

RNN을 이용한 유가 데이터 예측

김상협 박병준 정지윤

1조 GOAT



김상협

- RNN 모델 구성 및 학습
- 학습 결과 시각화

박병준

- RNN 코드 수정
- PPT 제작

정지윤

- 결과 보고서 작성
- 발표

유가 데이터는 대표적인 시계열 데이터



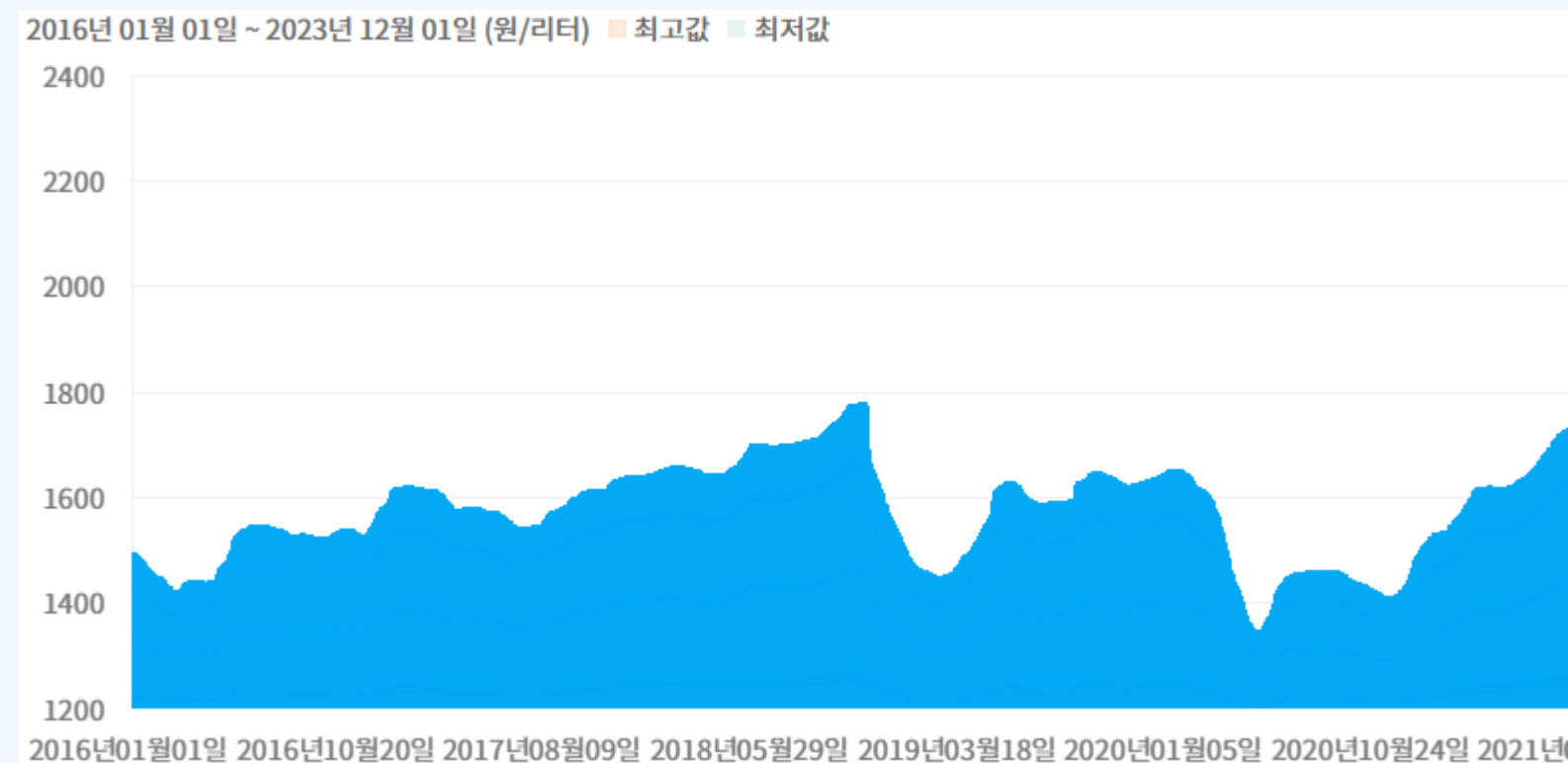
수치(의료, 복지, 기상), 음성, 텍스트, 이미지 등 다양한 데이터셋 후보군 선정



