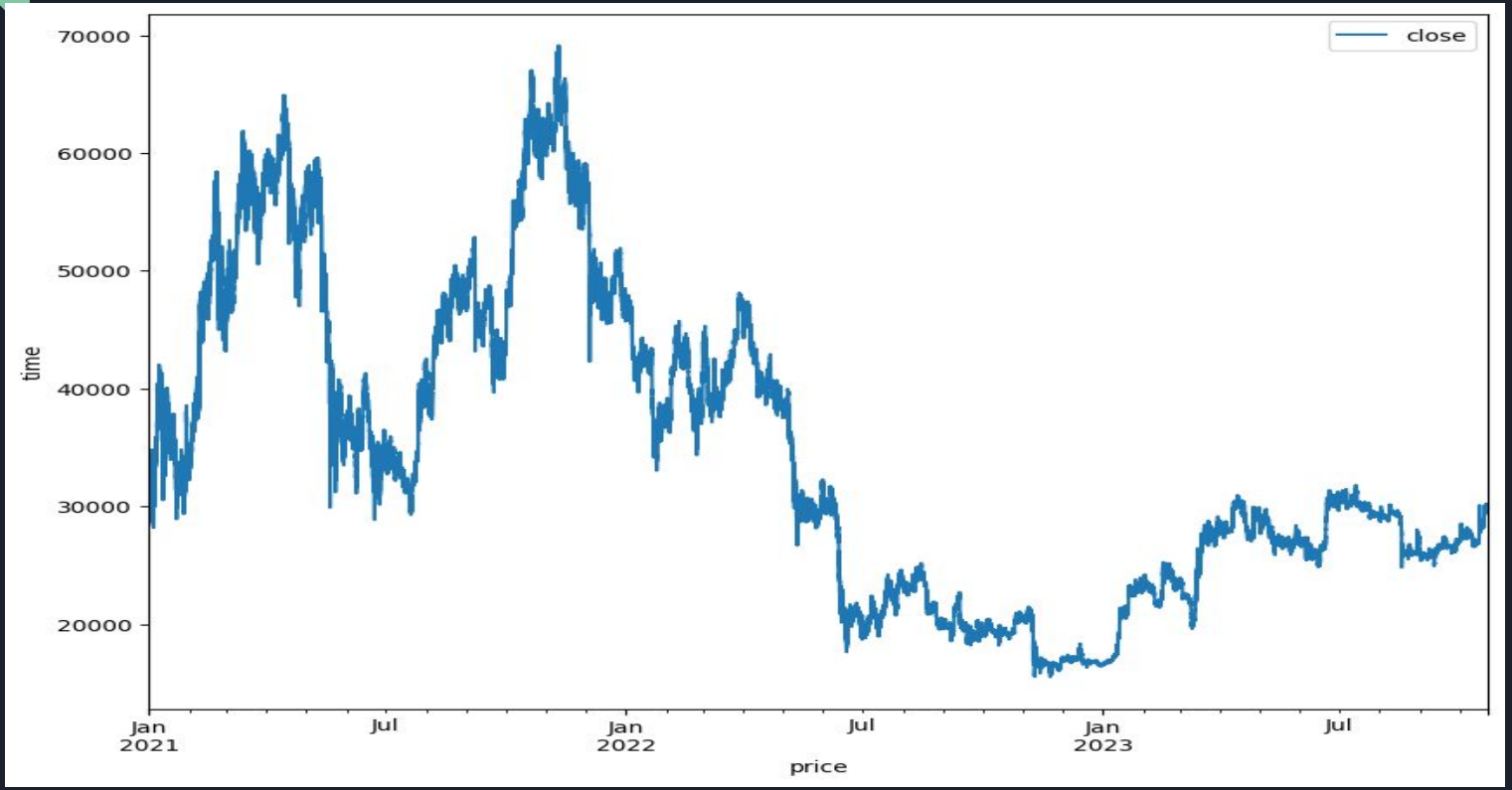
Using Price and Volume Data-based to Forecast Short-term Return in Bitcoin

by James Chiu, Johnson Hsiao
12, 22 , 2023

Q: Why doing this topic?
A: Make money easily with stats.



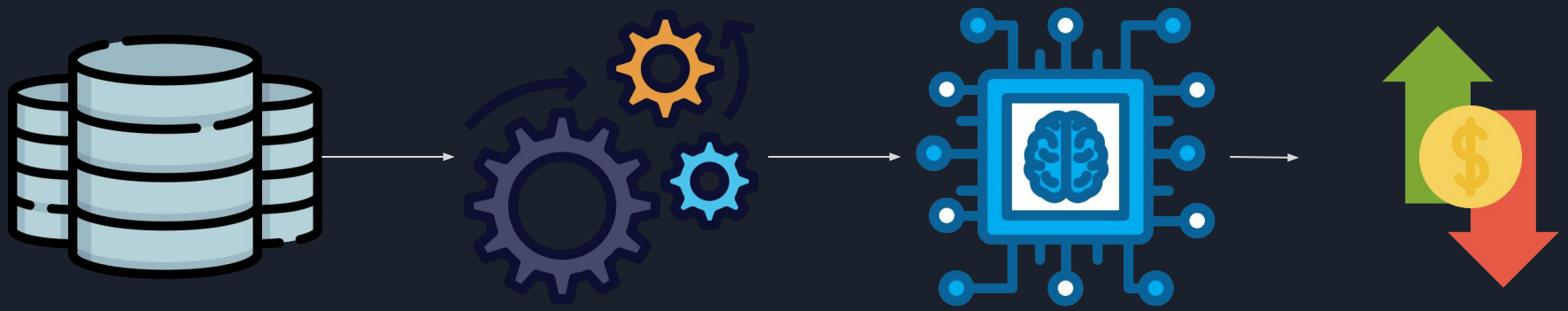
Preview and abstract

Base(Benchmark) → Long a unit evey minutes and close 1 hour later.

1. Win rate = win / total tades = 50.378%
   →It make sense because the market is the approximate 0-sum game.
2. Profit Facotr = 1.014251,
   →It means you can only earn 1.01425 dollar by losing 1 dollar in 3 years

Hence, our target is predict the correct direction of hourly return to imporove the win rate and PF.

Study Plan



price and volume data

feature engineering

ML models

price direction

DATA

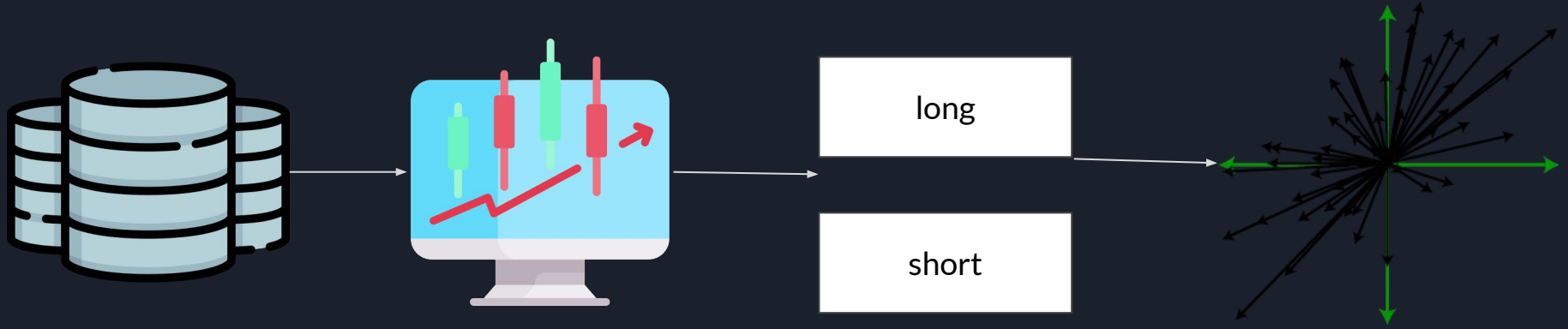
Original

Open, High, Low, Close

Indicator and Algorithm

rsi = momentum
amount_spread = volume differrence
bar_rtn_sum = momentum
volatility_0 = volatility
price_vol_corr = corr (price and volume)
zscore = rolling price change zscore
x_s_ratio = vwap / avg(rtn)
high_low_dis = high and low distance rolling sum

