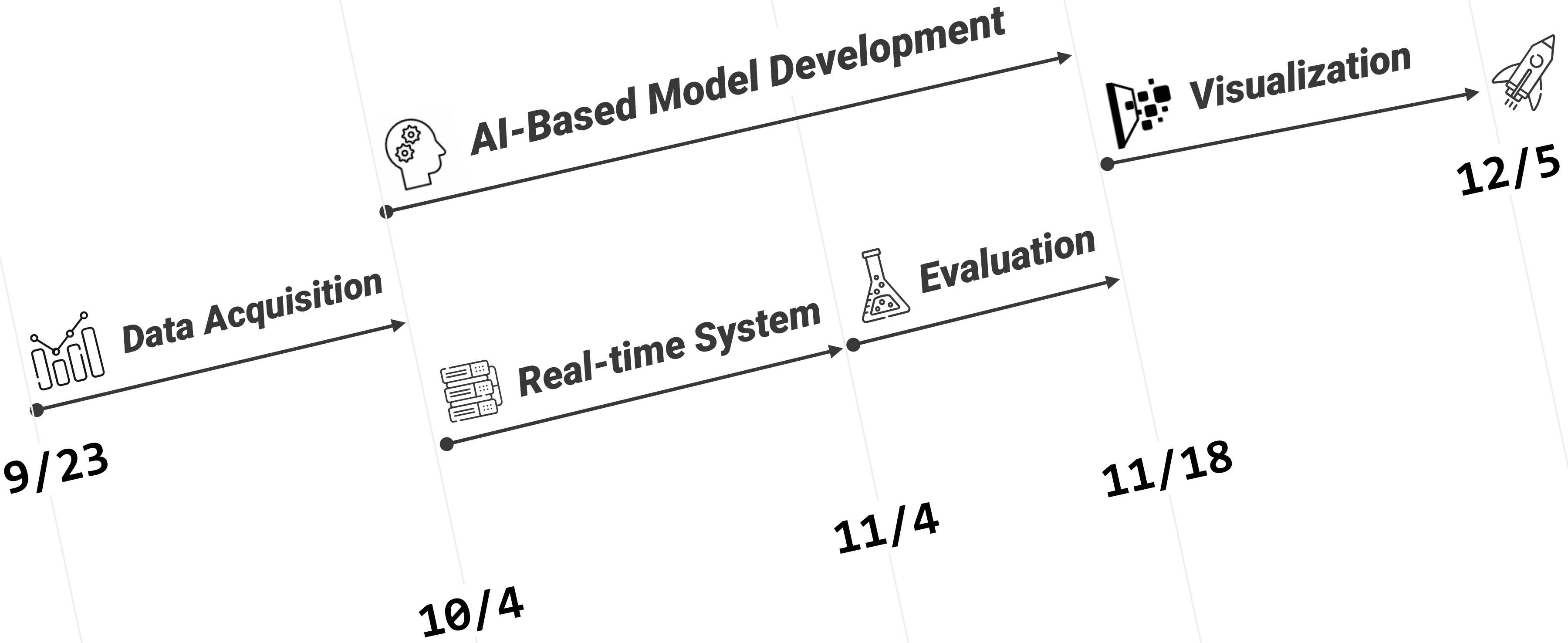
Timeline



Data Acquisition

AI-Based Model Development

Real-time System

Evaluation

Visualization

9/23

10/4

11/4

11/18

12/5

1. Subdivision of transaction situation (Heuristic approach)

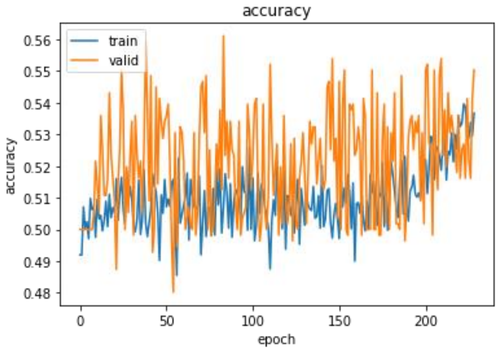
- Heuristic approach → **60.5%** Precision | Expected Annual Yield: **54%**

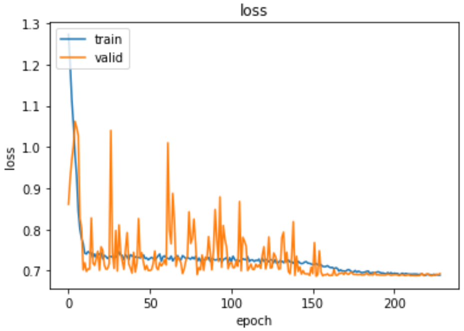
❖ Not enough data (Total 5867 data)

❖ Unreliable result

❖ To get a profitable data distribution

❖ Which has a high probability of yield case





➔ We need more data !

➔ Data should contain specific patterns.

1. **Subdivision of transaction situation** (problem of Heuristic approach)

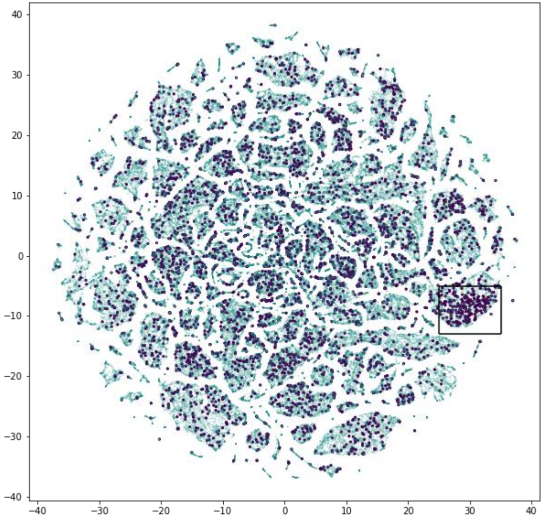
- Heuristic approach →
- Pattern clustering using Auto-Encoder

