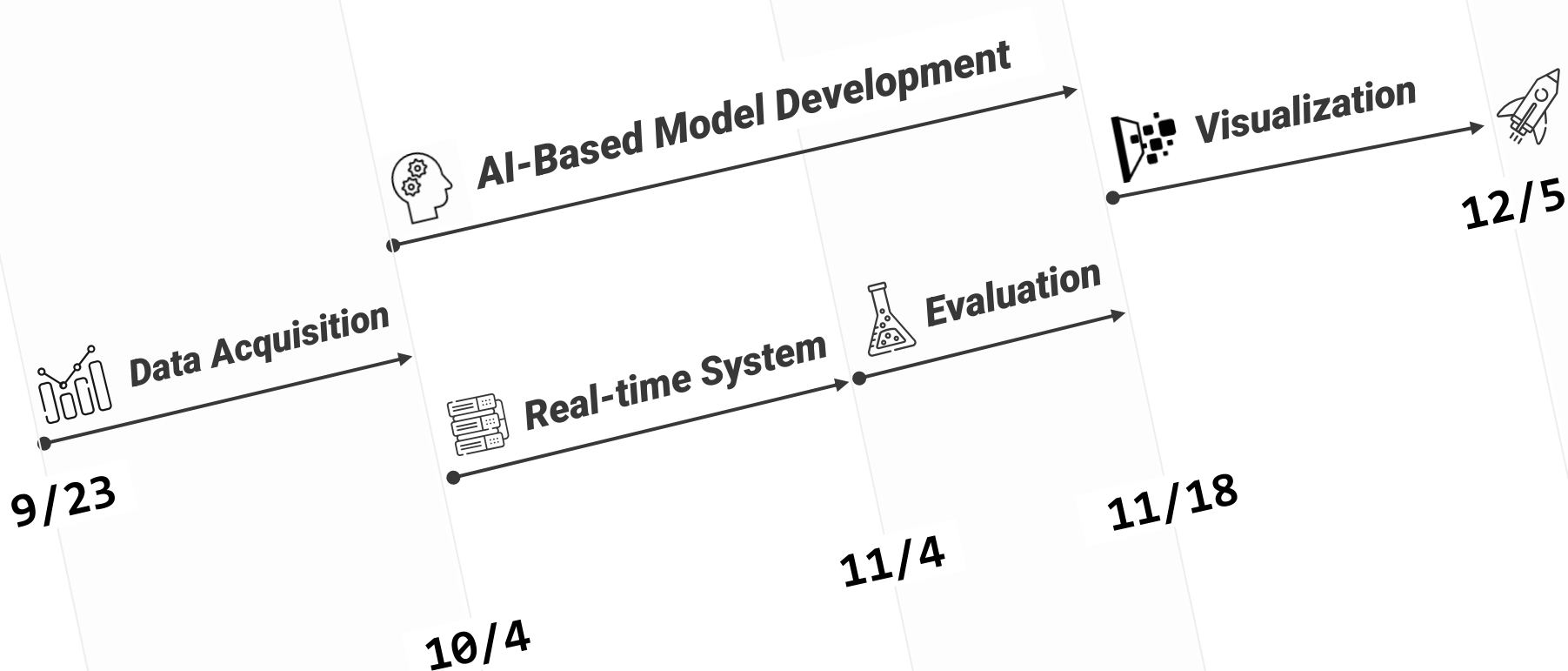




Timeline



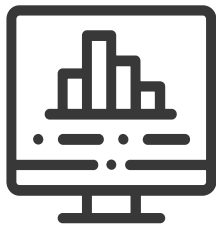


Real-time System

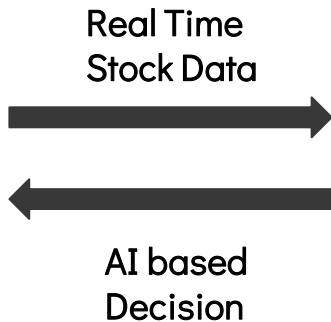
- The person in charge : **Dongyoung Choi**
- How?

Using KIWOOM Open API

- Connect Real Time Trading system based on Kiwoom API with AI model
- Do Real-time Mock Investment. A true evaluation!