Feature engineering Method 1



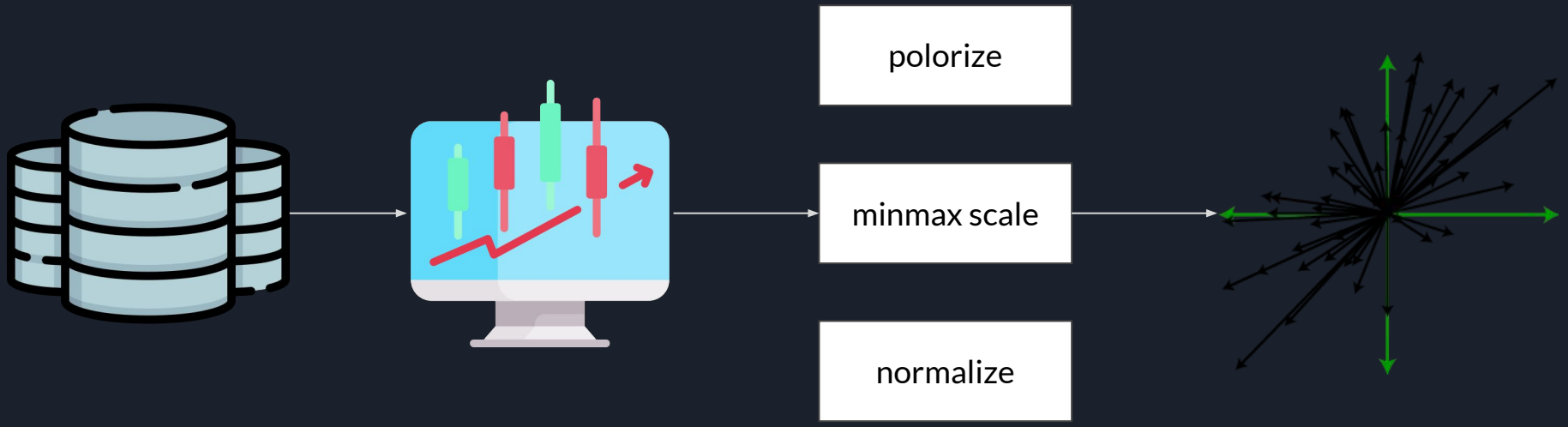
raw OHLCV data

technical indicators

Strategies

PCA

Feature engineering Method 2



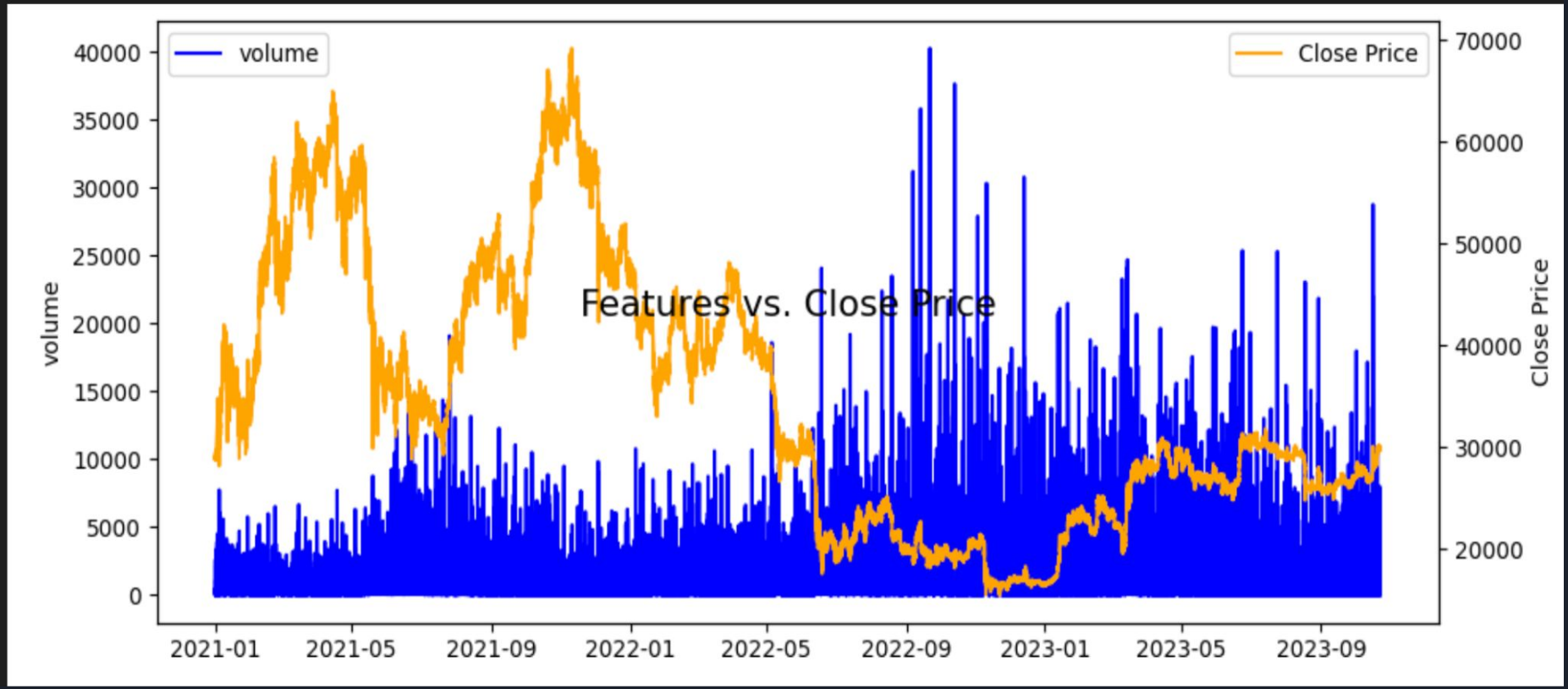
raw OHLCV data

technical indicators

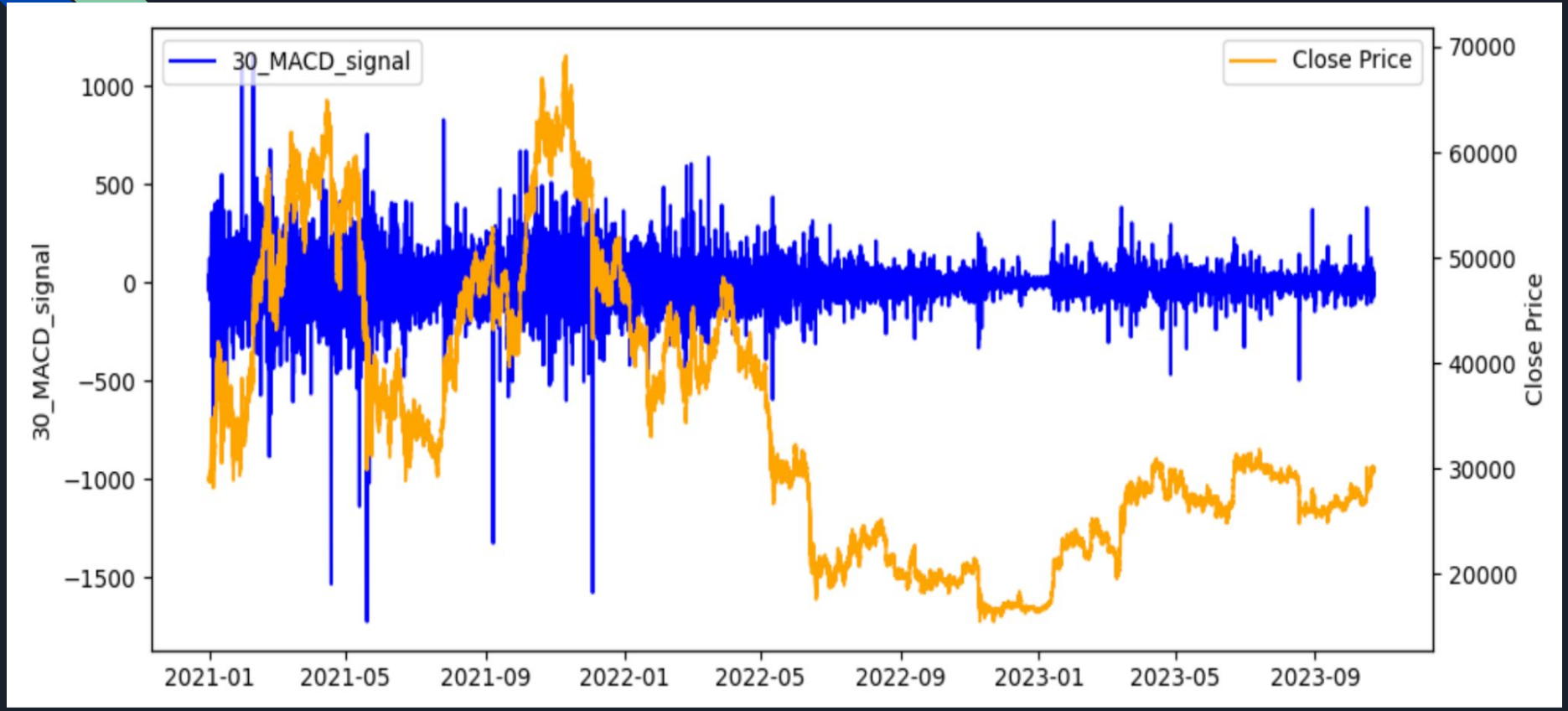
feature extension

PCA

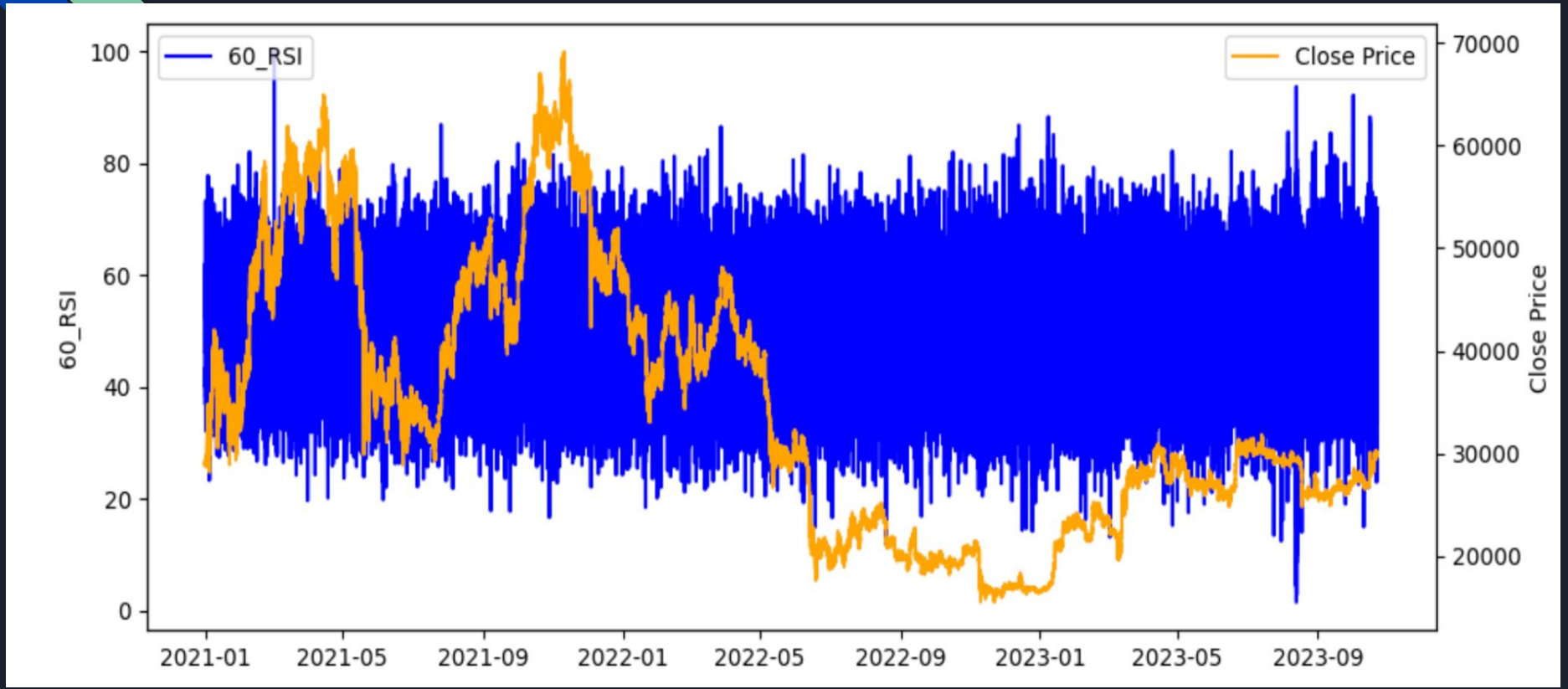
Overview


volume
Close Price
Features vs. Close Price