Basic Idea

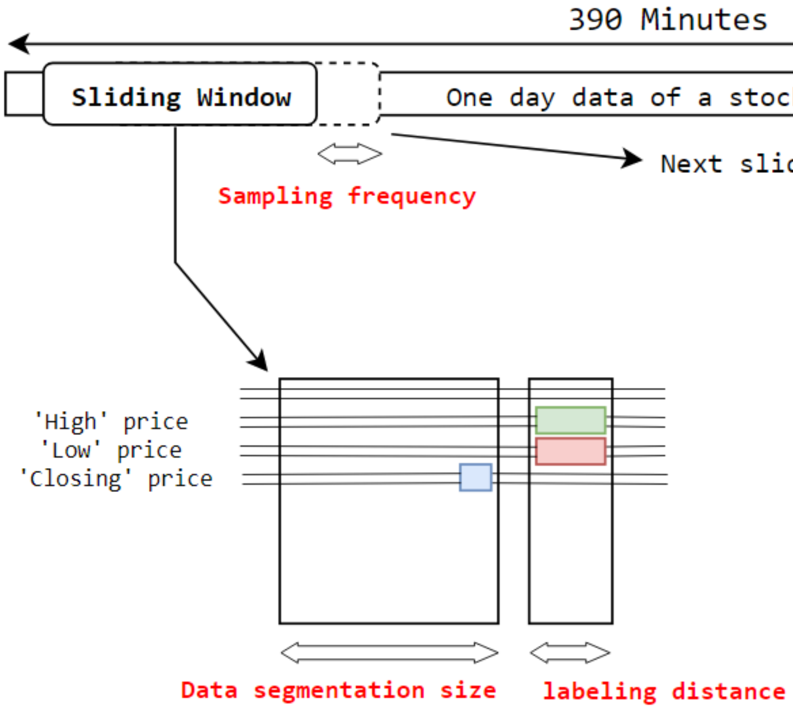
❖ figure out specific clusters where many yield cases exist

➔ Expected to get **much more data** for model training

➔ Data would contain **specific patterns** within its cluster



1. Subdivision of transaction situation (Data Preprocessing - Label)



Compare between
the **last closing price** 🟦
and **Low price in future** 🟥
and **High price in future** 🟩

In labeling distance,

1. last closing price * (1 + **labeling threshold**) < High price in future
>>> Labeled as '**profit**'

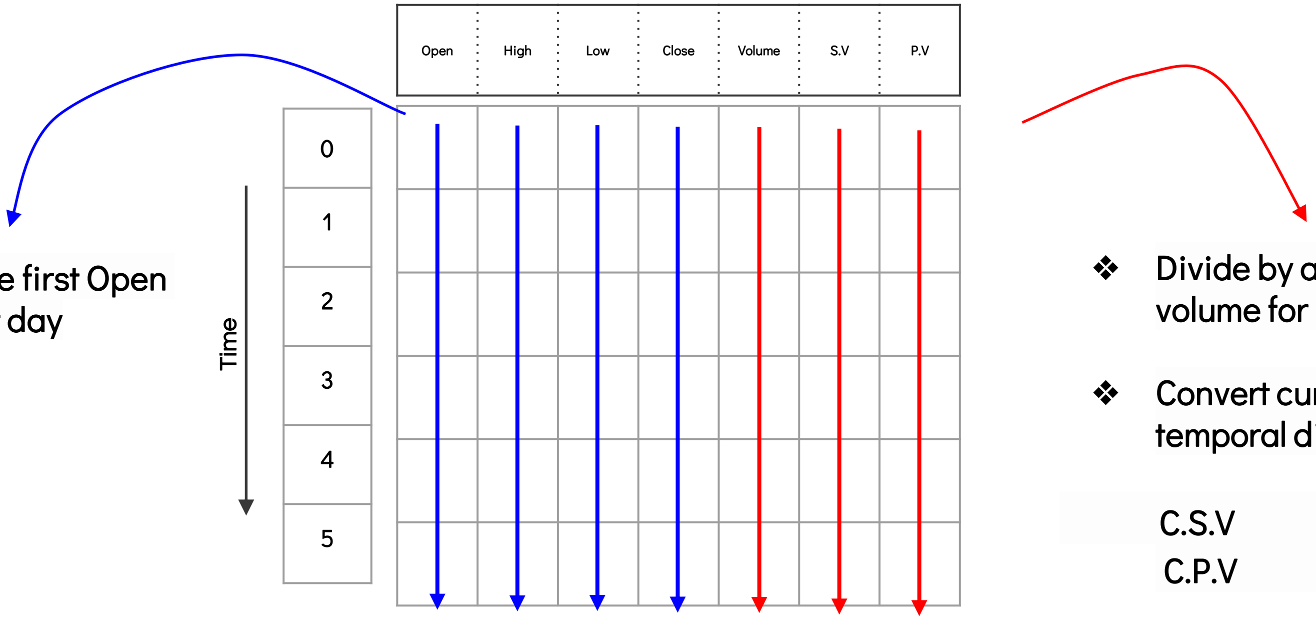
2. last closing price * (1 - **labeling threshold**) > Low price in future
>>> Labeled as '**loss**'

3. Neither 1 & 2
>>> Labeled as '**oscillation**'

❖ Four Parameters for preprocessing :
Sampling frequency, Data Segmentation size, Labeling Distance, Labeling Threshold

We use (2), (6), (6), (0.0075 - 0.75%)

1. **Subdivision of transaction situation** (Data Preprocessing - Scaling)



- ❖ Divide by the first Open price of that day

- ❖ Divide by average trading volume for recent 5 days.

