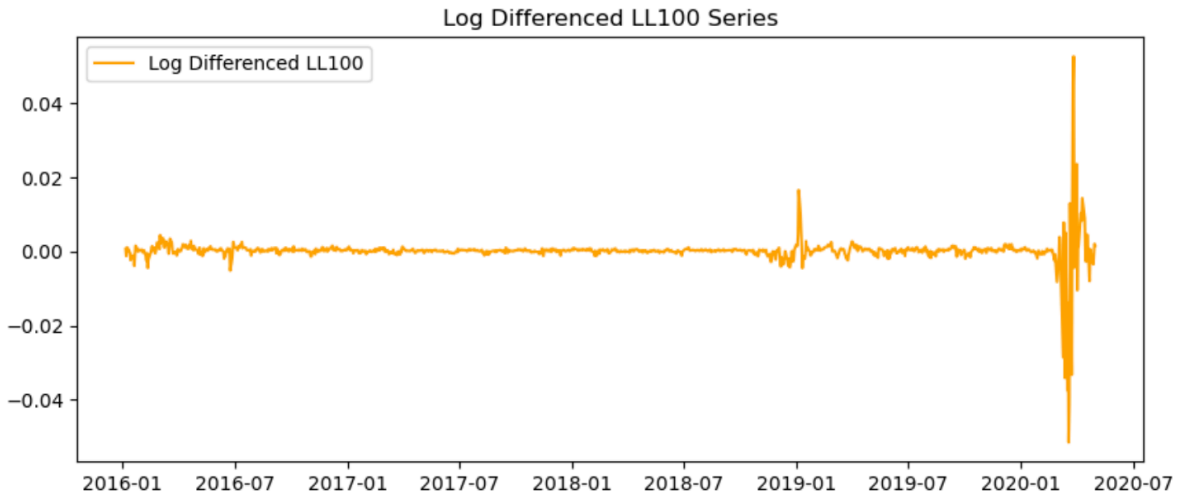
```
Testing stationarity of LL100:
ADF Statistic: -2.406256531890945
p-value: 0.13997832993914638
Critical Value 1%: -3.4364992651202697
Critical Value 5%: -2.8642551098431968
Critical Value 10%: -2.5682156239065925
The series is non-stationary.
```

This rules out the concerns for spurious regression when using OLS for prediction, but the stationary target variable can still cause inefficient coefficient result.

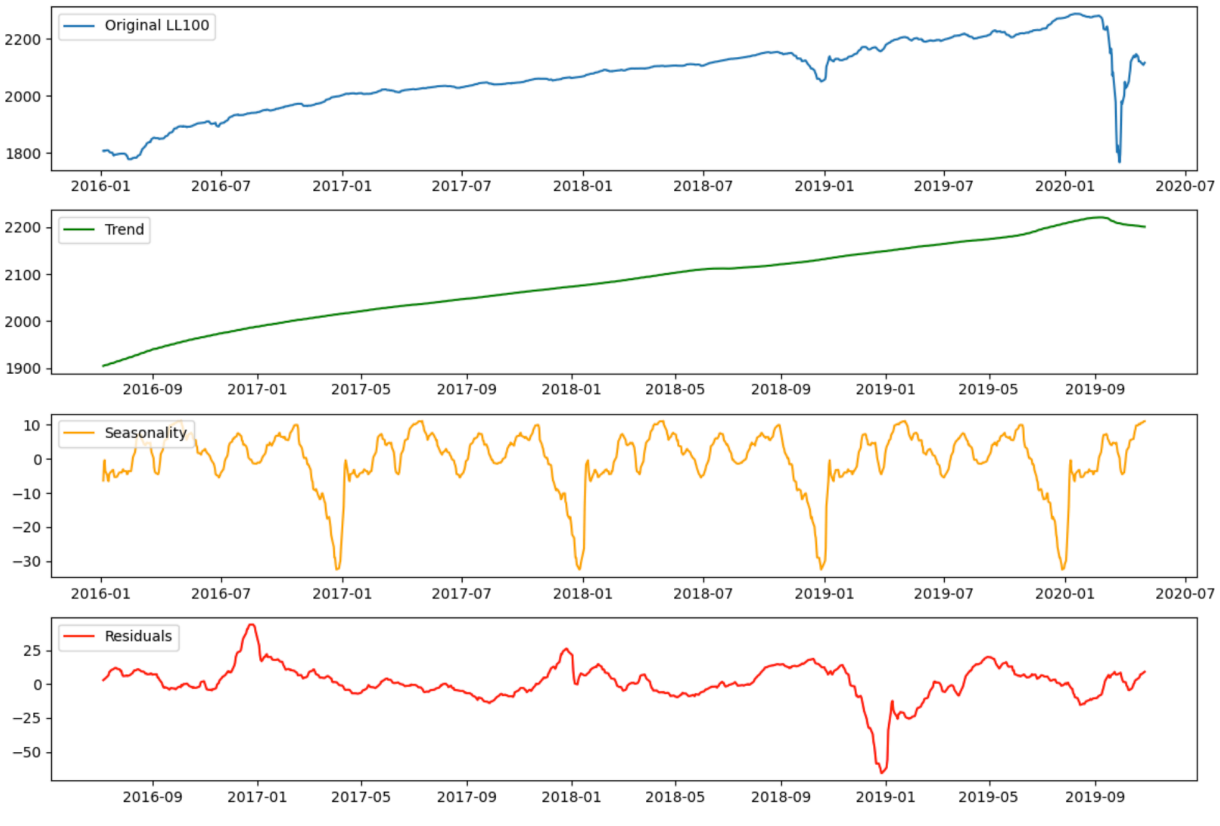
▼ **Approaches to Handle Non-Stationarity**

1. **Differencing:** Calculated the first difference of LL100, which successfully made the series stationary (ADF p-value < 0.05).
2. **Log Transformation + Differencing:** Log transformation followed by differencing also achieved stationarity.

```
Testing stationarity after log transformation:
ADF Statistic: -8.88006675875351
p-value: 1.325336476890943e-14
Critical Value 1%: -3.436510851955201
Critical Value 5%: -2.864260220574562
Critical Value 10%: -2.5682183458999943
The series is stationary.
```

