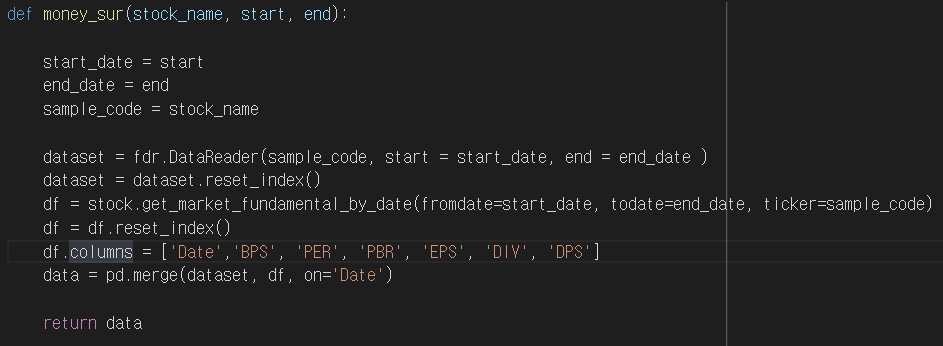
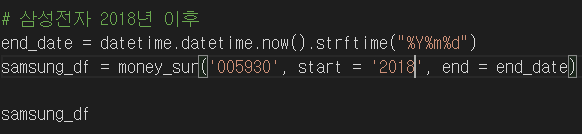
**코드 정의서**

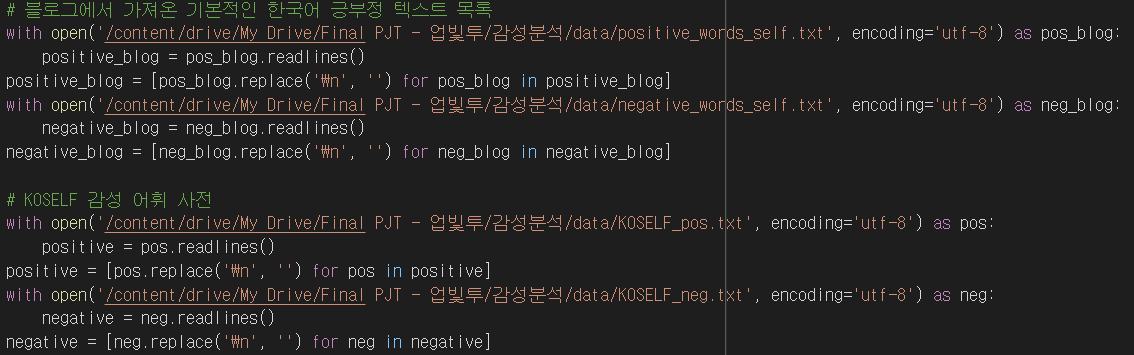
1. **데이터 로드 및 전처리**
   1. **주식 데이터 및 재무 지표 로딩 함수**



* 주식데이터는 FinanceDataReader 패키지를 통해서 해당 기간의 해당 업종 시가, 종가 등의 데이터를 가져오고, 재무 지표의 경우 pykrx 패키지를 통해서 ‘BPS’, ‘PER’, ‘PBR’ 등의 지표 데이터를 가져온다.
* 실제 로딩