- ❖ Convert cumulative data to temporal difference data

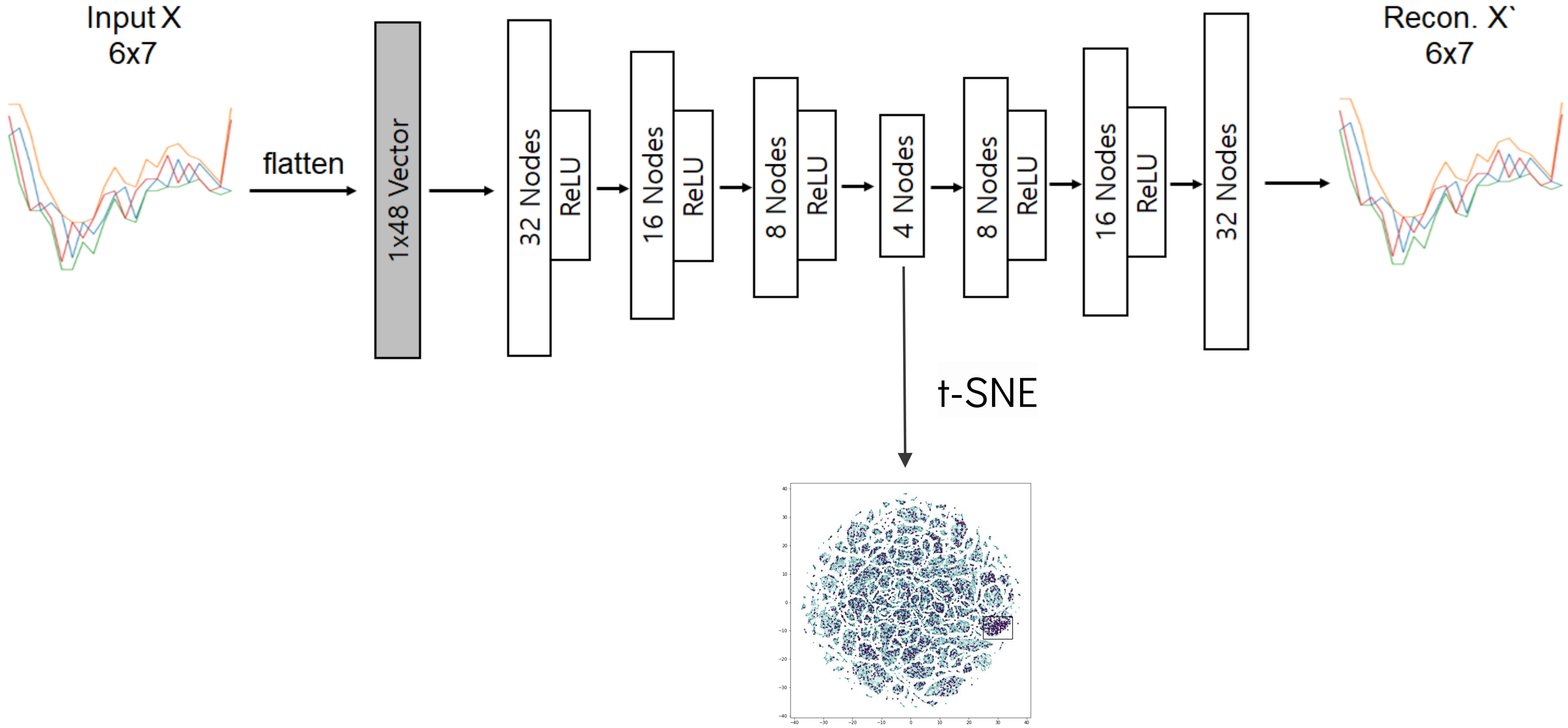
C.S.V > Δ C.S.V

C.P.V > Δ C.P.V

- ❖ S.V : Sales Volume
- ❖ P.V : Purchases Volume
- ❖ C.S.V : Cumulative Sales Volume
- ❖ C.P.V : Cumulative Purchases Volume

1. Subdivision of transaction situation (Pattern clustering using Auto-Encoder)

❖ Train AutoEncoder

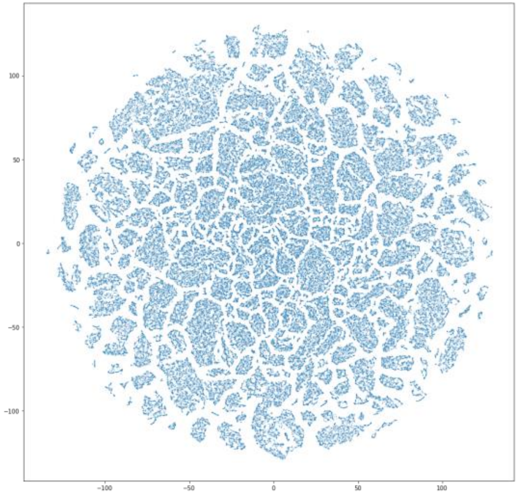


1. Subdivision of transaction situation (Pattern clustering using Auto-Encoder)

❖ Train an Auto-encoder model

- Train data : 30,000,000 X 6 X 7
  Validation data : 3,800,000 X 6 X 7

- t-SNE (5,000 iterations) with 3,800,000 latents vector (from validation data)



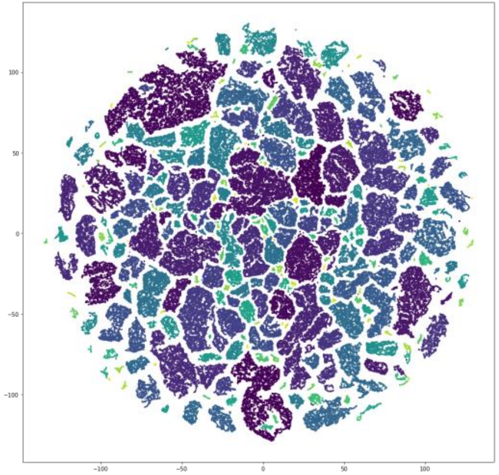
➢ t-SNE on latent vectors from trained Encoder

1. **Subdivision of transaction situation** (Pattern clustering using Auto-Encoder)

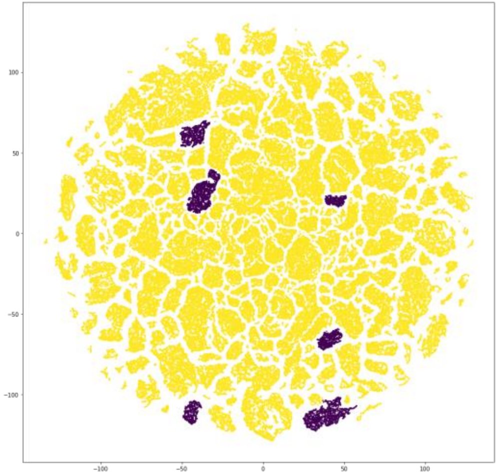
- Using DBSCAN, 300 clusters found.
  Time and Space complexity : **O(n)**

➔ **Suitable Algorithm for Large data !**

- Sort all clusters in order of size and yield data density whose sizes are over 1,000 minimally.



