“박수영의 PORTFOLIO”

Data Analysis & Statistics

Context

euphoria0-0

https://euphoria0-0.github.io/portfolio/
https://blog.naver.com/tutumd96
https://euphoria0-0.tistory.com/

All contents are in my github.

01

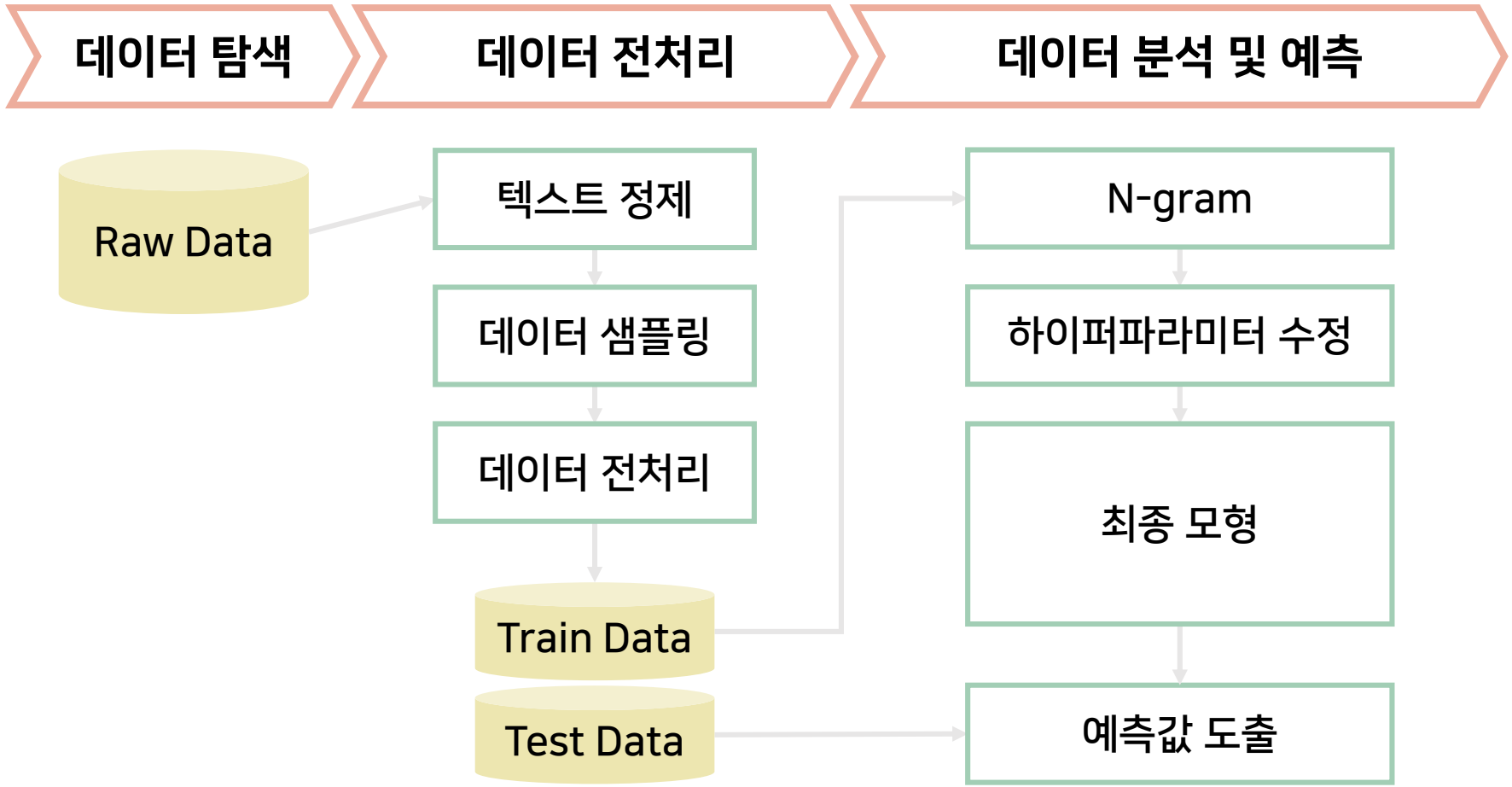
데이콘 금융문자 분석대회

1. 주제
   - 총 글자수 50,000,000개의 데이터를 활용해 스미싱 탐지 모델을 개발
2. 주최/주관: KB금융지주, DACON, KISA
3. 활용 데이터 설명
   - KB금융그룹 및 KISA(한국인터넷진흥원)에서 제공받은 정상문자와 스미싱 문자
   - 문제 및 답안 제출 : 해당 train,public_test.csv 파일을 활용하여,public_test.csv파일에서 없는 항목인 smishing 변수의 각 예측값 확률을 만들어 제출
   - 주의: 제공되는 데이터에는 개인정보 보호를 위해, 개인정보로 간주될 수 있는 이름, 전화번호, 은행 이름, 지점명은 X 혹은 *로 필터링 되어 제공되며 무단 배포를 금지합니다.

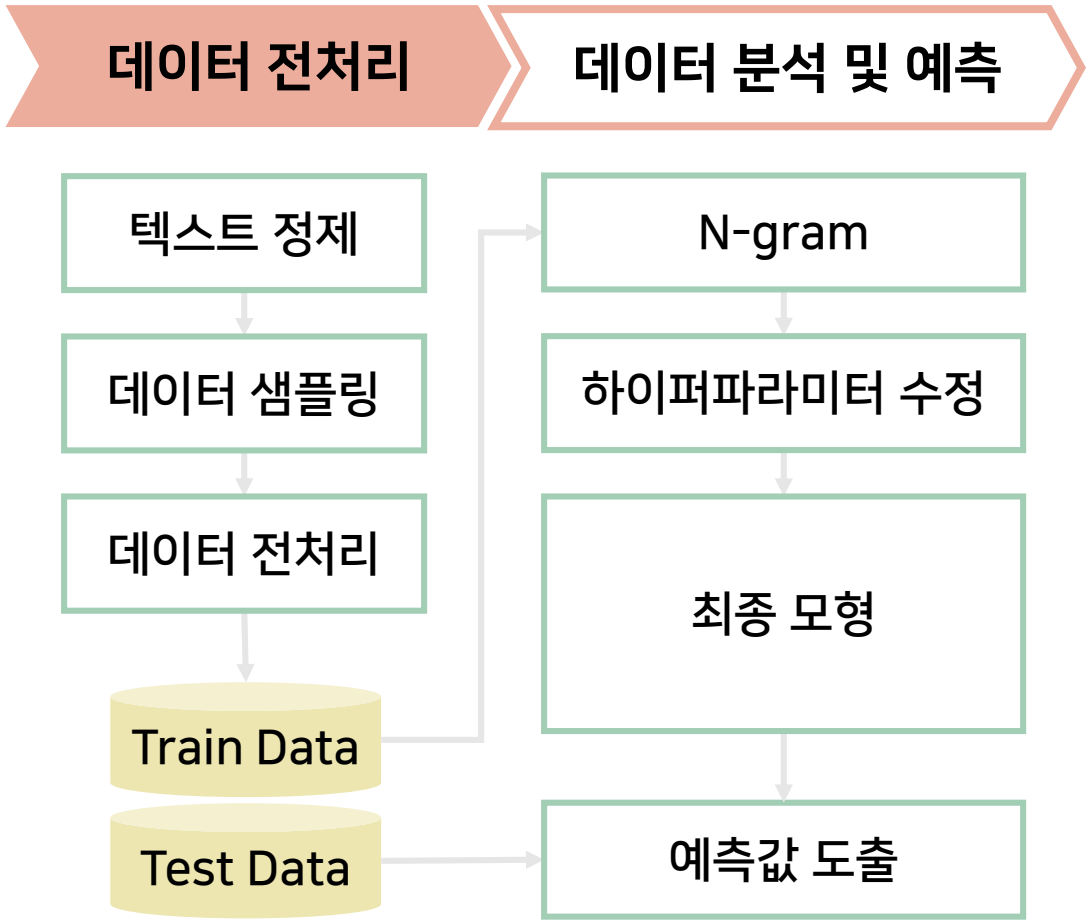
컬럼명	컬럼설명	상세 설명
ID	인덱스	각 문자가 가지고 있는 고유 구분 번호
year_month	일시	고객이 문자를 전송 받은 년도와 월
text	문자	고객이 전송받은 문자의 내용
smishing	스미싱	해당 문자의 스미싱 여부(0-정상,1-스미싱)