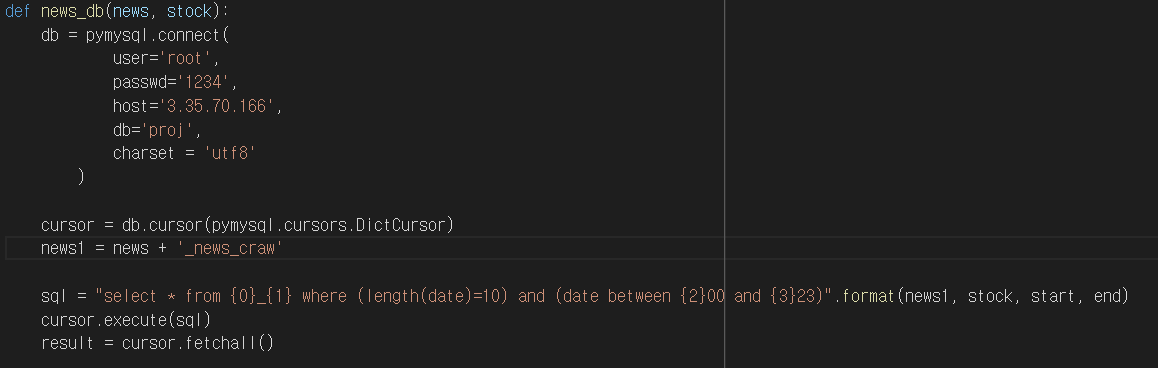


* 1. **텍스트 데이터 로딩 함수**
* 긍, 부정 텍스트 목록 및 KOSELF 감성사전 및 불용어 사전 로딩

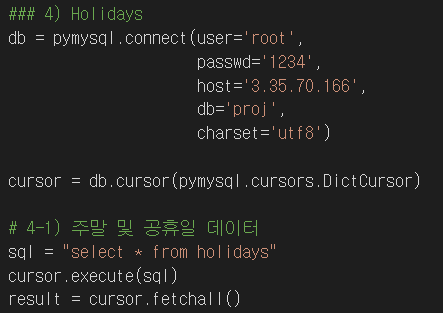




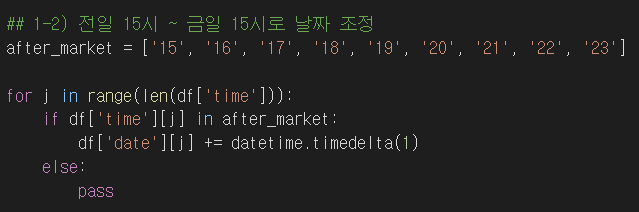
* txt 파일을 로딩 후 리스트로 전환
* DB에서 데이터 로딩

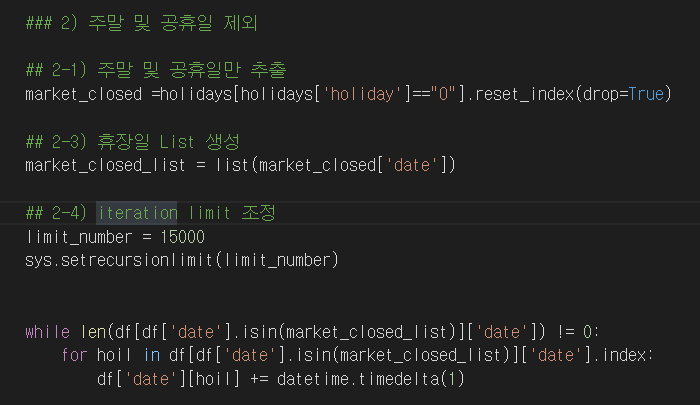


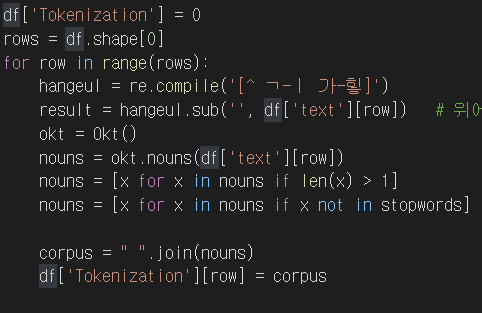
* 미리 크롤링해둔 데이터를 로딩 해서 사용
* 휴장일 데이터 로딩

****

* 1. **데이터 전처리**
* 휴장일 처리

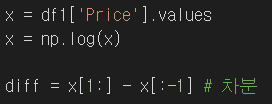




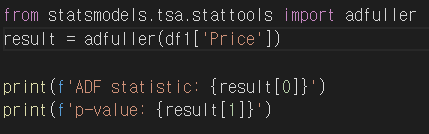
* 휴장일 및 시간에 따라서 날짜 조정
  1. **텍스트 토큰화**
* 텍스트 토큰화 및 불용어 처리
* 
* 한글 텍스트만 남긴 후 konlpy 패키지를 통해 명사만 추출
* 단어길이가 1이하인 단어는 삭제하고, 불용어 사전을 통해서 불용어 처리