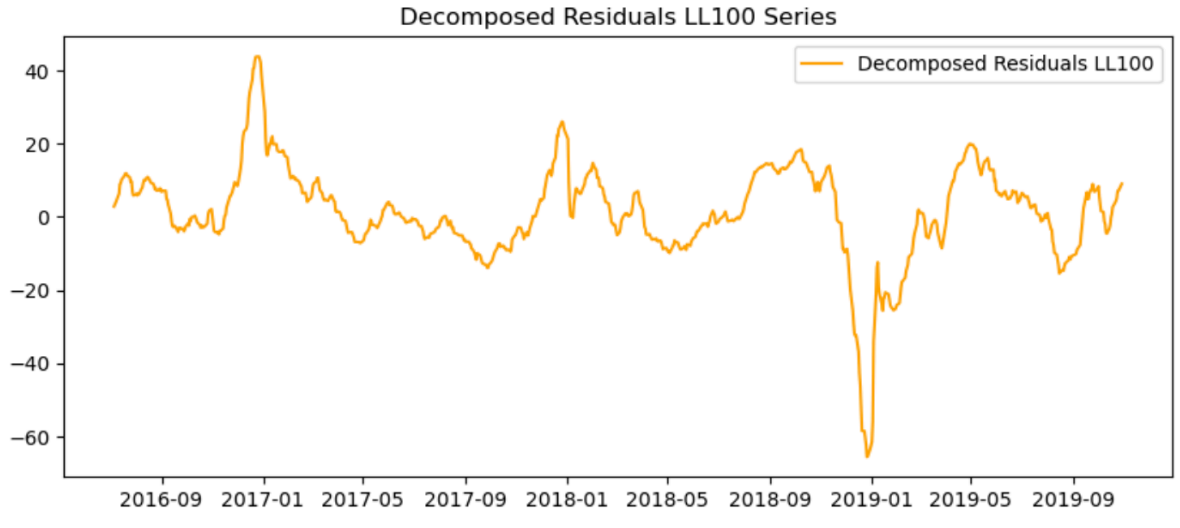


3. **Decomposition:**



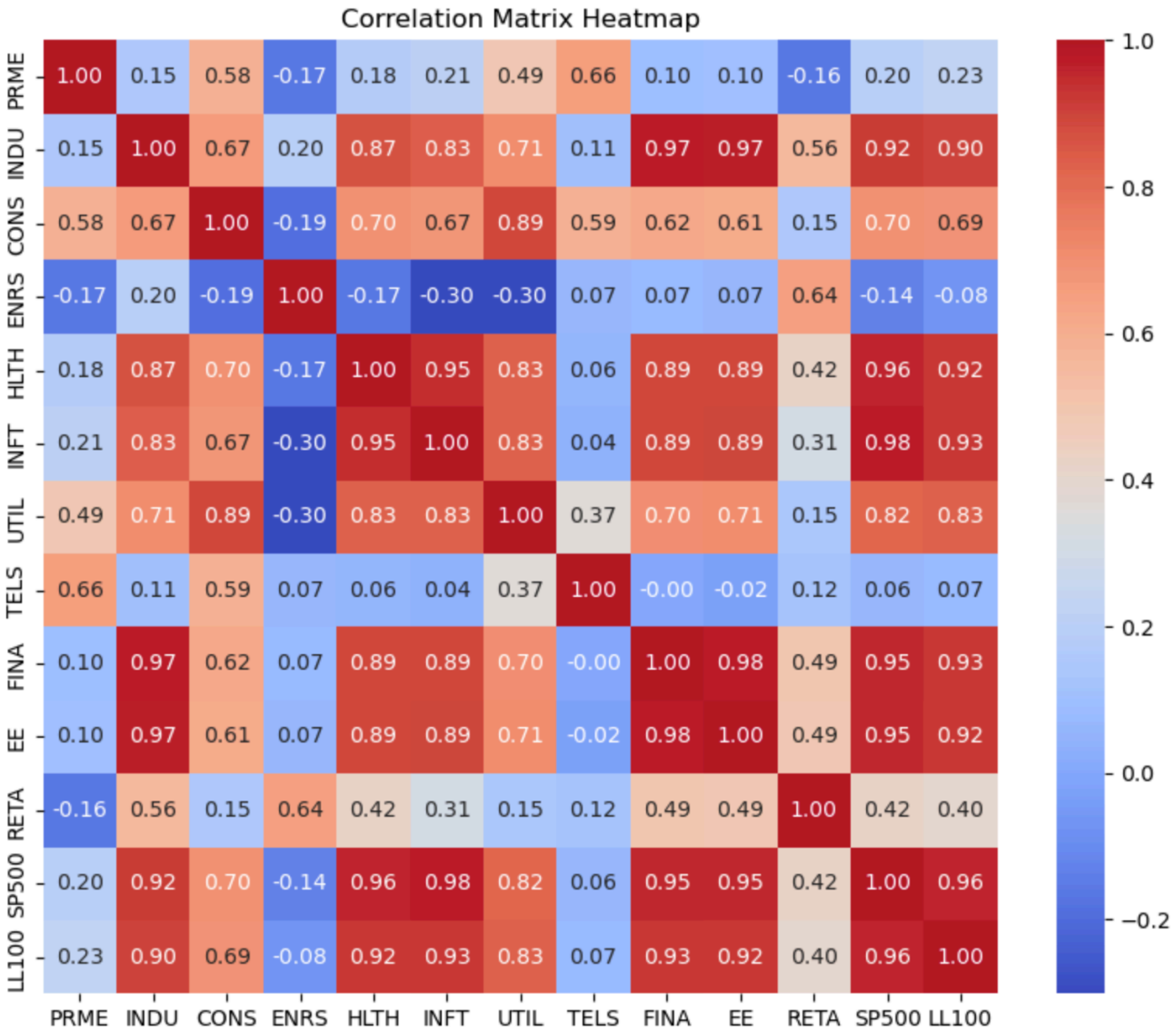
Seasonal decomposition of revealed:

- A clear upward trend: LL100 shows an overall upward trend, but there was a sharp decline from the end of 2019 to the beginning of 2020, possibly triggered by covid market crash.
- Seasonal pattern: LL100 exhibits clear periodic fluctuations, indicating that certain fixed time factors have a significant impact on the data.
- The residuals fluctuate around 0 for most of the time period, indicating that the trend and seasonality have explained the data well.



▼ 2. Causal Relationship Mapping

▼ Correlation Analysis



From the above correlation matrix, we can see there’re 15 possible casual relationship pairs with correlation above 0.9 shown as below.

Variable_1	Variable_2	# Correlation
INDU	FINA	0.9726425641371729
INDU	EE	0.9664953963246804
INDU	SP500	0.9150527789666218
INDU	LL100	0.9015433635719381
HLTH	INFT	0.9475405667162212
HLTH	SP500	0.9609092273912524
HLTH	LL100	0.9197770680106374
INFT	SP500	0.9787729737408286
INFT	LL100	0.9279095133711943
FINA	EE	0.9772374505004188
FINA	SP500	0.9494717983870558
FINA	LL100	0.9271543392487491
EE	SP500	0.9497365836576087
EE	LL100	0.9207592655583807
SP500	LL100	0.9637108997002228

- **Possible Lead-Lag relationship & Financial Perspective**

  The industrial sector (`INDU`) and the financial sector (`FINA`) exhibit a very high correlation (0.97). The industrial sector often relies on the financial sector for capital investment, loans, and credit. When the financial sector performs well (e.g., low-interest rates, strong credit growth), industrial companies often expand operations, leading to higher production and growth. Personally, I believe FINA potentially leading INDU due to its role in providing credit and liquidity to industrial firms.

  The information technology sector (`INFT`) has an extremely high correlation with the S&P500 (`SP500`) with a 0.98 correlation. This strong relationship reflects the increasing dominance of the technology sector within the S&P 500, particularly in recent years. Key players such as the `Tech 7` companies—Apple, Microsoft, Alphabet, Amazon, Meta (formerly Facebook), Tesla, and Nvidia—have collectively grown to represent a significant portion of the index's total market capitalization. Their outsized influence means that fluctuations in the performance of these companies can disproportionately impact the overall trajectory of the S&P 500, making

INFT a critical driver of market sentiment and investor behavior. This relationship highlights the tech sector's evolution from a high-growth niche to a cornerstone of the global economy, playing a central role in shaping the financial markets.

▼ **Granger Causality Test**

$$H_0 : \text{resstricted model, no causality}$$
$$Y_t = \alpha + \sum_{i=1}^{p} \beta_i Y_{t-i} + \epsilon_t$$
$$H_1 : \text{unrestricted Model, with causality}$$
$$Y_t = \alpha + \sum_{i=1}^{p} \beta_i Y_{t-i} + \sum_{i=1}^{p} \gamma_i X_{t-i} + \epsilon_t$$

- **Result:**
  - There’re certain levels of granger causality between each pair of variables.
  - `INDU` , `FINA` , `EE` , and `SP500` strongly lead `LL100` , passing the Granger Causality Test for lag periods between1-10 days.

▼ **VAR Model & Impulse Response Analysis**

▼ VAR

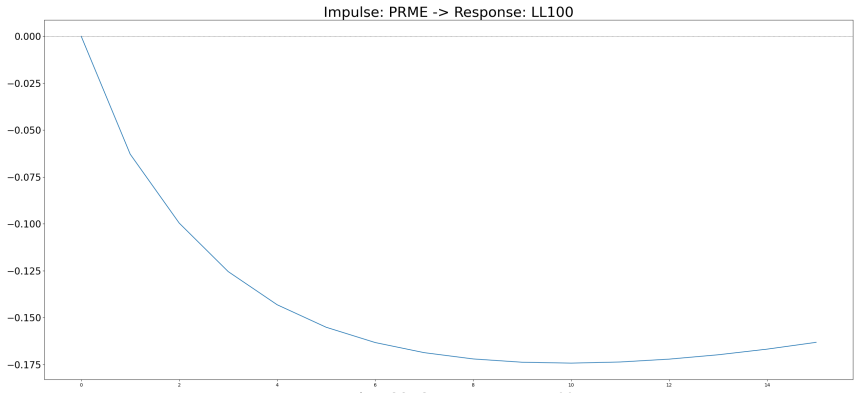
$$y_{\text{diff},t} = \beta_0 + \sum_{i=1}^{k} \beta_i y_{\text{diff},t-i} + \sum_{j=1}^{k} \gamma_j x_{t-j} + \eta_t$$

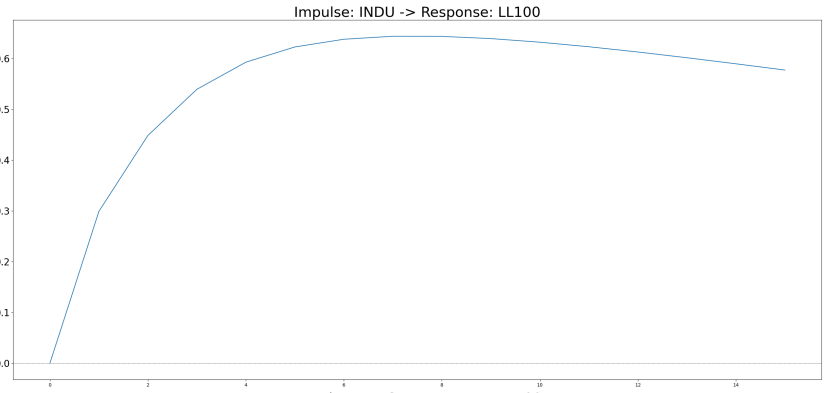
where k =2, $y_{\text{diff},t}$ represents the difference of all 13 variables, and $y_{\text{diff},t}$ represents the other 12 variables’ difference at time t