01

데이콘 금융문자 분석대회

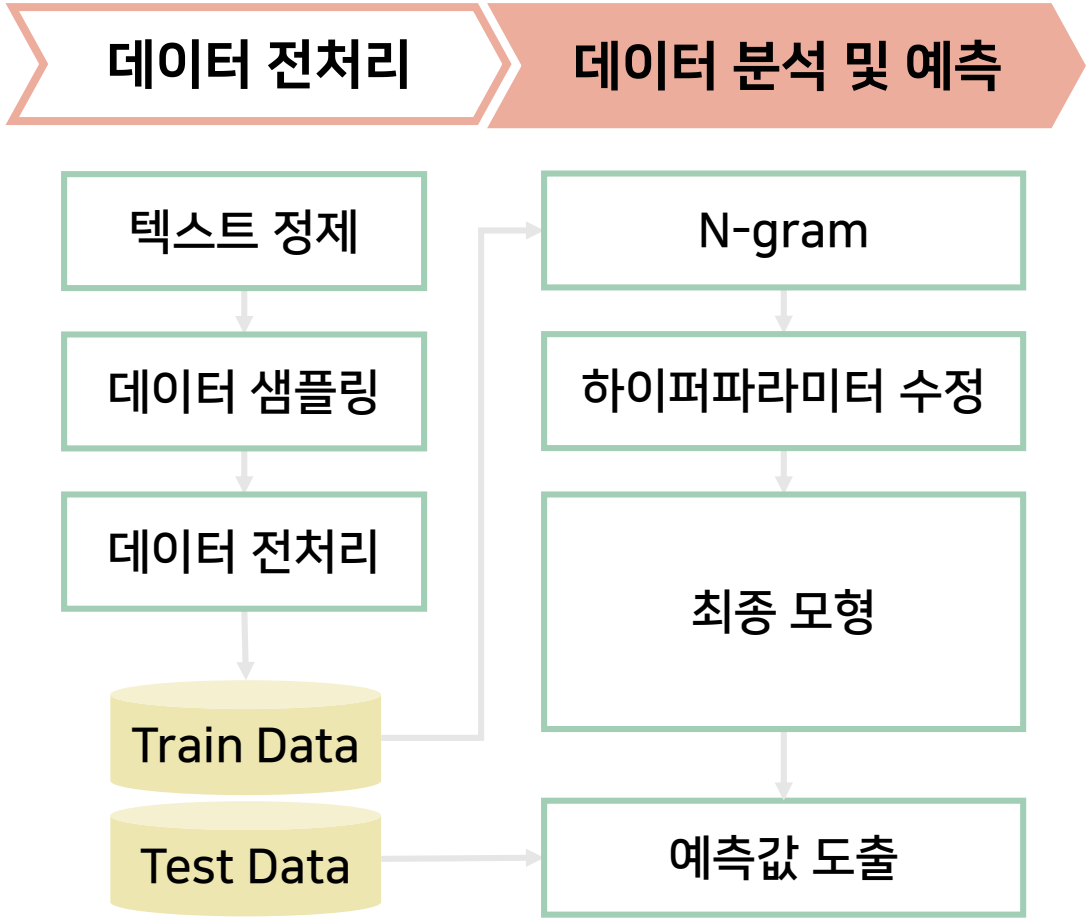


01 데이콘 금융문자 분석대회



1. 데이터 살펴보기
   - 데이터 불균형 문제: 약 97%대 3%
   - 스미싱 문자에 특정 단어의 출현 빈도가 높음
2. 텍스트 정제
   - 제거: 불용어, 한글 영문 외 문자, XXX 등 비식별처리 된 문자
   - Mecab을 이용해 tokenizing
   - tokenizing 후 unigram, bi-gram 저장
3. 데이터 샘플링
   - 데이터 불균형 문제 해결을 위한 Mixed Sampling
4. 데이터 전처리
   - 토큰화한 후 단어에 대하여 정수로 임베딩
   - 이때 모든 샘플의 길이를 800으로 동일하게 저장