1. **사용 모델 코드**
   1. **ARIMA**

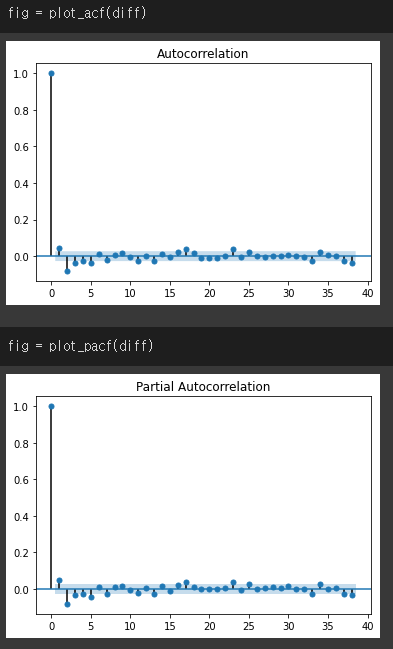
* 파라미터 조정
* 로그 변환과 차분과정



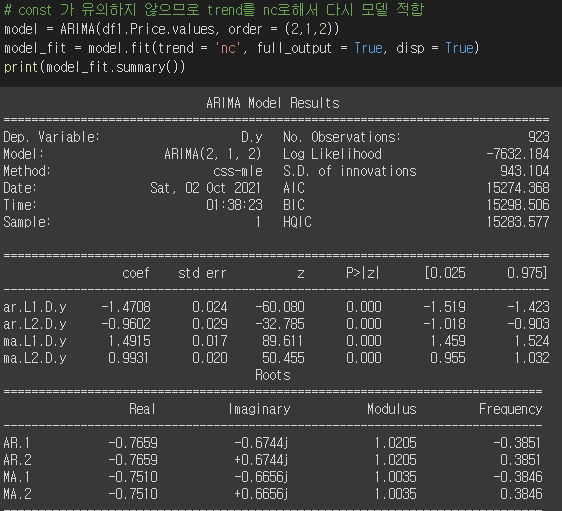
* 차분 후 정상성 검정(ADF 검정)



* 적절한 ar, ma 값 구하기



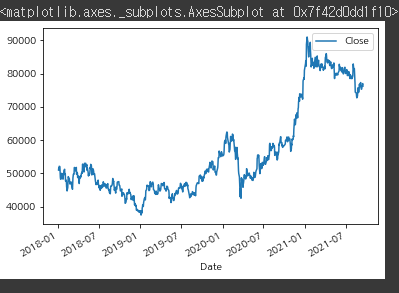
* adf, pacf 그래프를 통해서 적절한 ar,ma 값 구한다.
* 조정 후 모델 적합



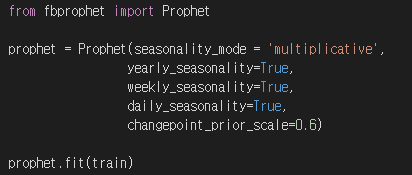
* 위에서 구한 파라미터를 조정하여 최종 모델을 만들고 훈련데이터 셋을 적합
* 예측



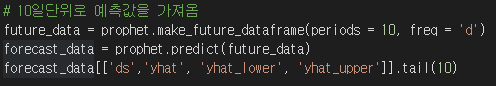
* 이후 10일을 예측
  1. **FBProphet**
* 파라미터 조정



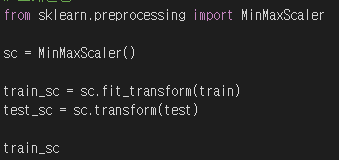
* 주식 그래프를 확인해 본 결과 진폭이 점점 증가하는 그래프이므로 seasonality\_mode 는 ‘multiplicative’ 로 결정, changepoint\_pior\_scale 의 경우 0.5, 0.6, 0.7를 해본 결과 0.6의 결과가 오차가 제일 적게 나옴을 확인
* 파라미터 조정 후 모델 적합

