

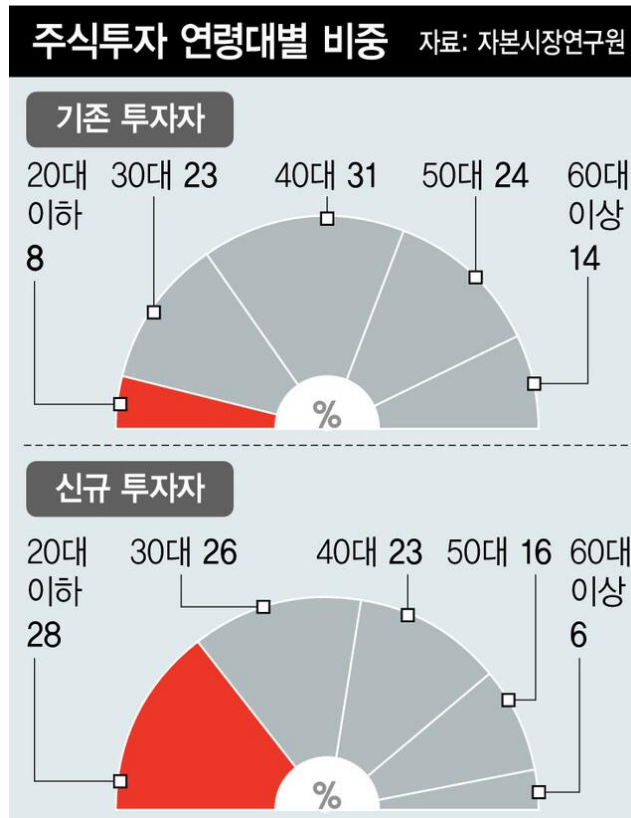
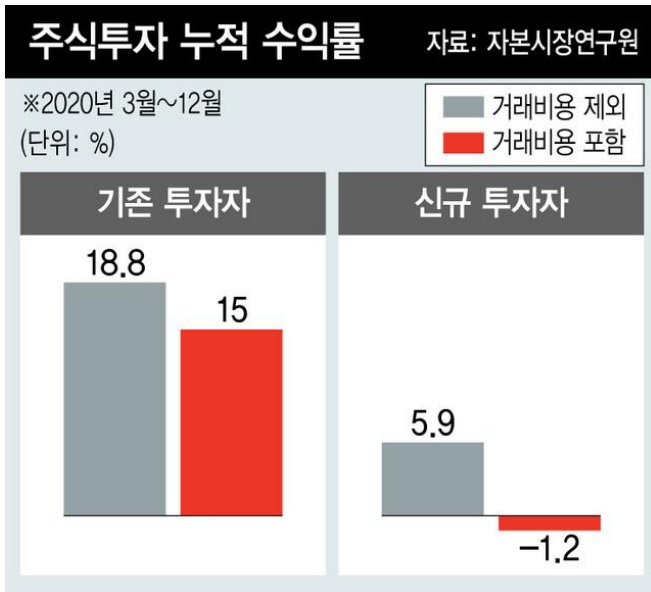
# 주가예측 시스템

최혜원

“주가를 예측하여 주식의 고수가 되자”

# 주제 선정 동기

## 주식을 로또처럼... '주린이' 3명 중 2명은 손실 봤다



# 개발 상황

중간발표 : 특정 기업에서 주가 예측 (APPL)

최종발표 : 현재 상황과 비교 시 얼마나 일치하는 지 파악  
(시간이 된다면)

(+) 본 기업의 주식을 살 지 팔 지 학습, 결정

(+) 타 기업도 적용 - 일반화시킬 수 있는지

# Dataset & Model

1. Yahoo finance  
(2020-07-27 ~ 2021-07-26, 1Y)

2. LSTM 을 이용할 예정  
(이전의 연속 시퀀스 데이터 이용)

Date	Open	High	Low	Close*	Adj Close**	Volume
Jul 22, 2021	145.93	148.20	145.81	146.80	146.80	74,144,058
Jul 21, 2021	145.53	146.13	144.63	145.40	145.40	74,915,000
Jul 20, 2021	143.46	147.10	142.96	146.15	146.15	96,238,600
Jul 19, 2021	143.75	144.07	141.67	142.45	142.45	121,434,600
Jul 16, 2021	148.46	149.76	145.88	146.39	146.39	93,100,300
Jul 15, 2021	149.24	150.00	147.09	148.48	148.48	106,820,300
Jul 14, 2021	148.10	149.57	147.68	149.15	149.15	127,050,800
Jul 13, 2021	144.03	147.46	143.63	145.64	145.64	100,827,100
Jul 12, 2021	146.21	146.32	144.00	144.50	144.50	76,299,700
Jul 09, 2021	142.75	145.65	142.65	145.11	145.11	99,788,400
Jul 08, 2021	141.58	144.06	140.67	143.24	143.24	105,575,500
Jul 07, 2021	143.54	144.89	142.66	144.57	144.57	104,911,600
Jul 06, 2021	140.07	143.15	140.07	142.02	142.02	108,181,800
Jul 02, 2021	137.90	140.00	137.75	139.96	139.96	78,852,600