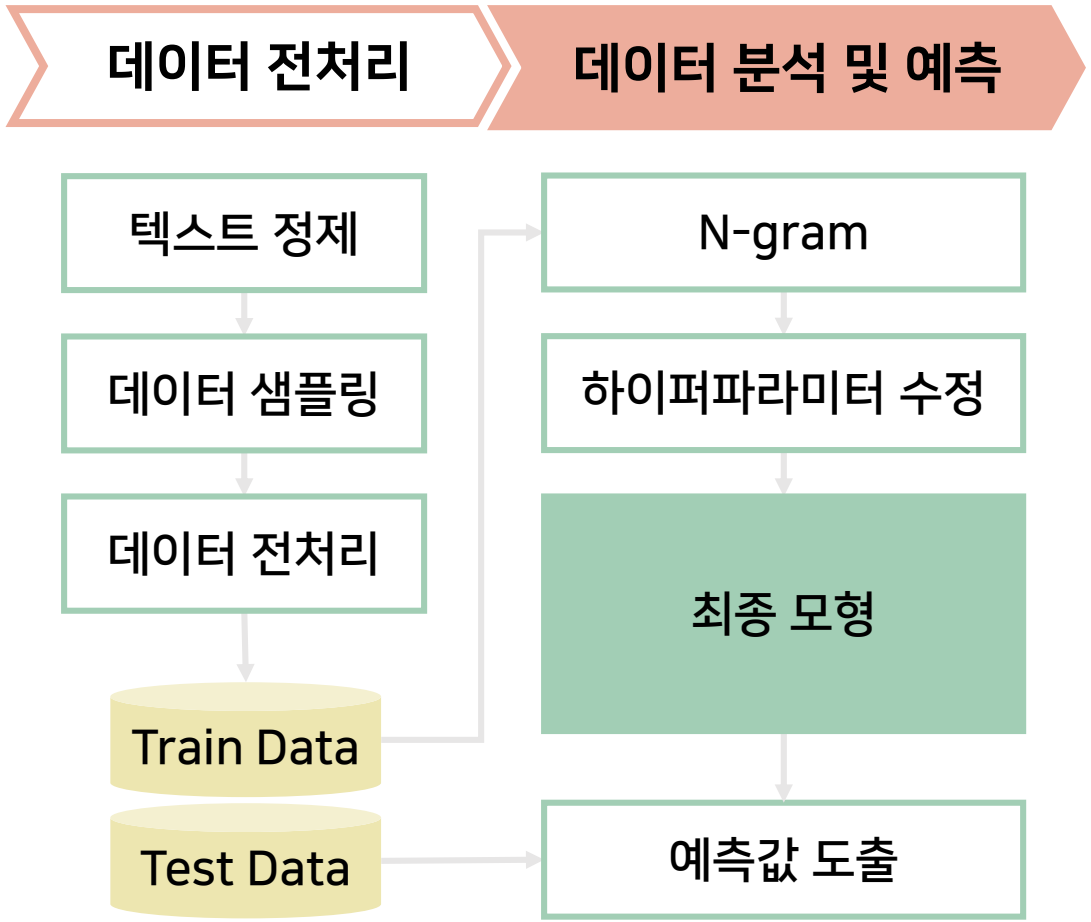
01 데이콘 금융문자 분석대회



1. 모델 구축
   - 평가지표는 AUC
   - 전처리된 Bi-gram 이용한 BiLSTM 모델 빌드

2. 모델 학습
   - Unigram, Bigram 등 중 Bigram 선택
   - simple한 BiLSTM 모델 적합
   - Epoch 등은 early stopping을 이용해 최적 학습 고려

01 데이콘 금융문자 분석대회



1. 최종 모형 선택 이유
   - Mecab: 형태소 분석 시간이 타 konlpy의 것보다 매우 짧음. Inference time이 평가 요소이므로 Mecab 선택
   - Hybrid Sampling: Under sampling은 정보손실 우려, Over sampling은 모형 과적합 우려, SMOTE 등의 기법은 텍스트의 특성 보존력 미약의 이유로 mixed 기법 사용
   - Bigram: 텍스트의 특성상 Word 기반의 Unigram 보다는 앞뒤 단어 관계를 고려하여 성능을 높임
   - BiLSTM: 텍스트의 앞뒤 관계가 유의미하고 이것이 실제 target에 영향을 준다고 파악하여 층이 많지 않은 (과적합 우려) Bi-Directional LSTM 모형 적합

2. 결과
   - Public 17위/438팀, private 10위
   - AUC 0.9903

02 제품 불량률 예측

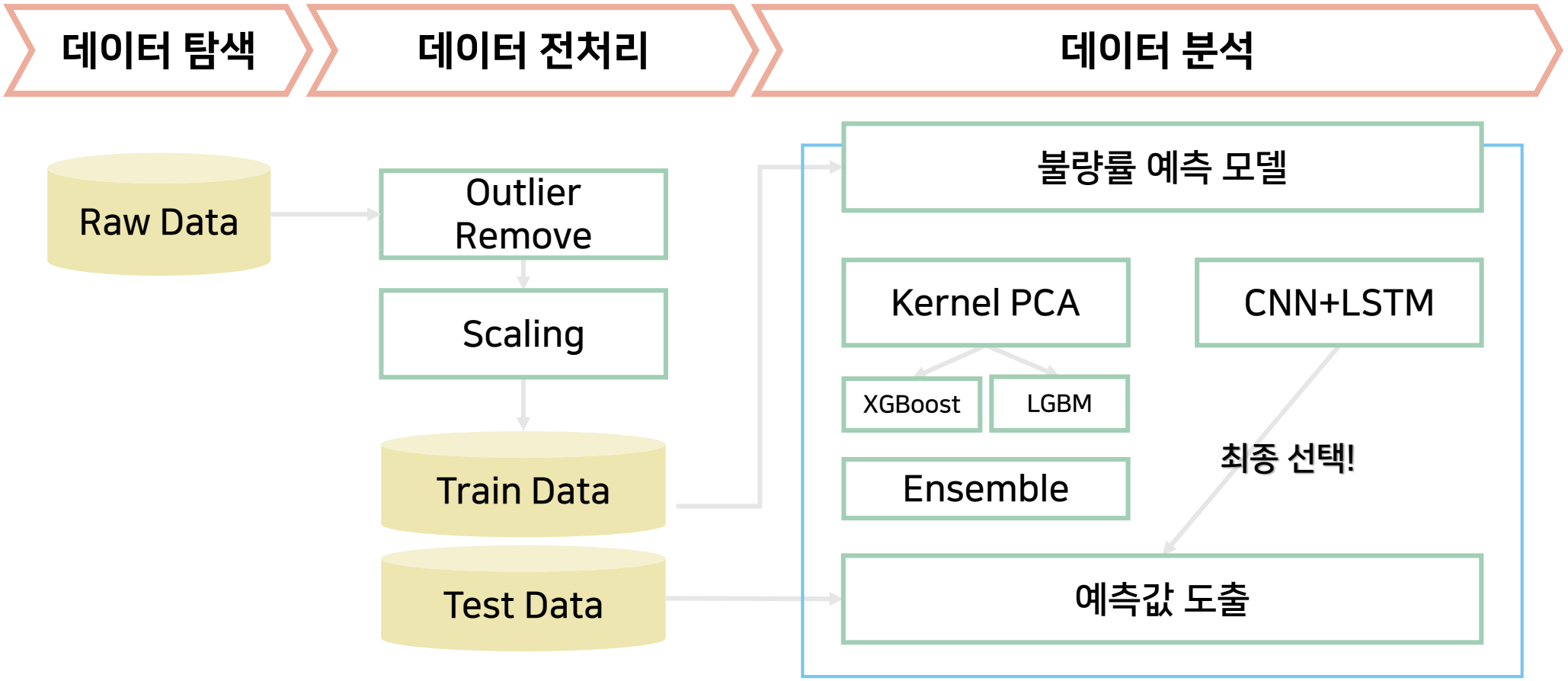
1. 주제: 제품을 생산하는 과정에서 발생하는 설비의 센서 측정 데이터와 해당 제품의 불량률 사이의 유의미한 연관성 분석 및 불량률 예측
   - 삼성 SDS Brightics AI 공모전
   - 기간: 2019.07.01. ~ 2019.08.31.
   - 툴: Python, Tensorflow, sklearn

2. 평가: 검증 데이터의 각 제품별 불량률에 대한 가중평균절대오차 (WMAE:Weighted Mean Absolute Error) 측정값을 산출

