목차 a table of contents

Part 1

연구목표

Part 1

연 구 목 표

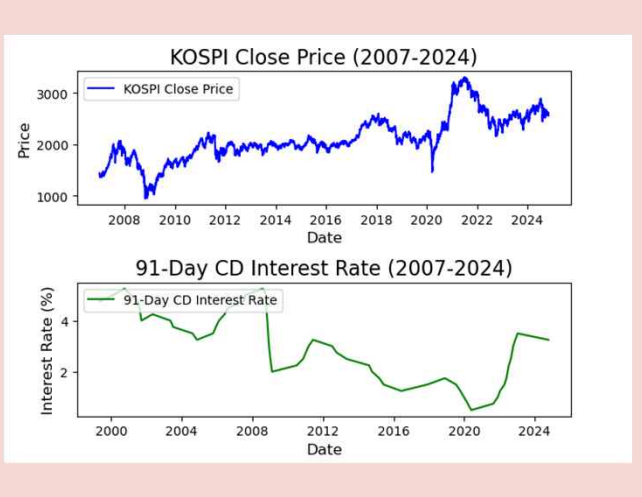
1.금리 상승/하락기의 주가 연관성 도출

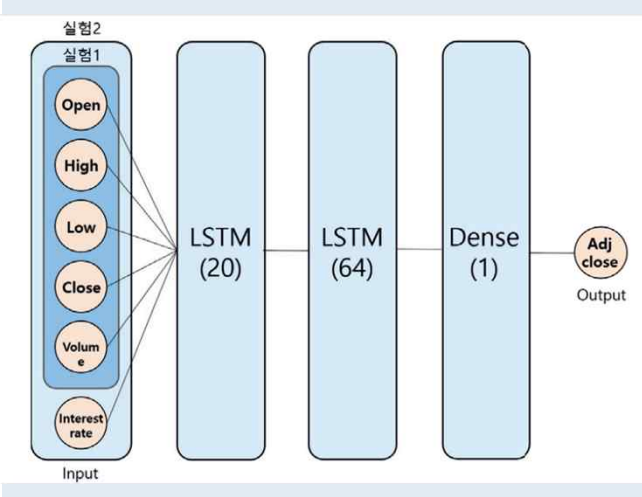
2.금리 유무의 영향 비교

3. 금융위기와 COVID-19 시기의 주가 변동성

Part 1

연구방법론





$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

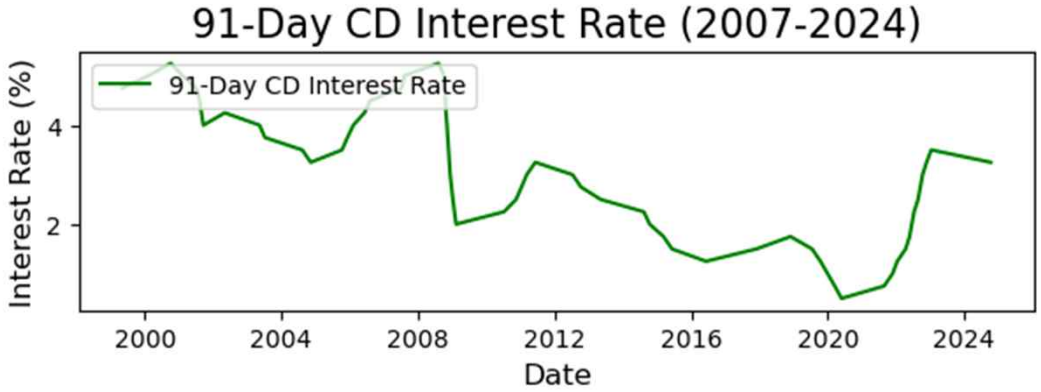
$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