volume
Close Price
40000
35000
30000
25000
20000
15000
10000
5000
0
70000
60000
50000
40000
30000
20000
2021-01
2021-05
2021-09
2022-01
2022-05
2022-09
2023-01
2023-05
2023-09

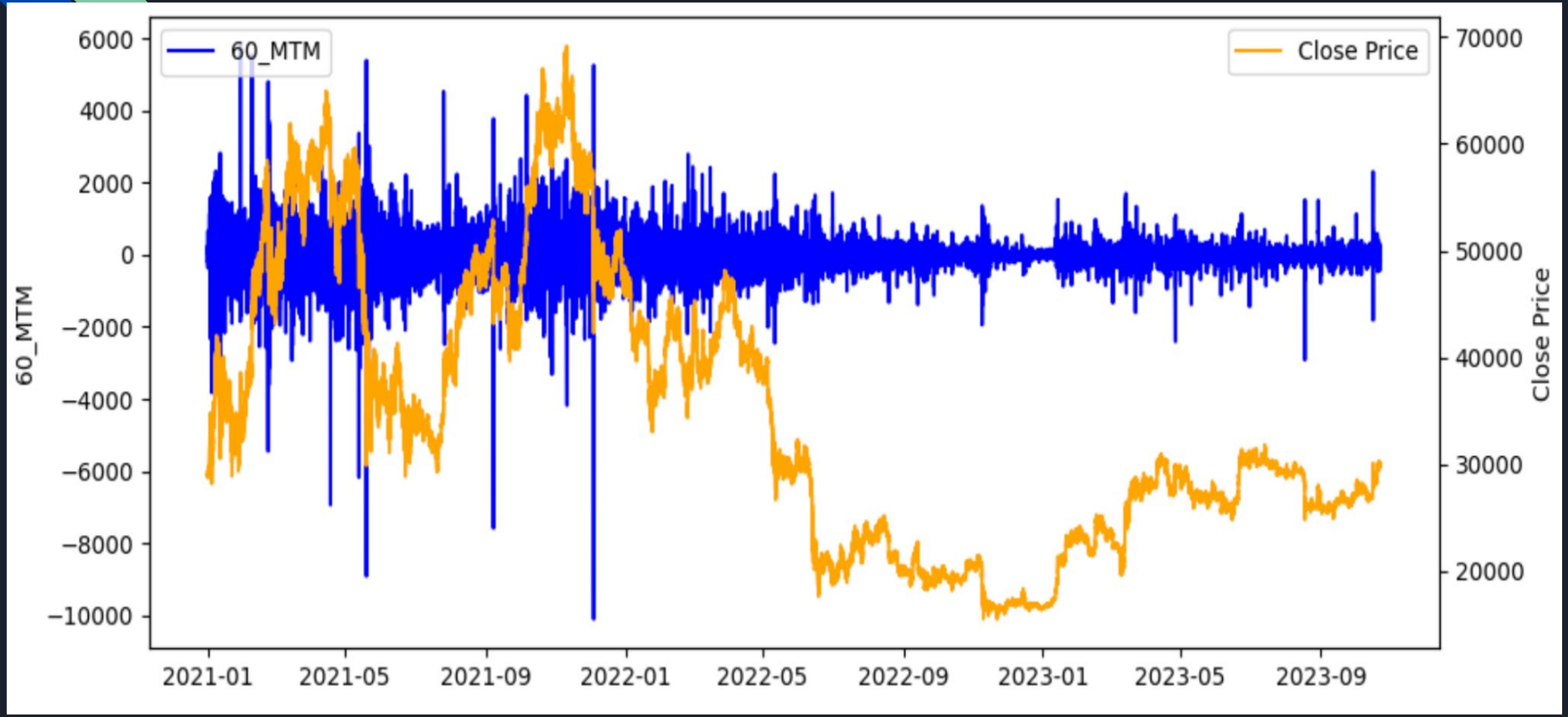
Overview


30_MACD_signal
Close Price
1000
500
0
−500
−1000
−1500
70000
60000
50000
40000
30000
20000
2021-01
2021-05
2021-09
2022-01
2022-05
2022-09
2023-01
2023-05
2023-09