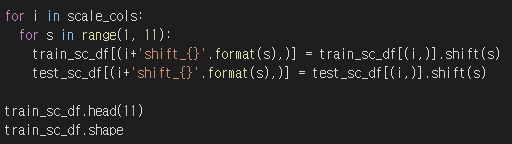


* 파라미터를 조정하여 모델 적합
* 예측

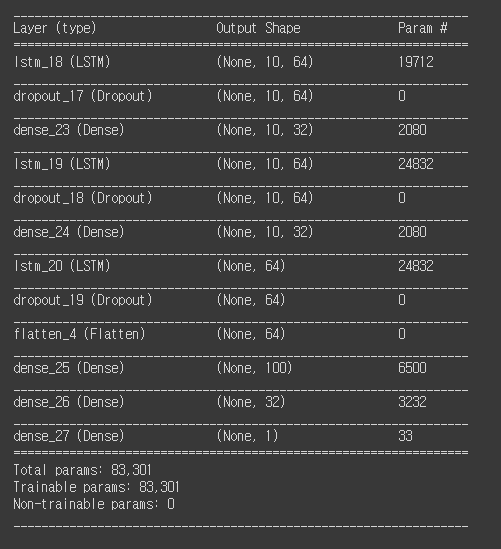


* 10일 단위로 예측값을 가져옴, 예측값 목록은 날짜, 예측값, 예측 하한, 상한 값으로 이루어져 있다.
  1. **LSTM**
* 데이터 전처리

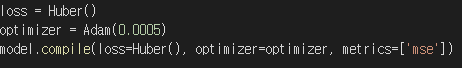




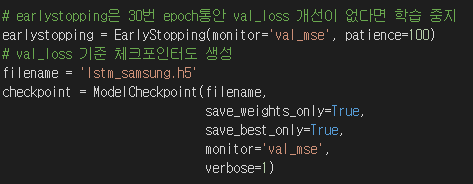
* 데이터 스케일링 및 window size를 10일 기준으로하여 데이터 전처리를 진행
* layer 확인

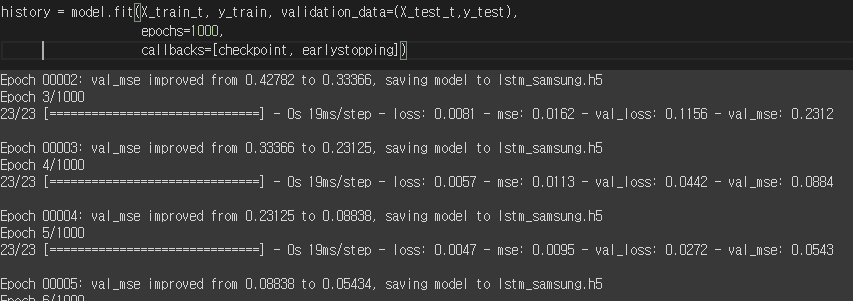


* lstm 층과 dense 층을 번갈아가면서 진행 과적합을 방지하기위해서 중간중간 dropout을 시행하였다. 또한 층의 activation 함수는 relu와 tanh를 사용하였다.
* compiler 조정

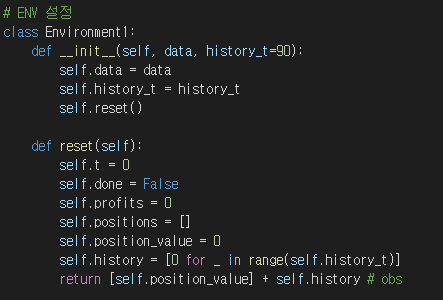


* loss 함수의 경우 adam과 비교하여 조금 더 좋은 성능을 보이는 huber 함수를 사용하였고, 평가 지표는 mse를 지정하였다.
* 모델 적합

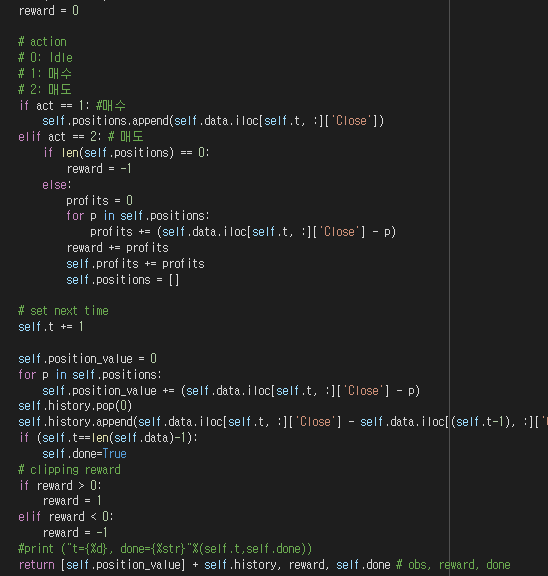




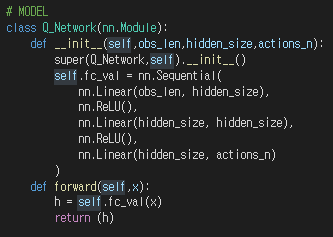
* 1000 epochs를 돌렸으나 checkpoint와 eariystoping을 지정하여 최적의 값이 나오면 모델을 저장후 정지하도록 하였다.
  1. **강화학습**
* 환경 설정