일상과 밀접한 주제



최근 운전을 시작한 조원 정지윤의 의견 피력

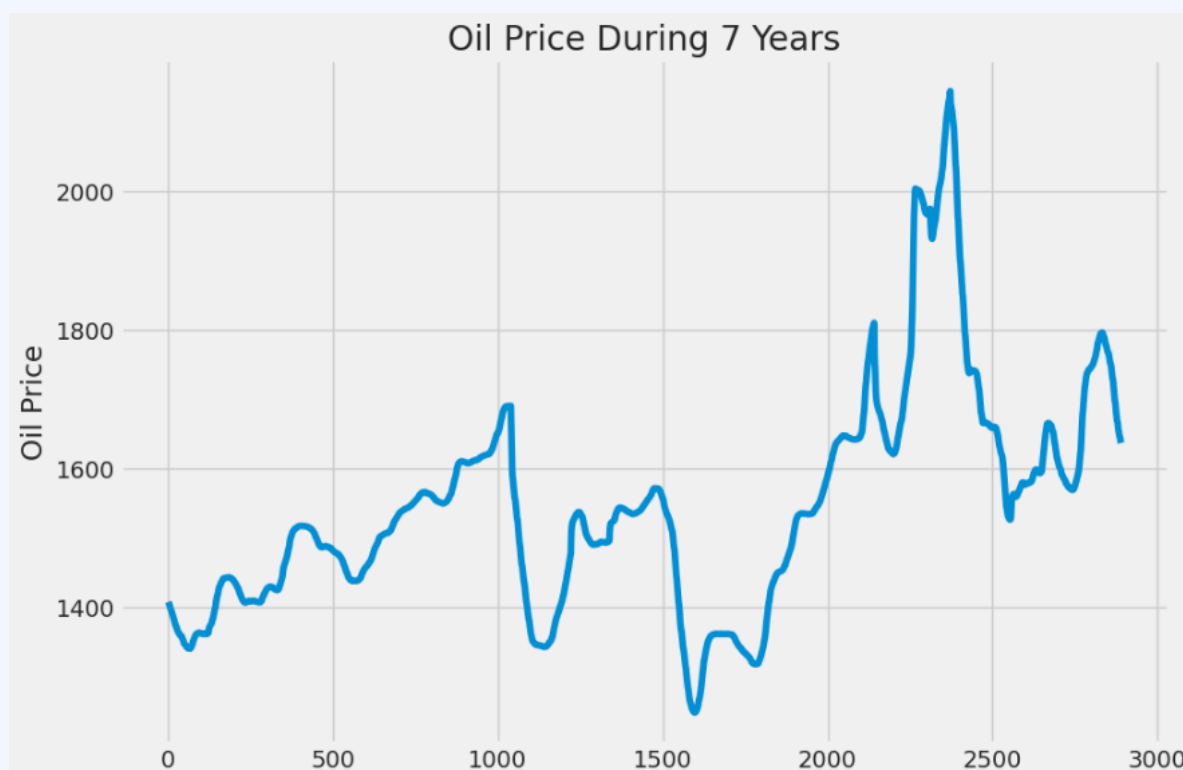


유가 데이터

1. 데이터 수집

- Opinet 주유소 통계 사이트
- 2016년 1월 1일 ~ 2023년 11월 30일
- 보통휘발유 가격(일간 평균 소수점 2자리)

3. 시각화



2. 데이터 전처리

- 결측치 제거

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 2891 entries, 0 to 2890  
Data columns (total 1 columns):  
#   Column  Non-Null Count  Dtype  
---  ---  
0   price    2891 non-null     float64  
dtypes: float64(1)  
memory usage: 22.7 KB
```

- 데이터 스케일링