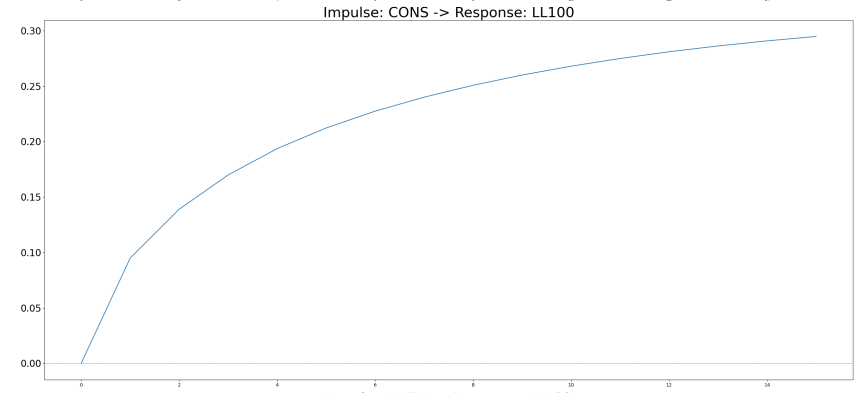
```
Results for equation LL100
================================================================================
             coefficient       std. error           t-stat            prob
--------------------------------------------------------------------------------
const           8.413622         3.756049            2.240           0.025
L1.PRME         0.019456         0.018054            1.078           0.281
L1.INDU        -0.061961         0.031268           -1.982           0.048
L1.CONS        -0.040834         0.028209           -1.448           0.148
L1.ENRS        -0.040646         0.018571           -2.189           0.029
L1.HLTH        -0.049985         0.019230           -2.599           0.009
L1.INFT        -0.117197         0.027896           -4.201           0.000
L1.UTIL        -0.089079         0.037141           -2.398           0.016
L1.TELS        -0.077167         0.049588           -1.556           0.120
L1.FINA        -0.080425         0.025543           -3.149           0.002
L1.EE          -0.003198         0.006178           -0.518           0.605
L1.RETA        -0.001666         0.001455           -1.146           0.252
L1.SP500        0.155004         0.038198            4.058           0.000
L1.LL100        1.570073         0.028172           55.731           0.000
L2.PRME        -0.009452         0.017985           -0.526           0.599
L2.INDU         0.063308         0.031290            2.023           0.043
L2.CONS         0.025359         0.028314            0.896           0.370
L2.ENRS         0.030094         0.018606            1.617           0.106
L2.HLTH         0.046182         0.019242            2.400           0.016
L2.INFT         0.111831         0.027829            4.019           0.000
L2.UTIL         0.088822         0.036899            2.407           0.016
L2.TELS         0.086513         0.049525            1.747           0.081
L2.FINA         0.075874         0.025431            2.984           0.003
L2.EE           0.004657         0.006210            0.750           0.453
L2.RETA         0.002065         0.001457            1.417           0.156
L2.SP500       -0.147783         0.038236           -3.865           0.000
L2.LL100       -0.574288         0.027970          -20.533           0.000
================================================================================
```

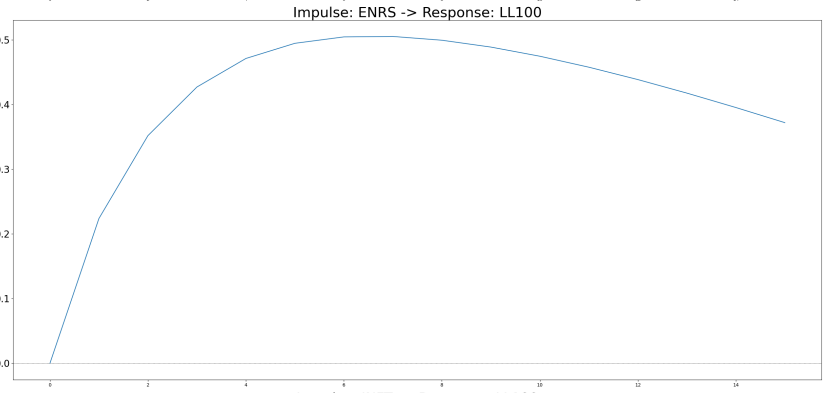
▼ IRF

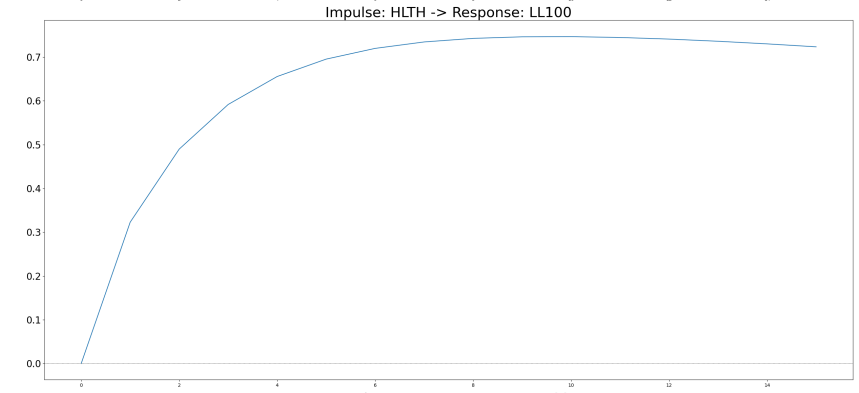
Based on this VAR model structure, I conduct impluse response analysis over 15 periods.

