



Team project 개발 계획서

# AI를 활용한 주식 예측 사이트

1조

김건우(PM)  
김다복  
박동혁  
이현정

# 목차

## 1. 소개

프로젝트 목적  
프로그램 구성



## 2. 기능 정의서

앱 별 기능 및 사용 AI  
전체 구성도



## 3. 추후 개발 계획

프로토 타입 모델  
프로젝트 업무 분담  
마일스톤



# 1. 소개

## 팀 프로젝트 목적

### 1. 다양한 데이터 분석을 통한 예측 모델 구성

- 데이터 분석 통해 **역량**을 키우고 **실무로 연결하는 능력** 강화

### 2. 협업 능력 강화

- **실무에 많이 쓰이는** 프로그램 및 도구 사용 기회

### 3. 효율적인 팀 프로젝트

- 효율적으로 팀 프로젝트를 수행하기 위한 **병렬 구조 작업 분담**

- 개개인 역량 강화 및 팀 프로젝트 공동 수행 가능



## 주제 선정

ProDS를 통해 배운 통계 분석, AI를 통한  
과거 **주식 분석**으로 미래 동향 **예측**

# 1. 소개

## 프로젝트 구성

Django를 통해 **프론트엔드**와 **백엔드**로 나누어 개발



로그인 앱



주식 분석 및 예측 앱



토론 게시판 앱

AI를 통한 **과거 주식 분석**을 통해 **미래 동향을 예측**하여  
주식 매매에 도움을 주는 것을 목표

## 2. 기능 정의서

### 앱 별 기능 및 사용 AI

▶ 메인 페이지로 이동하려면  
**로그인 필수**

▶ 회원 가입 이동 및 절차

1. 아이디 (E-mail)
2. 비밀번호 (영문/숫자 8-16자)
3. Nickname : 영/한/숫자 혼용(3-10자)  
**중복 금지**
4. **증권사** 정보  
(크롤링 데이터를 얻는 **사이트 패치에 적용**)

▶ **NaviBar** 다시 이동 가능

▶ **개인정보 변경** 가능

#### 1. 로그인 앱

▶ **케라스**를 통한 주식  
분석 자료 제공

▶ 과거 주식 정보 및 예측  
**데이터 시각화 자료** 제공

▶ 예측을 바탕으로 주식  
**매매/매도 여부 추천**

※ 과거 데이터를 기반으로 한 예측 및  
추천이므로 참고용 **맹신 금지!!**

#### 2. 주식 분석 예측 앱

▶ 네이버 지식in 형식으로  
**이용자들간 질의응답** 가능

▶ 게시판 인기/최신/조회순  
**분류 선택** 가능

▶ 메인 글쓰기 및 답글,  
추천 좋아요 기능,  
삭제 및 수정 기능

▶ 추가 기능 예정  
내가 쓴 글, 검색 기능

#### 3. 토론 게시판 앱



## 2. 기능 정의서

### 앱 별 기능 및 사용 AI

#### 프론트엔드 개발

- ▶ 메인화면 UI구성  
(로그인 앱과 주식화면 앱 연동)
- ▶ 주식목록에서 원하는 주식 선택  
(List HTML 구성)
- ▶ 메인 웹 페이지에 주식 분석 그래프 및 주가 예측 결과  
(그래프 시각화 자료 구성)
- ▶ 토론 게시판 인기순/추천순 분류 UI  
게시판 테마 Bootstrap 이용