3. 활용 데이터

컬럼명	컬럼설명	상세 설명
Time Index	인덱스	제품 제조 순서 (분석에 독립변수로 활용 가능)
X1 ~ X85	공정 센서 파라미터	제품 생산 시 파생되는 센서 변수값
Y	불량률	각 제품별 불량률 지표 (값이 클수록 불량도 심각)

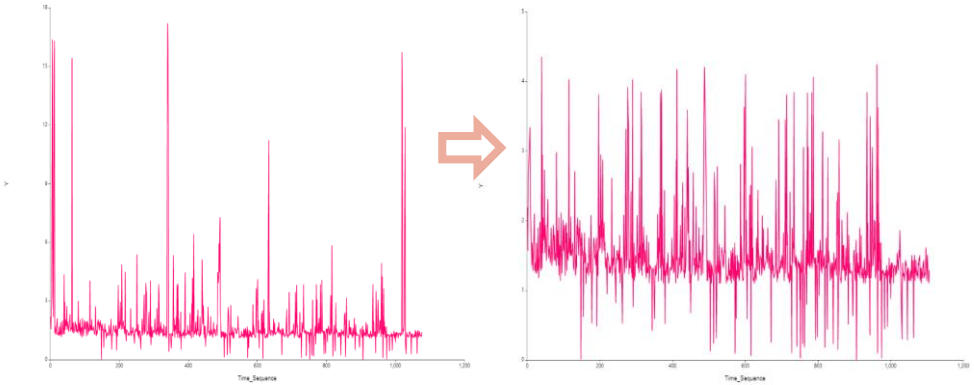
02 제품 불량률 예측


데이터 탐색
데이터 전처리
데이터 분석
Raw Data
Outlier Remove
Scaling
Train Data
Test Data
불량률 예측 모델
Kernel PCA
CNN+LSTM
XGBoost
LGBM
Ensemble
최종 선택!
예측값 도출

02 제품 불량률 예측

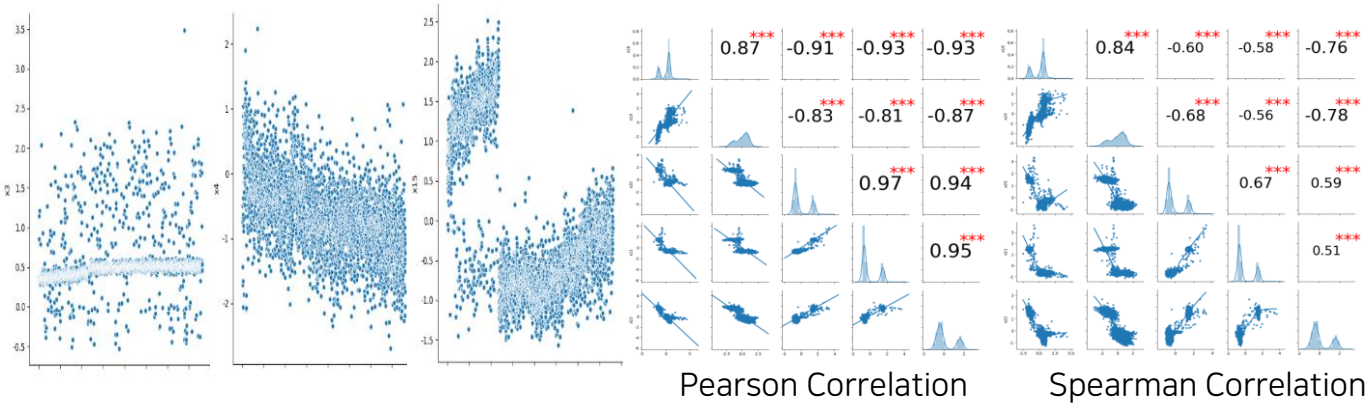
1. EDA

1. Target Variable - Outlier Detection



- Y 변수는 시계열에 크게 상관없어 보인다.
- 이상치가 존재하여 Tukey's method에 의해 제거하였다(Tukey's Statistics>10이면 제거)
- 약 1% 데이터가 제거되어 Train Data 3,507 중 3,472개의 레코드로 다음을 진행하였다.

2. Input Variables



Pearson Correlation Spearman Correlation

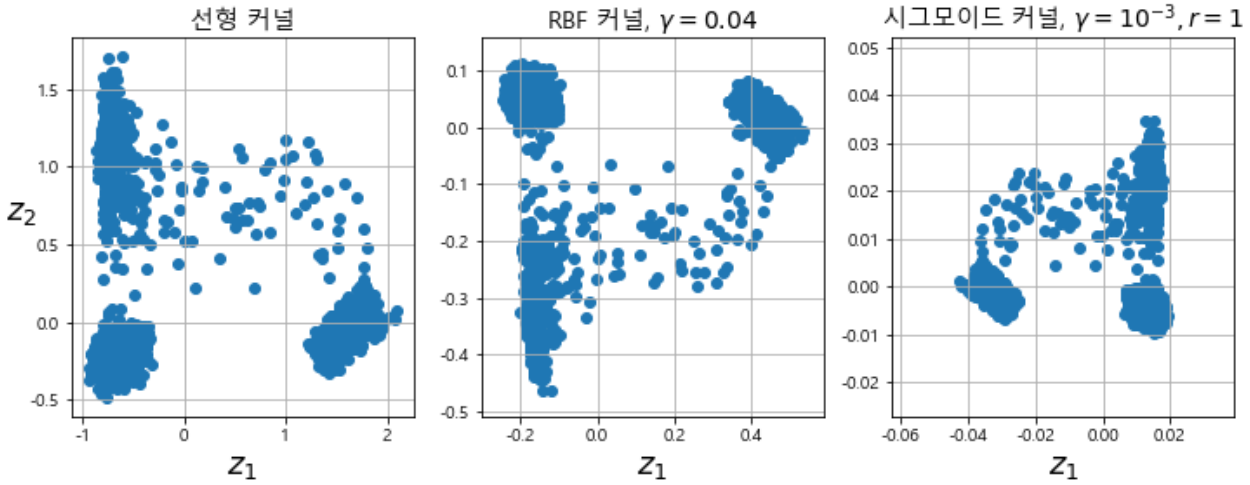
- 변수 간 다중공선성을 제거하기 위해
  ① 변수 선택을 한다
  ② 차원을 축소한다
- 모든 센서가 유의미한 역할을 가진다고 가정하고, 85개의 센서 데이터에 대하여 차원 축소를 수행한다.
- 차원 축소의 방법: PCA(linear, kernel, …), SVD, t-SNE, LLE, NMF, MDS, IsoMap, …
- 85개의 센서 변수들에 대하여 차원을 축소하여 예측 모델링을 수행한다.

02

제품 불량률 예측

2. Modeling – ① Dimension Reduction and XGBoost

- kernel PCA
  a. 변수 간 다중공선성이 존재하였다.
  b. 모든 변수(센서)가 유의미한 결과를 낼 수 있다고 가정한다. (변수선택을 사용하지 않은 이유)
  c. 변수 간 비선형적 관계를 고려하기 위해 비선형 투영(projection)으로 차원축소를 한다. 'rbf'함수를 커널 함수로 사용한 주성분분석을 실시하였다.
  d. Hyperparameter Tuning을 위해 GridSearchCV를 사용하였다.