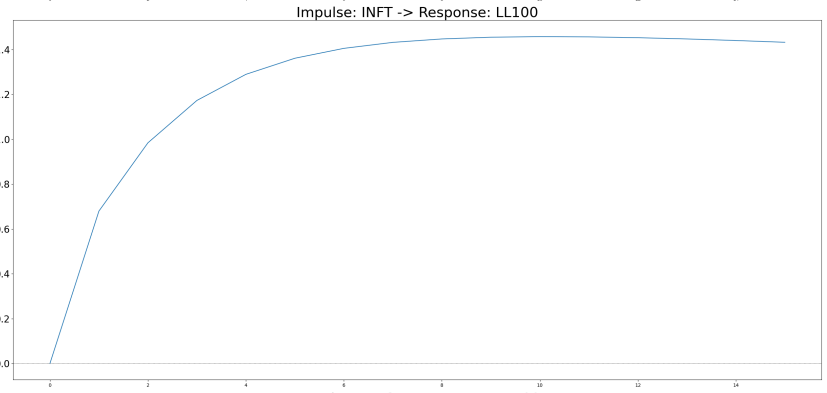

Impulse: PRME -> Response: LL100

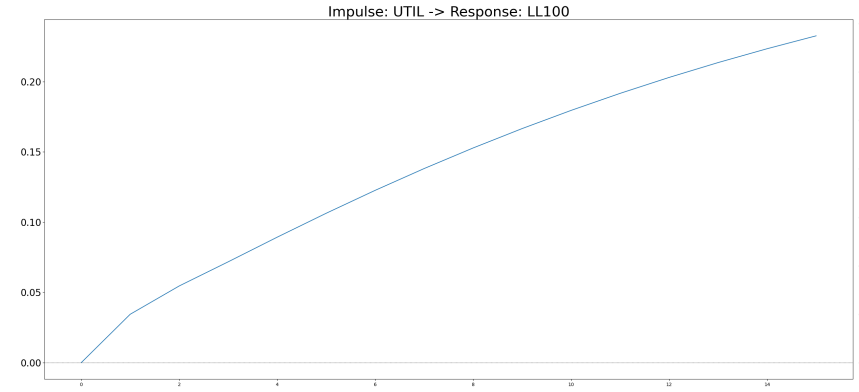

Impulse: INDU -> Response: LL100

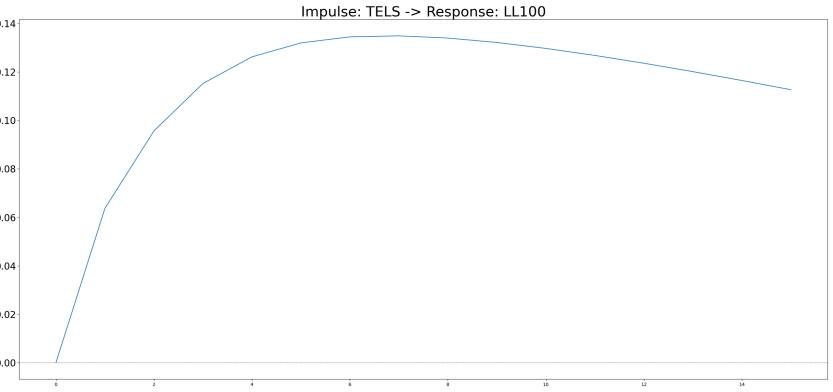

Impulse: CONS -> Response: LL100

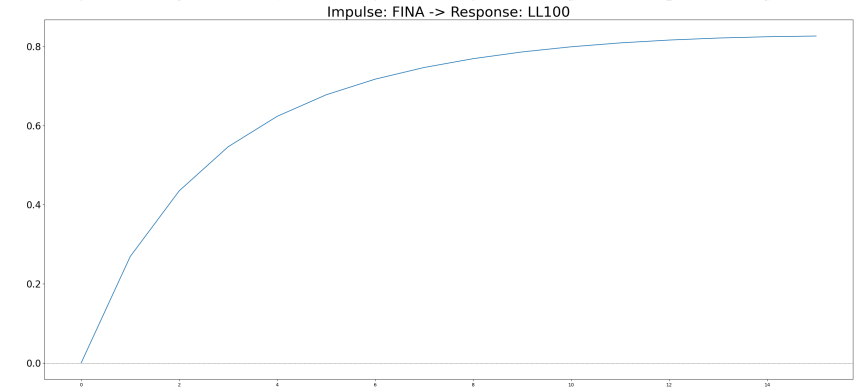

Impulse: ENRS -> Response: LL100

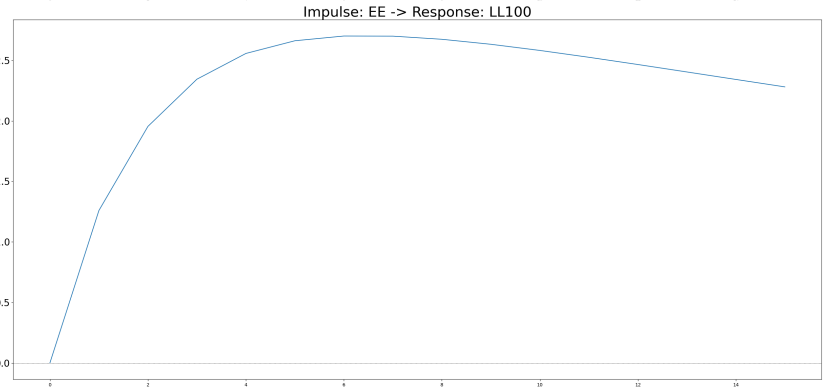

Impulse: HLTH -> Response: LL100

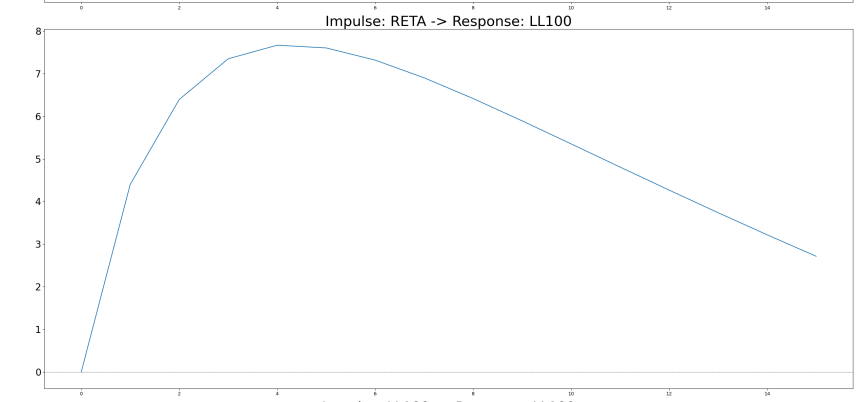

Impulse: INFT -> Response: LL100

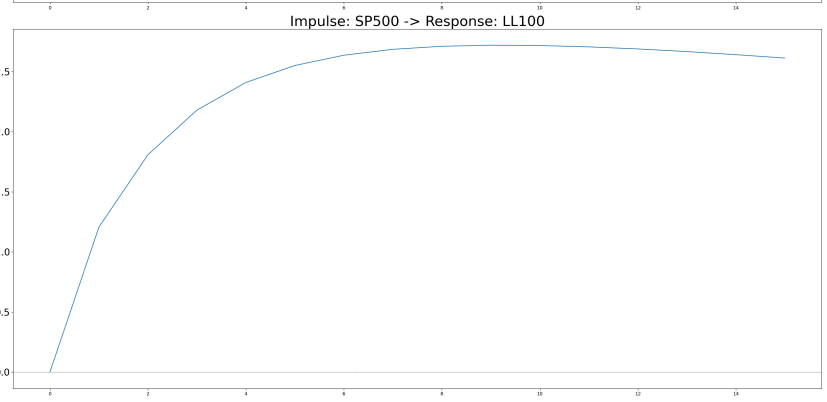

Impulse: UTIL -> Response: LL100

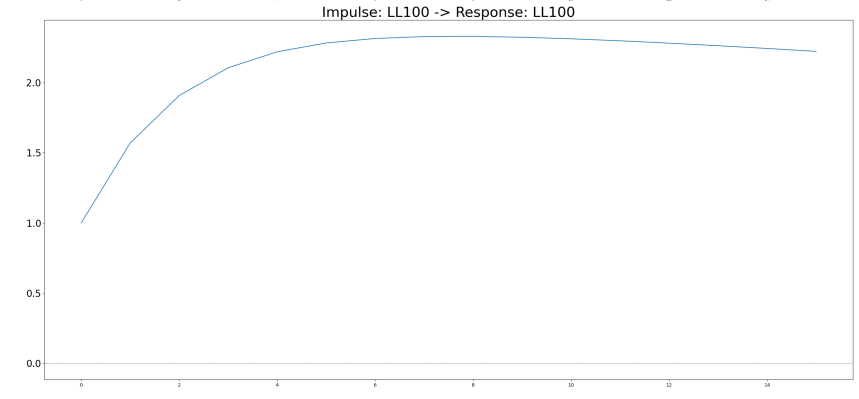

Impulse: TELS -> Response: LL100


Impulse: FINA -> Response: LL100


Impulse: EE -> Response: LL100


Impulse: RETA -> Response: LL100


Impulse: SP500 -> Response: LL100


Impulse: LL100 -> Response: LL100

▼ Results

VAR modeling revealed the following impacts on `LL100`:

- **Short-term impact:** `RETA` and `EE`.
  - Retail index has a significant short term impact (8) on LL100. Emerging Economies index has a short term impact (2.5) on LL100. Retail and emerging markets are more reactive to current economic conditions, making them significant short-term drivers of LL100. These sectors capture immediate changes in consumer behavior and global market sentiment.
- **Long-term impact:** `SP500` and `INFT`.
  - SP500, Information Technology index has a significant long term impact(2.5, 1.4) on LL100. The S&P 500 and the Information Technology index reflect structural and macroeconomic trends that shape the economy over time. Their influence on LL100 underscores the importance of equity markets and technological innovation in driving sustained economic performance.
- **Negative impact:** `PRME`.
  - Prime Index is the only one with a small negative impact on LL100.

▼ 3. Model Architecture

In this section, I explore econometric, machine learning and state-of-the-art models to predict LL100 index. I created a evaluation metrics with several indicators, aiming to identify which approach better captures market regime changes. I use the first 80% as train dataset, and the last 20% as test dataset.

▼ Evaluation Metrics

see `eval.py`

- **Mean Squared Error (MSE)**
- **Mean Absolute Error (MAE)**
- **Rolling Window MSE**
  - Use a rolling window of 30 days
- **R-squared (R^2)**
- **Explained Variance**
  - Measures the proportion of variance in the target variable that is explained by the model's predictions. Similar to *R*2, but focuses on the variance of the predicted values relative to the actual values.
- **Directional Accuracy**
  - Measures the percentage of times the model correctly predicts the direction of change (up or down) in LL100.

▼ OLS Regression

```
Ljung-Box Test Results:
        lb_stat  lb_pvalue
10  6269.362169        0.0
ADF Test Results:
ADF Statistic: -3.317074972178901
p-value: 0.014132263180800748
Critical Values:
 1%: -3.437923659686726
 5%: -2.8648832361839442
 10%: -2.5685501889710864
```

	OLS_train	OLS_test
MSE	537.92	7199.04
MAE	18.9	51.68
Rolling Window MSE	542.23	3369.62
R-squared (R^2)	0.95	0.09
Explained Variance	0.95	0.39
Directional Accuracy (%)	55.52	60.83

- Residuals were stationary (ADF p-value < 0.05) but showed autocorrelation (Ljung-Box p-value = 0.0), indicating OLS coefficient is consistent but inefficient.
- Performance on the test set is poor, prompting exploration of regularization techniques.