```
# scale the data  
from sklearn.preprocessing import MinMaxScaler  
sc = MinMaxScaler(feature_range=(0,1))  
ts_train_scaled = sc.fit_transform(ts_train)
```

RNN

Model: "sequential_4"

Layer (type)	Output Shape	Param #
simple_rnn_8 (SimpleRNN)	(None, 5, 32)	1088
simple_rnn_9 (SimpleRNN)	(None, 32)	2080
dense_4 (Dense)	(None, 2)	66

Total params: 3234 (12.63 KB)
Trainable params: 3234 (12.63 KB)
Non-trainable params: 0 (0.00 Byte)

GRU

Model: "sequential_16"

Layer (type)	Output Shape	Param #
gru_6 (GRU)	(None, 5, 32)	3360
gru_7 (GRU)	(None, 32)	6336
dense_16 (Dense)	(None, 2)	66

Total params: 9762 (38.13 KB)
Trainable params: 9762 (38.13 KB)
Non-trainable params: 0 (0.00 Byte)

GRU(R)

Model: "sequential_20"

Layer (type)	Output Shape	Param #
gru_20 (GRU)	(None, 5, 32)	3360
dropout_16 (Dropout)	(None, 5, 32)	0
gru_21 (GRU)	(None, 5, 32)	6336
dropout_17 (Dropout)	(None, 5, 32)	0
gru_22 (GRU)	(None, 5, 32)	6336
dropout_18 (Dropout)	(None, 5, 32)	0
gru_23 (GRU)	(None, 32)	6336
dropout_19 (Dropout)	(None, 32)	0
dense_20 (Dense)	(None, 1)	33

Total params: 22401 (87.50 KB)
Trainable params: 22401 (87.50 KB)
Non-trainable params: 0 (0.00 Byte)

LSTM, LSTM(R)

Model: "sequential_12"

Layer (type)	Output Shape	Param #
lstm_14 (LSTM)	(None, 5, 32)	4352
lstm_15 (LSTM)	(None, 32)	8320
dropout_3 (Dropout)	(None, 32)	0
dense_12 (Dense)	(None, 2)	66

Total params: 12738 (49.76 KB)
Trainable params: 12738 (49.76 KB)
Non-trainable params: 0 (0.00 Byte)

train : test 비율 = 85 : 15

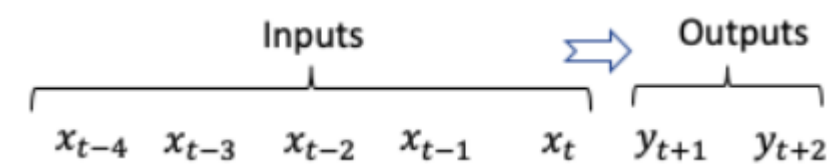
train -> 2016.1.1 ~ 2022.7.27

test -> 2022.7.28 ~ 2023.11.30

학습

input: 이전 가격 5일치

output: 이후 가격 2일치



학습 데이터 예시

	0	1	2	3	4	0	1
0	1407.15	1405.98	1405.33	1403.84	1402.25	1400.78	1399.15
1	1405.98	1405.33	1403.84	1402.25	1400.78	1399.15	1397.36
2	1405.33	1403.84	1402.25	1400.78	1399.15	1397.36	1395.62
3	1403.84	1402.25	1400.78	1399.15	1397.36	1395.62	1394.79
4	1402.25	1400.78	1399.15	1397.36	1395.62	1394.79	1393.53
...
2389	2034.36	2026.25	2013.01	2002.16	1989.93	1977.25	1964.60
2390	2026.25	2013.01	2002.16	1989.93	1977.25	1964.60	1959.42
2391	2013.01	2002.16	1989.93	1977.25	1964.60	1959.42	1948.86
2392	2002.16	1989.93	1977.25	1964.60	1959.42	1948.86	1936.63
2393	1989.93	1977.25	1964.60	1959.42	1948.86	1936.63	1926.81

2394 rows x 7 columns

RNN 코드 예시

```
my_rnn_model = Sequential()
my_rnn_model.add(SimpleRNN(32, return_sequences=True))
my_rnn_model.add(SimpleRNN(32))
my_rnn_model.add(Dense(2)) # The time step of the output

my_rnn_model.compile(optimizer='rmsprop', loss='mean_squared_error')