Kernel PCA에서 커널 선택을 통해 축소시킨 센서 데이터

02

제품 불량률 예측

2. Modeling – ① Dimension Reduction and XGBoost

- XGBoost, LightGBM
  a. 예측으로는 XGBoost와 LightGBM (Regressor)를 사용하였다.
  b. Hyperparameter Tuning을 위해 GridSearchCV를 사용하였다.
  c. 두 모델을 앙상블한 모델을 최종 예측에 사용하였다.

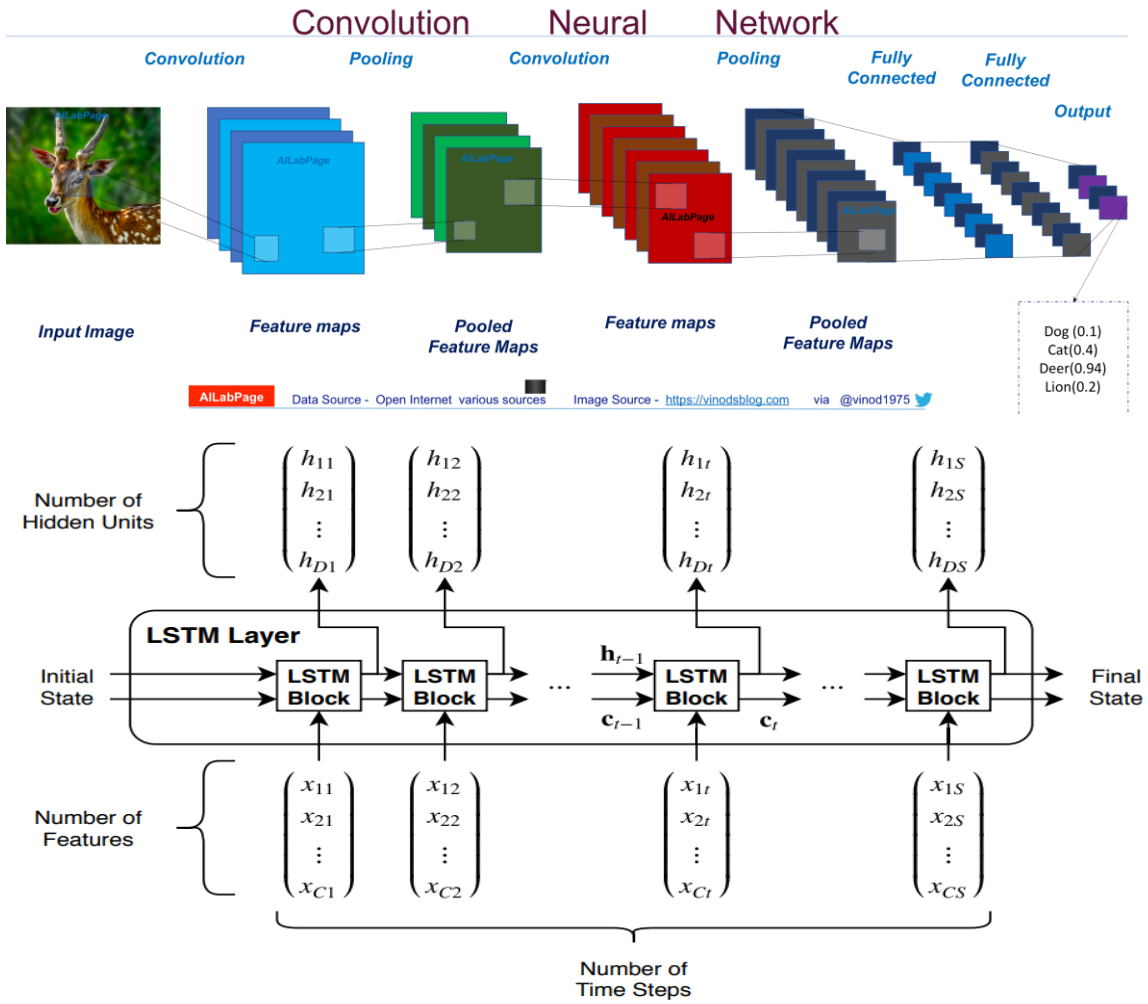
- Boosting Tree Model
  - 데이터에 트리 모델을 학습신 후 성능이 낮은 Weak learner를 보완하면서 순차적으로 학습하여 모델의 bias를 보완하는 모델이다.
  - XGBoost는 부스팅 트리 모델 중 가장 좋은 성능을 내는 앙상블 모델이다.
  - LightGBM은 XGBoost와 비슷한 성능을 보이지만 더 빠르게 학습을 할 수 있다.
  - 이 데이터에서 Y를 예측하는 데에 Boosting Tree Model이 가장 빠르고 성능이 좋아 이용하였다.

- 결과
  - WMAE = 1.29 (for test set)

02

제품 불량률 예측

2. Modeling – ② CNN+LSTM



- CNN의 특징
  - 주로 이미지를 인식하는데 사용, 이미지의 공간 정보를 유지하면서 특징을 효과적으로 학습
  - 컴퓨터 스스로가 특징표현을 만들어 냄

- CNN은 주로 이미지 학습에 많이 사용하나, 특징을 추출하기에 좋은 모델이다.
- X1~X86까지의 공간 정보와 시계열 정보를 학습하고, 86개의 센서데이터의 특징을 추출해내 사용하기 위해 CNN모델을 사용한다.

- LSTM의 특징
  - RNN의 특별한 모델로 이전 단계의 정보를 지속한다
  - RNN의 긴 의존 기간으로 인한 문제를 해결한다.
  - Time Series 데이터를 해결하기에 적합하다.