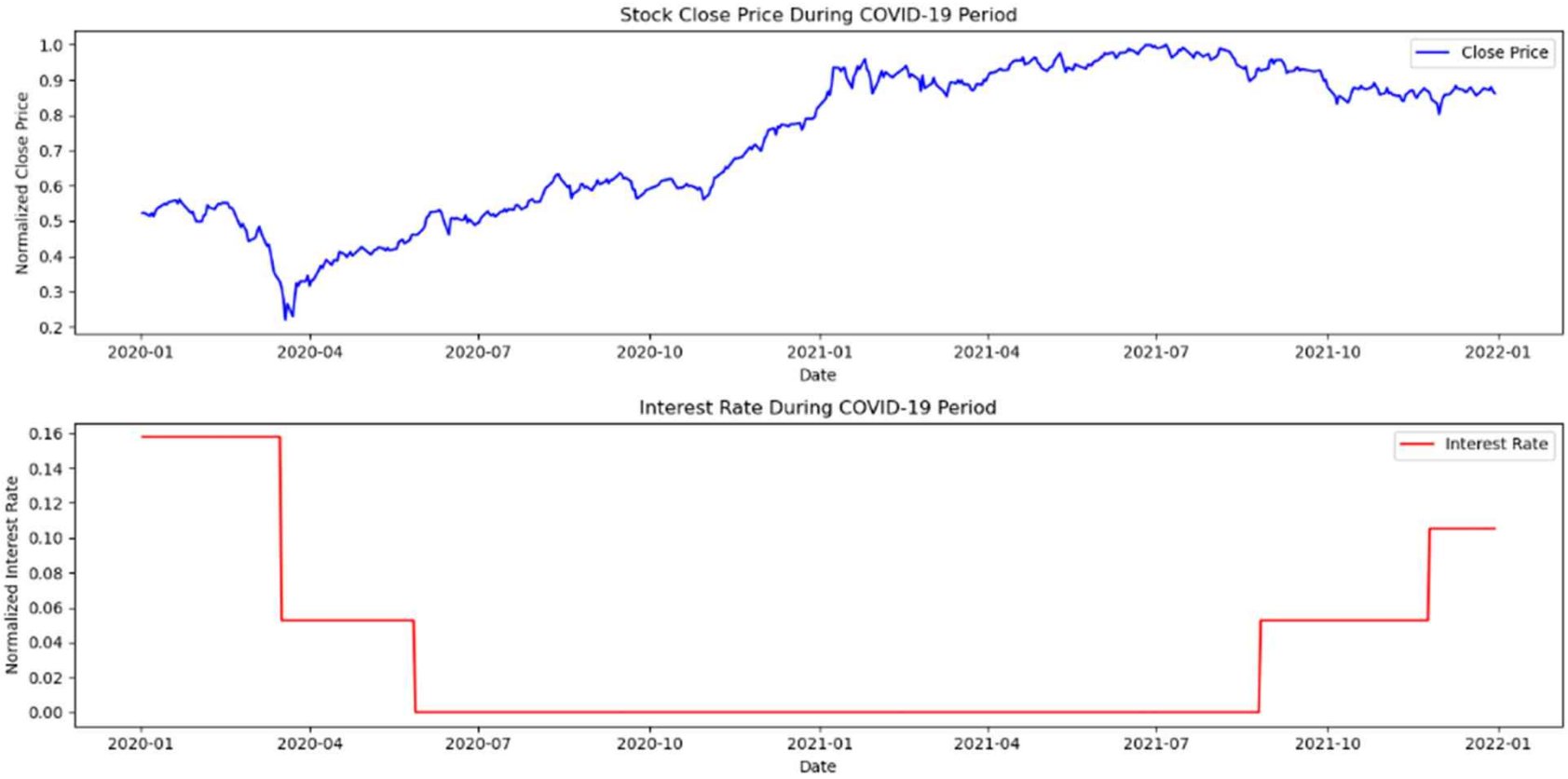
데이터

모델

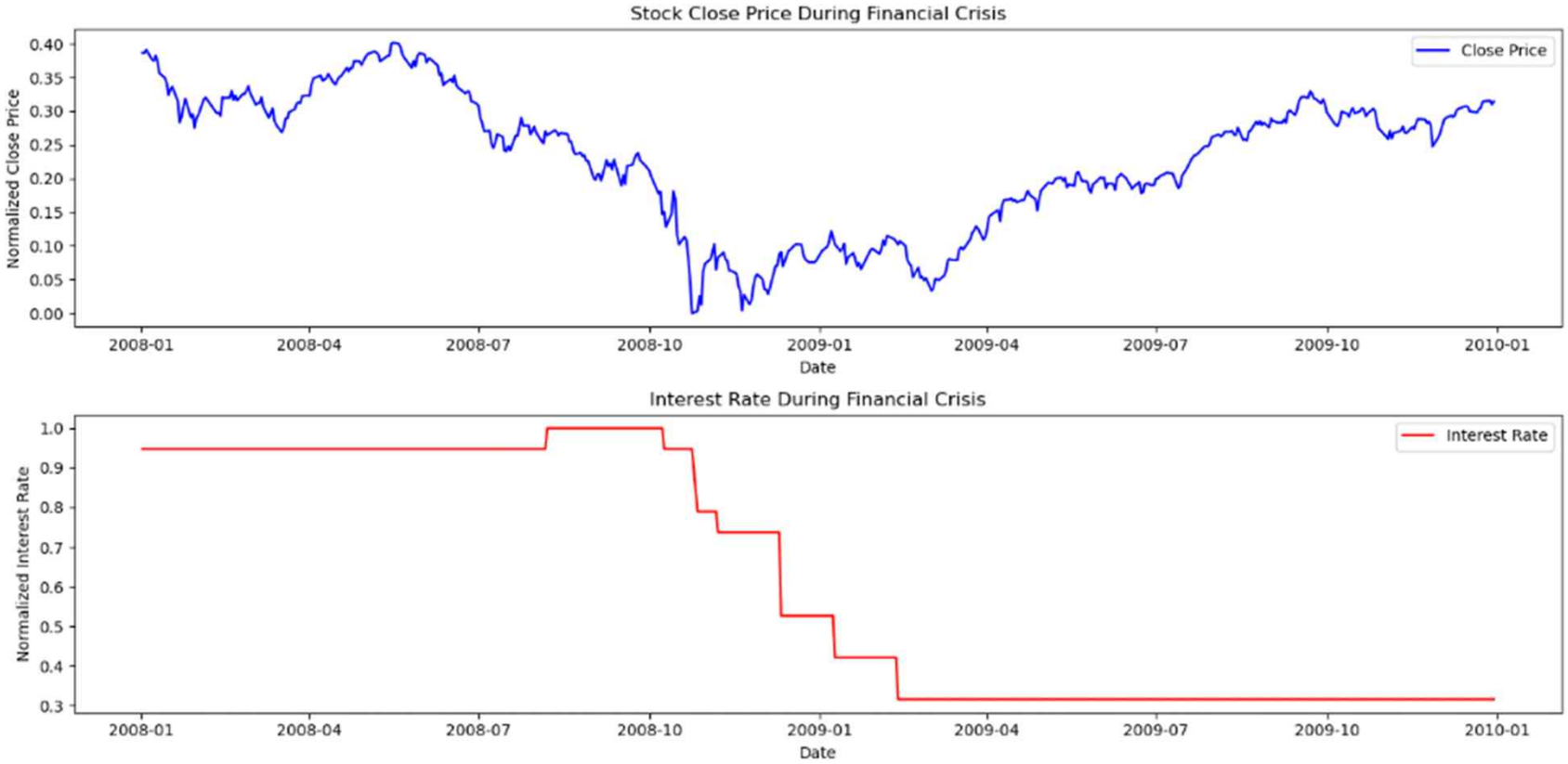
평가지표



코로나 기간 주가와 금리



금융위기기간 금리와 주가



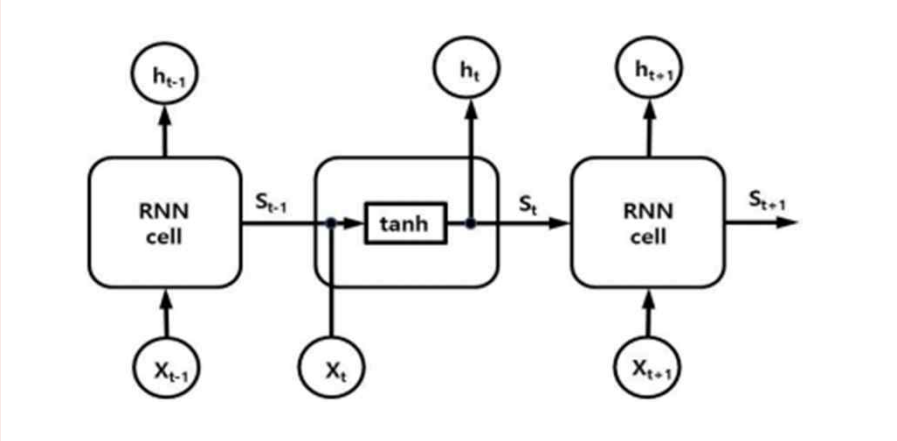
금리가 연속으로 내렸을(올랐을) 때 주가와 상관관계

	Close_normalized	Interest Rate_normalized
Close_normalized	1.00000	-0.78217
Interest Rate_normalized	-0.78217	1.00000

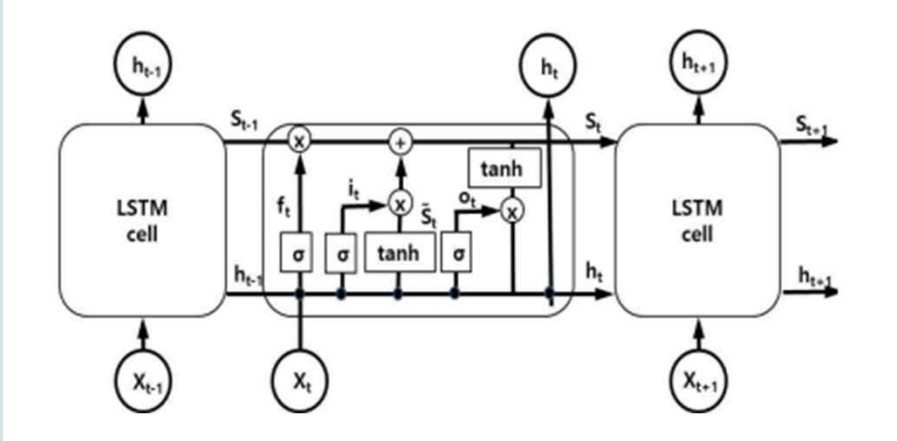
	Close_normalized	Interest Rate_normalized
Close_normalized	1.000000	-0.685661
Interest Rate_normalized	-0.685661	1.000000

