방학세미나 후기

즐거운 학회장팀
권남택
박서영

INDEX

1. 출제 의도

2. 주요 기법 리뷰

3. 공통 피드백

4. 1등 발표

1

출제 의도

Scania Truck data

Air pressure system failures in Scania trucks data from Kaggle

출제 의도

저번 방세와의 차이

- 저번 방세는 Train data가 1,000,000개로 엄청난 대용량 데이터
- 광고의 클릭여부를 예측하는 분류문제
- X들을 대부분 범주형 변수
- 9GB의 추가적인 join 가능 데이터 존재
- 탐탐에서 다같이 모델링 했었음…코로나 out…

출제 의도

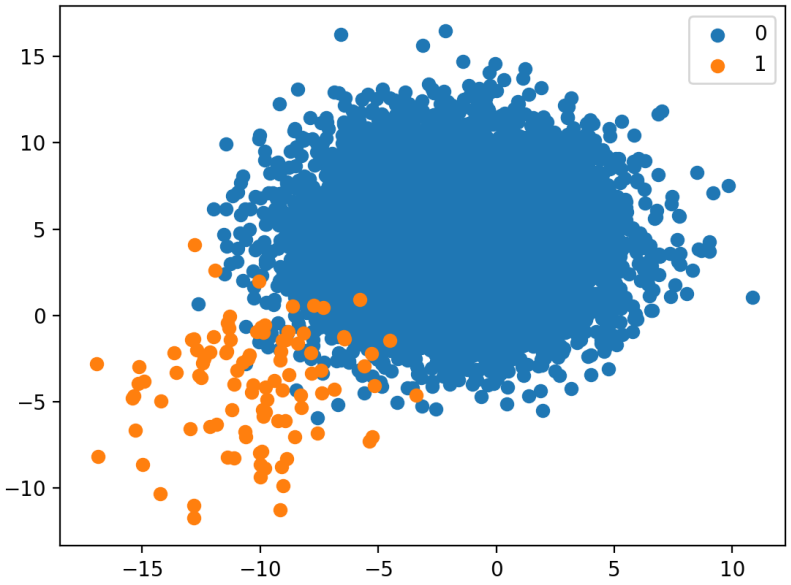
이번 방세가 집중한 부분

- Data의 수를 적절하게 관리해 모두가 자신의 노트북에서 모델링이 가능하도록…
- 이전 주제분석이 대부분 예측모델링과 거리가 있었기 때문에,
  신입학회원들이 들어오기 전에 예측모델링의 루틴에 익숙해질 필요성 존재
- 불균형 데이터 + NA 처리를 통해 예측 모델링에 익숙해지자!
- 서로 친해지자…but 온라인의 한계 존재 ㅠㅠ

출제 의도

클래스 불균형

- 클래스 불균형은 범주형 Y에 대해 빈번하게 일어나는 상황
- 범주팀 3주차에서 다루는 내용의 핵심은 결국 '클래스 불균형을 어떻게 대처할 것인가?'



출제 의도

클래스 불균형 - Cost

- 실제 상황에서 오분류에 대한 비용 동일하지 않고, 저범주가 더 큰 비용을 지님
- 이런 비용을 고려할 수 있는 문제로 Scania Truck Data를 사용함!

	Negative/healthy	Positive/cancerous
Number of cases	10,923	260
Category	Majority	Minority
Imbalanced accuracy	≈ 100%	0-10 %

Ex) 암종양이 실제로 양성인데 음성이라고 판단하면 큰일남…!

출제 의도

NA 처리

- 많은 경우 데이터는 NA를 가지고 있기 때문에 이를 적절하게 대체/삭제하는 것이 필요
- Scania Truck data는 수많은 NA를 가지고 있고, Test data에도 NA가 존재함
- 변수들이 Masking되어 있어서 NA의 패턴을 파악하기 어려운 것은 아쉽지만,
  NA 처리에 다양한 방법들이 있음을 알았으면 좋겠다는 측면에서 해당 데이터 사용
- 전 학회장 신성민의 MICE 발표도 생각했음…!!

그곳은 행복하니 성민아…?