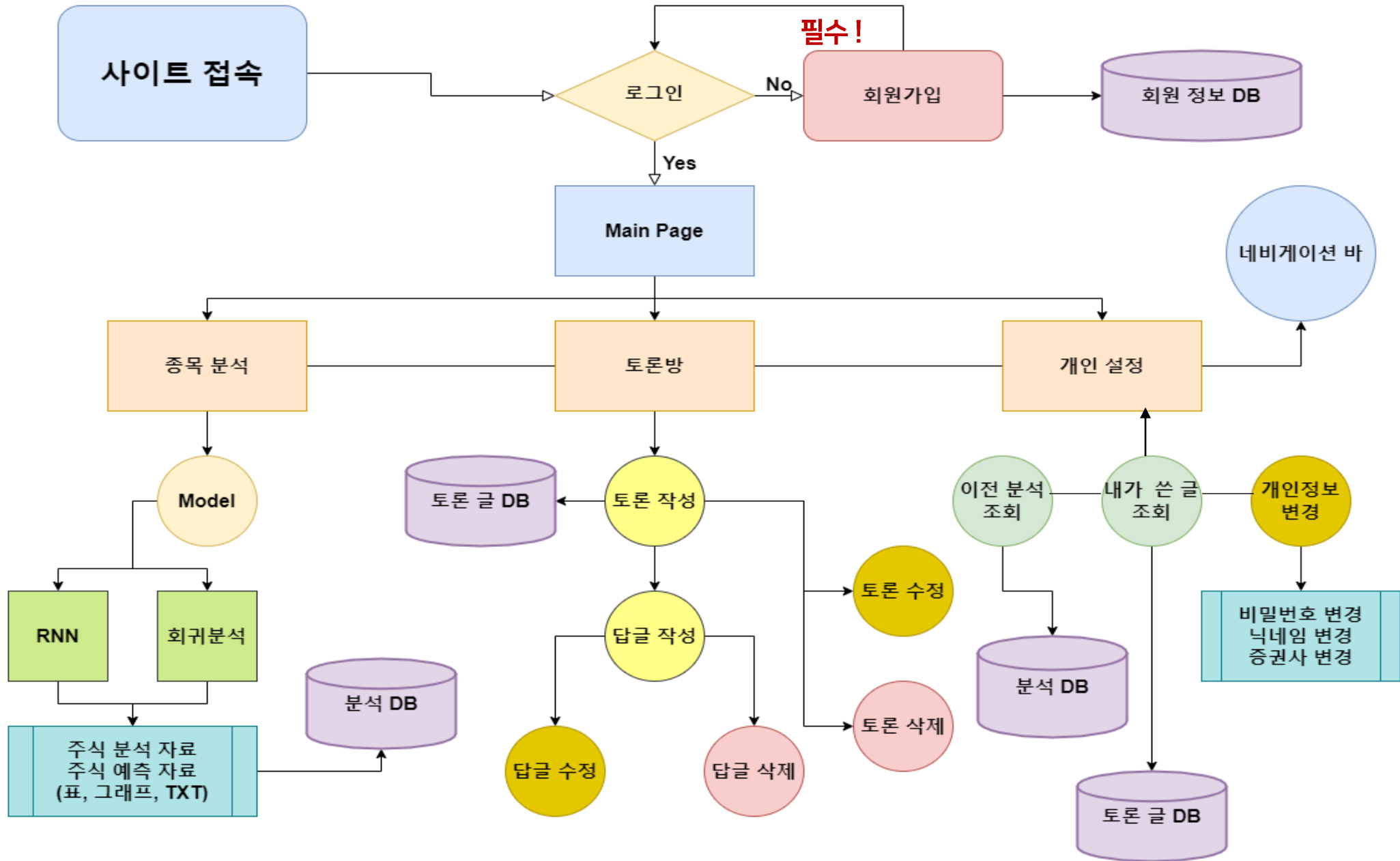
#### 백엔드 개발

- ▶ 데이터 크롤링 : 야후 주식 데이터 크롤링
- ▶ 모델 : 케라스 > **중추신경망** > train/test
- ▶ 토론방 DB : discussion\_id를 통한 질문/답변 구분,  
종목 표시, 사진 글 입력, 인기/추천순 분류, 검색 기능
- ▶ 회원 정보 관리 및 증권사 분석



