안녕하세요. 뉴스 기사를 이용한 넷플릿스 주가 예측 프로젝트를 발표할 시계열 1조입니다.

먼저 프로젝트에 대한 소개를 해드리고, 데이터 수집과 전처리 과정, 모델링, 결론과 한계점에 대해 설명드리겠습니다.

저희는 주가 예측이라는 다소 흔한 시계열 프로젝트를 선택하였지만 기존의 주가 예측 프로젝트들과는 2가지 관점에서 다른 시도를 해보는 것에 초점을 두었습니다.

우선 뉴스 기사를 활용하여 직접적으로 주가를 예측하고자 하였습니다.

아래 보이는 데이콘의 NH투자증권 대회 수상작들처럼 뉴스 기사를 주가 예측에 활용한 프로젝트들은 많이 있습니다. 그러나 대부분 직접적인 주가 예측 문제를 주식의 상승세 또는 하락세 이진분류문제로 우회하거나 인기있는 주식 종목을 추천해주는 프로젝트들이기 때문에 저희는 직접적으로 주식의 종가를 예측하는 시도를 하였습니다.

저희는 주식 종목을 넷플릭스 하나로 정했는데, 저희가 참고한 데이콘 대회에서는 뉴스 기사들의 url을 데이터로 제공해주었지만 저희의 경우에는 종목명으로 검색하여 직접 뉴스 기사를 수집해야 했기 때문입니다..

프로젝트 진행에 있어 두번째로 고려한 점은 주가의 장기적인 추세를 반영하는 것입니다. 깃헙이나 블로그, 또 교재에 있는 대부분의 주가 예측들이 1일에서 5일의 데이터를 입력받아 다음 하루를 예측하는 것들입니다. 이런 방식으로 모델을 훈련시키면 당연히 모델은 손실값을 최소화하기 위해 입력 시퀀스의 평균값으로 예측하는 가장 안전한 방법을 선택할 것입니다. 결과는 아래 그림처럼 이전날들의 가격을 그대로 수평이동시킨 그래프가 나타나게 됩니다.

따라서 이런 문제점이 발생하지 않도록 저희는 입력 시퀀스의 길이를 늘려서 모델이 주가의 장기적인 추세를 잘 반영하도록 의도했고, 그랬을 때 성능이 어떻게 변화하는지 실험을 통해 알아보았습니다.

프로젝트를 요약하자면, 최종 목표는 뉴스 데이터와 장기간의 입력 시퀀스를 통한 주가 예측에서 최적의 파라미터를 찾아내어 결론을 도출하는 것이고,

그 과정에서 LSTM, GRU, 트랜스포머 모델과 또 모델별로도 파라미터를 검증하며 여러 실험을 진행했습니다.

사용한 데이터는 주식 관련 데이터와 뉴스 관련 데이터 두가지입니다.

주가 데이터는 FinanceDataReader 라이브러리를 통해 2018년부터 2023년까지 6년 간의 시가, 고가, 저가, 종가, 거래량을 수집하였습니다. 또한 뒤에서 설명하겠지만, 데이터 전처리를 위해 넷플릭스와 유사한 사업을 하고 있고 또 주가의 동향도 비슷한 3개 종목의 종가 데이터도 같이 수집하였습니다.

뉴스 데이터에 대해선 처음에 웹 스크래핑을 통해 넷플릭스를 검색어로 하여 2018년부터 2023년까지의 뉴스 기사 17000여개를 수집하여 직접 금융분야에서 유명한 NLP모델인 FinBERT와 BERTopic 모델을 활용하여 감성분석과 토픽모델링을 시도했지만, 중간에 시행착오도 많았고 또 시계열 프로젝트에 집중하고자 API를 사용하여 데이터를 수집하는 것으로 우회하였습니다. 저희가 사용한 사이트는 stocknewsapi 라는 사이트로 주식 시장과 관련된 뉴스를 주로 취급하는 API 제공 사이트입니다.

API로 데이터를 수집한 과정은 다음과 같습니다. 해당 사이트에서는 넷플릭스 종목명으로 api를 요청하여 응답을 받았고 그 중에서 날짜와 토픽과 감성분석 결과를 수집하였습니다. 다만 토픽이 아무것도 나오지 않은 기사들이 있었고 이는 나중에 라벨인코딩 시 Undefined로 라벨링을 해주었습니다.