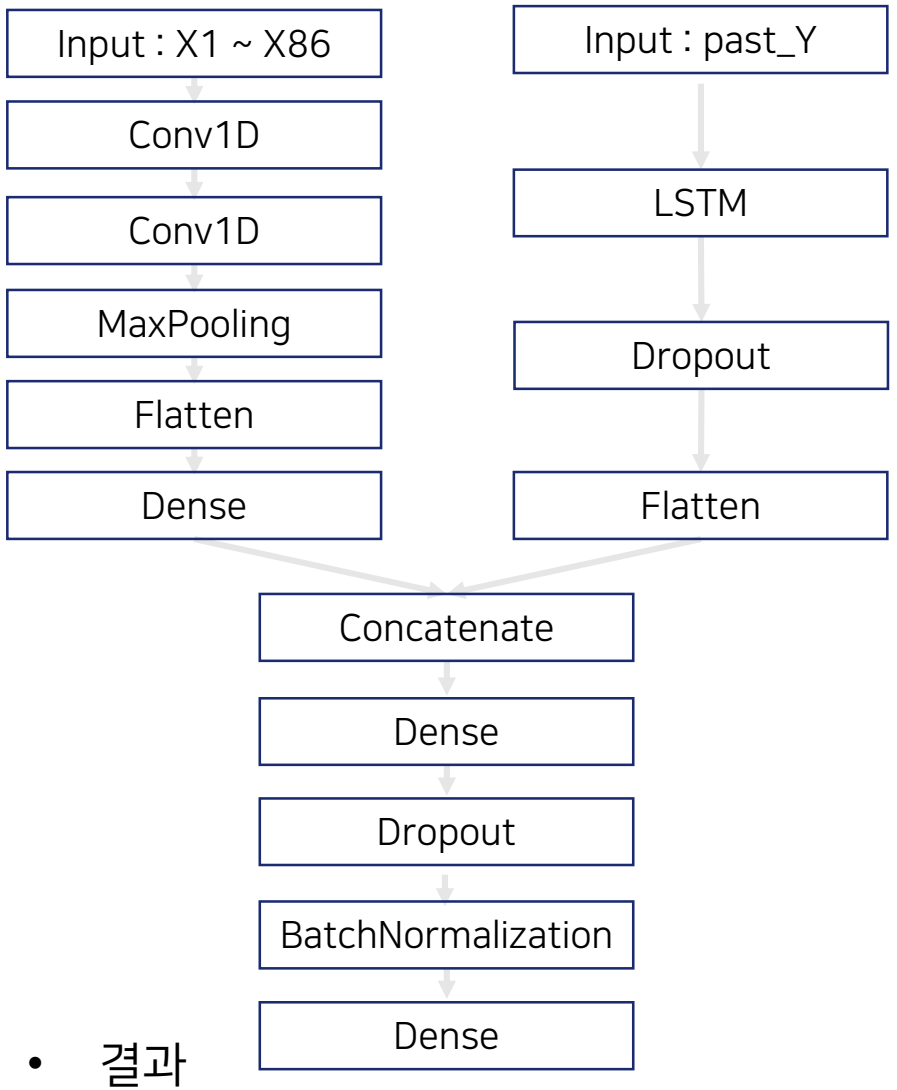
02

제품 불량률 예측

2. Modeling – ② CNN+LSTM

Layer (type)	Output Shape	Param #	Connected to
x_input (InputLayer)	(None, 85, 1)	0	
x_Conv1D_1 (Conv1D)	(None, 83, 32)	128	x_input[0][0]
x_Conv1D_2 (Conv1D)	(None, 81, 16)	1552	x_Conv1D_1[0][0]
x_MaxPooling (MaxPooling1D)	(None, 40, 16)	0	x_Conv1D_2[0][0]
x_Flatten (Flatten)	(None, 640)	0	x_MaxPooling[0][0]
y_input (InputLayer)	(None, 3, 1)	0	
x_Dense128 (Dense)	(None, 128)	82048	x_Flatten[0][0]
y_lstm (LSTM)	(None, 3, 32)	4352	y_input[0][0]
x_Dropout (Dropout)	(None, 128)	0	x_Dense128[0][0]
y_Flatten (Flatten)	(None, 96)	0	y_lstm[0][0]
concatenate_126 (Concatenate)	(None, 224)	0	x_Dropout[0][0] y_Flatten[0][0]
con_Dense64 (Dense)	(None, 64)	14400	concatenate_126[0][0]
dropout_357 (Dropout)	(None, 64)	0	con_Dense64[0][0]
con_Dense32 (Dense)	(None, 32)	2080	dropout_357[0][0]
dropout_358 (Dropout)	(None, 32)	0	con_Dense32[0][0]
batch_normalization_128 (BatchN	(None, 32)	128	dropout_358[0][0]
con_Dense1 (Dense)	(None, 1)	33	batch_normalization_128[0][0]

Total params: 104,721
Trainable params: 104,657
Non-trainable params: 64



- 결과
  - WMAE = 1.29 (for test set)

02 제품 불량률 예측

- 분석을 통해 얻을 수 있었던 점
  - 가장 좋은 성능을 내는 모델 하나를 선택하기 보다 여러 모델을 Ensemble한 모델을 선택하여 보다 모델의 bias를 보완하고 딥러닝의 오버피팅 문제를 해소한다.
  - 차원축소를 통해 85개의 센서데이터가 설명하는 양(분산량)을 주성분을 통해 나타낼 수 있어 단순하면서 재사용가능한 알고리즘이다.

- 현 분석의 한계
  - Deep Learning 모델이 복잡해짐에 따라 학습시간이 크게 증가한다. 그 때문에 적합한 모델을 찾고 Hyper Parameter를 찾는데 Grid Search를 사용하였고 많은 시간을 소요하였다.
  - 차원축소 기법에서는 시계열을 고려하지 않았다.
  - CNN+LSTM 모델에서 하이퍼파라미터의 기준을 WMAE가 아닌 MSE로 사용하였다(PCA, Boosting은 WMAE로 사용하여 다를 수 있다).

- 향후 개선 방향
  - 모델의 하이퍼파라미터를 찾는 데에 베이지안 최적화 방법을 이용하는 등 빠른 파라미터 서치가 필요하다.
  - 투영 차원축소를 이용하였고, 더 높은 성능을 확인할 수 있다면 매니폴드 방법도 시도할 수 있다.

03 종목분석 리포트 분석

1. 주제: 금융시장에서 데이터 과학을 이용한 투자전략 이끌어내기
   - 기간: 2019.05.01. ~ 2019.07.31.
   - 툴: Python, Tensorflow, Ubuntu

2. 분석 배경: 증권사에서는 투자자를 위해 종목부석 리포트를 제공한다. 그러나 금융 투자업계 구조 상 각 기업, 증권사 등의 이해관계가 얽혀있어 매도 의견을 자유롭게 낼 수 없다. 따라서 애널리스트의 매수추천 의견이 미래수익률과는 무관하게 발생되어 투자가치 및 신뢰성을 떨어뜨리는 결과를 낳기도 했다.

3. 분석 주제: 보고서의 신뢰성, 주가정보, 텍스트 표현 등의 요소가 투자자의 결정에 영향을 미친다고 가정한다. 그리고 리포트의 텍스트와 증권사의 투자의견을 이용하여 종목 선택에 유용성에 미치는 요인들을 탐색하고, 이를 바탕으로 유용성을 지표화 하고자 한다.