# 알고리즘

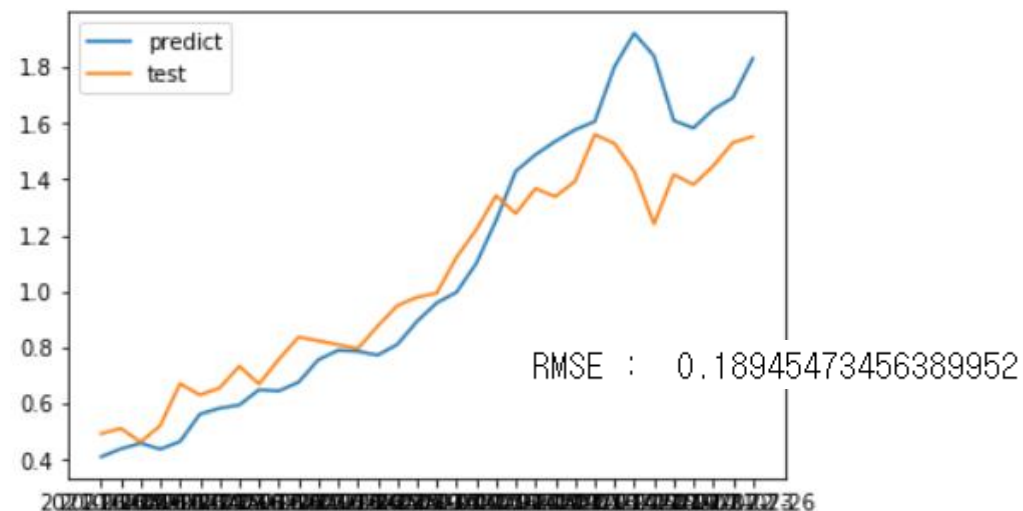
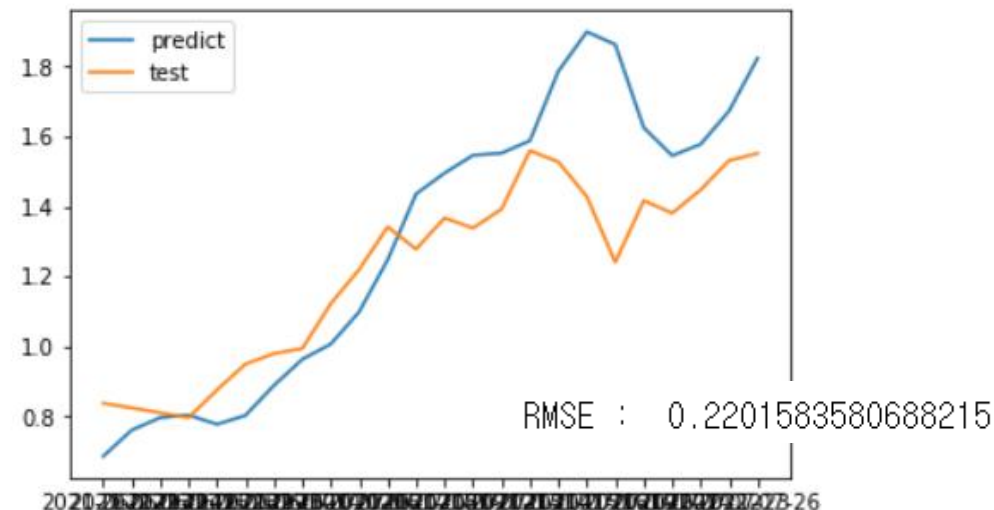
1. 이전의 기록들을 이용하여 그 다음의 그래프를 예측  
→ 주식이 월~ 금 이렇게 5일 단위로 바뀌기 때문에 이전 기록의 크기를 5로 맞춰주었다.