Part 1

RNN과 LSTM



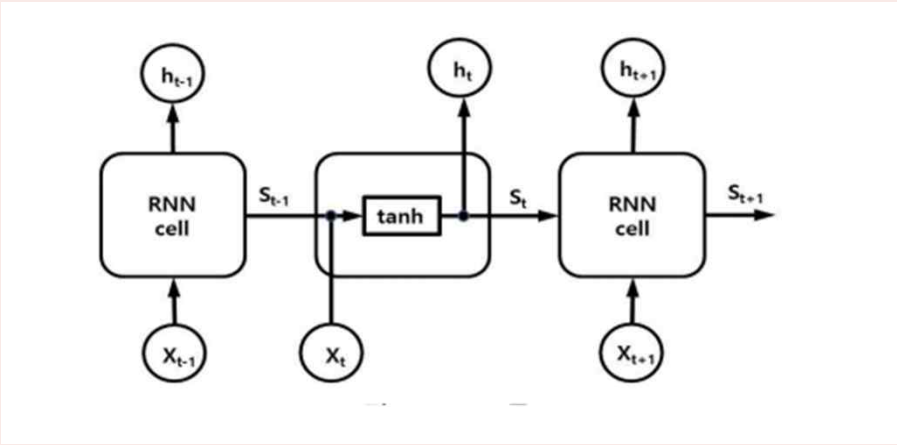
RNN



LSTM

Part 1

RNN과 LSTM

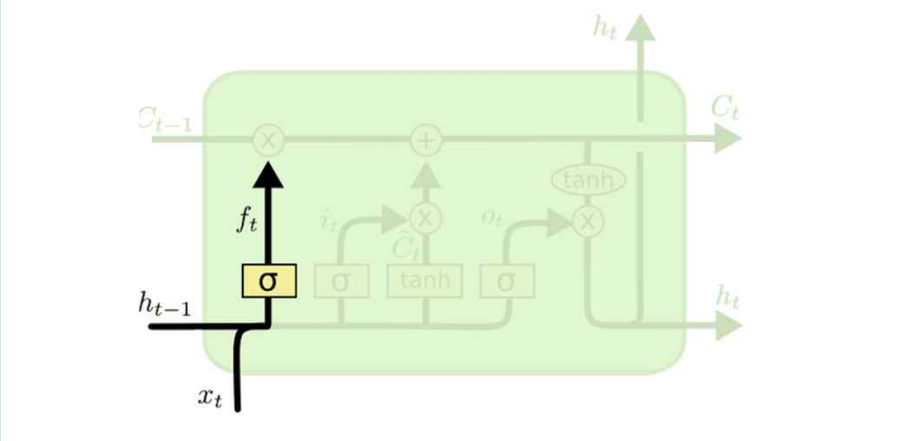


RNN

- DNN 병렬 체인 구조
- 딥러닝 네트워크
- 시계열 데이터 처리
- 의존성문제

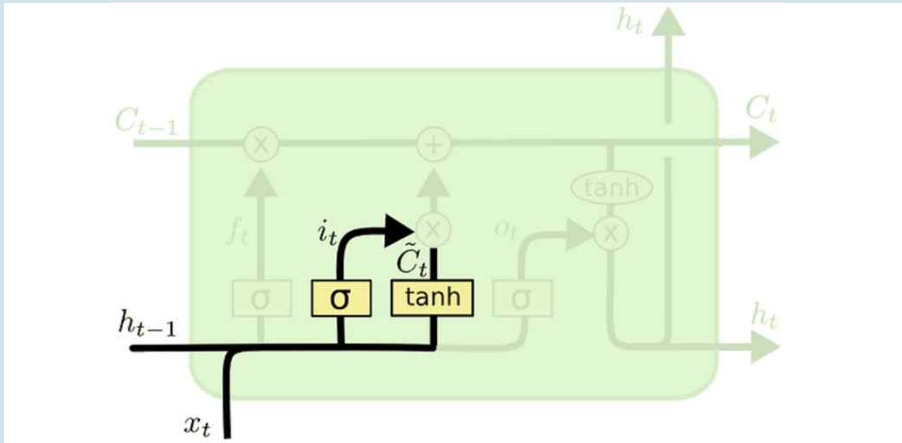
Part 1

RNN과 LSTM



- 1단계 : sigmoid 함수를 이용해 삭제할 정보를 결정한다.