# fit the RNN model
my_rnn_model.fit(X_train, y_train, epochs=epochs, batch_size=150, verbose=0)

# Finalizing predictions
rnn_predictions = my_rnn_model.predict(X_test)
rnn_predictions = sc.inverse_transform(rnn_predictions)
```

Optimizer

RMSProp

- learning rate: 0.01

Loss function

Mean Square Error

Epoch

1, 5, 10, 50 설정 후 비교

RNNgasprice v3.5.ipynp

RNNgasprice v3.ipynp

RNNgasprice v4.5.ipynp

RNNgasprice v4.ipynp

RNNgasprice v5.ipynp

RNNgasprice v7.5.ipynb

RNNgasprice v8.ipynb

RNNgasprice v9.ipynb

RNN

LSTM

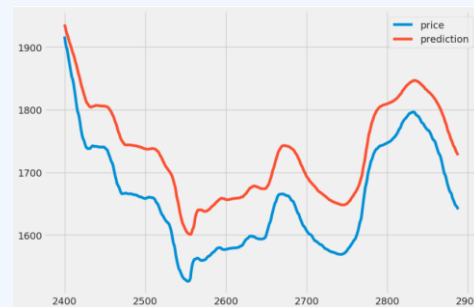
LSTM(R)

GRU

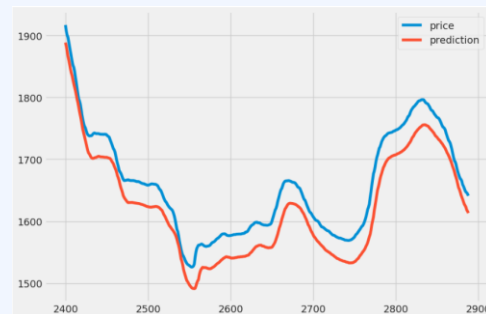
GRU(R)

Epoch

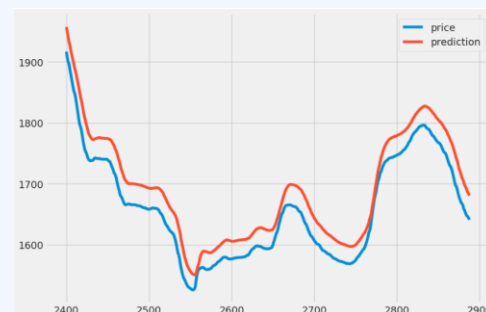
1



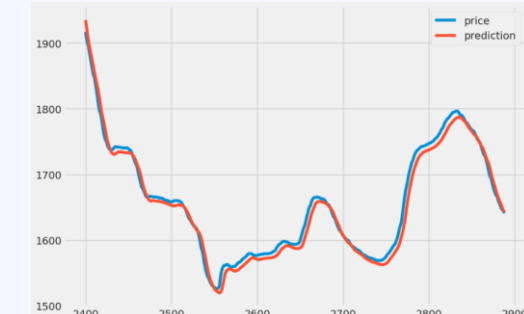
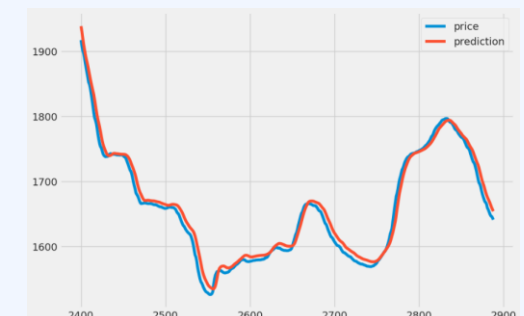
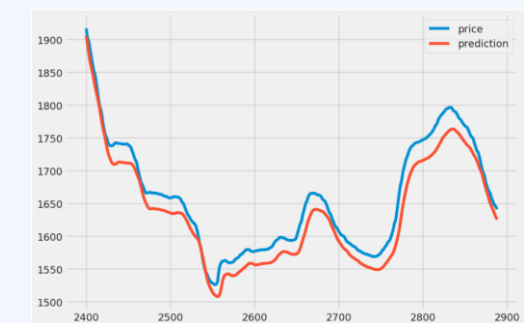
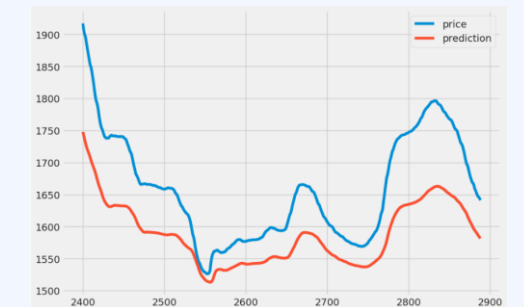
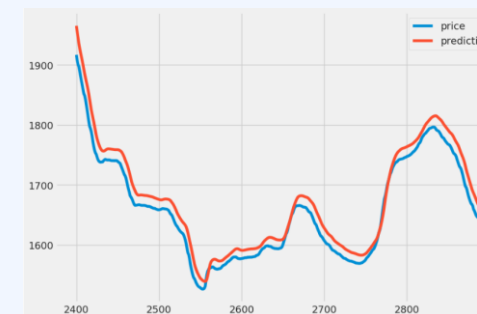