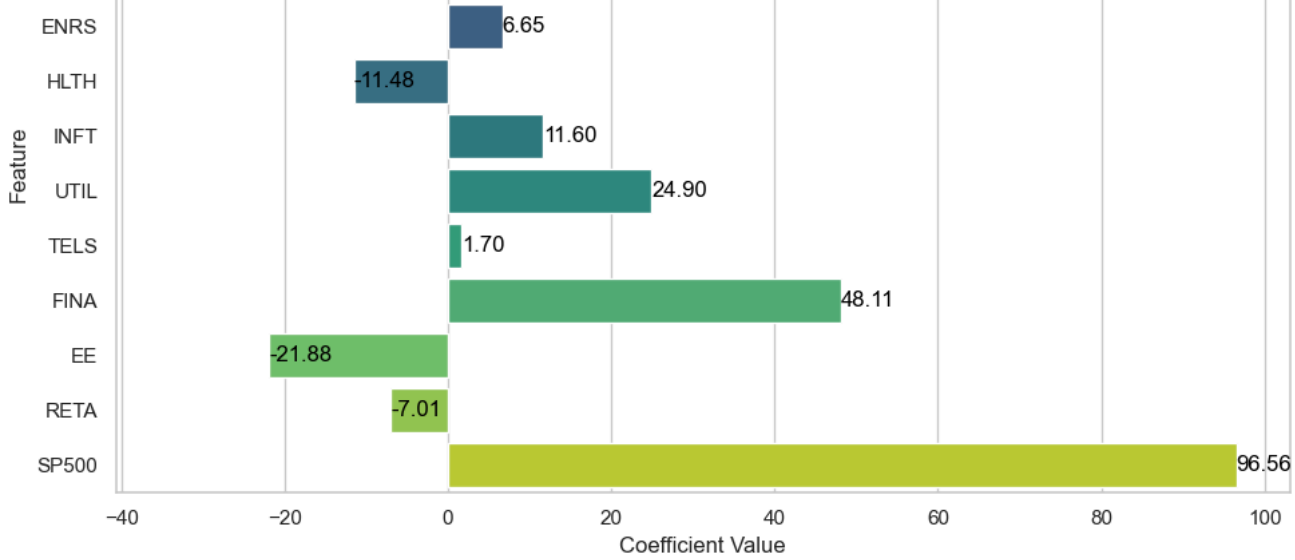
▼ Lasso, Ridge and Elastic Net

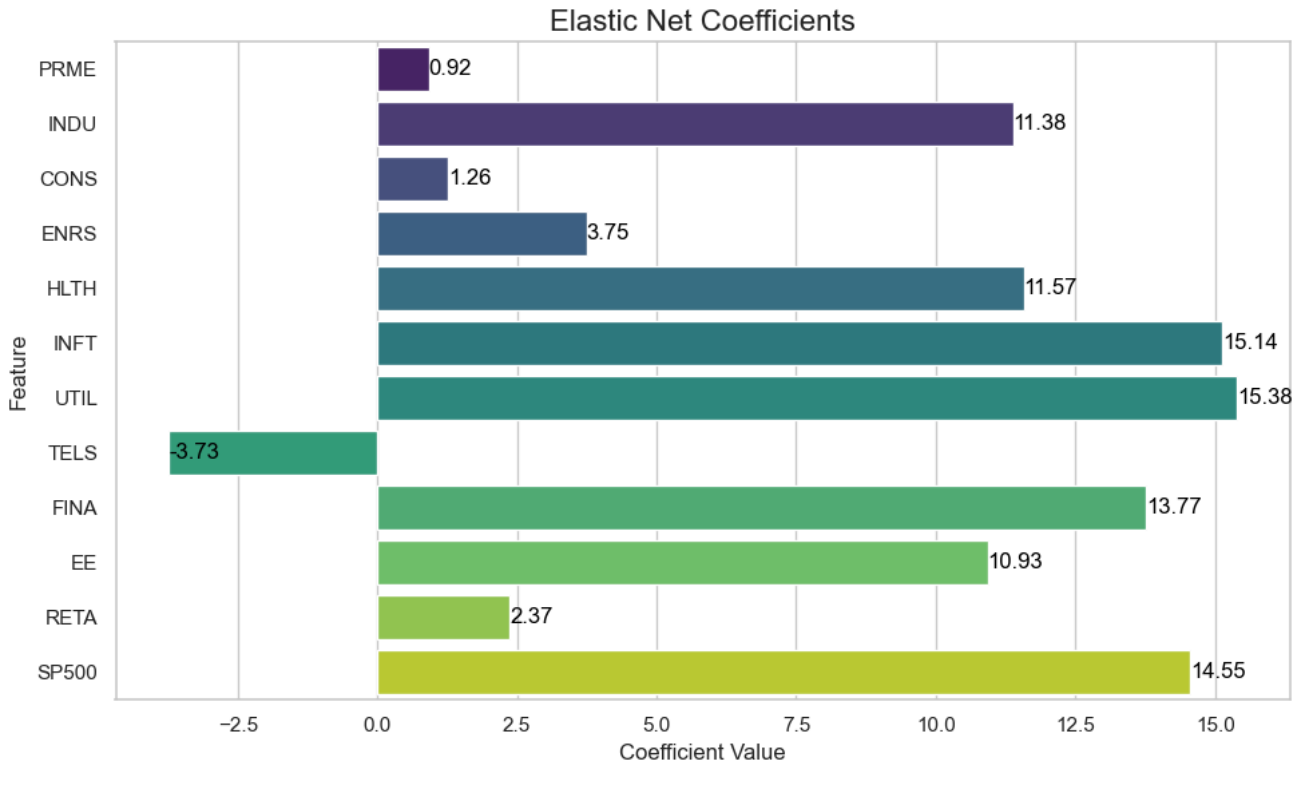
- Standardized 12 features as input

- The MSE and Rolling window MSE is large due to the data pattern in 2020 (test set) with a significant market crash. In the case, MAE is probably a better indicator than MSE for model evaluation.

	OLS_train	OLS_test	Lasso_train	Lasso_test	Ridge_train	Ridge_test	Elastic Net_train	Elastic Net_test
MSE	537.92	7199.04	653.59	2975.72	563.61	4892.34	869.4	1793.84
MAE	18.9	51.68	20.78	40.79	19.67	58.88	23.81	28.85
Rolling Window MSE	542.23	3369.62	654.67	3211.12	569.76	5193.78	822.55	1489.32
R-squared (R^2)	0.95	0.09	0.94	0.63	0.95	0.38	0.92	0.77
Explained Variance	0.95	0.39	0.94	0.76	0.95	0.82	0.92	0.8
Directional Accuracy (%)	55.52	60.83	56.21	63.13	57.01	62.67	56.9	61.75

- Elastic Net Achieved the best performance among linear models with R^2 = 0.77 and Explained Variance Score = 0.80 on the test set. Since it combines the strengths of both Lasso and Ridge regression. The regularization parameter alpha controls the trade-off between the L1 and L2 penalty terms.





▼ Random Forest, LightGBM, and XGBoost

- LightGBM achieved the best performance among ML models after parameter tuning with rolling window approach. (see `rolling_window_finetune.py`)
- Elastic Net still outperforms these models.

	RF_train	RF_test	RF_finetuned_train	RF_finetuned_test	LGB_train	LGB_test	LGB_finetuned_train	LGB_finetuned_test	XGB_train	XGB_test	XGB_finetuned_train	XGB_finetuned_test
MSE	4.69	4101.76	5.16	4281.36	3.29	2945.54	0.03	3204.95	0.03	4605.77	0.26	4037.92
MAE	1.08	46.43	1.24	46.81	1.1	43	0.13	42.79	0.11	46.07	0.39	44.49
Rolling Window MSE	4.73	4495.95	5.25	4689.02	3.26	3137.84	0.03	3508.16	0.03	5039.84	0.26	4415.76
R-squared (R^2)	1	0.48	1	0.46	1	0.63	1	0.6	1	0.42	1	0.49
Explained Variance	1	0.56	1	0.53	1	0.75	1	0.69	1	0.48	1	0.55
Directional Accuracy (%)	78.85	47.47	77.24	41.94	70.23	47.93	93.79	53.92	95.06	41.47	80.46	46.54

▼ LSTM

- LSTM struggled to perform well due to the small dataset and the presence of strong trends. Fine-tuning yielded marginal improvements but did not surpass Elastic Net or LightGBM.
- Below shows the LSTM result, the right two columns' predicted variable is LL100 removed of trend using decomposition, showing a significant improvement in model performance.

LSTM_finetuned_train	LSTM_finetuned_test	Detrend_LSTM_finetuned_train	Detrend_LSTM_finetuned_test
21048.72	5788.62	442.92	898.61
113.85	67.92	13.21	18.02
20005.66	6163.95	449.74	625.81
-0.98	0.27	-1.06	-23.09
-0.98	0.67	-1.06	-21.35
48.97	57.6	45.29	29.49

Conclusion

Elastic Net regression demonstrates the best predictive performance, followed by LightGBM. Therefore, I use Elastic Net model for predicting LL100 and building signals for the following implementation.