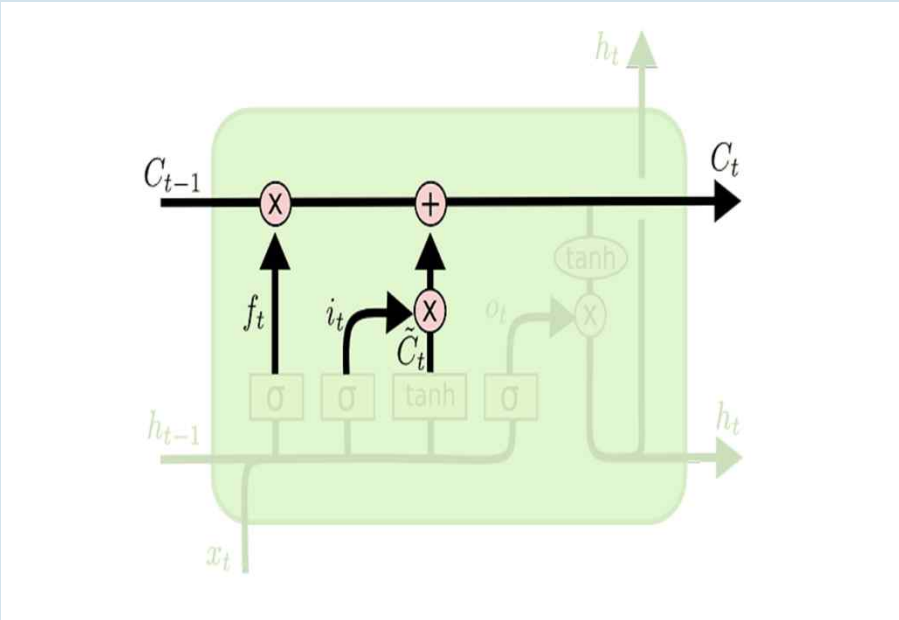
RNN과 LSTM

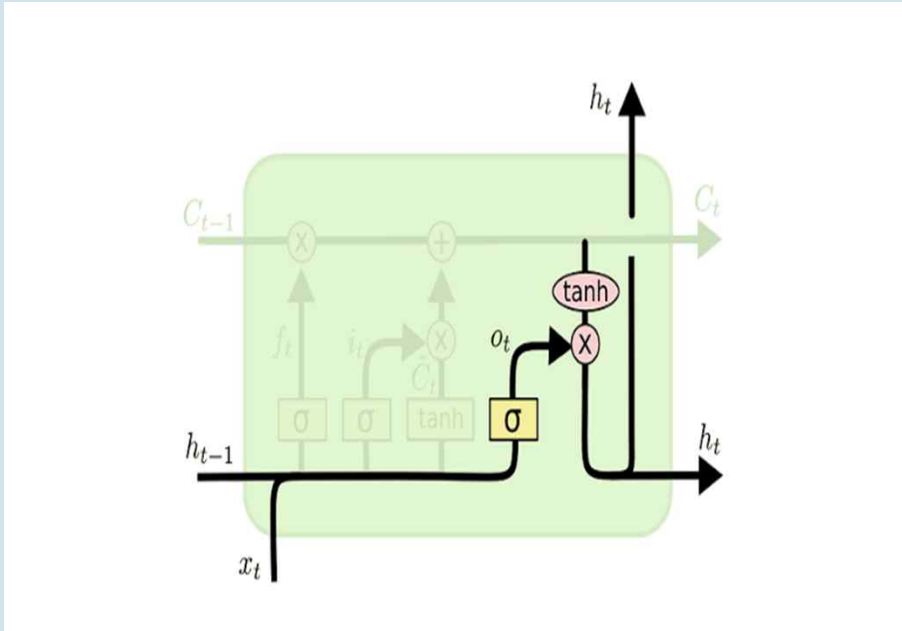


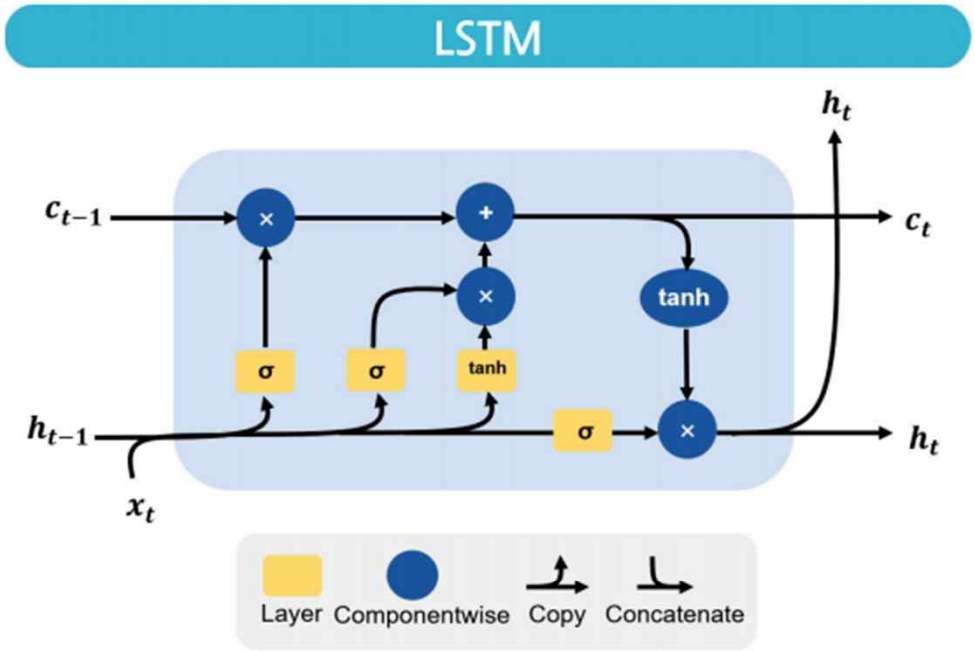
- 2단계 : sigmoid 함수와 tanh 함수를 이용해 새로운 정보가 셀 스테이트에 저장된다.

Part 1

RNN과 LSTM







LSTM은 RNN의 장기 의존성 문제(long-term dependencies)를 해결하기 위해서 나온 모델임
직전 데이터 뿐만 아니라 좀 더 거시적으로 과거 데이터를 고려해 미래 데이터를 예측함

Forget Gate

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

Input Gate

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\tilde{c}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Output Gate

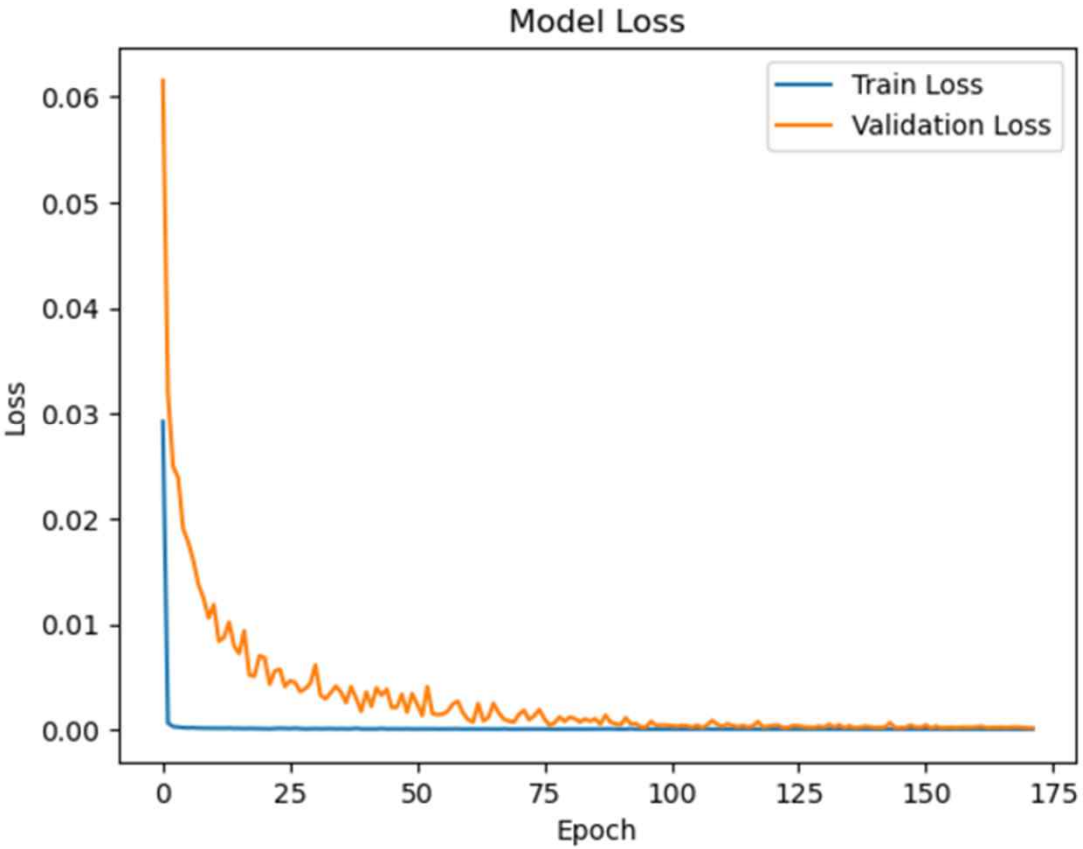
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$
$$h_t = o_t * \tanh(f_t * c_{t-1} + i_t * \tilde{c}_t)$$