다음은 데이터 전처리에 대해 설명드리겠습니다.

우선 주식의 종가의 경우 증감폭이 크고 불규칙적이기 때문에 정상성을 띄는 피처를 추출하기 위해 종가의 변화율과 종가에 대한 이동평균 피처를 생성했습니다.

2018년 1월 8일을 예시로 설명드리면, 1 day ROC는 1월 7일보다 1월 8일의 종가가 몇 퍼센트 상승했는지를 나타냅니다. 5day ROC는 영업일 기준으로 1월 8일로부터 5일전의 종가와 비교한 변화율을 나타냅니다. Moving Average도 시계열 예측에서 자주 사용하는 기법으로 1월 8일을 기준으로 5MA를 설명하자면, 1월 2일날부터 1월8일까지의 종가를 평균낸 값입니다.

저희가 훈련에 사용한 데이터는 2019년부터 2023년까지의 데이터인데, 이런 ROC와 MA 피처를 생성하기 위해 2018년의 데이터도 수집했습니다.

그럼 훈련 데이터가 2019년부터 2023년까지로 다소 짧다고 느끼실 수 있는데, 안타깝게도 뉴스 api에서 제공하는 데이터 중 제일 오래된 것이 2018년 데이터이기 때문에 데이터셋의 기간을 이렇게 설정하게 되었습니다.

다음으론 금융 시계열 데이터에 대한 기술 분석 라이브러리인 TA 라이브러리를 활용하여 추가적인 피처를 생성했습니다.

아래와 같이 많은 피처들 중에서

한국투자증권에서 지원하는 37개의 지표를 선택했습니다.

다음으로는 넷플릭스와 사업 영역이 겹치고 주가 동향이 유사한 3개 기업의 종가도 피처로 추가했습니다.

이 때, 핀터레스트는 2019년 4월에 상장한 주식이기 때문에 결측치는 최초 상장 시 주가로 대체하였습니다.

다음으로 상관계수 히트맵을 통해 상관관계가 높은 피처에 대한 처리를 진행했습니다. 핵심 피처인 주가 관련 피처들은 그대로 유지하고 TA 라이브러리를 사용해 추가한 피처 중에서 주가와 상관관계가 큰 피처들은 drop했습니다. 그리고 넷플릭스와 유사한 동향을 보이는 Peer stock도 drop하지 않고 유지했습니다.

다음으론 api를 통해 수집한 뉴스 토픽과 감성분석 데이터에 대한 전처리를 수행했습니다. 토픽은 총 15개의 클래스가 있어서 라벨인코딩을 수행했고 감성은 3개의 클래스만 있고 토픽보다 중요한 피처라고 생각하여 원핫인코딩을 했습니다.

일반적으로 그날 올라온 뉴스 기사가 바로 주가에 영향을 미치지는 않기 때문에 뉴스 데이터는 모두 이동평균으로 피처를 생성하였습니다. MM은 이동평균에서 평균을 최빈값으로 바꾼 피처입니다.

그래서 최종적으로 주가와 주식관련 지표들만 있는 stockOnly\_df 데이터셋,

감성분석과 토픽 모델링 정보까지 포함되어 있는 total\_df 데이터셋 2개를 준비했습니다.

다음은 모델링과 전체적인 실험 과정에 대해 설명드리겠습니다.

우선 시계열 예측의 프로세스에 대해 간단히 설명하자면, 모델은 훈련과정에서 우리가 설정한 sequence length W만큼을 입력값으로 받아서 K만큼의 예측값을 출력하게 됩니다.

전체적인 실험 과정은 크게 3가지 단계로 나누어집니다.

시퀀스 길이를 바꾸면서 실험을 진행하기 때문에 모델 정의 이전에 시퀀스 길이와 predict size만큼 데이터를 분할하는 작업이 선행하게 됩니다.

다음으로 LSTM, GRU, Transformer 모델을 정의합니다. (다등ㅁ어)

실험에 이 3가지 모델을 선택한 이유는 다음과 같습니다. 우선 LSTM은 기존 주가 예측 프로젝트에서 가장 많이 사용된 모델로 성능이 좋다고 평가되었지만, 입력 시퀀스 길이를 늘렸을 때도 그 성능이 좋을지 확인하기 위해 선택하였습니다.