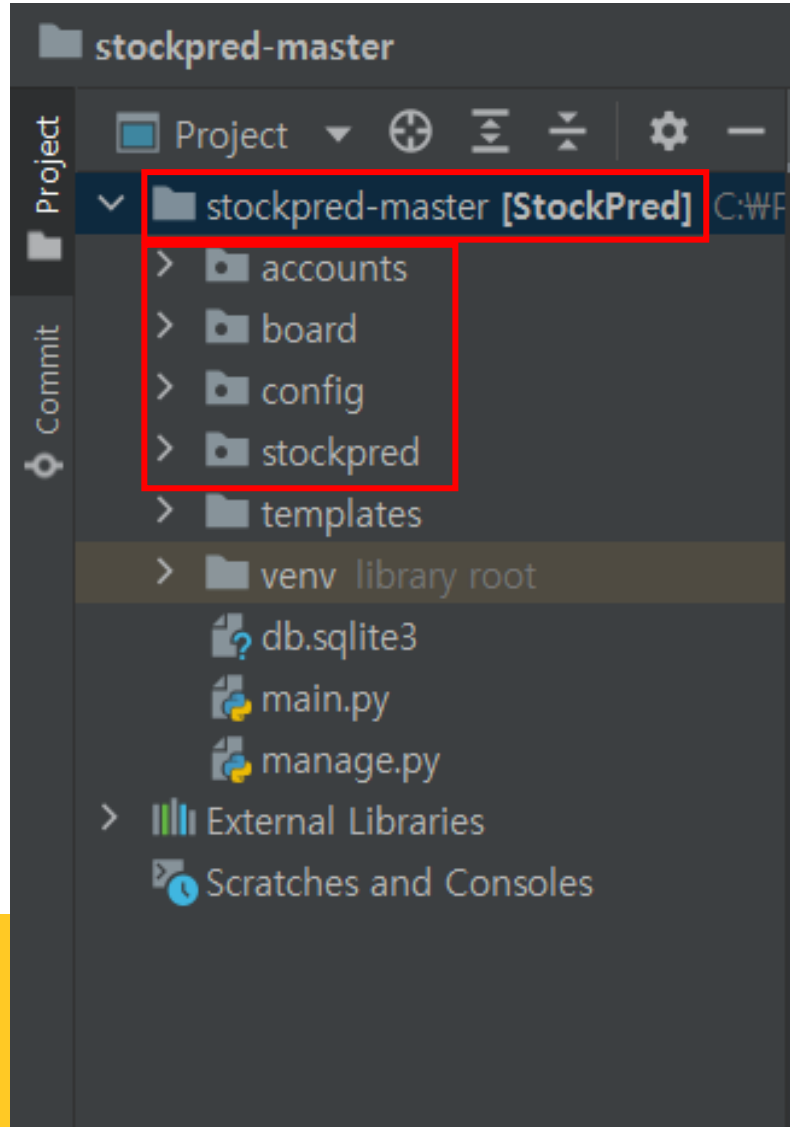
케라스 중추신경망 모델  
회귀분석 모델  
의사결정나무 모델

# 전체 구성도



### 3. 추후 개발 계획

#### 개발 사이트 환경 구성



#### 주식예측 프로젝트



로그인 앱

토론 게시판 앱

주식 분석 및 예측 앱

앞서 정리한 **Flow Chart**를 기반으로  
내부 모델 **프로토 타입** 구성



### 3. 추후 개발 계획

#### 프로토타입 모델

## 로그인 앱

```
accounts#views.py × question_detail.html × question_list.html × board#views.py ×
1 from django.shortcuts import render
2 from .forms import RegisterForm
3
4 # Create your views here.
5 def register(request):
6     if request.method == 'POST':
7         user_form = RegisterForm(request.POST)
8         if user_form.is_valid():
9             new_user = user_form.save(commit=False)
10             new_user.set_password(user_form.cleaned_data['password']) #암호화
11             new_user.save() # 데이터베이스 저장
12             return render(request, 'registration/register_done.html', {'form': user_form})
13     else:
14         user_form = RegisterForm()
15     return render(request, 'registration/register.html', {'form': user_form})
16
```

### 3. 추후 개발 계획

#### 프로토타입 모델

## 게시판 앱

```
accounts#views.py × question_detail.html × question_list.html × board#views.py ×
1 from django.shortcuts import render
2 from .models import Question
3
4 # Create your views here.
5
6 def index(request) : # 목록 출력
7     question_list = Question.objects.order_by('-create_date')
8     context = {'question_list' : question_list}
9     return render(request, 'board/question_list.html', context)
10
11 def detail(request, question_id) : # 내용 출력
12     question = Question.objects.get(id=question_id)
13     context = {'question' : question}
14     return render(request, 'board/question_detail.html', context)
```

토론 게시판 예시 화면

글 검색 기능

최신순

최신/인기/조회 순 분류 기능

찾기

번호	추천	제목	작성자	작성일시
3	1	test2	admin	2022년 1월 27일 3:44 오후
2	1	test1	admin	2022년 1월 27일 3:44 오후
1	1	test	admin	2022년 1월 27일 3:03 오후

이전

1

다음

질문 등록하기

질문 등록 기능



# 토론 게시판 글 작성 예시

discussion 변경

test

Subject: 제목

test

Content: 내용

test



Photo:

현재: image1.jpg ☐ 취소

변경:  선택된 파일 없음


이미지 파일 업로드 기능

Author: 작성자

admin ▼  

Voter:

admin ▲



하나 이상을 선택하려면 "Control" 키를 누른 채로 선택해주세요. Mac의 경우에는 "Command" 키를 눌러주세요.