키움증권



Trained Trading Model



Real-time System

- 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
-> The market price rose more than 1% from the previous day's closing price
-> The trading volume to the same time zone is more than 180% compared to previous day

The screenshot shows a stock conditional search interface. At the top, there's a header bar with a dropdown menu set to '대상변경' (Target Change) and a sub-menu '업종대상(전체)><제외없음><전체월결산>'. To the right are input fields for '0', '봉전기준' (Close as basis), and radio buttons for '매수' (Buy) and '매도' (Sell). Below this is a section for '일' (Daily) and '주기' (Period) set to '1' (1 day). The main search criteria are defined by two radio buttons: the first is selected, with a value of '1' and a unit of '%', and a dropdown set to '미상' (Unknown); the second is unselected, with a value of '5' and a unit of '%', followed by a tilde '~' and a value of '10' with a unit of '%'. To the right of these are buttons for '수정' (Modify) and '추가' (Add). At the bottom is a table with columns for '지표' (Indicator), '내용' (Content), and '값' (Value). The table contains two rows: Row A, '주가등락률:[일]1봉전(중) 증가대비 0봉전 시가등락률 1%이상' (Daily stock price change rate: [1 day] 1-day (middle) increase vs 0-day previous day's opening price change rate 1% or more), and Row E, '전일동시간대 대비 거래량비율 180%이상' (Trading volume ratio compared to the previous day's same time zone 180% or more). Both rows have a checked box in the '값' column and a set of directional arrows in the final column.

지표	내용	값	삭제	▲▼↑↓
<input checked="" type="checkbox"/>	A 주가등락률:[일]1봉전(중) 증가대비 0봉전 시가등락률 1%이상	<input checked="" type="checkbox"/>	X	▲▼↑↓
<input checked="" type="checkbox"/>	E 전일동시간대 대비 거래량비율 180%이상	<input checked="" type="checkbox"/>	X	▲▼↑↓



Kiwoom API

- PyQt5 based Open API
- Various functions of Kiwoom objects can be used through 'dynamic call'

```
class Kiwoom(QAxWidget):
    def __init__(self):
        super().__init__()
        self._create_kiwoom_instance()
        self._set_signal_slots()
        self.condition = {}
        self.codeList = []

    def _create_kiwoom_instance(self):
        self.setControl("KHOPENAPI.KHOpenAPICtrl.1")

    def _set_signal_slots(self):
        self.OnEventConnect.connect(self._event_connect)
        self.OnReceiveTrData.connect(self._receive_tr_data)

        ## 조건검색식 관련 추가
        self.OnReceiveConditionVer.connect(self.receiveConditionVer)
        self.OnReceiveTrCondition.connect(self.receiveTrCondition)
        self.OnReceiveRealCondition.connect(self.receiveRealCondition)
```

The screenshot shows the 'Kiwoom Open API' login window. At the top, it features the 'Kiwoom증권' logo and the text 'KIWOOM OPEN API'. Below this, the title 'Kiwoom Open API' is displayed in large orange letters. The login form includes three input fields: '고객ID' (Customer ID) with a masked value '*****', '비밀번호' (Password), and '인증비밀번호' (Authentication Password). There are three buttons: '로그인' (Login), '환경설정' (Settings), and '종료' (Exit). Below the buttons, there are two checkboxes: '고객 아이디 저장' (Save customer ID) and '모의투자접속' (Connect to simulation investment), both of which are checked. At the bottom, there are three buttons: '공동인증' (Joint authentication), '신규가입' (New registration), and 'ID/PW찾기' (Find ID/PW). A final instruction at the bottom reads 'ID 와 비밀번호를 입력해 주십시오.' (Please enter your ID and password).



Kiwoom API (cont.)

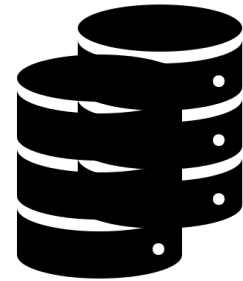
- Load conditional search result
- Subscribe the result stock to get real time trading information
- Wait and Get realtime data