➢ Result of clustering (301 Clusters total)



➢ Selected clusters for training **trader** model.

Contributor : Lee Minseung

1. **Subdivision of transaction situation** (Problem of using Autoencoder)

❑ Takes too much time and efforts

1. Training model 100 epochs **(4 Hours)**
2. Get latent vectors from validation dataset
3. t-SNE over **10%** of latent vectors for 5000 iterations **(5 Hours)**
   - ❖ Memory Issue
1. Cluster 2D Data (result of t-SNE) using DBSCAN

❑ Not much effective

- ❖ Why we used it
  1. Find out data cluster with high density of success case
  2. With relatively enough data

	N (Data)	Loss case ratio	Yield case ratio	Else
Before	380,000	5.2 %	4.6 %	90.1 %
After	20,000 (- 94.7%)	8.5 % (+ 63 %)	8.7 % (+ 89 %)	82.8 % (- 8.1 %)

1. **Subdivision of transaction situation** (Much simpler way)

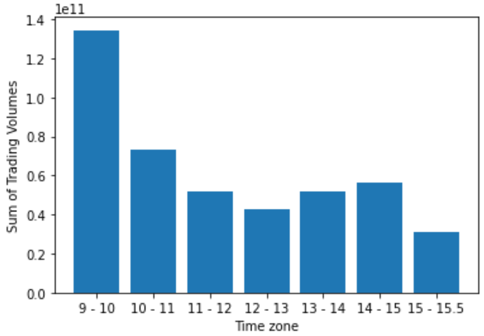
❑ Still most important factors

1. **Data Amount**
2. **Data distribution advantageous for model training**

➢ Choose **the time zone with the highest trading volume** during the day.

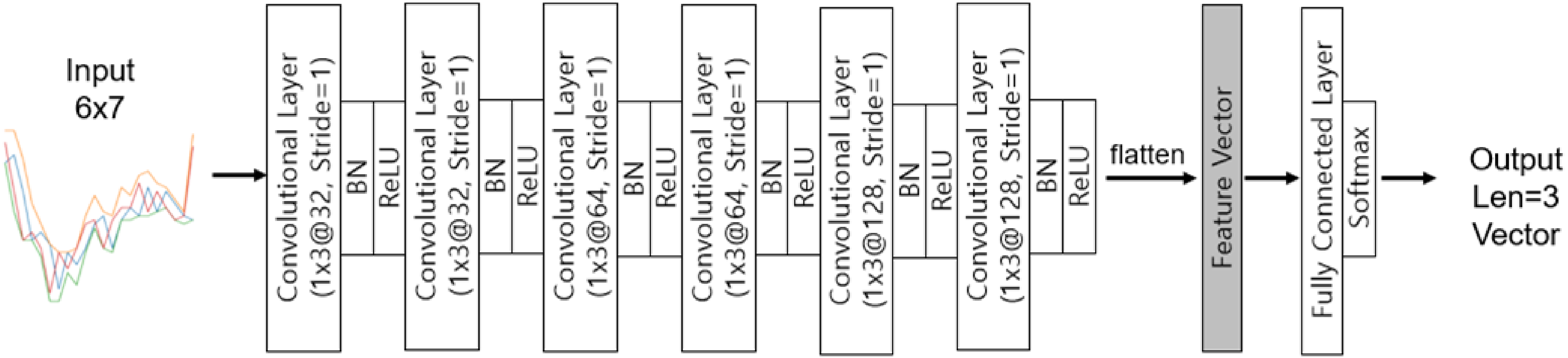
(09:01 AM - 10:00 AM)

We can deal with two factors at the same time!



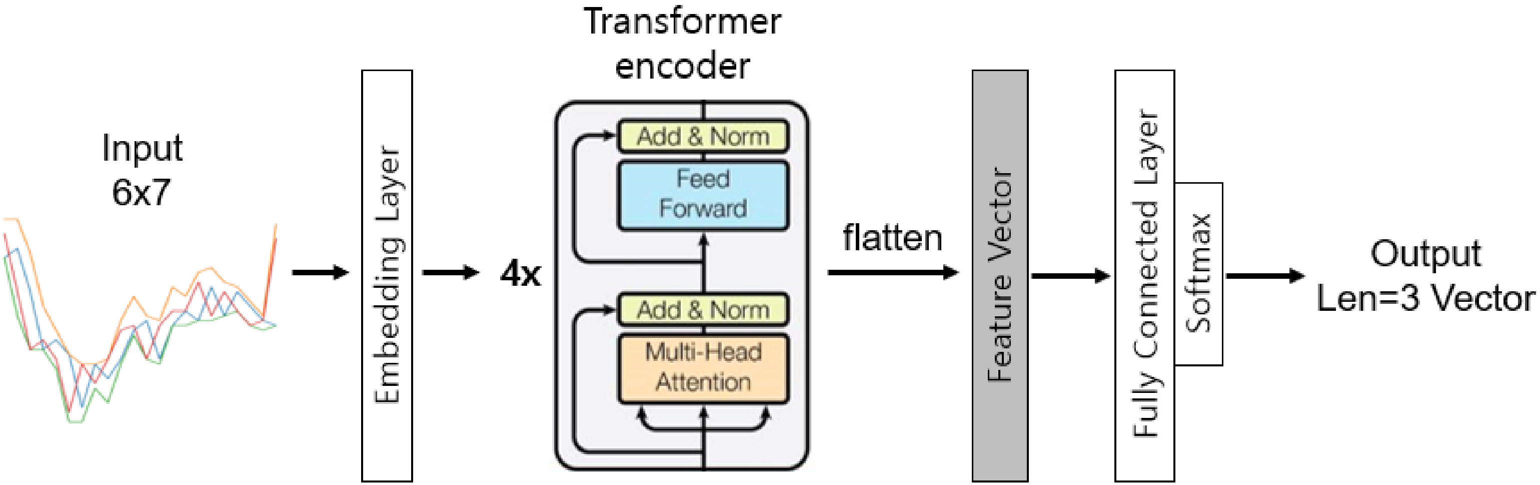
2. Train a Trading Model (1D-CNN)