Overview

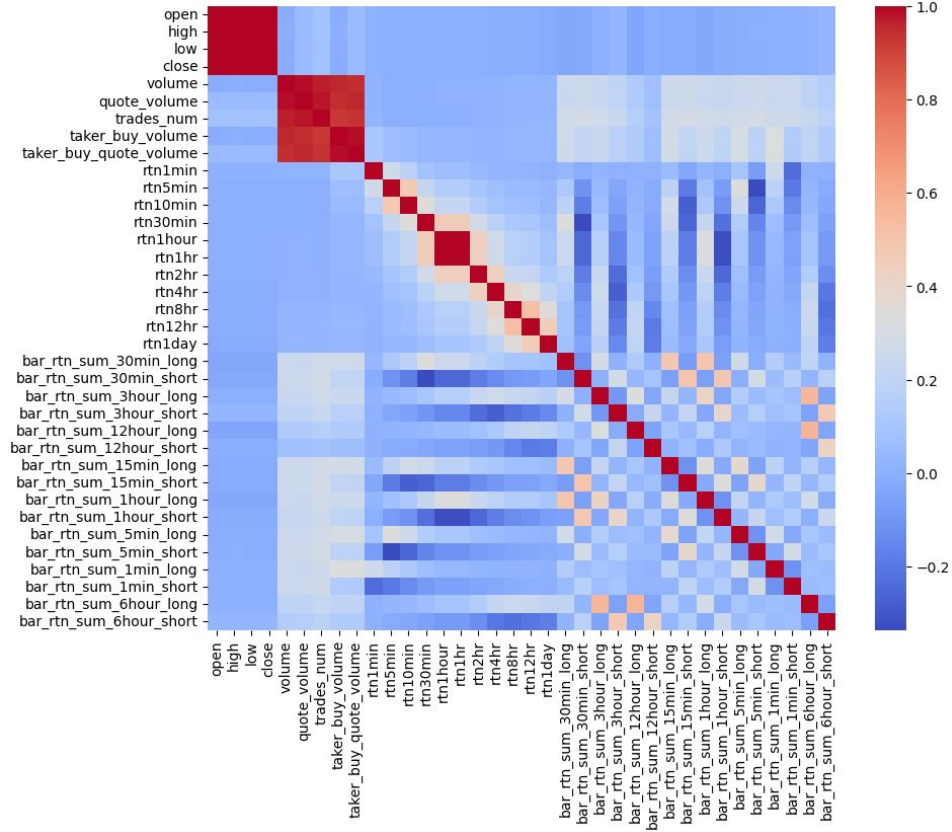

60_RSI
Close Price
100
80
60
40
20
0
70000
60000
50000
40000
30000
20000
2021-01
2021-05
2021-09
2022-01
2022-05
2022-09
2023-01
2023-05
2023-09

Overview

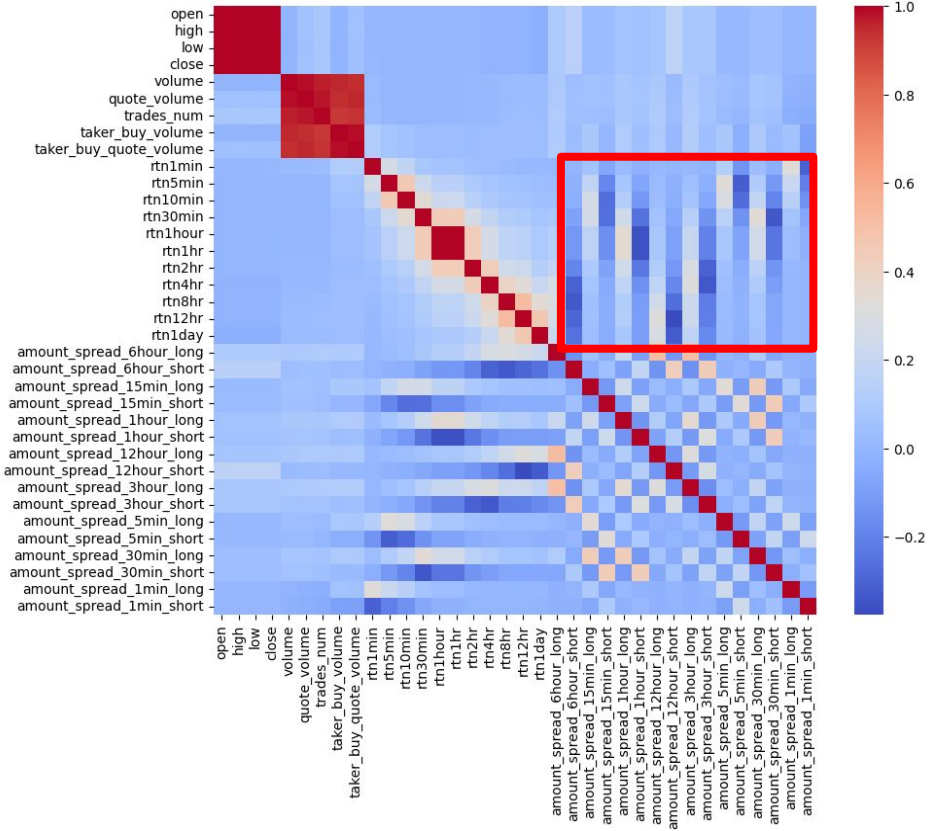

60_MTM
Close Price
6000
4000
2000
0
−2000
−4000
−6000
−8000
−10000
70000
60000
50000
40000
30000
20000
2021-01
2021-05
2021-09
2022-01
2022-05
2022-09
2023-01
2023-05
2023-09