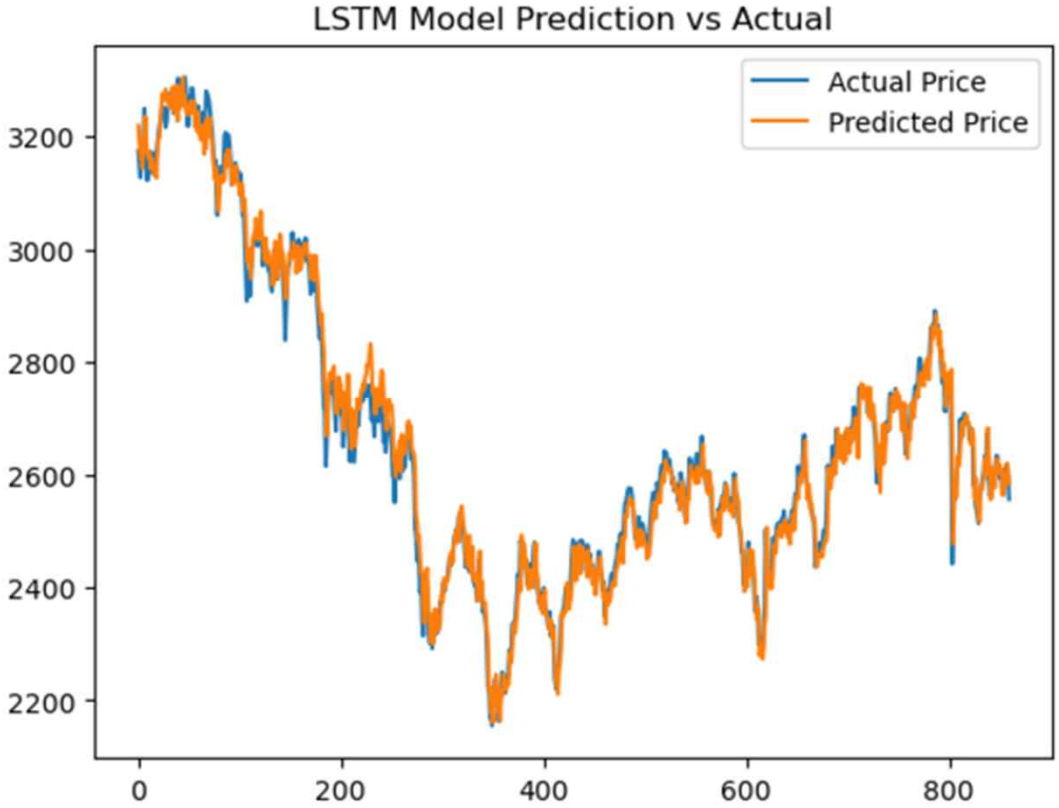
LSTM 모델훈련(금리포함)

```
# 입력 데이터를 LSTM에 맞는 형태로 변환
def create_dataset(X, y, window_size=20):
    Xs, ys = [], []
    for i in range(window_size, len(X)):
        Xs.append(X[i-window_size:i])
        ys.append(y[i])
    return np.array(Xs), np.array(ys)

# 훈련 데이터와 테스트 데이터를 LSTM 입력 형태로 변환
X_train_lstm, y_train_lstm = create_dataset(X_train, y_train, window_size=20)
X_test_lstm, y_test_lstm = create_dataset(X_test, y_test, window_size=20)

# LSTM 모델 구성
model = Sequential()
model.add(LSTM(units=50, return_sequences=False, input_shape=(X_train_lstm.shape[1], X_train_lstm.shape[2])))
model.add(Dense(units=1))

# 모델 컴파일
model.compile(optimizer='adam', loss='mean_squared_error')
```

```
from tensorflow.keras.callbacks import EarlyStopping

# EarlyStopping 콜백 정의
early_stopping = EarlyStopping(monitor='val_loss', patience=20, restore_best_weights=True)

# 모델 훈련 (EarlyStopping 콜백 포함)
history = model.fit(X_train_lstm, y_train_lstm, epochs=500, batch_size=32,
                    validation_data=(X_test_lstm, y_test_lstm), callbacks=[early_stopping])
```

모델의 손실 시각화





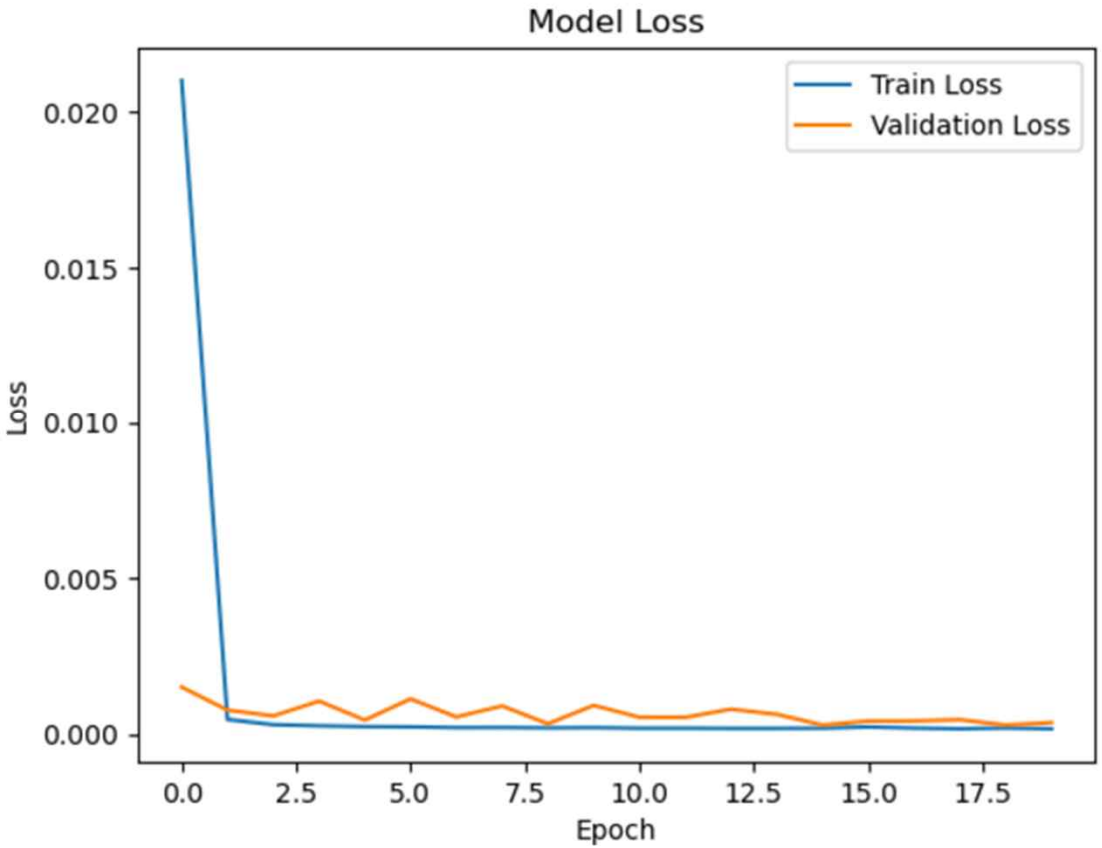
MAE: 25.887422215661328
MSE: 1144.0024635984448
R² : 0.9841189994352509