GRU는 LSTM의 복잡성 문제를 극복한 모델로 LSTM보다 과적합이 덜 일어날 것으로 기대되어 선택하였습니다.

Transformer는 NLP와 같이 순서가 있는 sequencial 데이터를 예측하는 과제에서 좋은 성능을 보이는 모델로, 주가 예측에서도 유의미한 성능을 보일지 확인하기 위해 선택했습니다.

마지막으로 파라미터를 조절해가며 3가지 모델에 대한 성능을 비교했습니다.

기존 주가 예측 프로젝트들처럼 종가를 타겟으로 한 것과 더 정상성을 보이는 피처인 1d\_ROC를 타겟으로 한 모델의 성능을 비교하였습니다.

다음으로 뉴스 데이터가 주가 예측에 도움이 되는지 알아보기 위해 앞서 준비했던 total\_df와 stockOnly\_df 를 비교하였습니다.

그리고 모델별 최적의 파라미터를 찾아내기 위해 여러 실험을 했습니다.

저희는 MSE를 손실함수로 사용했을 때 값이 너무 크게 나와 RMSE를 손실함수로 선택했습니다. 실험과정에서 비교한 파라미터들을 표로 정리하면 다음과 같습니다.

먼저 장기적인 추세를 비교하기 위해 시퀀스 길이는 최소 30부터 최대 120으로 설정하였고, batch size는 처음에 1,64,128로 실험을 했었는데 batch\_size가 작을수록 노이즈가 섞을 가능성은 증가하지만 그럼에도 성능이 좋아졌기 때문에 최종적으로는 1,4,8로 설정하였습니다. 모델 사이즈는 LSTM, GRU, tranformer 각 모델별 첫번째 레이어의 hidden size입니다. 처음엔 256도 실험했었는데 성능이 좋지 않고 훈련 시간이 오래 걸려 최종적으론 64와 128만 비교했습니다. 마지막으로 데이터셋을 뉴스가 포함된 total, 주식 데이터만 있는 stockonly로 훈련해가며 성능을 비교했습니다.

실험은 먼저 모든 파라미터 조합에 대해 모델별로 validation loss를 구하여 모델별 최적의 파라미터를 구하는 방식으로 이루어졌습니다. 설명을 위해 우선 단계적으로 결과를 보여드리고 최종 예측 결과를 발표하겠습니다.

우선 타겟을 종가로 할지 1d\_ROC로 할지 비교했습니다. 사진 예시는 데이터셋은 stockonly, 모델 은 LSTM, 모델 파라미터는 다음과 같습니다. 종가가 애초에 변동폭이 크기 때문에 1d\_ROC보다 RMSE값이 클 수밖에 없어 성능을 validation loss값으로 비교할 수는 없었고 따라서 그래프 개형을 통해 어떤 타겟값이 더 좋은 예측을 이끌어내는지 비교를 했습니다. 그 결과 모든 파라미터 조합에 대해 다음과 같이 타겟값이 종가일 땐 다소 완만한 예측선이 그려졌고 타겟값이 1d roc일 땐 유연한 예측선이 그려졌습니다.

데이터셋을 total\_df로 설정하고 비교했을 때도 마찬가지로 종가가 타겟일 땐 예측선이 완만하게 나왔고, 1d roc에서 더 증감 추세가 뚜렷하게 나타났습니다.

이는 GRU나 트랜스포머에서도 동일하게 나타난 특징이어서 최종적으로 타겟값은 1d\_ROC로 하는 것으로 확정 지었습니다.

다음은 타겟값은 1d\_ROC로 확정지은 채로 나머지 파라미터들인 시퀀스 사이즈, 배치 사이즈, 모델 사이즈, 데이터셋을 비교한 실험 결과입니다.

LSTM의 경우 시퀀스 길이가 짧을수록, 모델 사이즈가 작을수록 좋은 성능을 보이고 있습니다. 데이터셋은 stonk\_only가 더 성능이 좋게 나왔지만 타겟값인 1d\_ROC의 통계량을 보았을 때 최솟값이 -35이고 최댓값이 16이므로 값의 범위가 대략 50정도 됩니다. 이정도의 범위에서 RMSE 0.01 차이는 크지 않으므로 stock\_only 데이터셋의 성능이 아주 미세하게 좋긴 하나 크게 유의미하진 않은 것으로 보입니다.