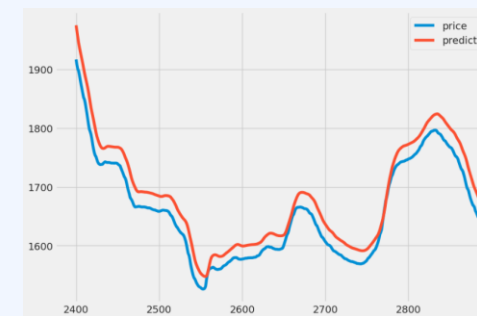
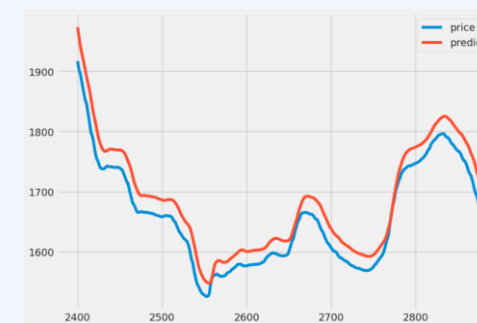
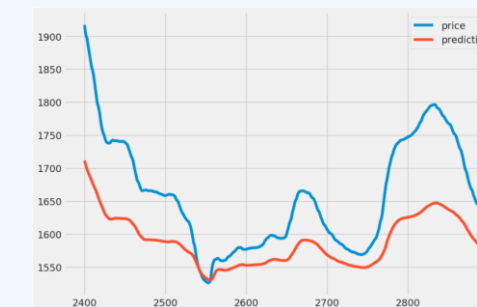
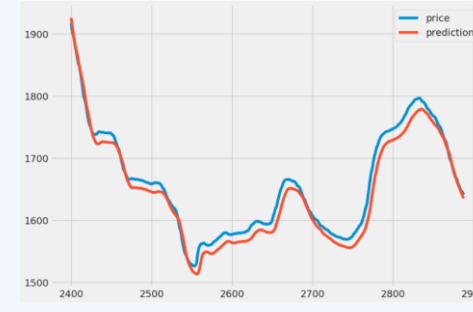
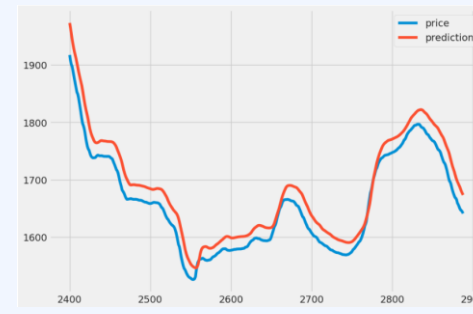
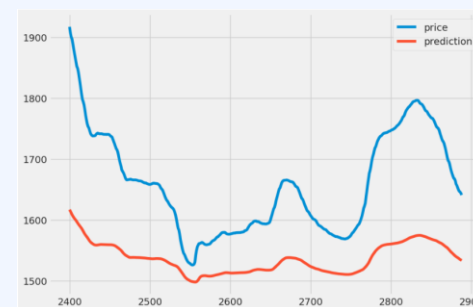
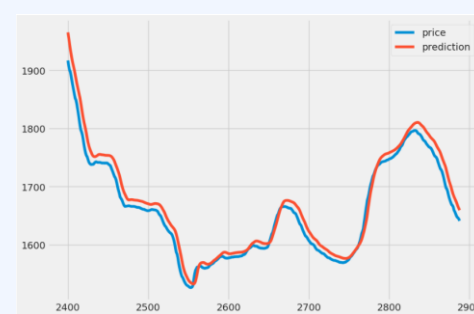
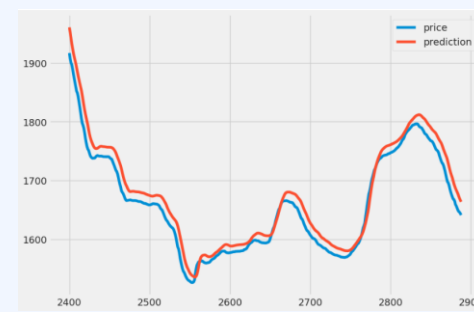
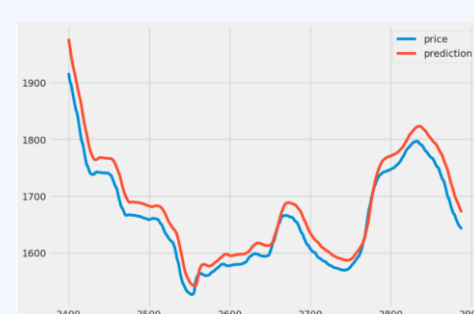
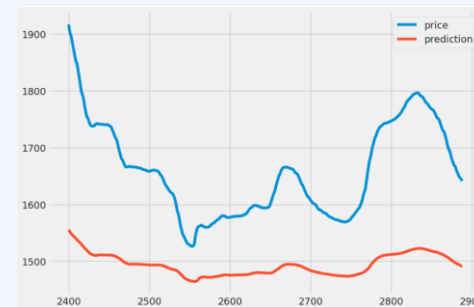