▼ 4. Practical Implementation

▼ Expanding Window Prediction

- Elastic Net Model Structure

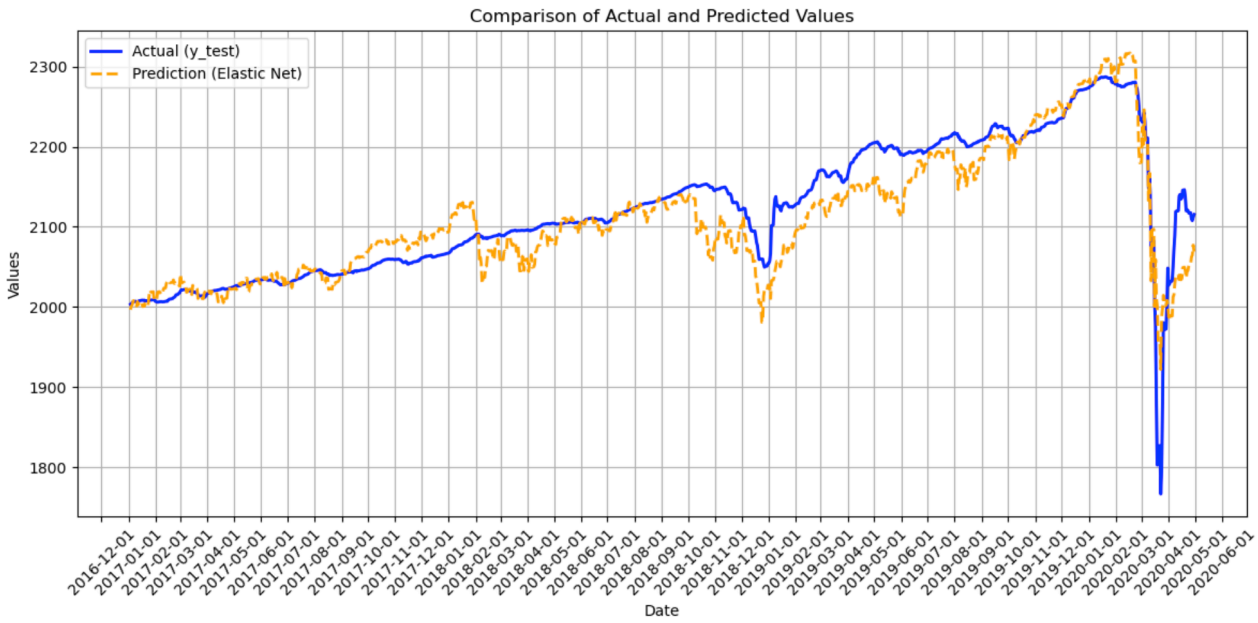
$$loss = \frac{1}{2n}\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p} X_{ij}\beta_j\right)^2 + \alpha((1 - l1ratio)\cdot\frac{1}{2}||\beta||^2 + l1ratio\cdot\frac{1}{2}||\beta||^1)$$
$$where\ \text{alpha} = 1.0, \text{l1ratio} = 0.5$$

- Expanding Window

  Start with data in `2016` as train set to predict the LL100 of `2017-01-01` , expanding the train set for daily LL100 prediction. Then shift the prediction for one day to avoid using future data.

- Use the distribution of `ll100_diff` to add values to fill NaN values for weekends, accounting for weekend transaction for LL100.

Performance Metrics for Elastic Net (test)	
	Elastic Net_test
MSE	1089.40
MAE	23.87
Rolling Window MSE	1048.68
R-squared (R^2)	0.84
Explained Variance	0.85
Directional Accuracy (%)	69.91



▼ Momentum-based Strategy

The trading strategy uses a rolling prediction signal to identify potential trends in the market and generates buy or sell signals based on the momentum of the predictions. The strategy incorporates risk management techniques, including `trailing stop-loss` , `take-profit levels` , and a `maximum holding period` . All trades are executed using the entire available capital, and the strategy is benchmarked against a simple buy-and-hold approach.

▼ Trading Logic

Momentum Signal:

The signal is calculated as the difference between the current and previous rolling predictions:

$$\text{Signal} = \text{Rolling Prediction}_t - \text{Rolling Prediction}_{t-1}$$

- If the signal > 0, it indicates upward momentum, generating a buy signal.
- If the signal < 0 , it indicates downward momentum, generating a sell signal
- If the signal = 0 , no trade executed.

Risk Management:

The strategy exits a position based on the following conditions:

- Trailing Stop-Loss:
  - For a long position, if the price falls below the peak price by more than the specified `stop_loss_trailing` percentage, the position is exited.
  - For a short position, if the price rises above the peak price by more than the specified `stop_loss_trailing` percentage, the position is exited.

- Take-Profit:
  - For a long position, if the price rises above the entry price by the specified `take_profit` percentage, the position is exited.
  - For a short position, if the price falls below the entry price by the specified `take_profit` percentage, the position is exited.

Capital Allocation

At the time of entry, the entire available capital is allocated to the position. The position size is calculated as:

$$\text{Position Size} = \frac{\text{current capital}}{\text{entry price}}$$

▼ **Parameters**

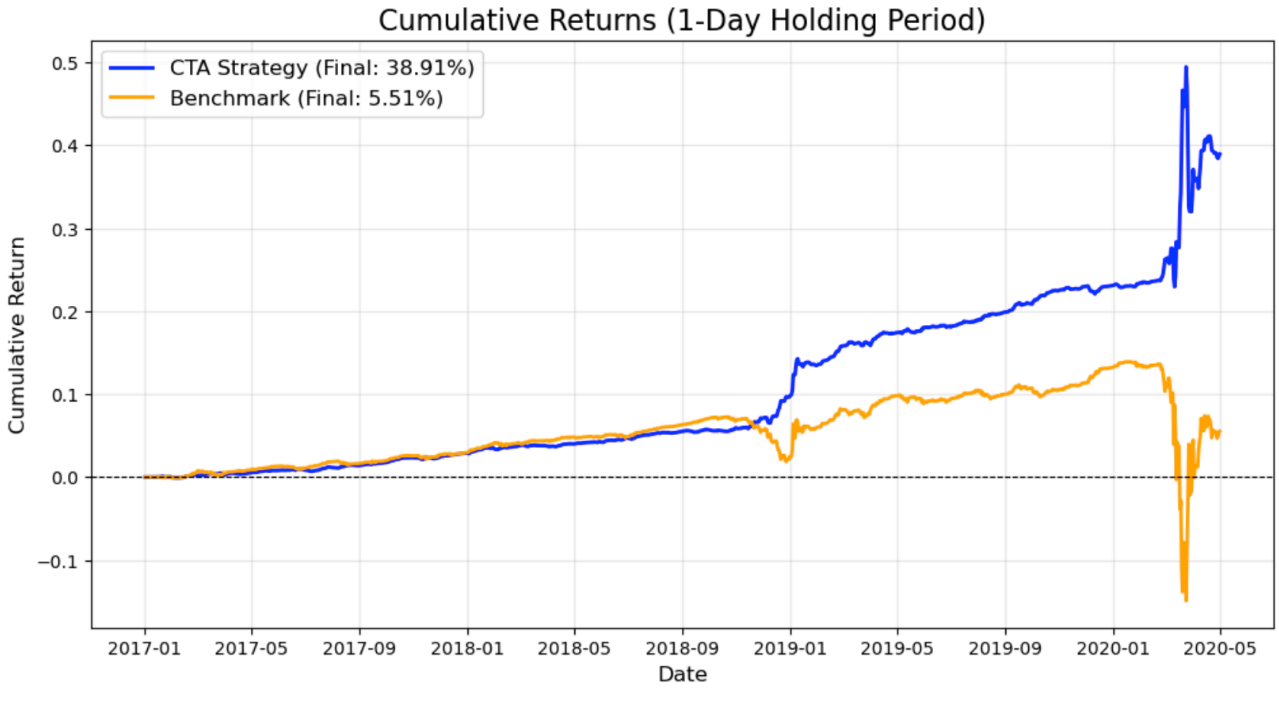
1. Initial Capital (`initial_capital=100000`):
2. Rolling Window (`rolling_window=10`):
   - Used to calculate the average of the LL100 predictions.
3. Trailing Stop-Loss (`stop_loss_trailing=0.05`):
   - Exits a trade if the price moves against the position by 5% from the peak price.
4. Take-Profit (`take_profit=0.1`):
   - Exits a trade when the price moves in favor of the position by 10%
5. Maximum Holding Period (`holding_period`):
   - The maximum number of days a position can remain open.
6. Transaction Cost (`fee_rate=0.1%`)
   - Assume transaction cost is 0.1% of the capital at each trade