	shift_1	shift_2	shift_3	shift_4	shift_5
Date					
2021-02-08	0.970043	1.000000	0.835949	0.885878	0.845459
2021-02-09	0.977176	0.970043	1.000000	0.835949	0.885878
2021-02-10	0.934379	0.977176	0.970043	1.000000	0.835949
2021-02-11	0.904898	0.934379	0.977176	0.970043	1.000000
2021-02-12	0.892535	0.904898	0.934379	0.977176	0.970043
...	...	...	...	...	...
2021-05-25	0.510699	0.431289	0.520685	0.396101	0.403709
2021-05-26	0.501189	0.510699	0.431289	0.520685	0.396101
2021-05-27	0.498811	0.501189	0.510699	0.431289	0.520685
2021-05-28	0.424156	0.498811	0.501189	0.510699	0.431289
2021-06-01	0.392297	0.424156	0.498811	0.501189	0.510699

- loss func : mean\_squared\_error  
→ activation = relu

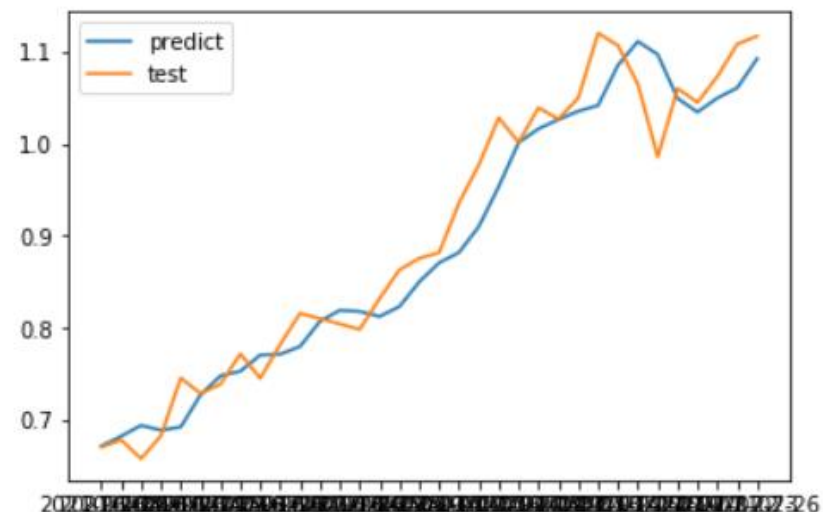
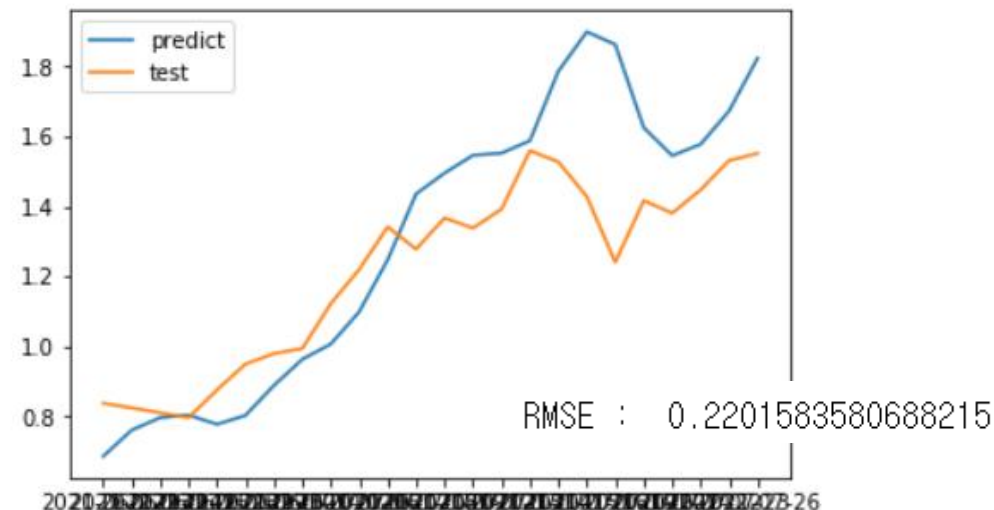
# 실험 기록-1

- train : '2021-02-01' ~ '2021-06-14'  
 test : '2021-06-15' ~ '2021-07-26'  
 loss func = mean\_squared\_error  
 optimizer = adam  
 activation = 'relu'  
 epochs = 200  
 model.add(layers.Dense(5))  
 model.add(layers.Dense(1))
- train : '2021-02-01' ~ '2021-06-14'  
 test : '2021-06-15' ~ '2021-07-26'  
 loss func = **mean\_absolute\_error**  
 optimizer = adam  
 activation = 'relu'  
 epochs = 200  
 model.add(layers.Dense(5))  
 model.add(layers.Dense(1))