- 1D-CNN model


Input
6x7
Convolutional Layer (1x3@32, Stride=1)
BN
ReLU
Convolutional Layer (1x3@32, Stride=1)
BN
ReLU
Convolutional Layer (1x3@64, Stride=1)
BN
ReLU
Convolutional Layer (1x3@64, Stride=1)
BN
ReLU
Convolutional Layer (1x3@128, Stride=1)
BN
ReLU
Convolutional Layer (1x3@128, Stride=1)
BN
ReLU
flatten
Feature Vector
Fully Connected Layer
Softmax
Output
Len=3
Vector

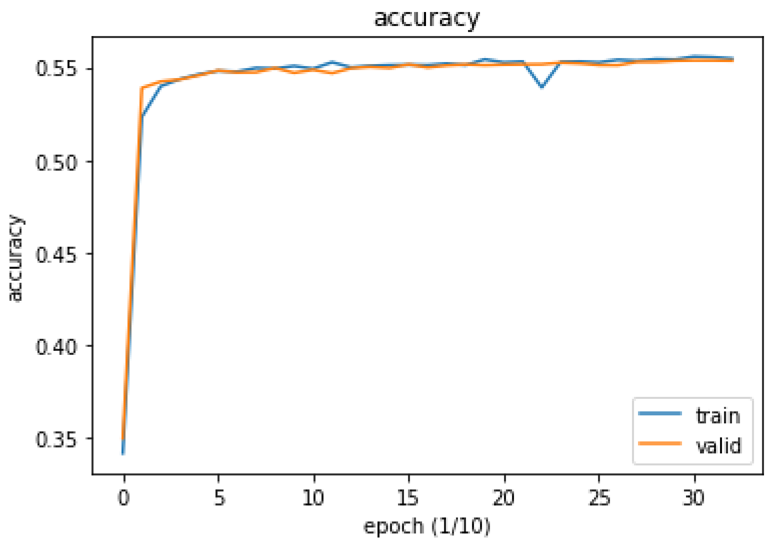
2. **Train a Trading Model** (Transformer)

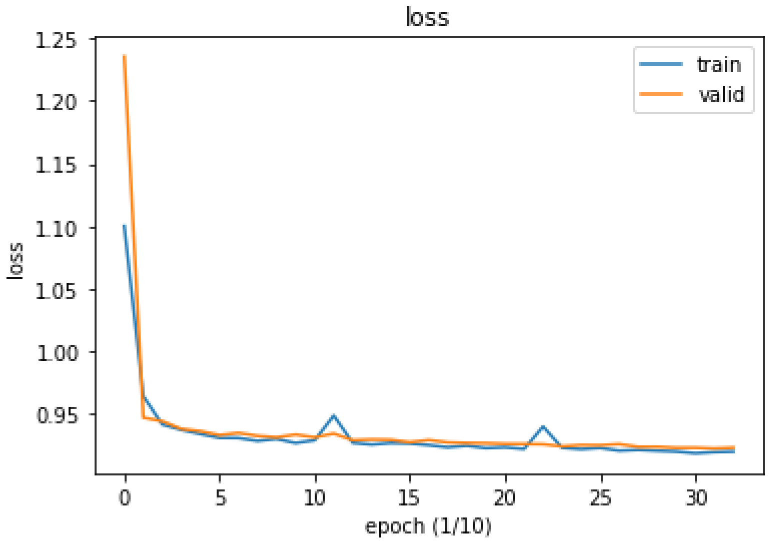
- Transformer model



2. **Train a Trading Model** (Result of 1D-CNN Model)

- 1D-CNN model





Total data size : 5,760,000 (80% train, 10% valid, 10% test)

2. **Train a Trading Model** (Result of 1D-CNN Model)

- 1D-CNN model (test set size = 576,011)

	+0.375% gain	between	-0.45% loss
BUY	91896	27086	42753
HOLD	107781	147950	158535

Win rate = 91896/(91896+27086+42753) = **56.8%**

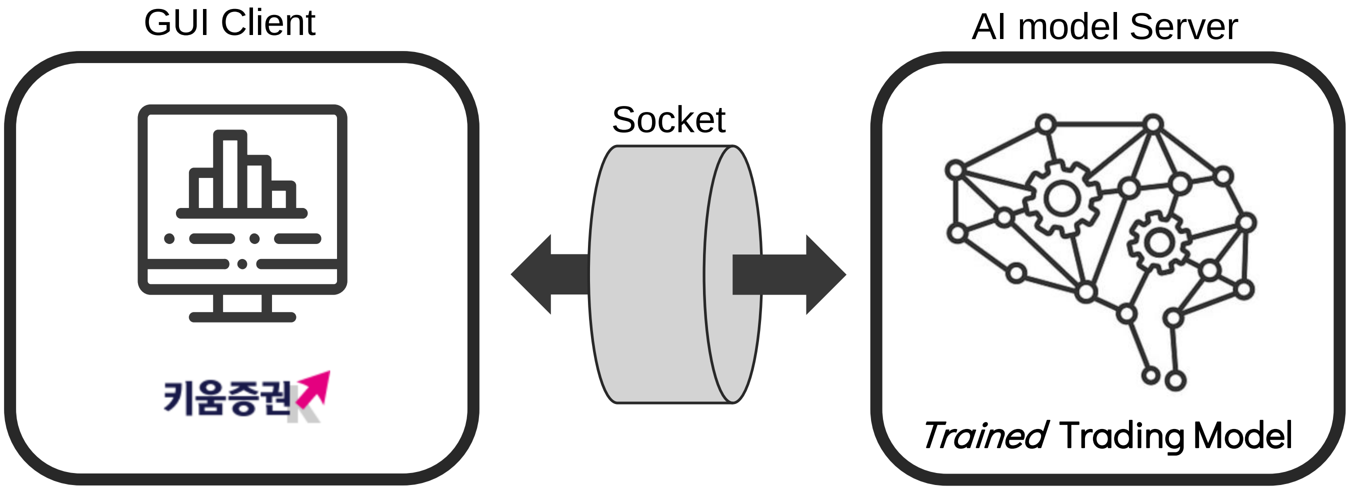
Loss rate = 42753/(91896+27086+42753) = **26.4%**