Feature

RSI

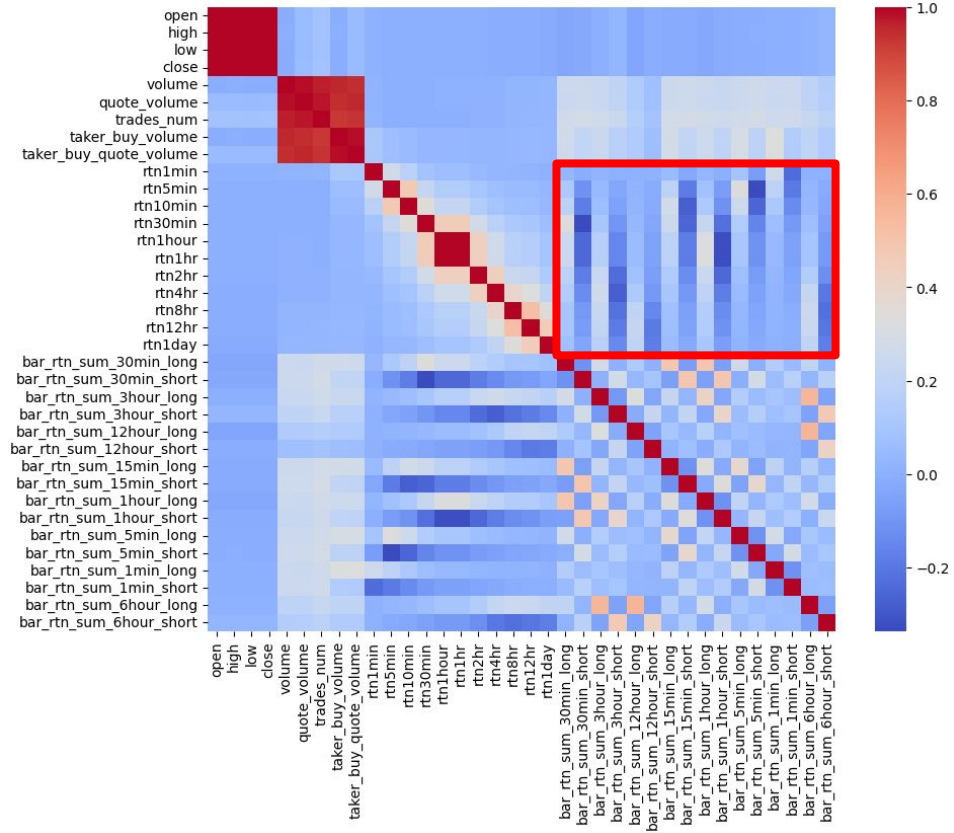


amount_spread

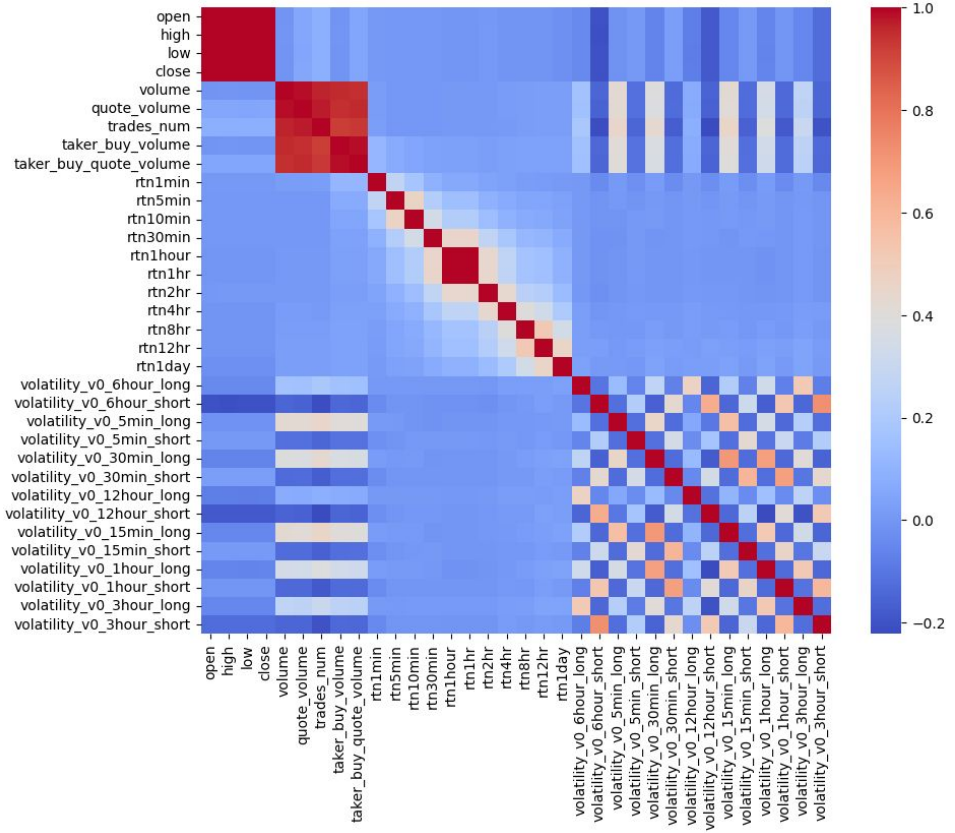


Feature

Kbar_rtn_sum

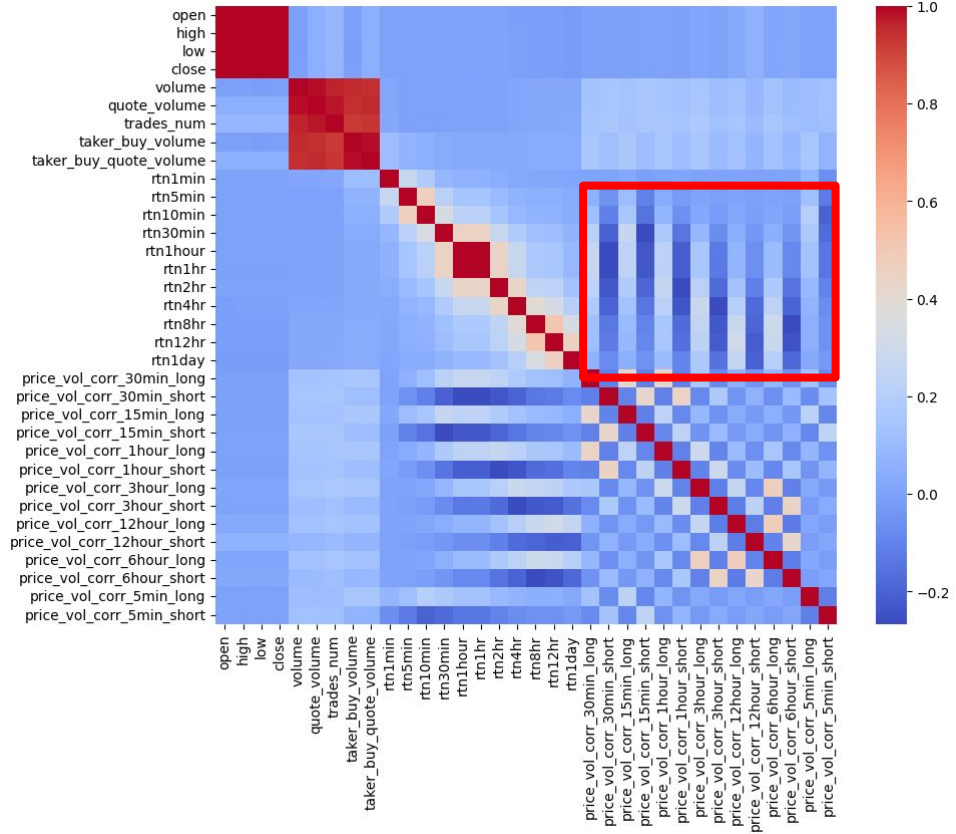


Volatility

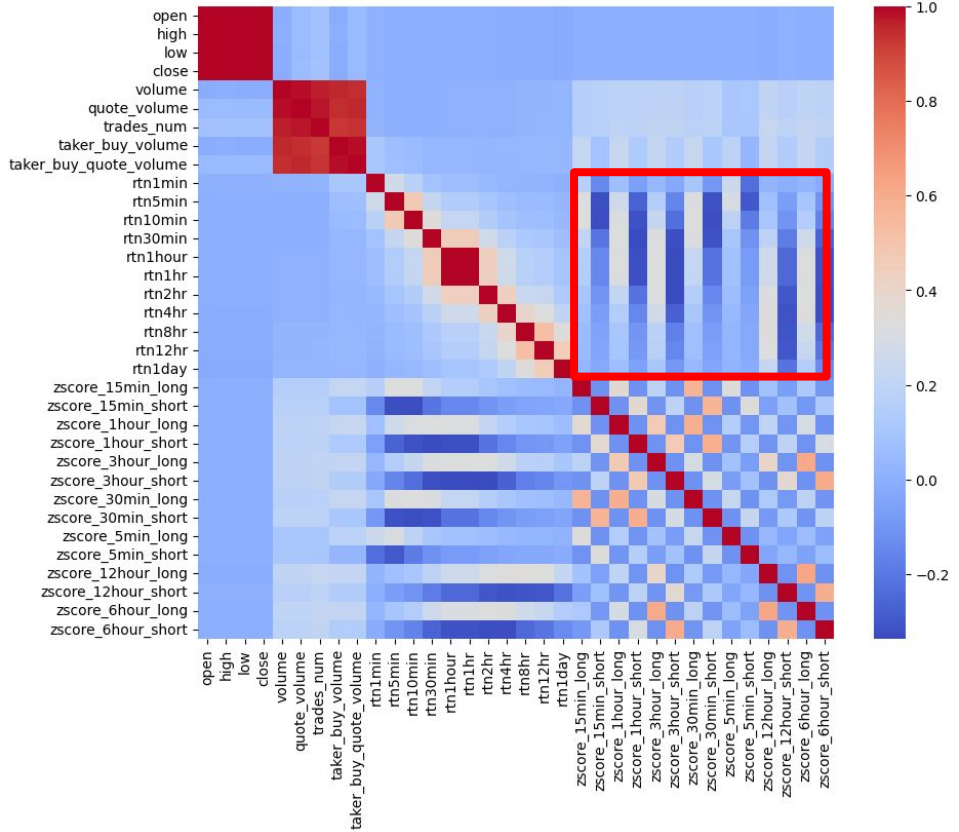


Feature

Price_vol_corr

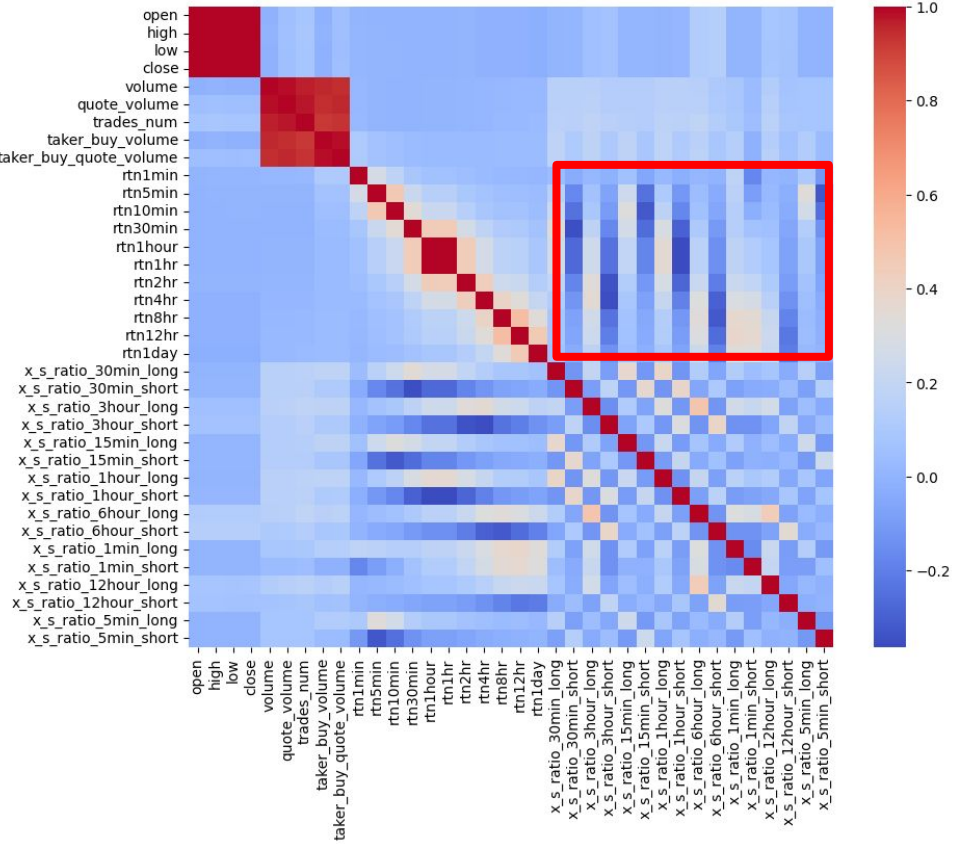


Zscore

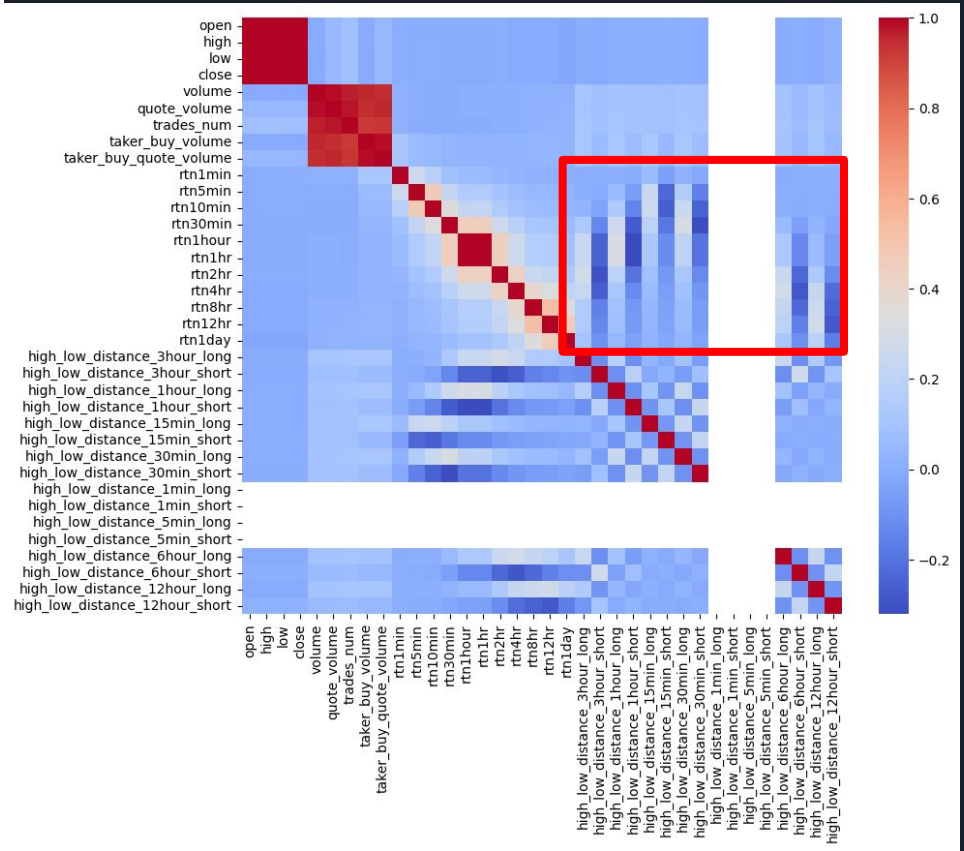


Feature