삭제

# RNN 프로토타입 모델

```
code = '005930.KS' # 종목코드  
start = '2021-01-01' # 조회 시작일  
end='2022-01-26' # 조회 종료일  
epochs = 100 # 학습 반복수
```

- 사용자로부터 종목코드, 조회시작일/ 종료일  
등을 입력받음

```
from keras.models import Sequential  
from keras.layers import SimpleRNN, Dense  
from sklearn.preprocessing import MinMaxScaler  
from sklearn.metrics import mean_squared_error  
from sklearn.model_selection import train_test_split  
import math  
import numpy as np  
import matplotlib.pyplot as plt
```

- 분석 및 시각화에 필요한 모듈 import

```
import time  
import pandas_datareader as pdr # 주식데이터 크롤링  
  
from jupyterthemes import jtplot # 시각화 테마 변경  
jtplot.style(theme = 'grade3')
```

```
def rnn(code, start, end): # 순환신경망(RNN)분석 함수  
    df = pdr.get_data_yahoo(code, start=start, end=end).reset_index() # 주식 데이터 크롤링  
    dataset = df['Close'].values # 종가 데이터 추출  
    dataset = dataset.reshape(dataset.shape[0], 1) # 1차원배열을 2차원으로 변경  
    dataset = dataset.astype('float32') # int -> float변환  
    scaler = MinMaxScaler(feature_range=(0, 0.9)) # 최대값을 0.9로 설정  
    Dataset = scaler.fit_transform(dataset)  
    train_data, test_data = train_test_split(Dataset, test_size=0.2, shuffle=False) # 8:2로 train, test 분리
```

- 사용자로부터 입력받은 데이터를 이용  
- 'Yahoo Finance'에서 데이터 크롤링/필요 데이터 추출  
- 분석에 필요한 데이터 형태로 가공  
(차원, int->float, Min-Max 변환)  
- train, test 데이터로 분리

```
def create_dataset(dataset, look_back): # 종속변수와 독립변수를 생성하는 함수  
    x_data = [] # 빈 리스트 생성  
    y_data = []  
  
    for i in range(len(dataset) - look_back):  
        data = dataset[i: (i + look_back), 0]  
        x_data.append(data)  
        y_data.append(dataset[i + look_back, 0])  
    return np.array(x_data), np.array(y_data)
```

- RNN 분석이 가능하도록 x, y를 생성하는 함수

```
look_back = 7 # 7개의 데이터로 예측
x_train, y_train = create_dataset(train_data, look_back) # train_data 생성
x_test, y_test = create_dataset(test_data, look_back) # test 데이터 생성

X_train = np.reshape(x_train, (x_train.shape[0], 1, x_train.shape[1])) # 3차원으로 변경
X_test = np.reshape(x_test, (x_test.shape[0], 1, x_test.shape[1])) # 3차원으로 변경
```

- 7일간의 데이터를 이용하여 다음 날을 예측
- 위에서 만든 함수를 이용하여 x, y 생성
- 순환신경망 학습을 위해 3차원으로 변환

```
model = Sequential() # model 옵션
model.add(SimpleRNN(3, input_shape=(1, look_back)))
model.add(Dense(1, activation="linear"))
model.compile(loss='mse', optimizer='adam')

model.fit(X_train, y_train, epochs=epochs, batch_size=1, verbose=1) # train_data 학습
```

- 모델 학습 옵션 지정 및 train data를 이용하여 학습

```
trainPredict = model.predict(X_train) # train_data를 이용하여 예측
testPredict = model.predict(X_test) # test_data를 이용하여 예측

TrainPredict = scaler.inverse_transform(trainPredict) # Min-Max 변환된 값을 원래대로 되돌림
Y_train = scaler.inverse_transform([y_train]) # Min-Max 변환된 값을 원래대로 되돌림
TestPredict = scaler.inverse_transform(testPredict) # Min-Max 변환된 값을 원래대로 되돌림
Y_test = scaler.inverse_transform([y_test]) # Min-Max 변환된 값을 원래대로 되돌림

train_RMSE = math.sqrt(mean_squared_error(Y_train[0], TrainPredict[:, 0])) # train_RMSE
test_RMSE = math.sqrt(mean_squared_error(Y_test[0], TestPredict[:, 0])) # test_RMSE
```

- 학습된 모델을 이용하여 예측 데이터 생성
- Min-Max 변환된 값을 원래 값으로 변환
- True값과 예측값을 이용하여 RMSE 구하기

```
x_pred = np.zeros(look_back * look_back, dtype = float) # 예측값을 저장할 array 생성
x_pred = x_pred.reshape(look_back, 1, look_back) # 3차원으로 변환

for i in range(look_back):
    if i == 0:
        x_pred[0][0][: -1] = Dataset[-6:].reshape(6) # 최근 6일 데이터를 pred의 1번째 row, 1~6번째 인덱스에 저장
        x_pred[0][0][-1] = model.predict(Dataset[-7:].reshape(1, 1, look_back)) # 최근 7일 데이터를 이용한 예측값을 pred의 1번째 row, 7번째 인덱스에 저장
    elif i > 0:
        x_pred[i][0][: -1] = x_pred[i-1][0][1:] # pred의 이전 row의 2~7번째 데이터를 다음 row의 1~6번째 인덱스에 저장
        x_pred[i][0][-1] = model.predict(x_pred[[i-1]]) # pred의 이전 row의 1~7번째 데이터를 이용하여 데이터 예측 후, 다음 row의 7번째 인덱스에 저장
```