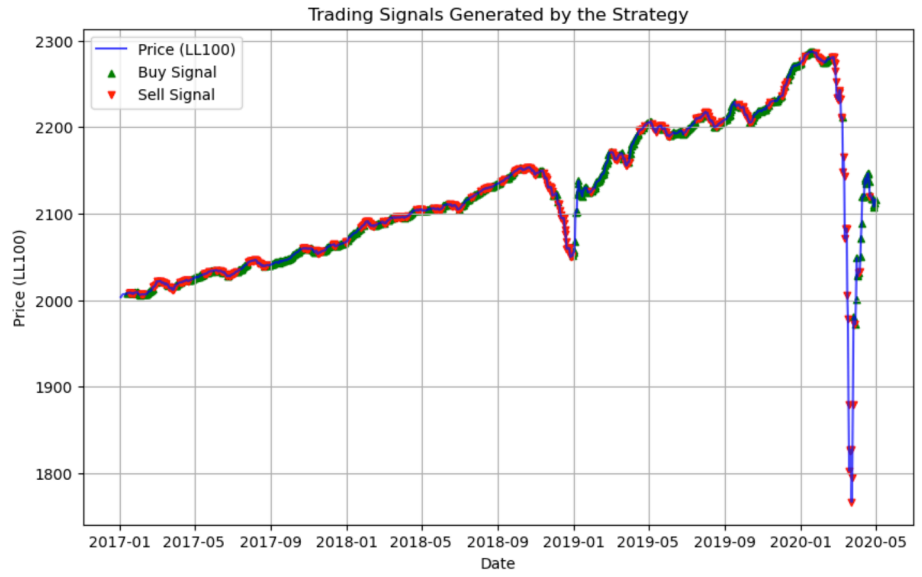
▼ **Backtest Result**

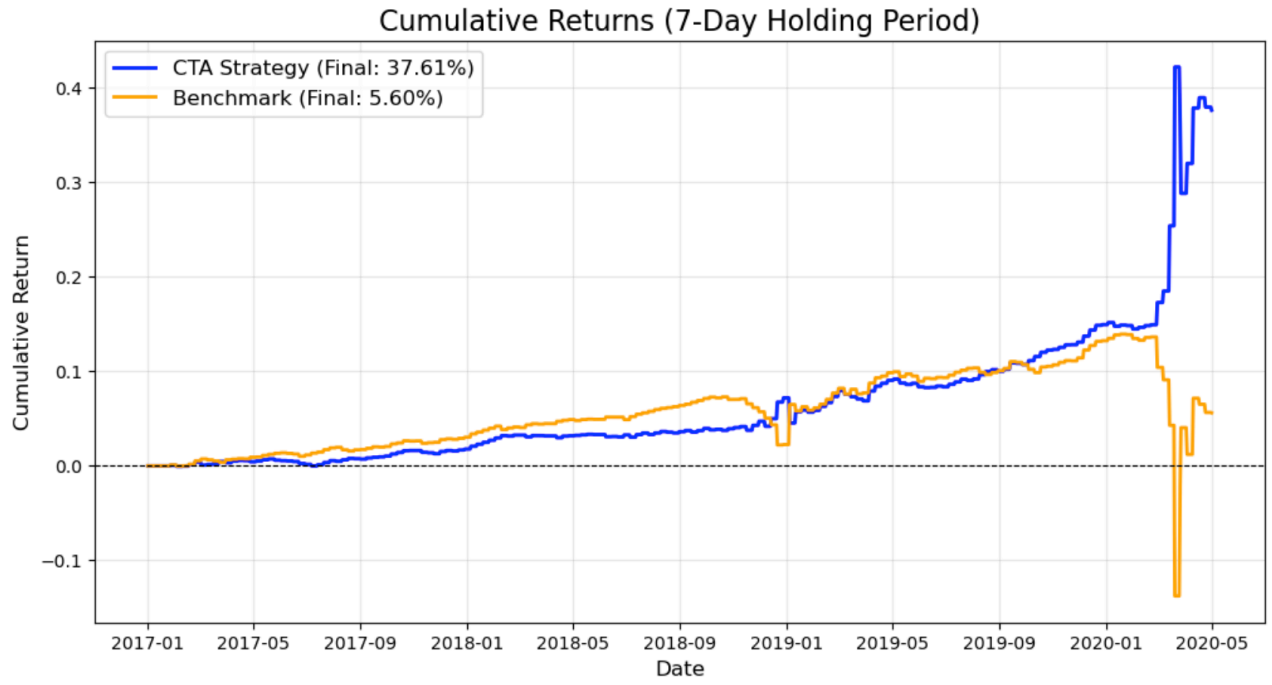
see `backtest.py` ,Benchmark is a simple buy-and-hold strategy.

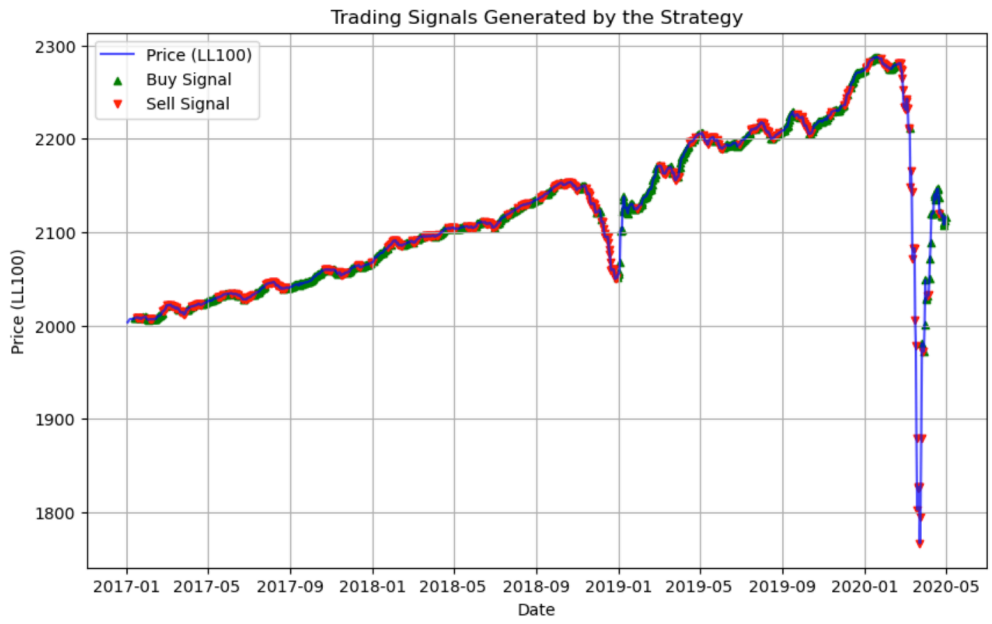
▼ `2017-01 - 2020-04` :

201701-202004	MOM_1d	MOM_7d	Benchmark
Annualized Return	0.1065	0.1055	0.0265
Sharpe Ratio	1.1351	0.8302	0.0445
Max Drawdown	-0.1168	-0.0940	-0.2432
Volatility	0.0715	0.0967	0.1374