# 실험 기록-1

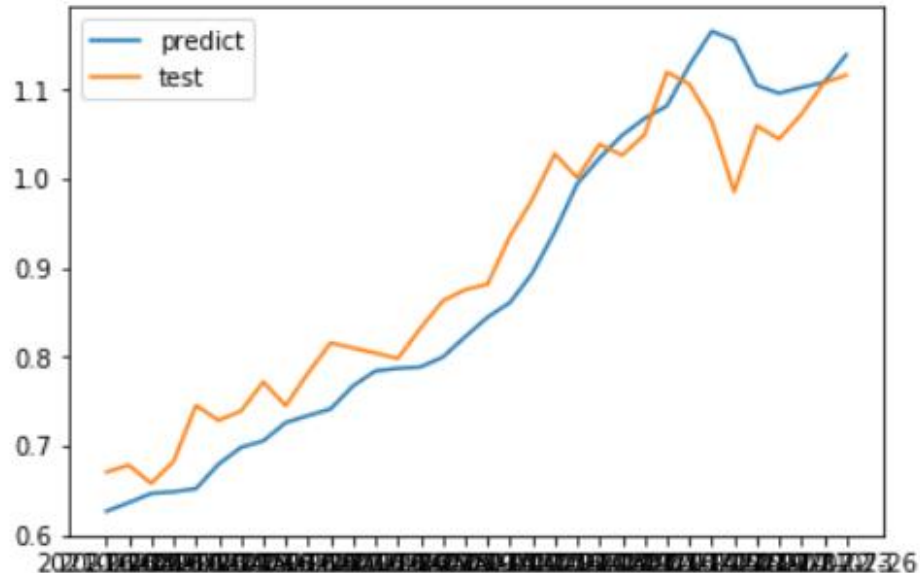
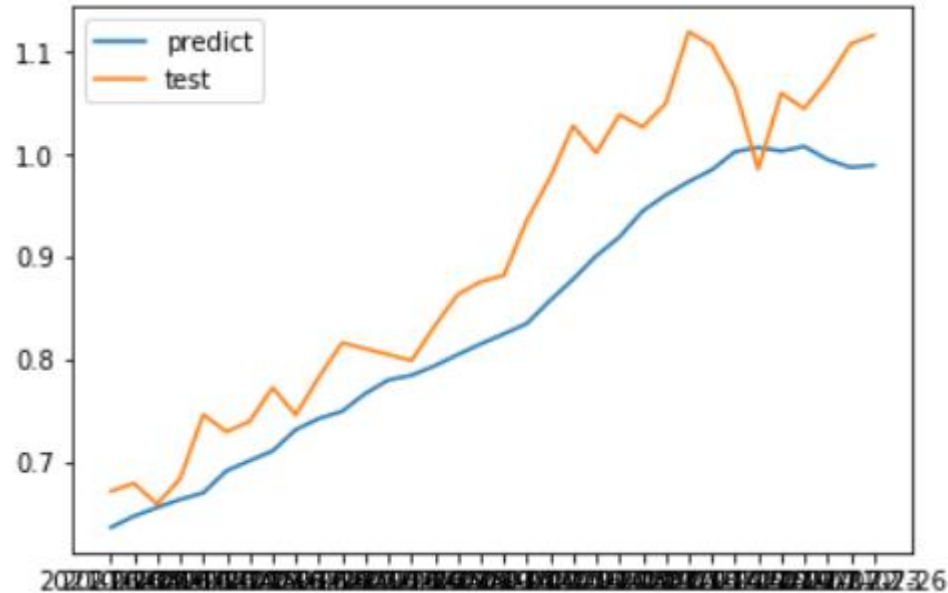
- train : '2021-02-01' ~ '2021-06-14'  
test : '2021-06-15' ~ '2021-07-26'  
loss func = mean\_squared\_error  
optimizer = adam  
activation = 'relu'  
epochs = 200  
model.add(layers.Dense(5))  
model.add(layers.Dense(1))
- train : '2020-07-26' ~ '2021-05-31'  
test : '2021-06-01' ~ '2021-07-26'  
loss func = mean\_squared\_error  
optimizer = adam  
activation = 'relu'  
epochs = 200