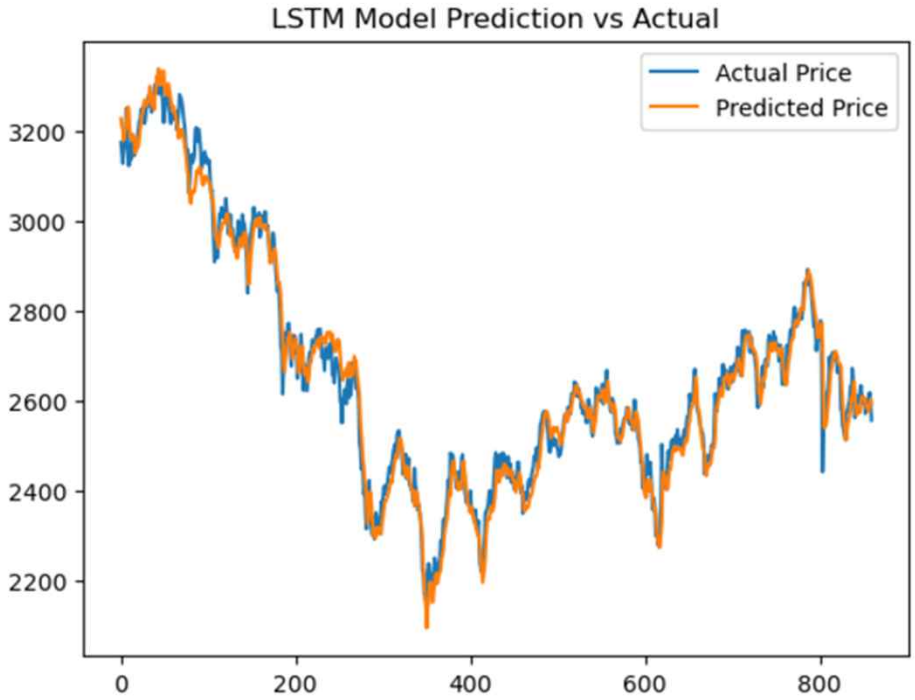
LSTM 모델훈련(금리 미포함)

```
# 금리 데이터를 제외하고 입력 데이터 구성
X_no_interest = data[['Open', 'High', 'Low', 'Close', 'Volume']].values
y_no_interest = data['Close'].values.reshape(-1, 1)

# 데이터 스케일링
X_no_interest_scaled = scaler_x.fit_transform(X_no_interest)
y_no_interest_scaled = scaler_y.fit_transform(y_no_interest)

# 데이터 분할 (훈련 데이터와 테스트 데이터)
train_size = int(len(data) * 0.8)
X_train_no_interest, X_test_no_interest = X_no_interest_scaled[:train_size], X_no_interest_scaled[train_size:]
y_train_no_interest, y_test_no_interest = y_no_interest_scaled[:train_size], y_no_interest_scaled[train_size:]
```

```
# 입력 데이터를 LSTM에 맞는 형태로 변환
X_train_lstm_no_interest, y_train_lstm_no_interest = create_dataset(X_train_no_interest, y_train_no_interest, window_size=20)
X_test_lstm_no_interest, y_test_lstm_no_interest = create_dataset(X_test_no_interest, y_test_no_interest, window_size=20)

# LSTM 모델 구성
model_no_interest = Sequential()
model_no_interest.add(LSTM(units=50, return_sequences=False, input_shape=(X_train_lstm_no_interest.shape[1], X_train_lstm_no_interest.
model_no_interest.add(Dense(units=1))

# 모델 컴파일
model_no_interest.compile(optimizer='adam', loss='mean_squared_error')

# 모델 훈련
history_no_interest = model_no_interest.fit(X_train_lstm_no_interest, y_train_lstm_no_interest, epochs=500, batch_size=32,
                                            validation_data=(X_test_lstm_no_interest, y_test_lstm_no_interest),
                                            callbacks=[early_stopping])
```

모델의 손실 시각화



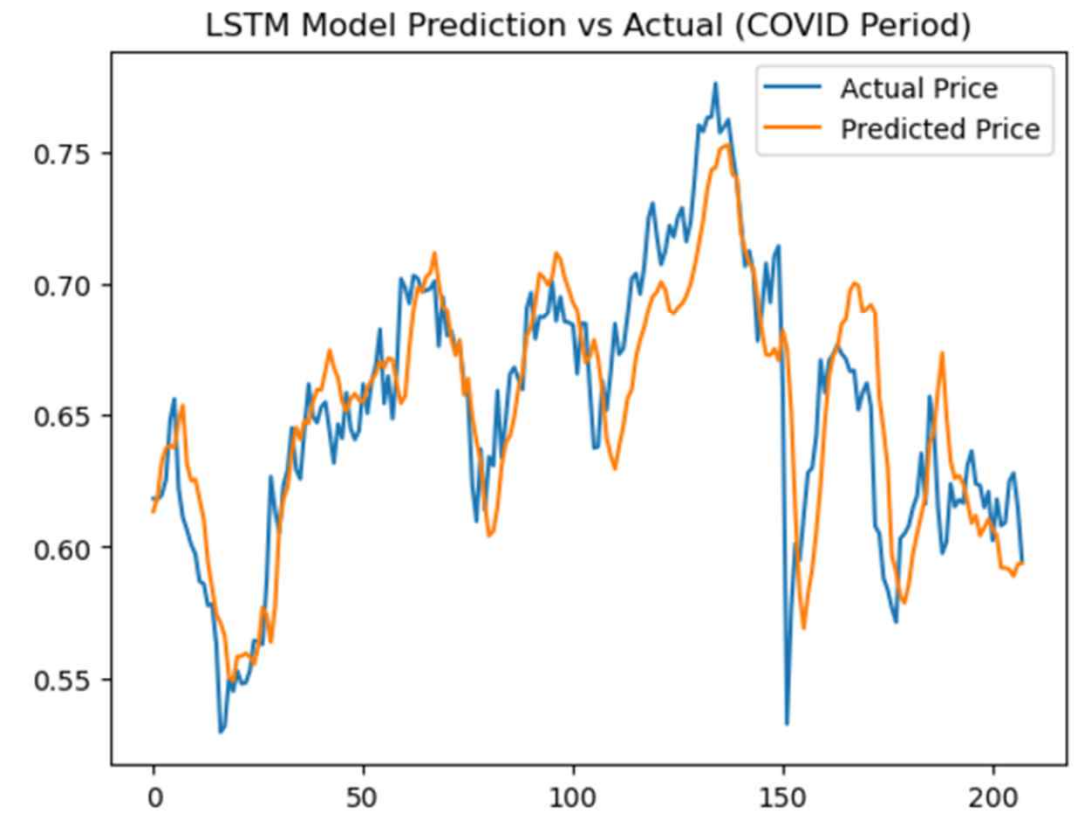



```
MAE: 30.535062568132258
MSE: 1569.6348086820092
R² : 0.9782103867113012
```