Expected Annual Yield = **15.4%** *(1 Purchase per day)*

Contributor : Choi Dongyoung

Real-time System

- GOAL
    1. **Real time trading based on AI model**
    2. **Display explanation of model's decision**
- Connect Real Time Trading GUI based on Kiwoom API with AI model
- Use socket programming to communicate GUI and AI model Server



GUI Client

Socket

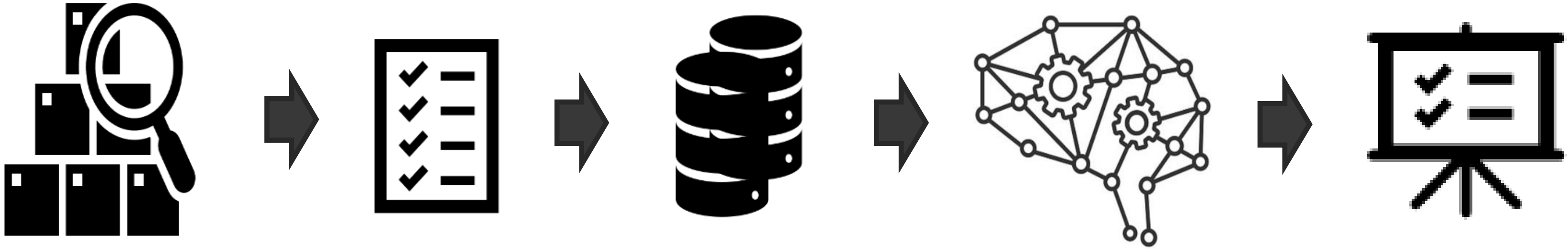
AI model Server

Trained Trading Model

Contributor : Choi Dongyoung

Real-time System(Cont.)

1. Load conditional search result
2. Subscribe the result stock to get real time trading information
3. Get real time data and display it on GUI
4. Send real time data to pretrained AI model after preprocessing
5. Send model result to GUI
6. GUI will buy or ignore target stocks followed by AI decision



Search active stocks

Subscribe those stocks

Get real time data and display it on GUI

Send real time data to AI model

Send model result to GUI

Contributor : Choi Dongyoung

Conditional Search

- ❑ Too many stocks to test through AI model in real time (more than 3000)

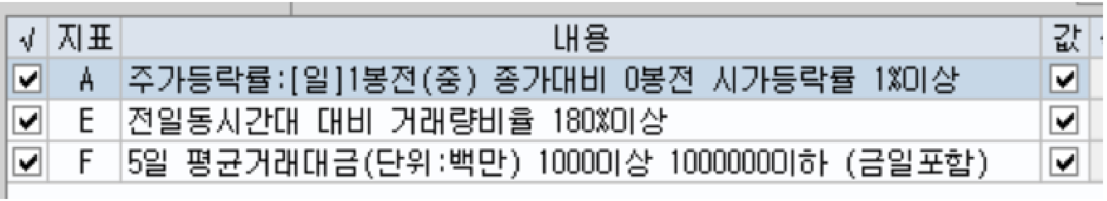
  -> Use **Conditional Search** supported by 키움 API

- ❑ Search the most popular and active stocks that satisfies certain conditions

  -> **Market Price > 1.01 x closing price of yesterday**

  -> **Trading volume > 1.8 x Trading volume of yesterday (at the same time zone)**

  -> **Average stock trading amount > 1 Billion won**

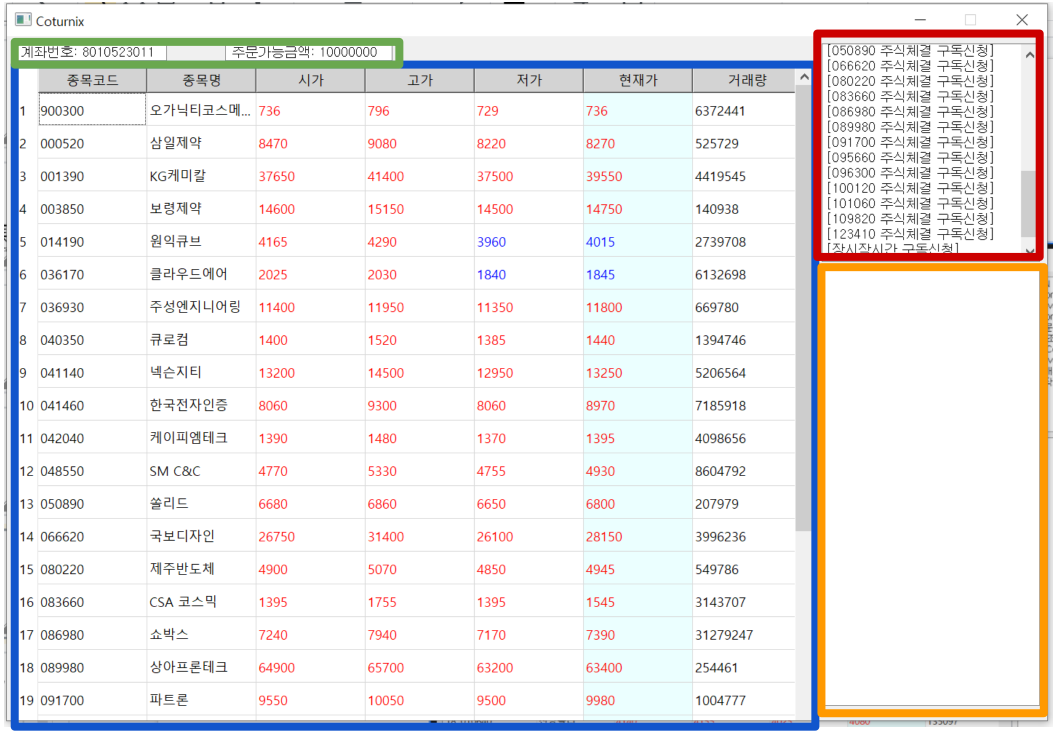


√	지표	내용	값
☑	A	주가등락률:[일]1봉전(중) 종가대비 0봉전 시가등락률 1%이상	☑
☑	E	전일동시간대 대비 거래량비율 180%이상	☑
☑	F	5일 평균거래대금(단위:백만) 1000이상 1000000이하 (금일포함)	☑