또 다른 특징으로는 시퀀스 길이가 60, 120으로 길어져도 RMSE 차이는 크지 않다는 점이 있습니다. 시간이 더 충분하여 240일, 360일을 입력 시퀀스로 하여 실험해보아도 의미있는 결과를 기대할 수 있을 것으로 보입니다.

GRU는 다른 모델들과 달리 모델 사이즈가 128일 때의 성능이 가장 좋게 나왔습니다. 마찬가지로 시퀀스 길이와 배치 사이즈는 30과1이 최적의 파라미터로 나타났습니다. LSTM과 GRU 모두 공통적으로 시퀀스가 60일 때는 stock\_only 데이터셋이 더 좋은 성능을 보인 반면 시퀀스 길이가 120일 때, 주로 total 데이터셋이 좋은 성능을 보였습니다.

트랜스포머는 GRU와 마찬가지로 total 데이터셋의 성능이 대체로 좋게 나타났으며 LSTM과 같이 seq 30 batch 1 model 64일 때의 성능이 가장 좋았습니다. 그리고 공통적으로 시퀀스 길이가 60일 때보다 30과 120일 때의 성능이 더 좋았습니다.

다만 여기서 알 수 있는 점은 기존의 시퀀스 길이가 1~5일 정도로 짧았던 프로젝트들과 마찬가지로 시퀀스 길이가 짧을수록 당연히 손실값이 작은 특징을 보이고 있습니다. 따라서 수치만으로 결론을 짓기는 어렵고 최적의 파라미터들을 토대로 그래프를 그려 결론을 짓기로 방향을 잡았습니다.

다음은 파라미터별로 평균값으로 실험결과를 요약한 그래프들을 살펴보겠습니다. 모델로는 LSTM이 가장 평균 validation loss가 작게 나타났습니다. 데이터셋의 경우 평균적으로 total\_df의 validation loss가 작게 나타났습니다.

seq 30, batch 1, model 64가 좋은 결과를 나타냈고 gru만 예외적으로 모델사이즈가 클수록 성능이 좋아졌습니다.

시퀀스 길이의 경우 앞에서도 언급했지만 주목할만한 점이 120일 때의 평균 성능이 시퀀스 길이가 60일 때보다 좋다는 점입니다. 일반적으로 예상했을 때 시퀀스 길이가 짧을수록 손실값이 작을 것이라 기대할 수 있는데 저희의 실험에서는 오히려 시퀀스 길이 120의 성능이 좋게 나와서 이후에 시퀀스 길이를 240, 360 등으로 키워 실험을 진행해보는 것도 의미가 있을 것 같습니다.

이렇게 진행한 실험 결과를 토대로 RMSE를 기준으로 모델별 최적의 파라미터를 선정하여 최종 예측을 진행하였습니다. 최종 예측은 최적의 파라미터를 기준으로 모델을 50번 훈련시켜 50번의 예측값에 대한 평균값으로 그래프를 그렸습니다. 이렇게 한 이유는, 저희가 선정한 모델별 파라미터들의 batch\_size가 모두 1로 아주 작기 때문에 모델이 훈련과정에서 local minimum에 빠질 가능성이 커서 예측값의 결과가 정확하지 않을 수 있기 때문입니다. 따라서 앙상블 모델을 만드는 것이 정석적인 방법이지만, 시간 관계상 50번의 예측값을 평균내는 것으로 대체하였습니다.

최종 예측에서 성능 평가를 위해 새로운 평가지표를 정하였습니다. 주식에서 주로 얘기하는 것이 수익률이기 때문에 체감이 어려운 RMSE보다는 평균 오차율로 성능을 보는 것이 효과적이라 봅니다.

그래서 최종적으로 모델별로 다음과 같이 데이터셋, seq length, batch size, model size 파라미터를 설정하여 50번의 예측에 대한 평균값으로 평균 오차율을 계산하였습니다. 의외로 수치상으론 트랜스포머가 우세한 모습을 보였으나 주가 예측에서는 수치만으로 평가하기 어렵다는 결론이 실험과정에서 드러났으므로 그래프를 그려 최종 결과를 확인했습니다.