x_s_ratio



high_low_dis



ML Model

Logistic Regression

Functionality:
Models the probability that an instance belongs to a particular category.

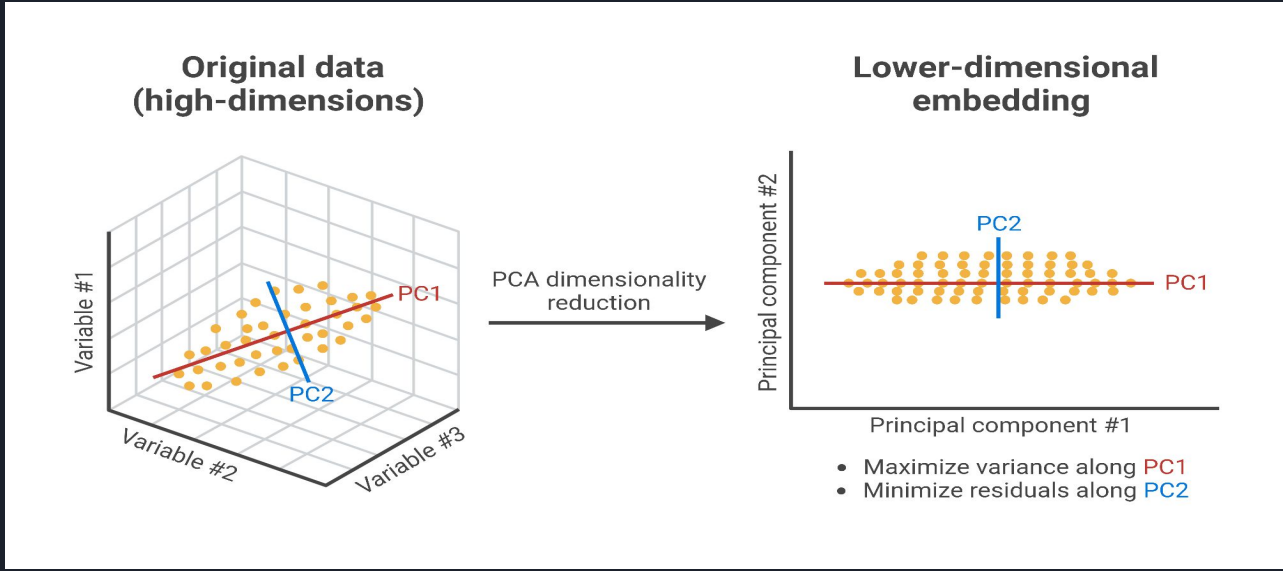
Random Forest

Functionality:
Constructs a multitude of decision trees during training and outputs the mode of the classes or mean prediction of the individual trees.