- 예측값으로 다시 예측하기 위한 새로운 array 생성
- 반복문을 통해 '예측하고 예측된 값으로 다시 예측' 반복 및 결과 저장



```

trainPredictPlot = np.zeros(len(dataset) + 7)
trainPredictPlot = trainPredictPlot.reshape(trainPredictPlot.shape[0], 1)
trainPredictPlot[:, :] = np.nan
trainPredictPlot[look_back:len(TrainPredict) + look_back] = TrainPredict

testPredictPlot = np.zeros(len(dataset) + 7)
testPredictPlot = testPredictPlot.reshape(testPredictPlot.shape[0], 1)
testPredictPlot[:, :] = np.nan
testPredictPlot[len(TrainPredict) + (look_back) * 2 : len(dataset), :] = TestPredict

predPlot = np.zeros(len(dataset) + 7)
predPlot = predPlot.reshape(predPlot.shape[0], 1)
predPlot[:, :] = np.nan
predPlot[-7:] = x_pred[6].reshape(x_pred.shape[0], 1)
predPlot = scaler.inverse_transform(predPlot) # Min-Max변환된 값을 원래대로 되돌림

```

- True값, train data/test data를 이용한 예측값, 일주일간의 미래 예측값을 시각화 하기위해 데이터 구조 변경 및 데이터 생성

```

plt.figure(figsize = (20, 10))
plt.plot(dataset, label = "true", c = 'green')
plt.plot(trainPredictPlot, label = 'train_predict')
plt.plot(testPredictPlot, label = 'test_predict')
plt.plot(predPlot, label = 'pred')
plt.legend()

plt.show()

```

- 시각화

```

result = np.round(predPlot[-7:].reshape(look_back), 2)
result_max = result.max()
result_min = result.min()

|

return result, result_max, result_min, train_RMSE, test_RMSE

```

- result(7일의 예측데이터) 및 예측값의 최대/최소값
- Return : 1. 예측데이터 2. 최대값, 3. 최소값, 4. 신뢰 수준을 판단할 수 있는 RMSE(train, test)

# RNN 프로토타입 모델



프로토 타입 AI모델이  
예측한 주식 전망

코드 진행시간(초) : 85.67079615592957



코드 진행시간을 낮추는 것을 목표

prediction : [74224.63 74152.24 74190.75 74357.08 74448.61 74438.99 74444.67]

result\_max : 74448.61

result\_min : 74152.24

train\_RMSE : 990.561335713837

test\_RMSE : 1121.9886763639058

## 프로젝트 및 업무 분담

※ 프로젝트가 진행하면서 업무 추가 및 변동 예상


### 1조 팀 구성



김건우 (PM)

게시판 앱 담당  
기능 정의서 작성  
일정 정리 및 구성


&



박동혁

주식 분석 및 예측  
앱 담당  
순환신경망  
프로토타입 구성


&



김다복

게시판 앱 담당  
전체 프레임 워크  
구성

&



이현정

로그인 앱 담당  
개발 기획서 PPT  
제작  
회의록 서기

효율적인 업무 진행을 위한 병렬 구조 업무 분담 추구

## 마일 스톤 일정 계획

※ 일정에 기반하여 조율 가능

마일스톤																										
project timeline			1월									2월														
단계	진행업무	상세업무	24일	25일	26일	27일	28일	29일	30일	31일	1일	2일	3일	4일	5일	6일	7일	8일	9일	10일	11일	12일	13일	14일	15일	
1	프로젝트 계획 수립																									
	프로젝트 계획서 작성	계획서, prototype				발표																				
2	프로젝트 시작	기본 양식 작성				4~8																				
		숙제				구현하고 싶은 양식을 생각																				
		프론트엔드							앱 별로 개발 예정																	
		백엔드							앱 별로 개발 예정																	
		aws 배포 및 피드백															배포									
		피드백 수정																								
		구현 확인 및 ppt																								
		발표																								

Team Project 개발 계획서

# THANK YOU

1조