# 실험 기록-1

- train : '2020-07-26' ~ '2021-05-31'  
test : '2021-06-01' ~ '2021-07-26'  
loss func = mean\_squared\_error  
optimizer = adam  
activation = 'relu'  
epochs = 50
- train : '2020-07-26' ~ '2021-05-31'  
test : '2021-06-01' ~ '2021-07-26'  
loss func = mean\_squared\_error  
optimizer = adam  
activation = 'relu'  
epochs = 100



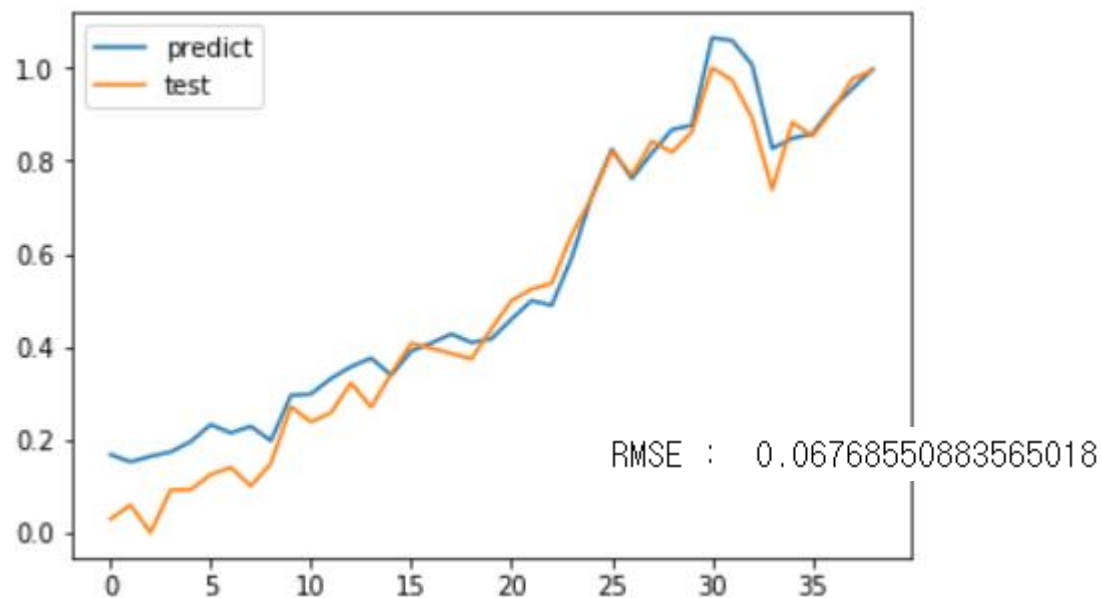
# 알고리즘

2. Train data 의 Feature 를 ' Open ' , ' High ' , ' Low ' , ' Volume ' 으로 하고  
이후의 predict 값이 어떻게 나오는 지 고찰

(	Open	High	Low	Volume	Date	
Date					2021-06-01	124.279999
2021-02-01	133.750000	135.380005	130.929993	106239800	2021-06-02	125.059998
2021-02-02	135.729996	136.309998	134.610001	83305400	2021-06-03	123.540001
2021-02-03	135.759995	135.770004	133.610001	89880900	2021-06-04	125.889999
2021-02-04	136.300003	137.399994	134.589996	84183100	2021-06-07	125.900002
2021-02-05	137.350006	137.419998	135.860001	75693800	2021-06-08	126.739998
...	...	...	...	...	2021-06-09	127.129997
2021-05-25	127.820000	128.320007	126.320000	72009500	2021-06-10	126.110001
2021-05-26	126.959999	127.389999	126.419998	56575900	2021-06-11	127.349998
2021-05-27	126.440002	127.639999	125.080002	94625600	2021-06-14	130.479996
2021-05-28	125.570000	125.800003	124.550003	71311100	2021-06-15	129.639999
					2021-06-16	130.149994
					2021-06-17	131.789993

# 실험 기록-2

- train : '2021-02-01' ~ '2021-05-28'  
test : '2021-06-01' ~ '2021-07-26'  
loss func = mean\_squared\_error  
optimizer = adam  
activation = 'relu'  
epochs = 100  
model.add(layers.Dense(1))



# 앞으로의 계획

최종발표 :

- 모델 구체화, 정리, 좀 더 많은 데이터로 K-겹 교차검증 적용
- **현재 상황과 비교** 시 얼마나 일치하는 지 파악
- 본 기업의 주식을 살 지 팔 지 학습, 결정

(시간이 된다면)

(+) 타 기업도 적용 - 일반화시킬 수 있는지

감사합니다.