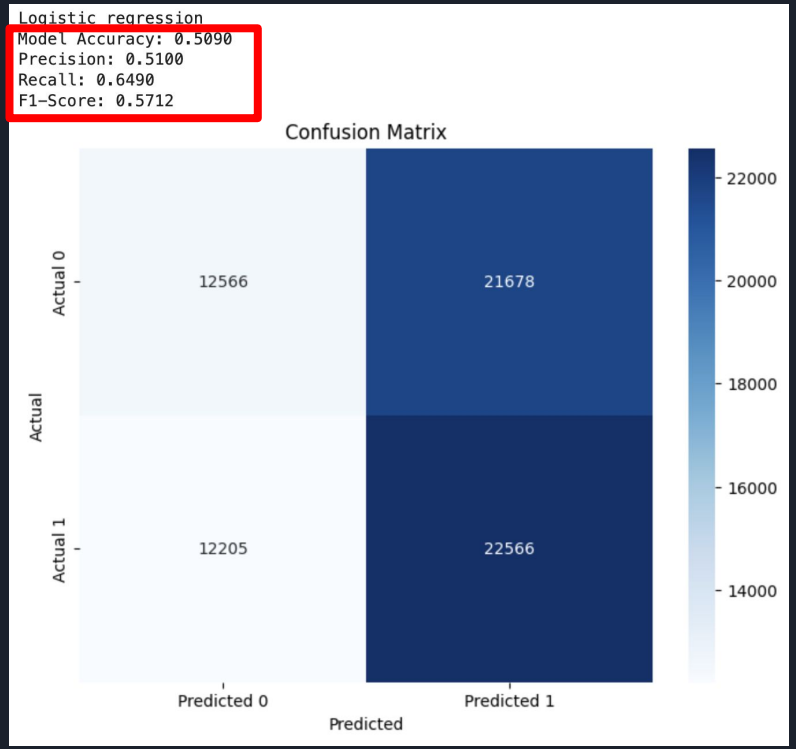
XGBoost

Functionality:
Builds a series of weak learners (usually decision trees) sequentially, where each new learner corrects the errors made by the previous ones.

Principal Component Analysis

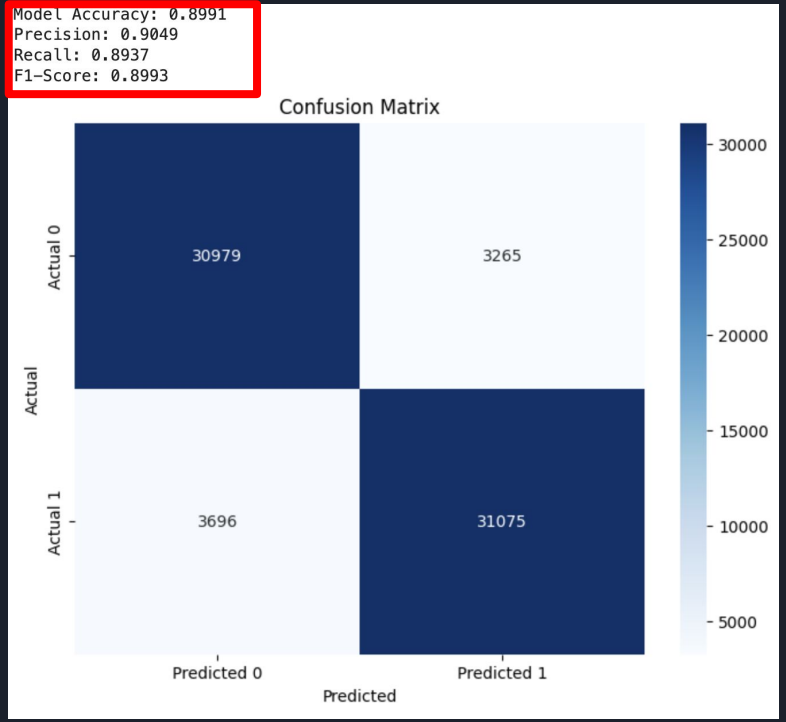


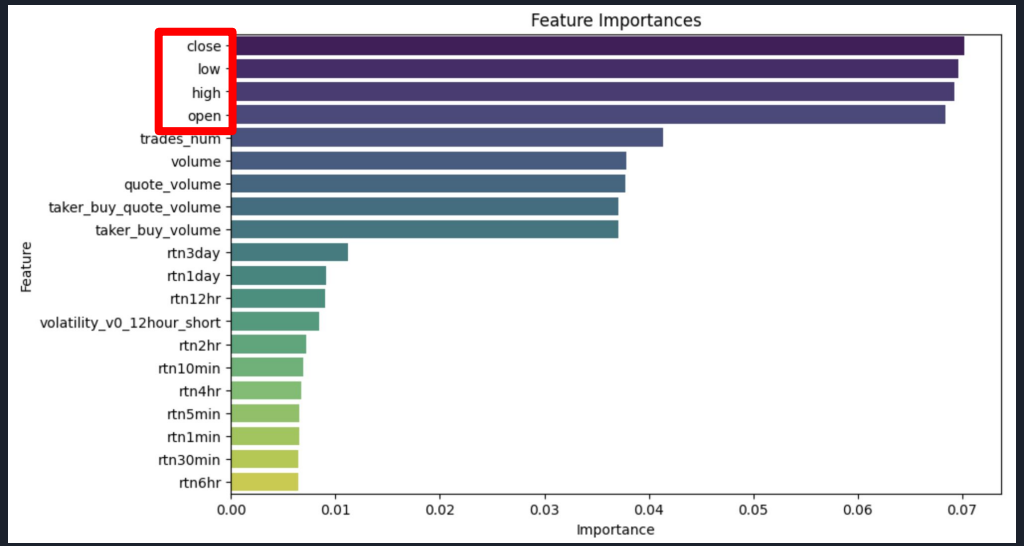
Logistic Regression



Random Forest

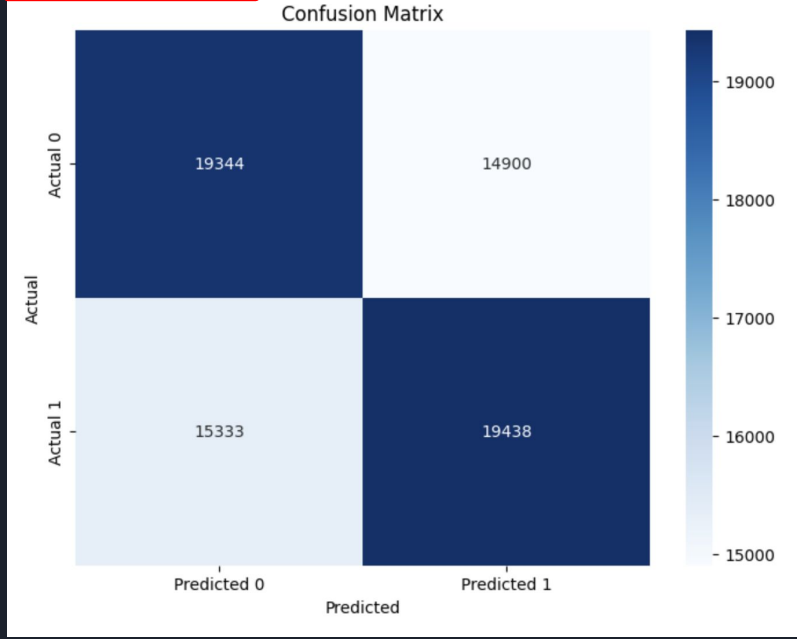
```
Model Accuracy: 0.8991
Precision: 0.9049
Recall: 0.8937
F1-Score: 0.8993
```