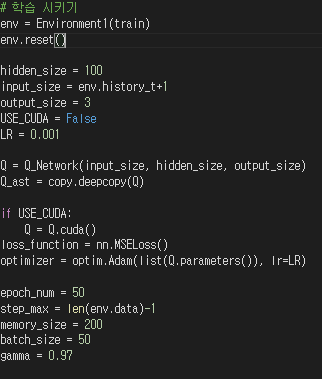
* 모델에 적합할 수 있게 환경을 설정 해준다.



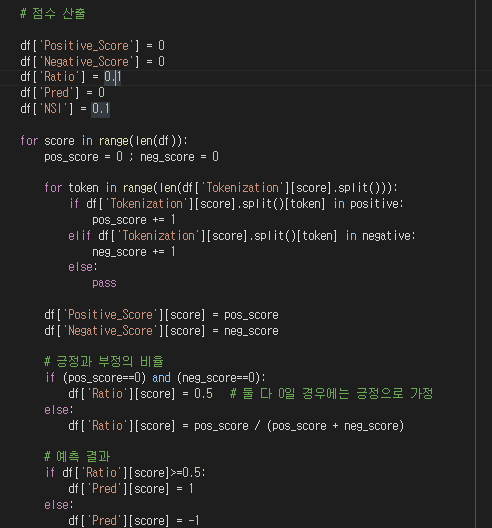
* 모델의 step 당 reward 설정
* model



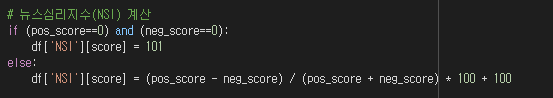
* 모델 설정 간단하게 linear와 relu를 반복하는 layer로 구성
* 학습



* 데이터 환경을 조정한 후에 모델에 적합
  1. **뉴스 텍스트 긍부정 라벨링**
* 긍 부정 점수 산출



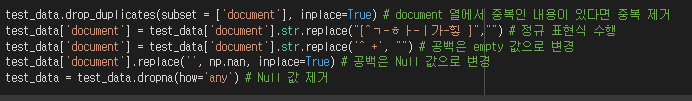
* 구축된 사전을 통해서 긍정, 부정 점수 산출 및 긍정 비율 산출
* 뉴스심리지수 NSI 구하기

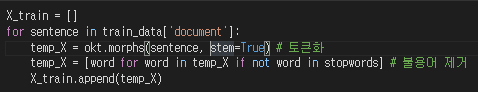


* 문장 수를 단어 수로 변환하여 계산, 공식은 기존 NSI 지수 산출법과 같다.
  1. **유트브 텍스트 긍부정 라벨링**
* 훈련 데이터 로딩

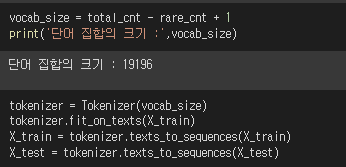


* 유튜브 특성상 맞춤법이 맞지 않고, 대화식으로 진행될 것으로 판단하여 라벨링된 데이터 중 네이버 영화 리뷰 데이터 사용하였다.
* 전처리 및 불용어 처리

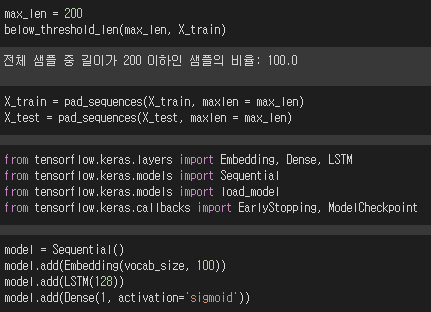




* 전처리 방식은 위 텍스트 전처리 방식과 동일
* Tokenizer 정의



* 단어 집합 크기를 구한 후에 집합 크기에 맞게 Tokenizer를 정의, 훈련데이터에 적합
* 모델 적합



* max\_len = 200으로 하여 pad\_sequences를 진행한 후에 모델에 적합
* layer는 Embedding을 진행한 후에 lstm층을 쌓고 sigmoid 함수를 통해 긍/부정을 도출