그래프를 그렸을 때 오차율에 큰 차이나 난 이유를 알 수 있는데, 바로 예측을 시작하는 날인 1월 2일에 주가가 급락했기 때문입니다. 따라서 최종적으로도 결국 수치로 결과를 100% 해석하는 것은 어렵다는 결론이 났고, 그래프의 개형을 통해 모델별로 몇가지 특징을 확인할 수 있었습니다.

LSTM의 경우, 3가지 모델 중 가장 주가의 전체적인 추세를 잘 예측하였습니다. 1월 2일의 급락을 반영하여 빨간색 예측선을 아래로 수직이동시켰을 때, 가장 시작점과 끝점이 실제값과 유사하게 나타났고, 실제값에서도 1월2일부터 1월16일까지 전체적으로 보면 증가하는 추세이며 예측선도 마찬가지로 전체적으로 증가하는 것으로 예측하였습니다.

GRU의 경우에도 처음엔 감소하는 것으로 잘못 예측했으나 전체적으로는 증가하는 추세를 잘 반영하였습니다. 그래프에 나온 평균 오차율 대로라면 GRU가 LSTM보다 성능이 좋아보이지만, 빨간색 예측선의 시작을 1월2일로 내려서 주가 급락 같은 예측하기 어려운 변수를 제외하고 본다면, LSTM보다는 평균 오차율이 커서 LSTM보다 성능이 무조건 좋다고 보긴 어렵습니다. 따라서 LSTM과 GRU를 더 정확히 비교하기 더 많은 실험이 필요합니다.

트랜스포머는 LSTM이나 GRU와 달리 시작점과 끝점이 거의 동일선상에 있는 전체적으로 완만한 예측을 보여주고 있습니다. 따라서 전체적인 추세 예측에서는 성능이 떨어진다고 볼 수 있으나, 세 모델 중에서 가장 높은 정확도의 등락 분류를 해내고 있습니다. 전체 예측 일수가 열흘이니 증가와 감소 구간은 9개가 있는데 이 중 7개의 구간에서 주가의 등락을 정확히 맞추고 있습니다. 만약 저희가 프로젝트를 주가 예측이 아닌 주가의 등락 예측으로 잡았다면 트랜스포머가 가장 성능이 좋은 모델이었을 것입니다.

이를 통해 낼 수 있는 결론은, 주식의 장기적인 흐름은 LSTM이나 GRU를 사용하여 예측하는 것이 효과적이고, 단기적으로 주가의 등락을 예측하고자 할 땐 트랜스포머가 더 효과적이라는 결론을 도출할 수 있습니다.

한계점

직접 주가를 예측하는 데 있어 큰 어려움이 있었던 부분은 성능을 평가하는 것이었습니다. 손실값이 작더라도 예측 그래프를 그려보면 추세를 정반대로 예측하고 있는 경우가 있었고, 예측 그래프가 실제 주가와 많이 떨어져 있어서 손실값은 크지만 증감 추세는 잘 반영하고 있는 경우가 있었습니다. 이런 문제를 해결하기 위해 최종 결론을 도출해내는 단계에서 같은 파라미터로 50번의 학습을 진행하여 그 평균치를 최종 결과로 도출하는 방법을 사용하긴 했으나 더 정교한 앙상블 모델을 설계하여 정확한 결과를 얻었으면 좋았겠다는 아쉬움이 있습니다.

또한 뉴스 감성분석이나 토픽 모델링을 API를 사용하느라 2018년부터 23년의 데이터밖에 사용하지 못했는데, NLP쪽을 공부할 시간이 충분했다면 웹 스크래핑을 통해 더 많은 기간의 뉴스들도 수집할 수 있었기 때문에 더 많은 훈련 데이터로 모델의 정확도를 높일 수 있었을 것 같습니다.

주가 예측 프로젝트가 어렵다고 평가받는 이유는 암 진단 프로젝트와 마찬가지로 높은 재현율을 요구하기 때문입니다. 예측 주가와 실제 주가의 오차율이 평균 5% 이하라고 해도, 주식 시장에서는 큰 손실 또는 이익으로 평가받기 때문에 성능 평가와 마찬가지로 수치만으로 프로젝트의 성공 여부를 결정하기가 어렵다는 한계가 있습니다.