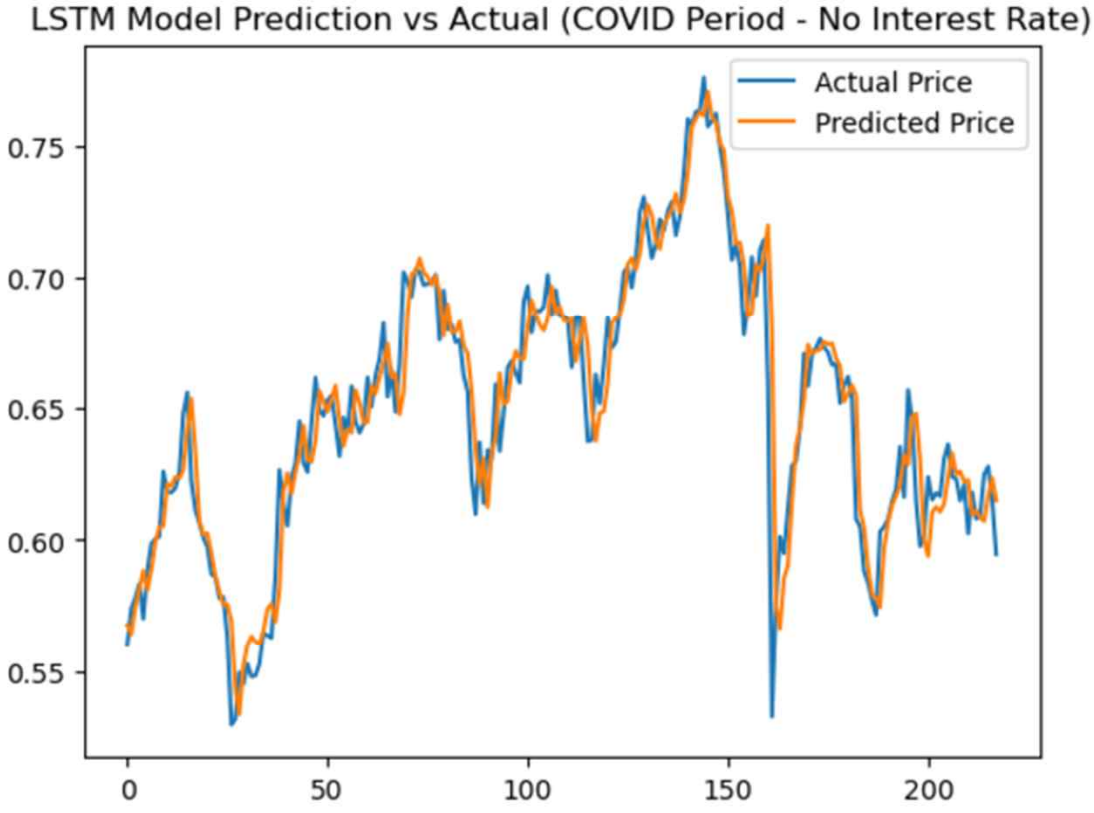
MAE 15%
MSE 27%
R-square 0.5%

금리가 포함되었을 때 평가지표가 개선되었다.

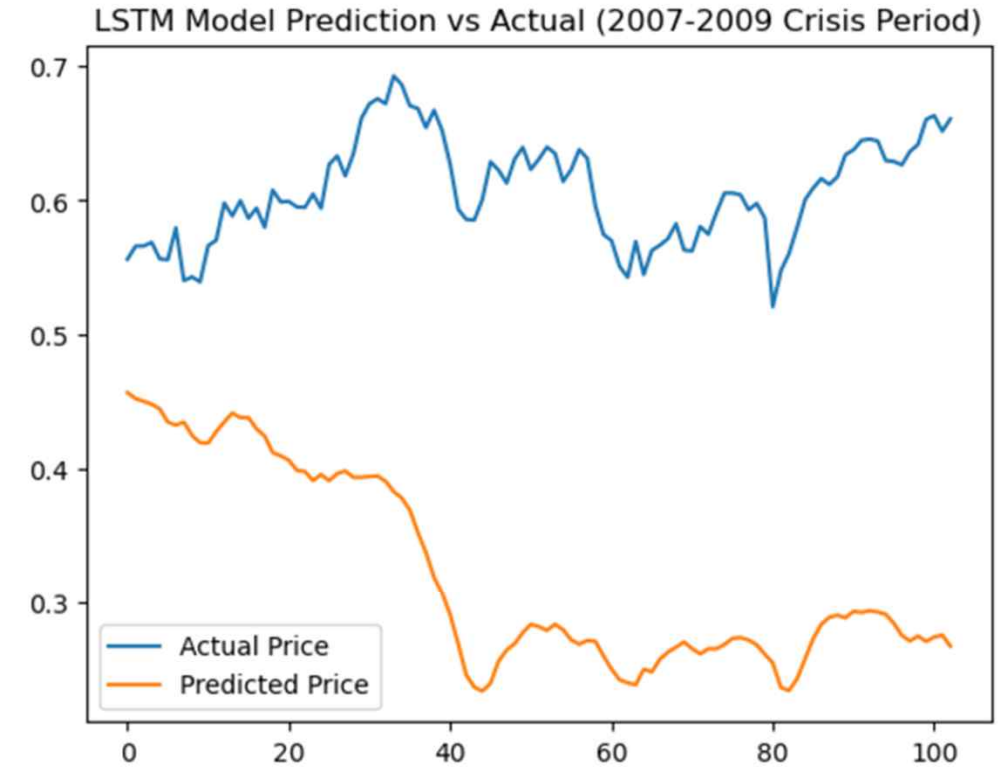
$$\text{개선율} = \frac{(\text{기존 모델의 성능} - \text{개선된 모델의 성능})}{\text{기존 모델의 성능}} \times 100$$



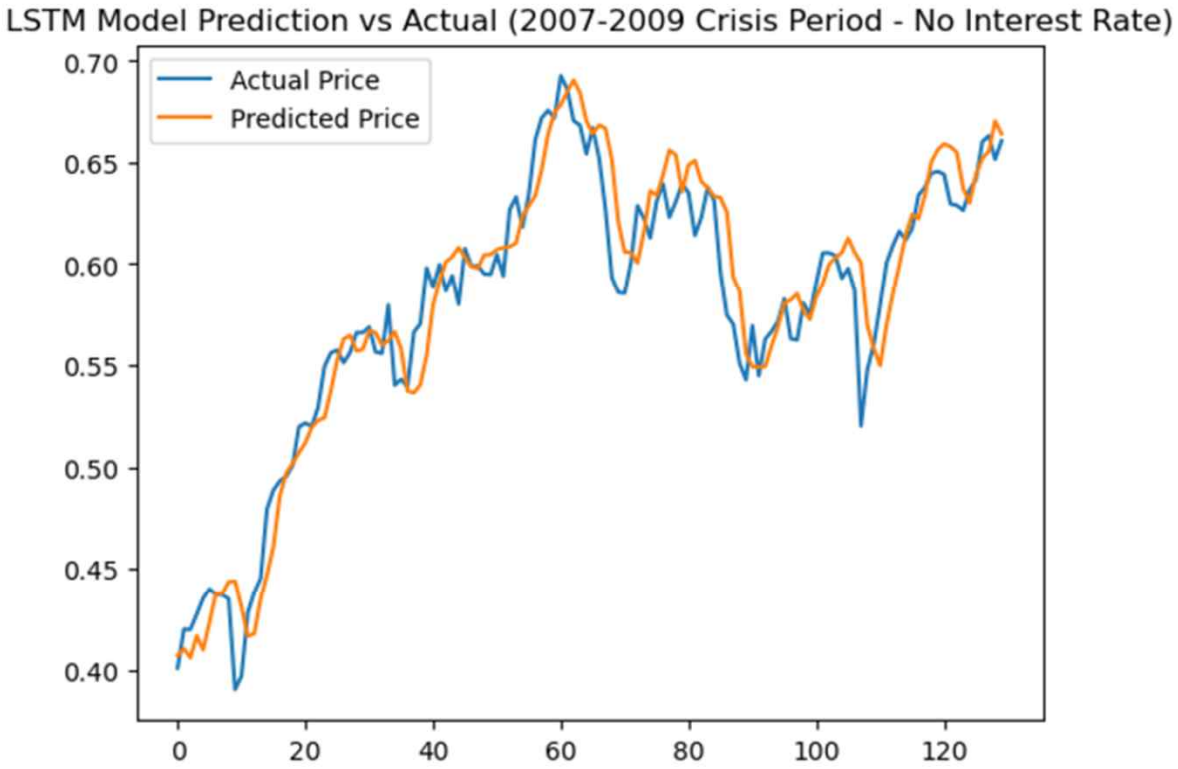
MAE (COVID Period): 0.020182289865038585
MSE (COVID Period): 0.0007190563460088061
R² (COVID Period): 0.7222140736508684