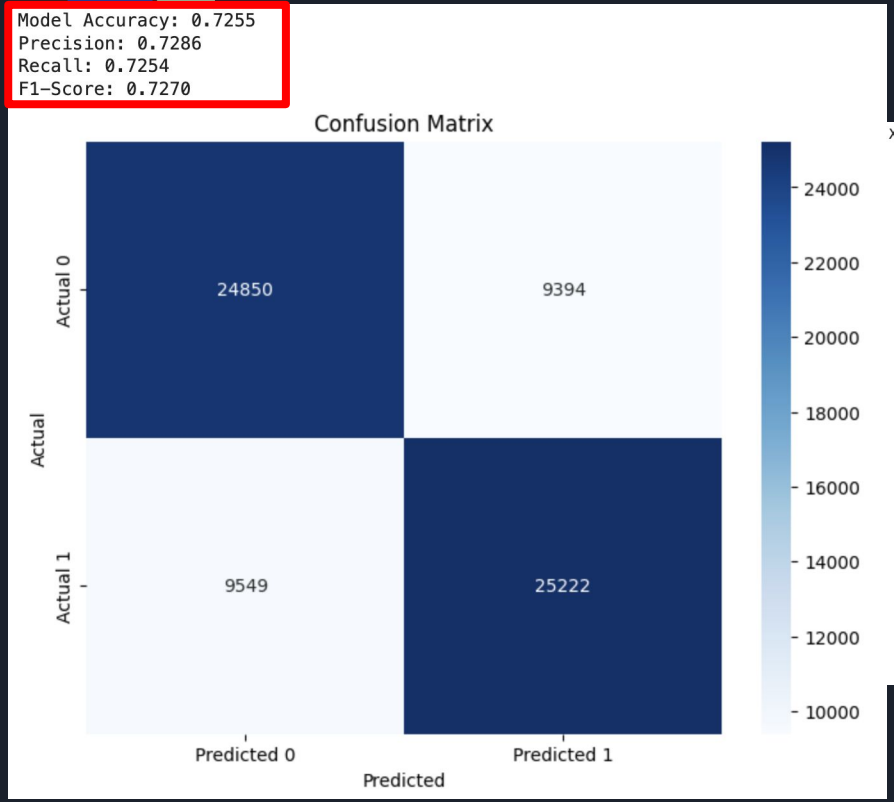
Random Forest (only use OHLC)

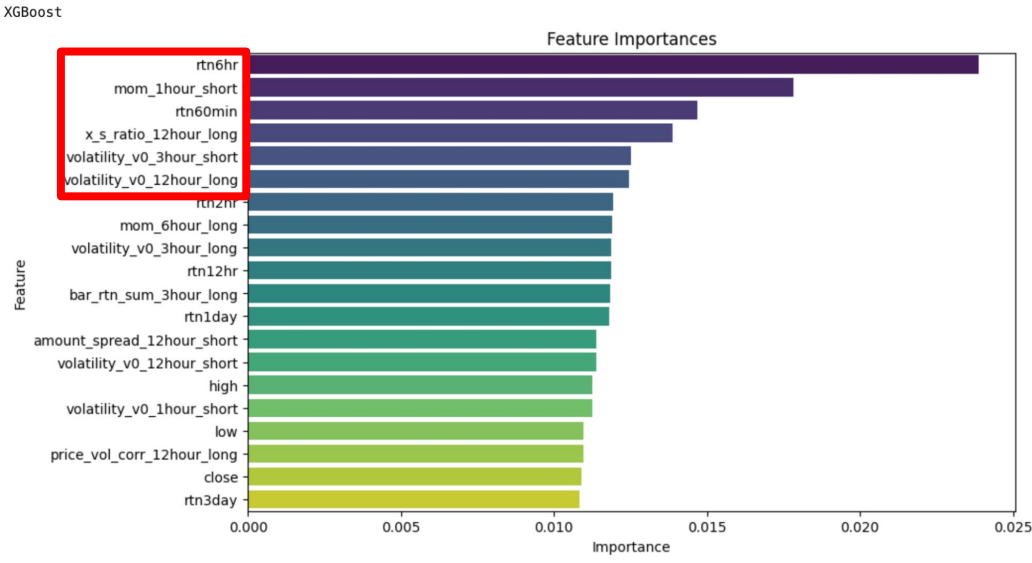
```
Index(['open', 'high', 'low', 'close'], dtype='object')
random_forest
Model Accuracy: 0.5619
Precision: 0.5661
Recall: 0.5590
F1-Score: 0.5625
```



XGBoost

```
Model Accuracy: 0.7255
Precision: 0.7286
Recall: 0.7254
F1-Score: 0.7270
```

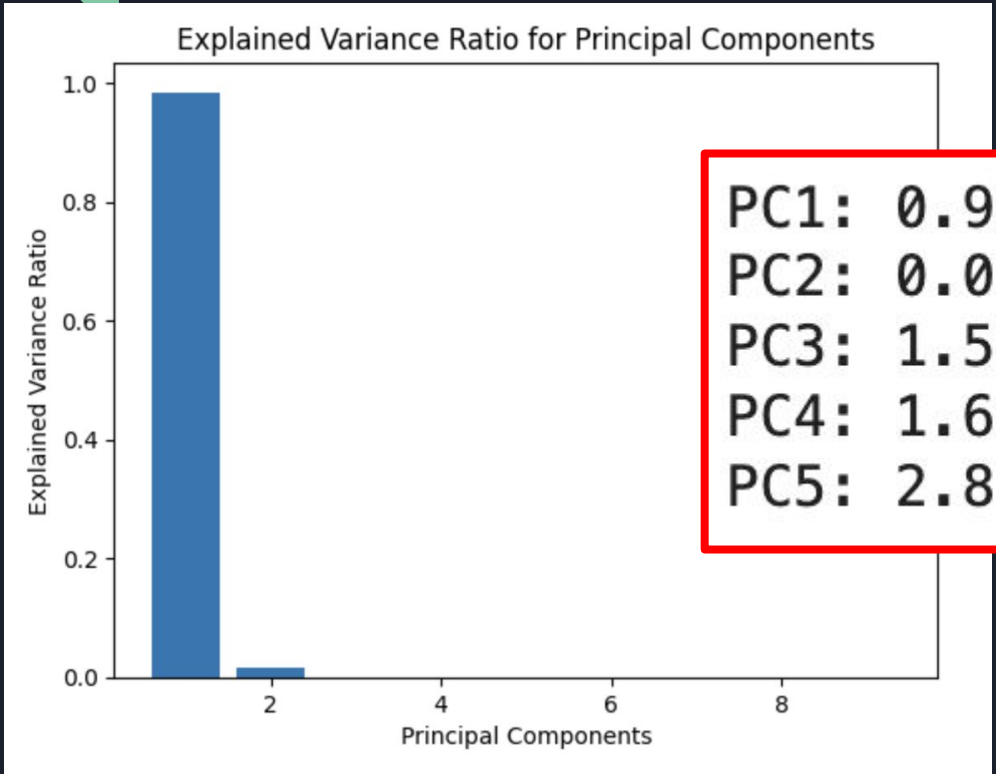




Result (Without PCA)

	Accuracy	Precision	Recall	F1 score
Logistic Regression	0.509	0.5100	0.6490	0.5712
Random Forest	0.8991	0.9049	0.8937	0.8993
XGBoost	0.7255	0.7286	0.7254	0.7270

PCA



PC1: 0.9847721605154578
PC2: 0.015227678278119187
PC3: 1.595094682428055e-07
PC4: 1.6674910168497199e-09
PC5: 2.8728703150190032e-11

Choosing N

	Logistic Regression	Random Forest	XGBoost
10 principal component	0.5067	0.5532	0.5563
50 principal component	0.5067	0.6952	0.6263
100 principal component	0.5067	0.7286	0.6365
158(all) principal component	0.5067	0.7584	0.6608

Result (With PCA)

	Accuracy	Precision	Recall	F1 score
Logistic Regression	0.5067	0.5165	0.3282	0.4014
Random Forest	0.8991	0.9049	0.78937	0.8993
XGBoost	0.6608	0.6643	0.6608	0.6625

Result

Strategy: If predict is 1, long the position and close it 1 hour later.
Win rate is improved from 50% to 69.35%
Profit Facotr is improved from 1.01 to 2.996
Which is the more profitable strategy by Machine Learning.

$$PF = \frac{avg(profit\ per\ trades)}{avg(loss\ per\ trades)} = 2.99615391136621$$

Win rate = win / total tades = 0.693475