Contributor : Choi Dongyoung

GUI

- PyQt5 based GUI



계좌번호: 8010523011 | 주문가능금액: 10000000

	종목코드	종목명	시가	고가	저가	현재가	거래량
1	900300	오가닉티코스메...	736	796	729	736	6372441
2	000520	삼일제약	8470	9080	8220	8270	525729
3	001390	KG케미칼	37650	41400	37500	39550	4419545
4	003850	보령제약	14600	15150	14500	14750	140938
5	014190	원익큐브	4165	4290	3960	4015	2739708
6	036170	클라우드에어	2025	2030	1840	1845	6132698
7	036930	주성엔지니어링	11400	11950	11350	11800	669780
8	040350	큐로컴	1400	1520	1385	1440	1394746
9	041140	넥슨지티	13200	14500	12950	13250	5206564
10	041460	한국전자인증	8060	9300	8060	8970	7185918
11	042040	케이피엠테크	1390	1480	1370	1395	4098656
12	048550	SM C&C	4770	5330	4755	4930	8604792
13	050890	쏠리드	6680	6860	6650	6800	207979
14	066620	국보디자인	26750	31400	26100	28150	3996236
15	080220	제주반도체	4900	5070	4850	4945	549786
16	083660	CSA 코스믹	1395	1755	1395	1545	3143707
17	086980	쇼박스	7240	7940	7170	7390	31279247
18	089980	상아프론테크	64900	65700	63200	63400	254461
19	091700	파트론	9550	10050	9500	9980	1004777

```
[050890 주식체결 구독신청]
[066620 주식체결 구독신청]
[080220 주식체결 구독신청]
[083660 주식체결 구독신청]
[086980 주식체결 구독신청]
[089980 주식체결 구독신청]
[091700 주식체결 구독신청]
[095660 주식체결 구독신청]
[096300 주식체결 구독신청]
[100120 주식체결 구독신청]
[101060 주식체결 구독신청]
[109820 주식체결 구독신청]
[123410 주식체결 구독신청]
[장시작시간 구독신청]
```

Account number and Deposit

Stock table

System Log

Stock Trading Log

Contributor : Choi Dongyoung

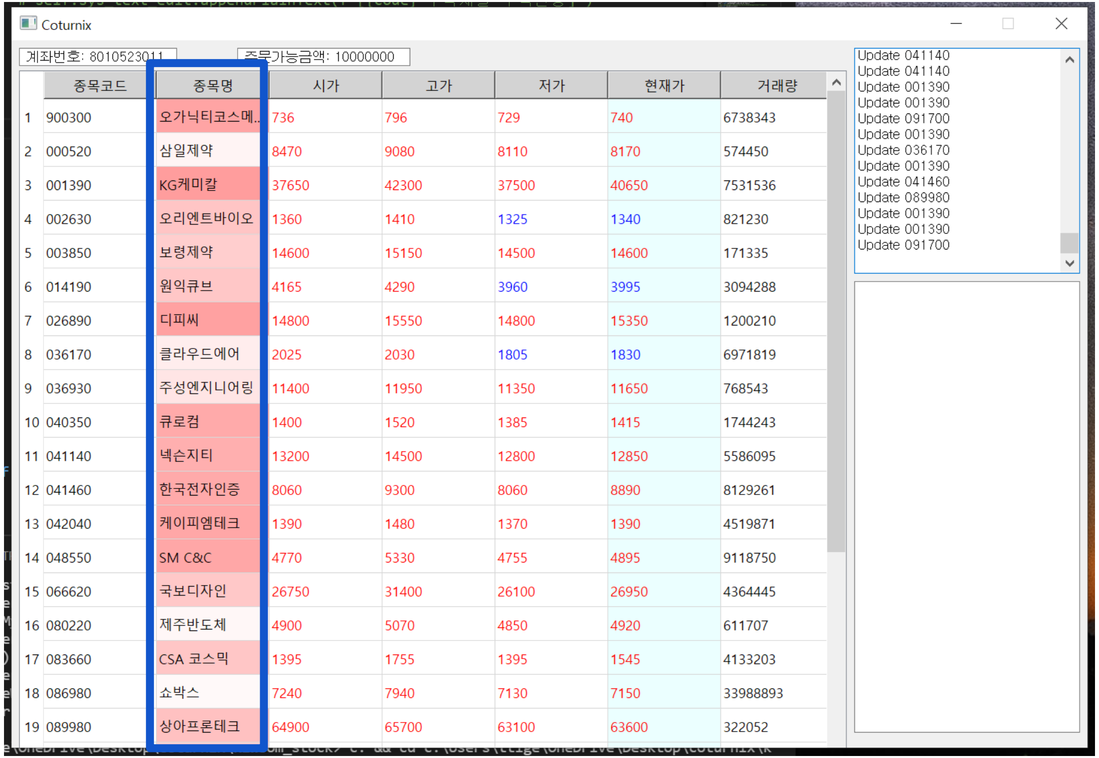
GUI

- ❑ PyQt5 based GUI

Contributor : Choi Dongyoung

GUI (Scheduled)