03 종목분석 리포트 분석

4. 활용 데이터

- 네이버 금융에서 제공하는 증권사별 종목분석 리포트를 크롤링하여 텍스트 데이터를 추출
- 시가 총액 상위 50 개 종목 중 우선주를 제외한 48개 종목
- 2009년 1월 1일 이후부터 2019년 6월 30일 까지의 10년 6개월의 보고서를 수집하여 총 13,508개의 종목분석 리포트를 이용
- 리포트의 애널리스트 투자의견의 하향 건수는 20, 중립 혹은 의견없음은 164, 나머지 13324는 매수추천 의견이었다. 또한 한달 후 주가 상승 여부에서 상승한 경우는 5116건, 내린 경우는 8392건

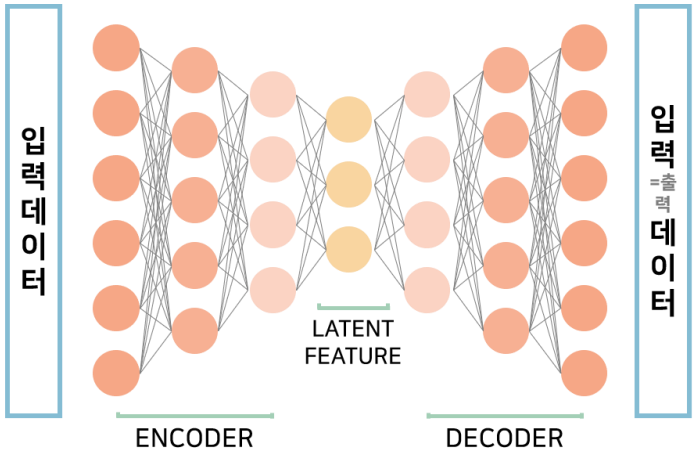
5. 분석 방법

- Dimension Reduction: AutoEncoder & Doc2Vec
- Prediction: Logistic Regression

03 종목분석 리포트 분석

- Auto Encoder
  다차원 입력 데이터를 hidden layer를 이용하여 저차원 부호로 바꾸고(Encoder) 다시 저차원 부호를 처음 입력한 다차원 데이터로 바꾸면서(Decoder) 특징점 (latent feature)들을 찾아내는 Unsupervised Learning이다. 각 문서의 차원을 축소시켜 유사도를 비교한다.

- Doc2Vec
  개별 문서를 임베딩하여 저차원의 featur를 포함하는 Document Vector 생성
  리포트들의 군집 형성 여부를 보기 위해 임베딩된 문서벡터의 분포를 t-SNE로 시각화



03 종목분석 리포트 분석

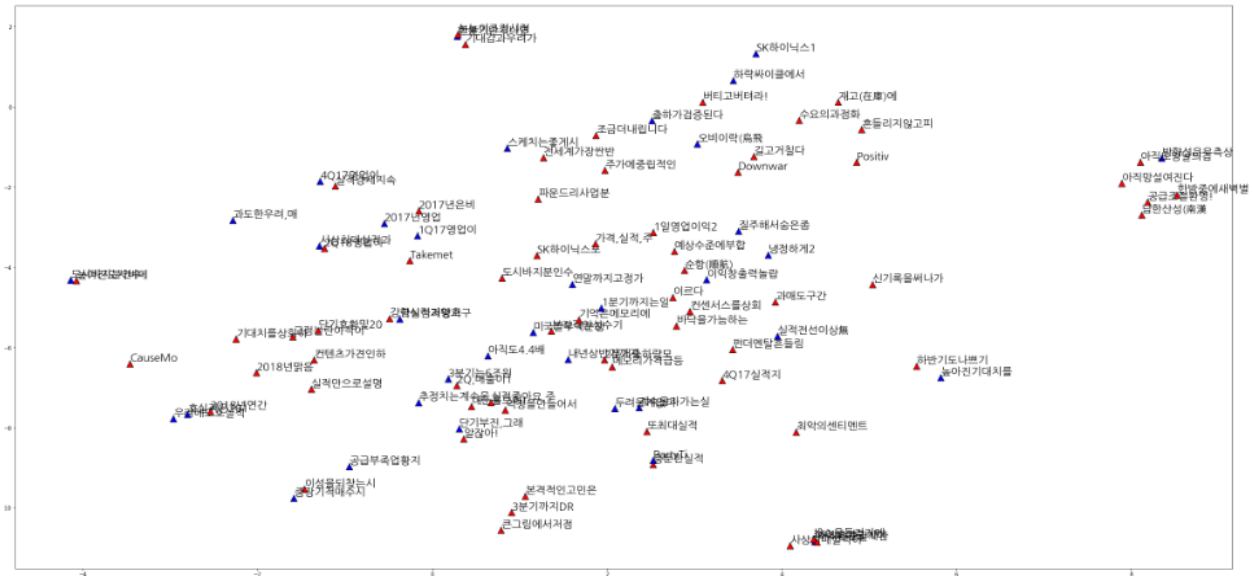
1. Doc2Vec 시각화

- 리포트의 텍스트와 투자의견, 그리고 한달 후 주가 등락여부 데이터를 이용하여 Doc2Vec 알고리즘 사용 후 t-SNE 시각화
- 아래 그림은 실제 문서의 투자의견(매수, 의견없음, 매도)과 한달 후 주가 상승 여부(상승, 하락)
- 실제로 서로 다른 두 의견을 가진 리포트는 Doc2Vec의 임베딩으로는 비슷하게 나타나나 오른쪽에는 주가가 상승한 리포트들이 모였다.

- 그 의미는

① 리포트에 담긴 임베딩 단어의 분포는 매수 리포트와 중립 혹은 매도 리포트 모두 비슷하다. 즉, 매수추천 의견을 가진 리포트는 중립 혹은 매도 의견의 텍스트 어조와 크게 다르지 않다.

② 리포트의 텍스트 정보 및 투자의견이 한달 후 주가 등락 여부와는 뚜렷한 상관관계가 없다.