Search the active stocks list



Subscribe those stocks

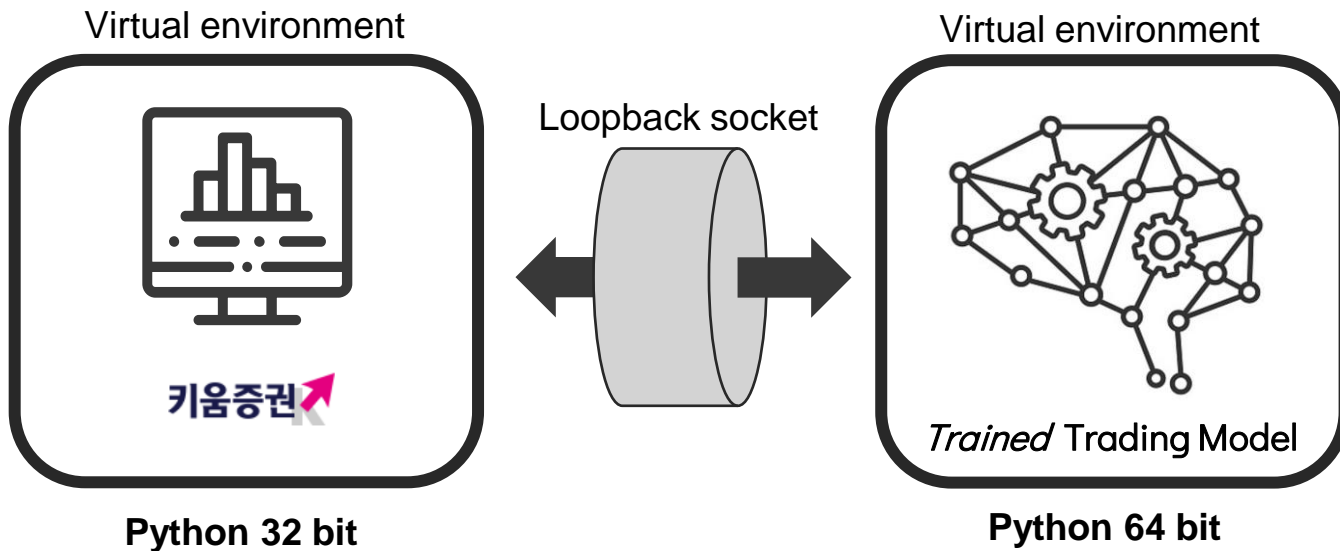


Get real time data



Real-time System (cont.)

- Kiwoom API only supports 32-bit, while Pytorch supports only 64-bit
-> Use two virtual environment and communicate with two programs through packet





Real-time System (cont.)

- Real time trading information of subscribed stocks

```
--waiting  
코드: 001140 시간: 143447 시가: +1705 고가: +2140 저가 +1700  
코드: 060240 시간: 143447 시가: +5460 고가: +6660 저가 -5370  
코드: 003530 시간: 143447 시가: +5100 고가: +6070 저가 -4950  
코드: 003530 시간: 143447 시가: +5100 고가: +6070 저가 -4950  
코드: 007110 시간: 143447 시가: +2850 고가: +2965 저가 -2775  
코드: 100130 시간: 143447 시가: +5920 고가: +6310 저가 +5910  
코드: 257720 시간: 143447 시가: +20800 고가: +22700 저가 +20200  
코드: 100130 시간: 143447 시가: +5920 고가: +6310 저가 +5910  
코드: 025750 시간: 143447 시가: +2030 고가: +2190 저가 +1930  
코드: 274090 시간: 143447 시가: +16900 고가: +17600 저가 -15900  
코드: 003530 시간: 143447 시가: +5100 고가: +6070 저가 -4950  
코드: 003530 시간: 143447 시가: +5100 고가: +6070 저가 -4950  
코드: 003530 시간: 143447 시가: +5100 고가: +6070 저가 -4950  
코드: 104200 시간: 143447 시가: +7720 고가: +8980 저가 +7540  
코드: 060240 시간: 143447 시가: +5460 고가: +6660 저가 -5370  
코드: 257720 시간: 143447 시가: +20800 고가: +22700 저가 +20200  
코드: 007660 시간: 143447 시가: +4040 고가: +4050 저가 -3780  
코드: 060240 시간: 143447 시가: +5460 고가: +6660 저가 -5370  
코드: 101360 시간: 143447 시가: +38550 고가: +41300 저가 +38400  
코드: 100130 시간: 143447 시가: +5920 고가: +6310 저가 +5910  
코드: 025750 시간: 143447 시가: +2030 고가: +2190 저가 +1930  
코드: 003530 시간: 143447 시가: +5100 고가: +6070 저가 -4950  
--waiting
```



AI Model Development

- Zero-pad Empty time serial data
- Data Labeling:
With Hyper parameter
rising (-1%), dropping percentage(+1%), Length of looking forward (3)
- Data cut:
With Hyper parameter
Data length (6), Sampling frequency (2)



AI Model Development

- Auto-encoder Network (Vanilla FC layers)

```
AutoEncoder(  
    (encoder): Sequential(  
      (0): Linear(in_features=48, out_features=32, bias=True)  
      (1): ReLU()  
      (2): Linear(in_features=32, out_features=16, bias=True)  
      (3): ReLU()  
      (4): Linear(in_features=16, out_features=8, bias=True)  
      (5): ReLU()  
      (6): Linear(in_features=8, out_features=4, bias=True)  
    )  
    (decoder): Sequential(  
      (0): Linear(in_features=4, out_features=8, bias=True)  
      (1): ReLU()  
      (2): Linear(in_features=8, out_features=16, bias=True)  
      (3): ReLU()  
      (4): Linear(in_features=16, out_features=16, bias=True)  
      (5): ReLU()  
      (6): Linear(in_features=32, out_features=48, bias=True)  
    )  
  )  
)
```

Next Steps

1. Find better model : 1D - CNN , transformer
1. Analysis on Trained unsupervised Model
Like t-SNE, PCA on latent vectors.