- We will express the degree of confidence that it will rise through colors



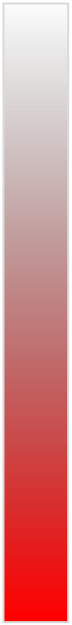
Coturnix

계좌번호: 8010523011

주문가능금액: 10000000

	종목코드	종목명	시가	고가	저가	현재가	거래량
1	900300	오가닉티코스메..	736	796	729	740	6738343
2	000520	삼일제약	8470	9080	8110	8170	574450
3	001390	KG케미칼	37650	42300	37500	40650	7531536
4	002630	오리엔트바이오	1360	1410	1325	1340	821230
5	003850	보령제약	14600	15150	14500	14600	171335
6	014190	원익큐브	4165	4290	3960	3995	3094288
7	026890	디피씨	14800	15550	14800	15350	1200210
8	036170	클라우드에어	2025	2030	1805	1830	6971819
9	036930	주성엔지니어링	11400	11950	11350	11650	768543
10	040350	큐로컴	1400	1520	1385	1415	1744243
11	041140	넥슨지티	13200	14500	12800	12850	5586095
12	041460	한국전자인증	8060	9300	8060	8890	8129261
13	042040	케이피엠테크	1390	1480	1370	1390	4519871
14	048550	SM C&C	4770	5330	4755	4895	9118750
15	066620	국보디자인	26750	31400	26100	26950	4364445
16	080220	제주반도체	4900	5070	4850	4920	611707
17	083660	CSA 코스믹	1395	1755	1395	1545	4133203
18	086980	쇼박스	7240	7940	7130	7150	33988893
19	089980	상아프론테크	64900	65700	63100	63600	322052

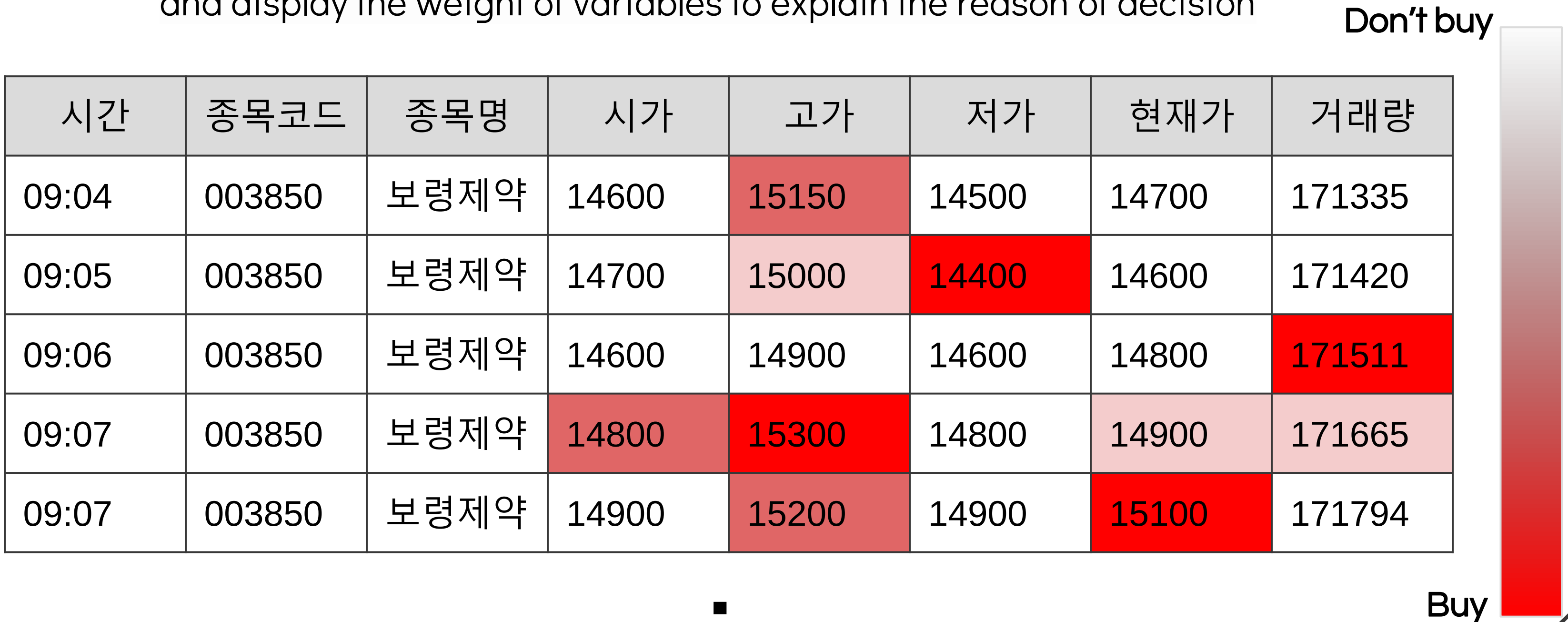
```
Update 041140
Update 041140
Update 001390
Update 001390
Update 091700
Update 001390
Update 036170
Update 001390
Update 041460
Update 089980
Update 001390
Update 001390
Update 091700
```



Contributor : Choi Dongyoung

GUI (Scheduled)

❑ Show the cumulative stock data that is double clicked and display the weight of variables to explain the reason of decision


Don’t buy
Buy

시간	종목코드	종목명	시가	고가	저가	현재가	거래량
09:04	003850	보령제약	14600	15150	14500	14700	171335
09:05	003850	보령제약	14700	15000	14400	14600	171420
09:06	003850	보령제약	14600	14900	14600	14800	171511
09:07	003850	보령제약	14800	15300	14800	14900	171665
09:07	003850	보령제약	14900	15200	14900	15100	171794

⋮

Next Steps

1. Fine tuning
   - Hyperparameters (data preprocessing)
   - Transfer learning

2. Verification (Mock investment)

3. Explainability (Grad-CAM or Attention)

How to calculate the expected annual yield.

e.g) Precision for +1% yield : 60.5%

0.03% Fee

250 Transactions (250 weekdays in a year)

Win : 250 * 0.605 = 151

Loss : 250 * 0.395 = 99

$$(1 + 0.01 - 0.0003)^{151} \times (1 - 0.01 - 0.0003)^{99} = 1.54$$

>> **54% Annual Yield**