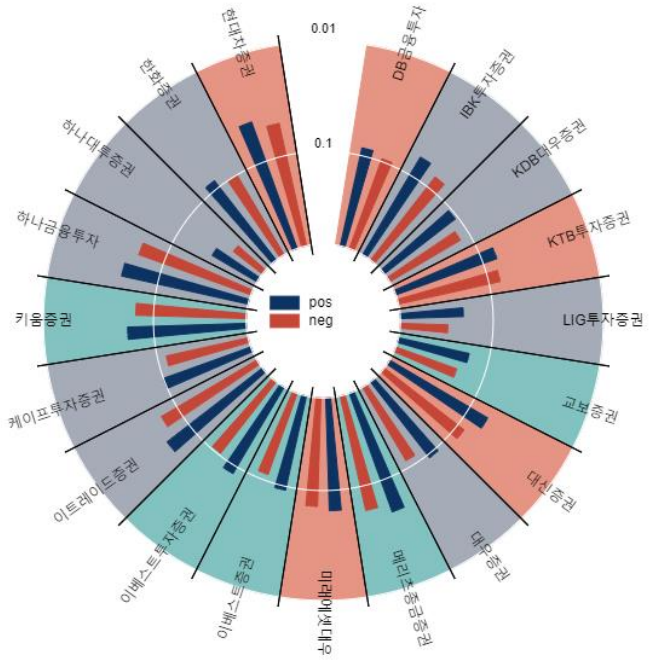
t-SNE를 이용한 Doc2Vec 시각화

03 종목분석 리포트 분석

2. AutoEncoer & Classification

- 사용한 변수: 리포트의 투자의견, 리포트 발간 날짜 기준 up_down 증권사와의 베타계수, 리포트 감정(긍/부/중), 강조 단어
- 변수 파생: 리포트는 약 한 달에서 두 달마다 발간된다. 이를 기준으로 주가는 상승한 일수, 발간일 기준 증권사와의 베타계수, 서울대 감성 사전 기준 긍부정 , 강조단어의 빈도를 계산하였다.
- 분석 방법: 로지스틱 회귀를 이용해 주가 상승을 설명하는 변수들을 찾아낸다. 보고서의 텍스트를 전처리하고 Auto Encoder를 이용해 워드 임베딩의 차원을 축소시켜 투자 의견이 다른 두 문서의 유사도를 비교
- 로지스틱 회귀 결과, 모델 설명력이 낮았다. 변수들은 강조어조의 수, 투자 의견이 주가가 상승한 일수에 영향을 주었다.
- 오토인코더 결과, 이를 시각화하여 표현하였다. 매수 의견을 지닌 보고서의 수는 매매 의견의 수보다 25%였으며 매매가 매수보다 더 넓게 퍼져 있다. 문서들 간 잘 뭉쳐져 보이진 않는다.
- 오른쪽 파이차트는 각 증권사 별 베타계수를 기준으로 감성(긍부정) 추이를 표현하였다. 감성에 대한 차이를 보이진 않았다.

	coef	std err	z	P>|z|	[0.025	0.975]
beta	0.1349	0.109	1.237	0.216	-0.079	0.349
length	-0.3027	0.362	-0.837	0.403	-1.012	0.406
pos	-0.3477	0.391	-0.890	0.373	-1.113	0.418
neg	0.1474	0.475	0.310	0.756	-0.784	1.079
neut	-0.1365	0.229	-0.597	0.551	-0.585	0.312
intensity	0.8225	0.347	2.368	0.018	0.142	1.503
comment	0.2301	0.111	2.068	0.039	0.012	0.448



각 증권사에서 pos, neg 값의 차가 크지 않기 때문에 중요한 변수가 아님을 확인할 수 있다.

03

종목분석 리포트 분석

Conclusion

- 본 연구의 목적은 보고서의 투자판단에의 유용성을 탐색하는 것이었다.
- 주가 상승에 영향을 주는 보고서의 요인을 찾고자 한 로지스틱 회귀는 모델 설명력이 낮았다. 생성한 파생변수가 주가 상승 수를 설명하지 못했다. 오토인코더는 input data의 30% 정도만 재현했다. 그 이유는 첫째, 텍스트 분석 시 시퀀스가 아닌 단일 단어만을 고려하여 빈도를 계산하였다. 둘째, 금융 도메인인 텍스트만의 특성을 고려할 필요가 있다. 마지막으로 본 연구에서 유용성이라는 없는 지표를 찾으면서 옳은 모델인지 확인하기 어려웠다.

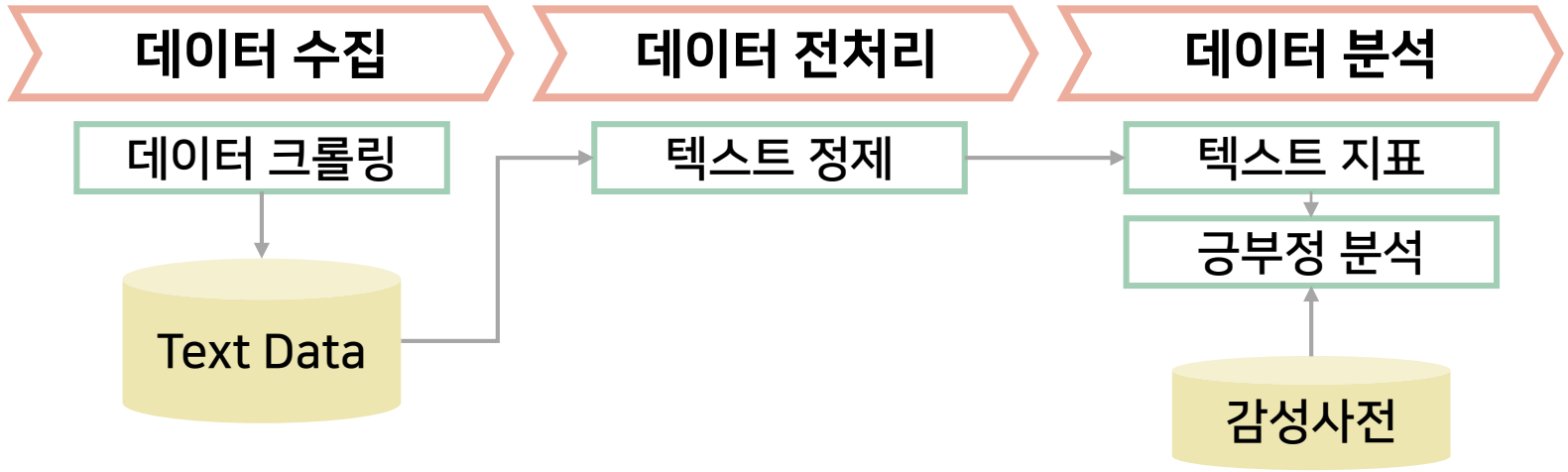
- 따라서, 보고서의 텍스트만으로는 보고서의 투자 유용성을 분석하기엔 부족했다.

04 DART 사업보고서 분석

1. 주제: 사업보고서 분석
   - 기간: 2019.01.01. ~ 2019.02.28.
   - 툴: Python

2. 활용 데이터
   - DART에 공시된 사업보고서를 OPEN API를 이용하여 크롤링
   - 사업보고서 내 ≪이사진의 경영진단 및 분석≫ 부분의 텍스트

3. 분석 방법



04 DART 사업보고서 분석

1. 데이터 크롤링
   ① dart.fss.or.kr에서 OPEN API를 이용해 웹에 접근
   ② 종목코드와 색인 날짜로로 해당 기업의 사업보고서 리스트를 가져옴
   ③ 같은 방법으로 모든 기업의 기간 내 모든 사업보고서 목록을 가져옴
   ④ 해당 사업보고서 url로 원하는 부분의 텍스트를 가져옴

2. 변수 생성
   - 텍스트 길이
   - 텍스트 내 단어 수
     - 한글 영어 등 모두를 포함한 단어 수를 셈
   - 긍정 단어 수
   - 부정 단어 수
     - 긍부정 단어 사전을 이용해 단어 수를 셈
     - 긍부정 단어 사전은 KNU Dictionary 이용

02

그 밖1: Data Mining

- 통계자료분석 캡스톤디자인 프로젝트
  - https://github.com/euphoria0-0/etc/blob/master/paper.pdf

- Bank TeleMarketing Analysis
  - https://github.com/euphoria0-0/etc/blob/master/bank%20marketing_pt.pdf

03

그 밖2: Text Mining

- IMDB movie review helpfulness index
  - https://github.com/euphoria0-0/imdb_tm/blob/master/imdb_pt.pdf
- 대선 토론회
  - https://github.com/euphoria0-0/debate_tm/blob/master/debate_report.pdf
- Yelp Text Analysis
  - https://github.com/euphoria0-0/etc/blob/master/yelp%20tm_pt.pdf

“감사합니다”