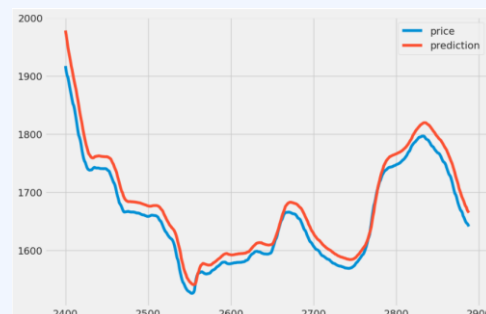
5

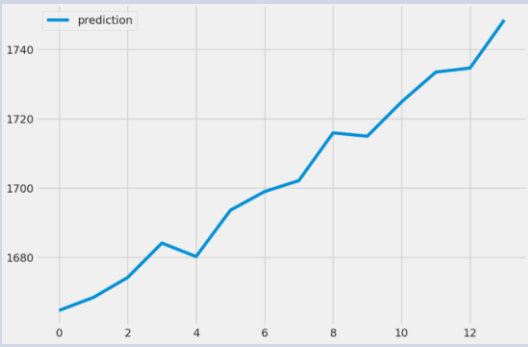
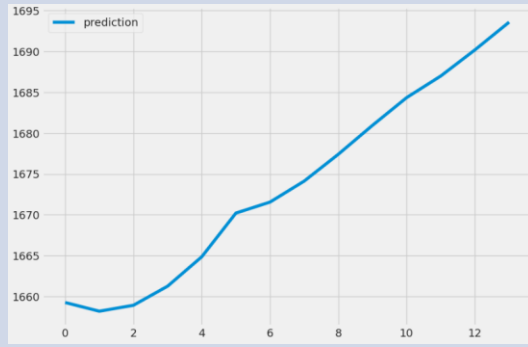
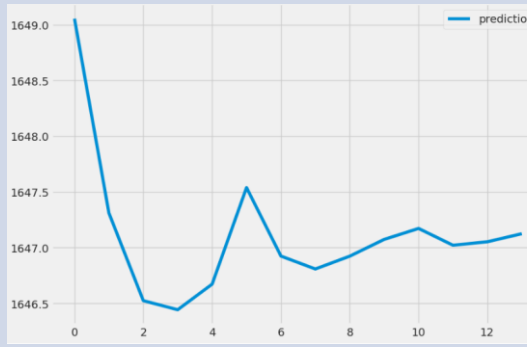
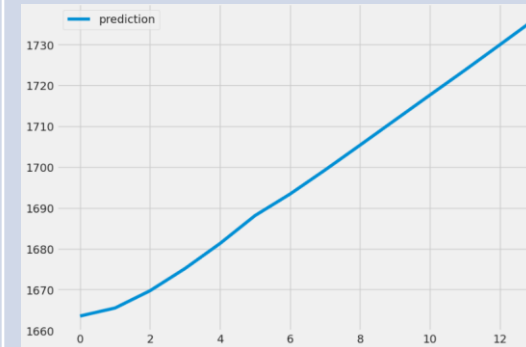
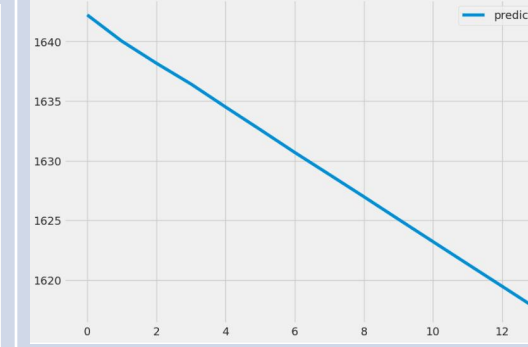


10



50

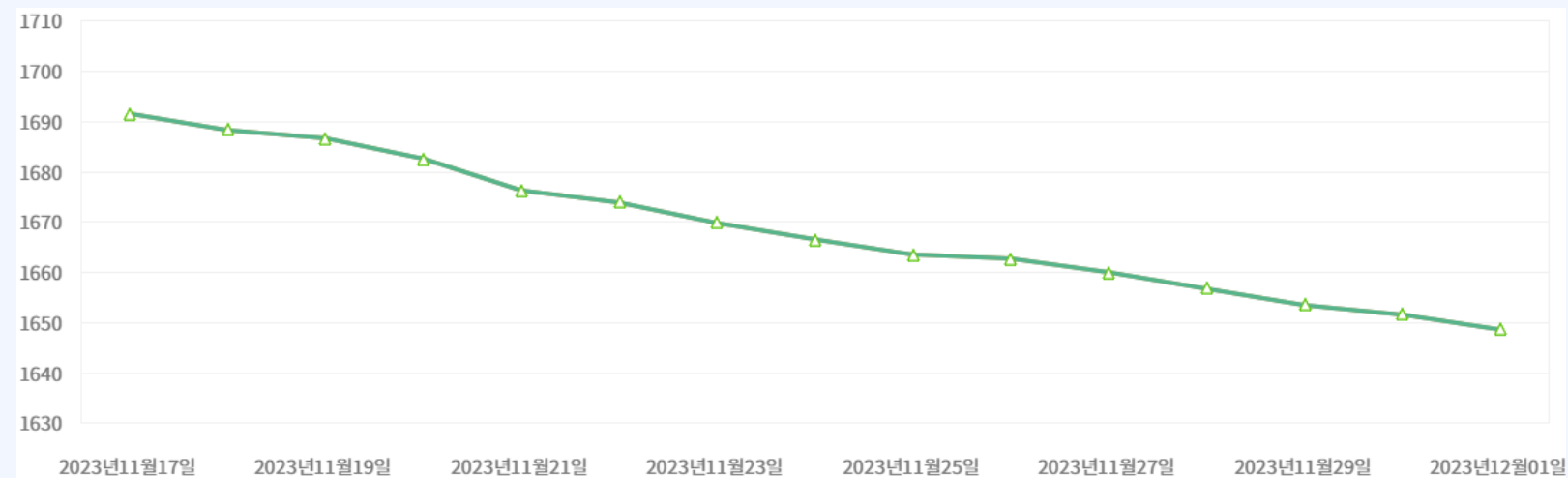


	RNN	LSTM	LSTM(R)	GRU	GRU(R)
미래 예측 시각화					
RMSE	22.266472496177794	17.062587596953378	11.121208140208152	20.085190201586553	10.78635695817664