MAE (COVID Period - No Interest Rate): 0.011750266930740065
MSE (COVID Period - No Interest Rate): 0.00032208991549272705
R² (COVID Period - No Interest Rate): 0.8792034472102574



MAE: 0.28523242868552123
MSE: 0.08827565611890059
R² : -59.843217246512985



MAE: 0.015126554216065949
MSE: 0.0004048505836859476
R² : 0.9146349203609007

Part 1

결 론

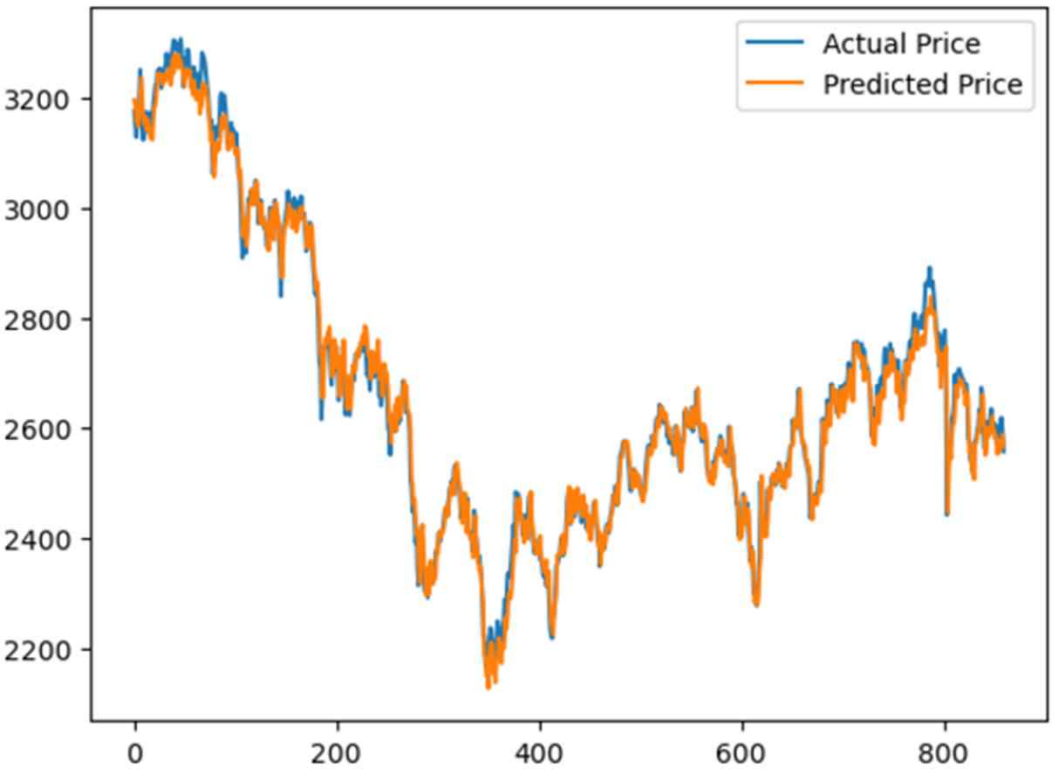
1. KOSPI 지수의 기본적 정보(시가, 고가, 저가, 종가, 거래량)를 이용해 익일 조정 종가의 등락을 예측하는 모델에서 금리가 어떤 영향을 주는지 파악함

2. 금리를 추가한 모델을 제시하고 테스트 결과에서 실제 주가와 비교해 성능 비교함

3. 연속적으로 금리가 상승(또는 하락)되었을 때 주가와 상관성이 있다.

4. 장기적인 측면에서 금리는 주가에 영향을 미친다.

5. 단기적인 측면에서는 금리는 주가에 영향을 미치지 않는다.



MAE: 24.73375897074854
MSE: 1026.8061197437514
R² : 0.9857459148153879

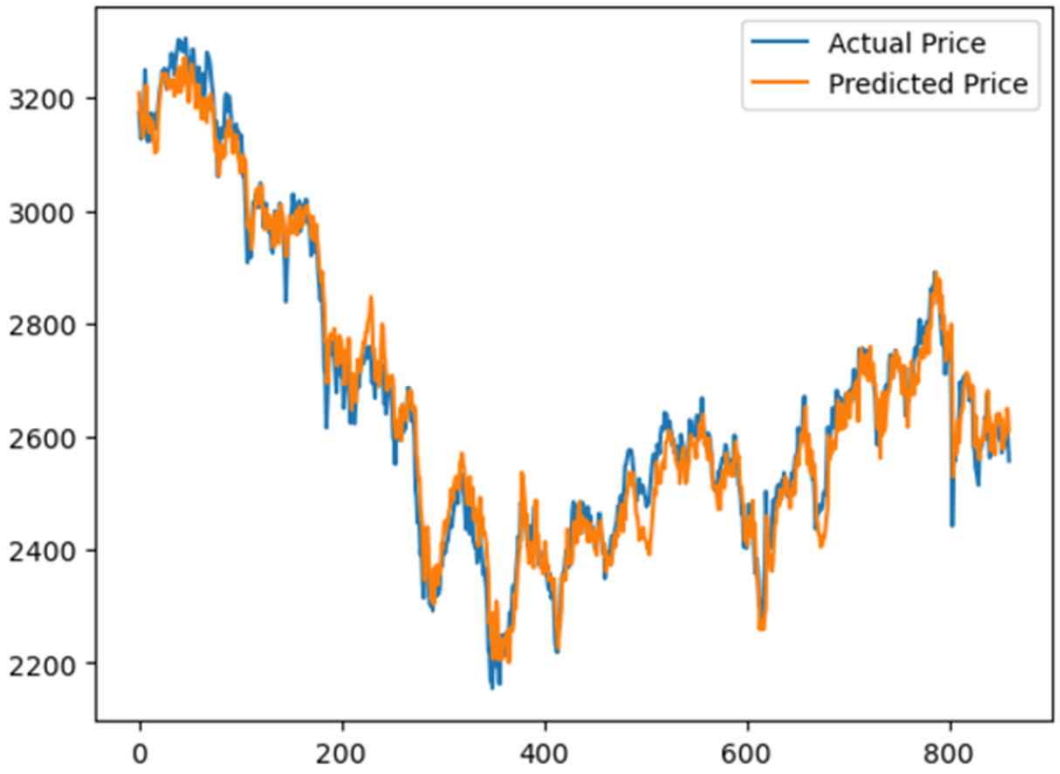
환율 포함

MAE: 25.887422215661328
MSE: 1144.0024635984448
R² : 0.9841189994352509

금리포함

MAE: 30.535062568132258
MSE: 1569.6348086820092
R² : 0.9782103867113012

금리&환율 미포함



MAE: 35.91142890397892
MSE: 2051.6550347865314
R² : 0.9715189994752039