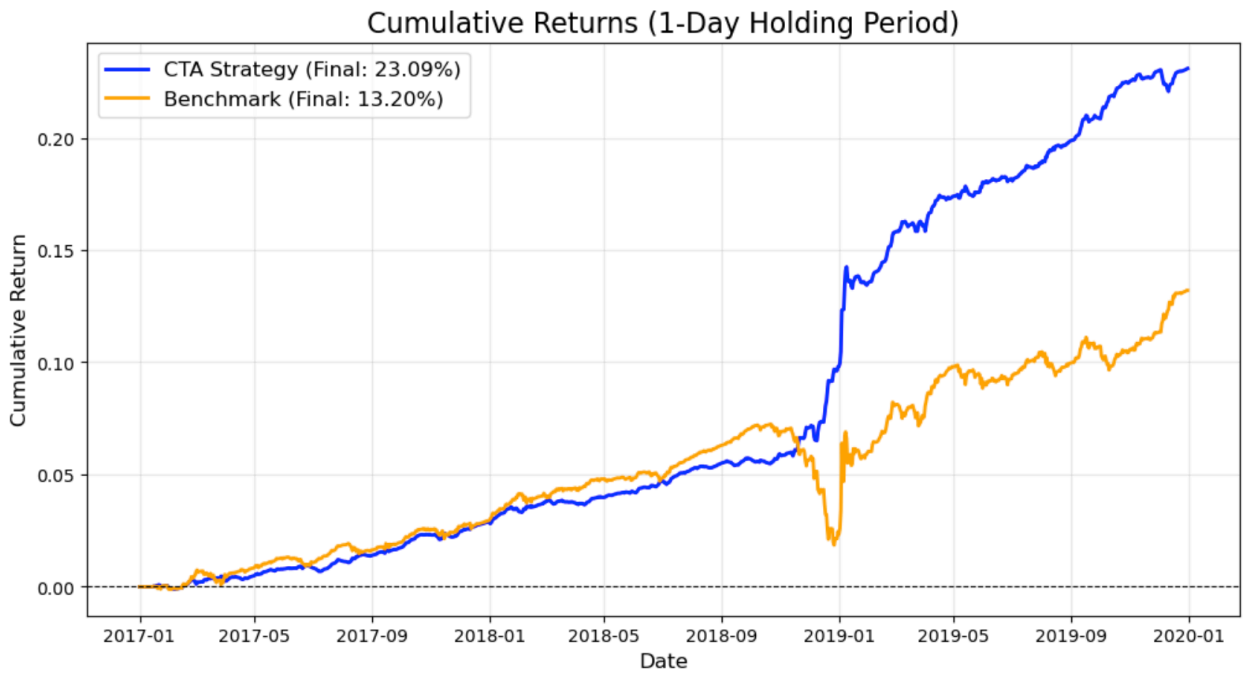


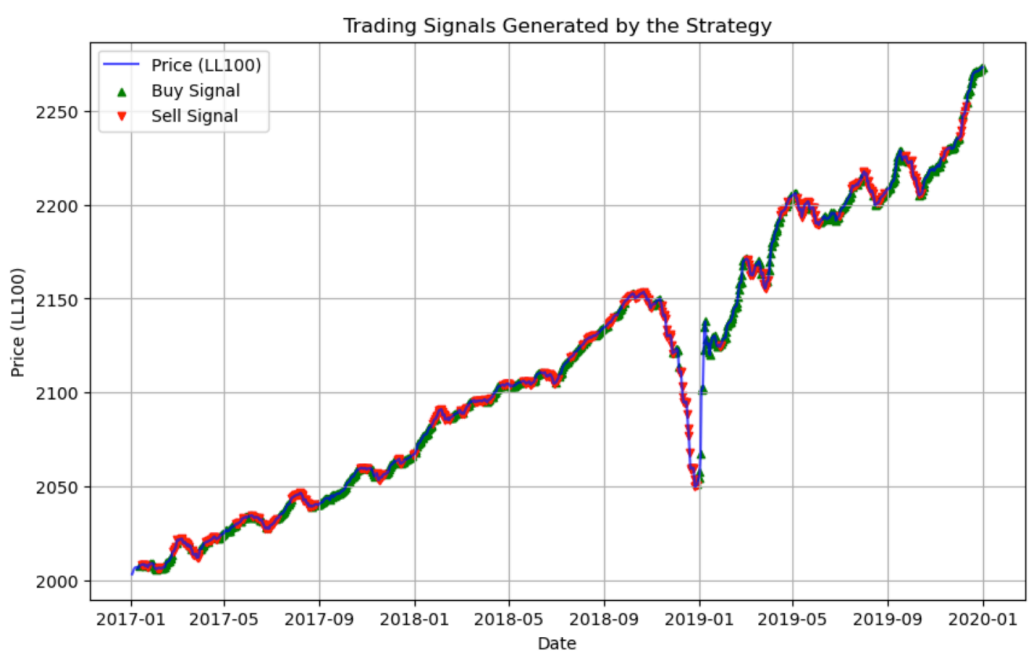




▼ 2017-01 - 2019-12 :

201701-201912	MOM_1d	Benchmark
Annualized Return	0.0718	0.0427
Sharpe Ratio	2.9833	0.7039
Max Drawdown	-0.0084	-0.0503
Volatility	0.0166	0.0310





▼ **Limitations**

1. Sideways Markets: The strategy may underperform in range-bound or choppy market conditions, where trends are weak or nonexistent.
2. Parameter Sensitivity: The performance of the strategy is sensitive to the choice of parameters (e.g., rolling window size, stop-loss percentage).
3. Transaction Cost and Slippage: While I set a fixed transaction cost rate, real-world markets often experience slippage, especially during periods of high volatility or low liquidity. This can further impact net returns and should be accounted for in the backtest.

Conclusion

- The MOM_1d strategy consistently outperformed the benchmark, achieving higher annualized returns, Sharpe ratios, and lower drawdowns and volatility.
- During 2017-2019, MOM_1d demonstrated exceptional risk-adjusted performance with a Sharpe ratio of 2.9833, far surpassing the benchmark's Sharpe ratio (0.7) with low drawdown.

▼ **5. Model Improvements**

- Additional Data Sources
  - Macroeconomic indicators (e.g., GDP, interest rates, inflation).
  - Stock prices related to LL100 index.
  - Commodity prices related to LL100 index.
  - Alternative data (e.g., news sentiment, social media trends).
- Use larger datasets with longer time periods to train state-of-the-art models.
- Incorporate cointegration analysis to inform pairs-trading strategy.

▼ Cointegration Analysis

```
''' Cointegration Test '''
from statsmodels.tsa.vector_ar.vecm import coint_johansen

variables = data[feature_names]
result = coint_johansen(variables, det_order=0, k_ar_diff=1)

'''
Johansen Cointegration Test
H0: r=0 ; r<=1; r<=2; ...; r<=11
'''
print(result.lr1)  # Trace statistic - r (number of cointegration)
print(result.cvt)  # Critical values - [90%、95%、99%]
```

```
[3.95189408e+02 2.78529126e+02 2.09988919e+02 1.63554260e+02
 1.22907502e+02 8.89812094e+01 6.18562227e+01 4.04860355e+01
 2.44366853e+01 1.12517888e+01 4.74811961e+00 3.54398118e-01]
[[326.5354 334.9795 351.215 ]
 [277.374  285.1402 300.2821]
 [232.103  239.2468 253.2526]
 [190.8714 197.3772 210.0366]
 [153.6341 159.529  171.0905]
 [120.3673 125.6185 135.9825]
 [ 91.109   95.7542 104.9637]
 [ 65.8202  69.8189  77.8202]
 [ 44.4929  47.8545  54.6815]
 [ 27.0669  29.7961  35.4628]
 [ 13.4294  15.4943  19.9349]
 [  2.7055   3.8415   6.6349]]
```

From the Johansen Cointegration test above, 8898<190.8714, indicating there’re at least 5 cointegration relationship in 12 features, suggesting possible pairs trading opportunities.

▼ Z-Score Strategy

Below use the entire dataset `2016-2020` to determine the cointegration coefficience for pair-trading strategy, which is no realistic in real life. Suppose there are previous data from before 2016, we can use the same strategy to avoid any future data leakage.

The z-score-based strategy described in the code employs a statistical arbitrage approach, focusing on trading three assets: **LL100**, **EE**, and **INFT**. The strategy is designed to exploit mean-reversion behavior in the residuals of these assets. Below is a summary of the trading logic.

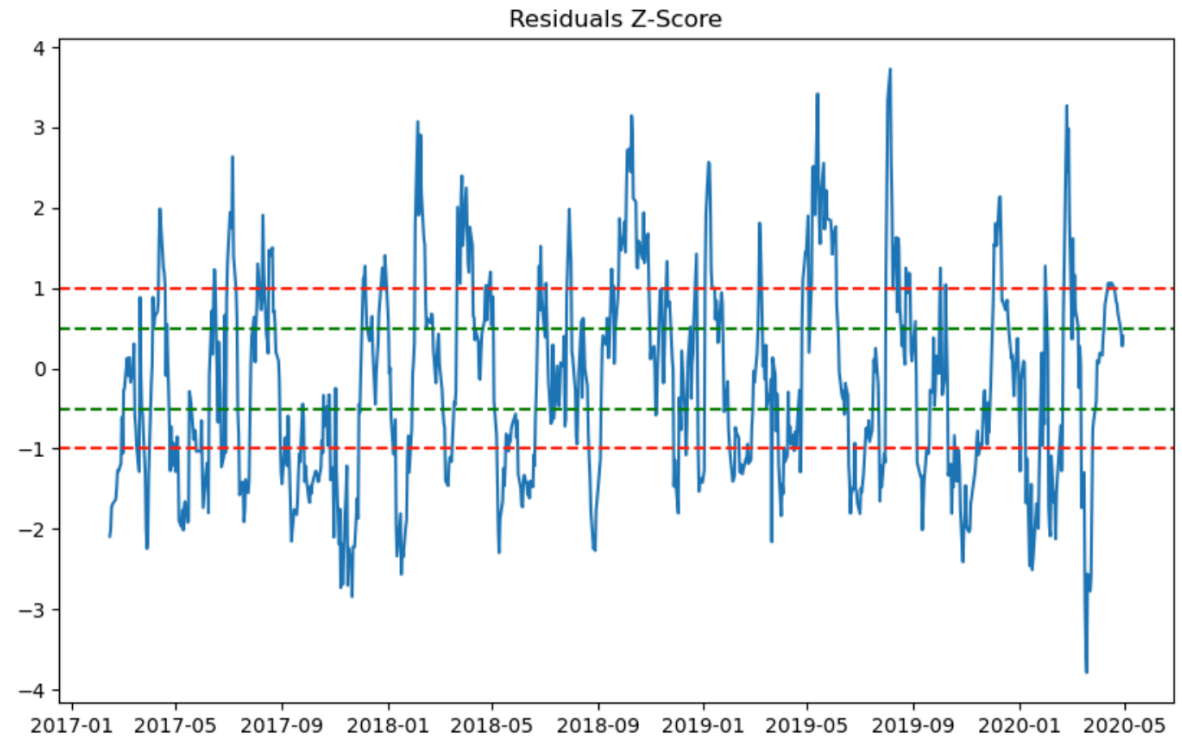
▼ **Trading Logic**

- Found significant cointegration relationships of `LL100` with `INFT` and `EE`.

```
ADF Statistic for residuals: -2.9543750819242813
p-value for residuals: 0.039380490868017495
```

- **Z-Score Calculation:**

$$z\_score = \frac{\text{residuals} - \text{rolling mean}}{\text{rolling standard deviation}}$$
$$\text{where window is } 30$$



- **Entry Conditions:**
  - **z_score > 1:**
    - Short 1 unit of LL100. Long on 0.2 units of EE and 0.2 units of INFT (proportional to the available capital).
  - **z_score < -1:**
    - Long 1 unit of LL100. Short 0.2 units of EE and 0.2 units of INFT.
- **Exit Condition:**
  - **0.5 < z_score < 0.5:**
    - Close the position when the z-score reverts to the range. At this point, calculate the pnl for all three assets. Add the total pnl to the capital.

▼ **Backtest Result**

201701-202004	Z-score_Strat	Benchmark
Annualized Return	0.0547	0.0265
Sharpe Ratio	0.7585	0.0445
Max Drawdown	-0.0296	-0.2432
Volatility	0.0438	0.1374