성능 순위	5	3	2	4	1
파라미터 수	3234	12738	12738	9762	22401

*X_train(5일), y_train(2일), X_test(22.7.22~)로만 train test split했기 때문에 model.evaluate로 accuracy 측정 불가

*RMSE만으로 모델 성능 평가

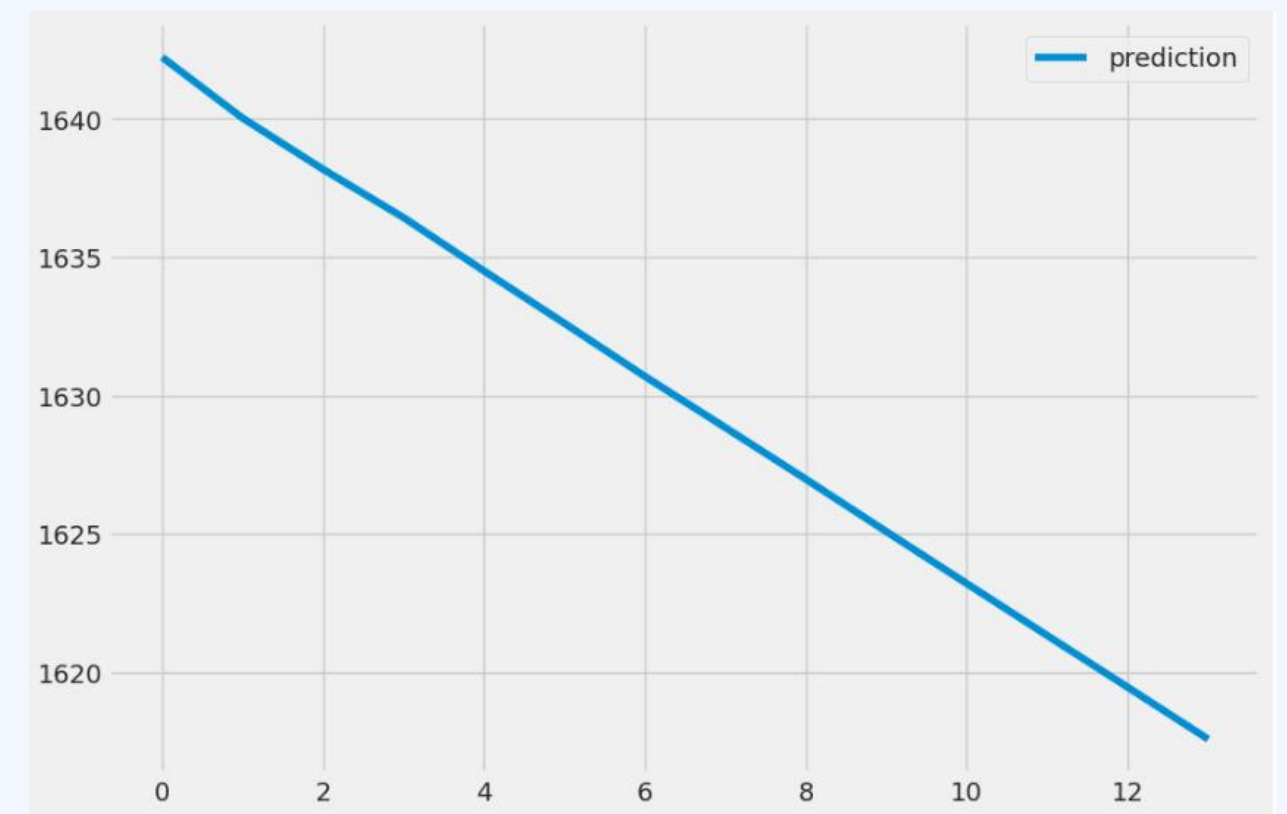
지난 2주 간 유가 그래프



	11-24	11-25	11-26	11-27	11-28	11-29	11-30
예측 값	1641.73	1639.38	1637.01	1633.49	1630.28	1628.82	1626.18
실제 값	1649.56	1647.05	1645.97	1643.95	1641.03	1638.49	1636.75

앞으로 2주

GRU(R)



12월 1일

12월 14일

감사합니다