2

주요 기법 리뷰

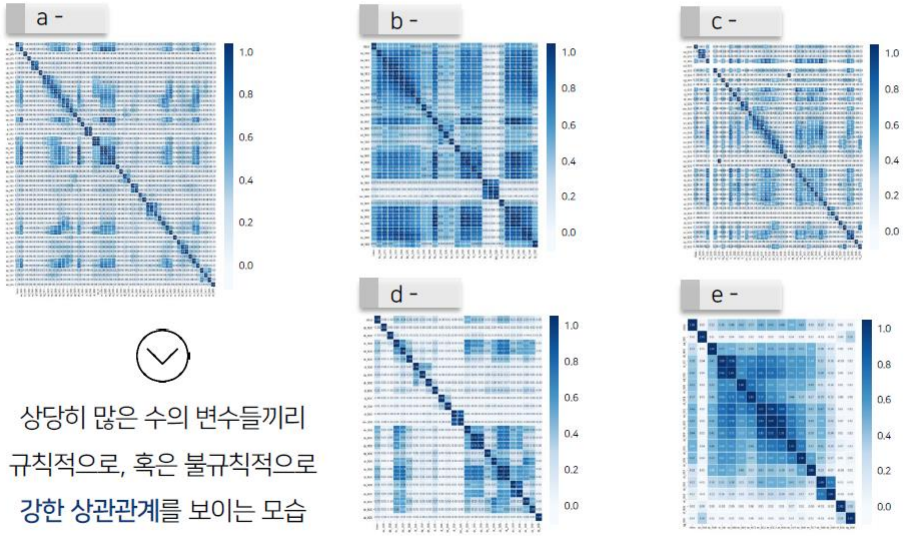
변수 필터링

- 현재 주어진 변수가 170개로 절대 적지 않은 상황
- 모든 변수들을 사용하는 것도 방법이지만,
  적은 변수들로도 더 좋은 성능을 찾는 것도 하나의 방법이 될 수 있음.
- 미리 불필요한(Y와 상관이 떨어지는) 변수들,
  모델을 만드는데 악영향 (ex 다중공선성) 을 주는 변수들을
  선제적으로 제거하는 것이 좋음!
- 이를 위한 필터링(Filtering)을 사용하는 것을 고려해볼 수 있다!

변수 필터링

1 데이터 전처리

상관행렬 plot



상당히 많은 수의 변수들끼리
규칙적으로, 혹은 불규칙적으로
강한 상관관계를 보이는 모습

1팀의 경우 X들 사이의 강한 상관관계가 어떤 패턴을 가지고 나타나는 것을 인식

불필요한 X들을 지우려는 시도가 인상적이었음!

변수 필터링

- 범주형 변수 Y와 X 사이의 연관성을 찾는 것은 상대적으로 조금 귀찮은 일.
- 연속형 변수간에는 피어슨/스피어만 상관계수를 통해 제거하는 것이 가능.
- 범주형 Y에 대한 변수 필터링을 시행하는 알고리즘으로는 'Relief' 알고리즘이 존재.
- 혹은 randomforest나 다른 부스팅 모델에서의 변수중요도를 사용할 수도 있음.
  (완전히 근본없는건 아님!)

통계적 모델링과 머신러닝 실습에서
Filtering & Variable Selection 부분!

변수 필터링

- 물론 Y와 X1, X2간의 상관관계가 크면 불필요한 변수들이 걸러지지 않고,
  Y와 X_j의 개별적인 관계만을 관찰할 수 있다는 한계 존재

- 또한 필터링에 명확한 기준이 없기 때문에,
  너무 많은 변수를 한번에 필터링하는 것은 좋지 않음

- 하지만 변수(차원)를 줄임으로서 불필요한 정보로부터 모델을 지킬 수 있다는 장점!

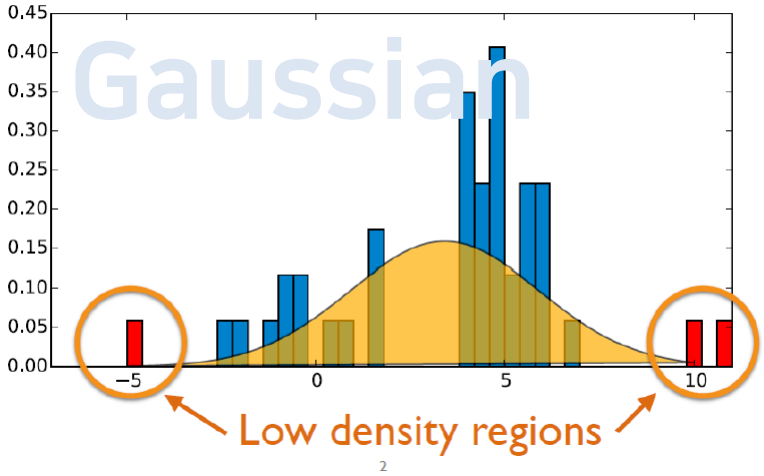
https://en.wikipedia.org/wiki/Relief_(feature_selection)
An Introduction to Variable and Feature Selection (Journal of Machine Learning Research)

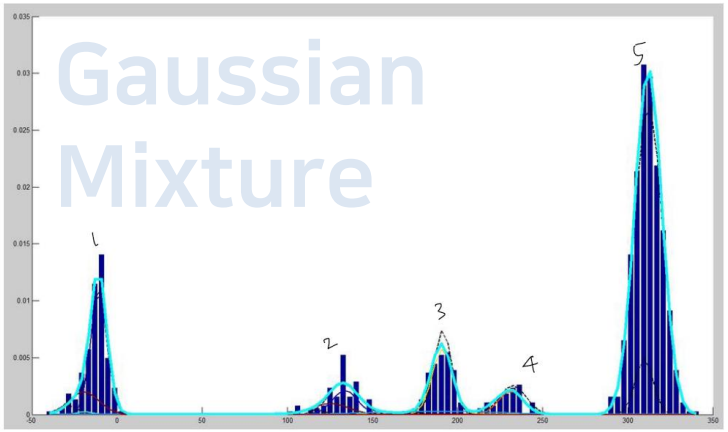
이상치 탐지

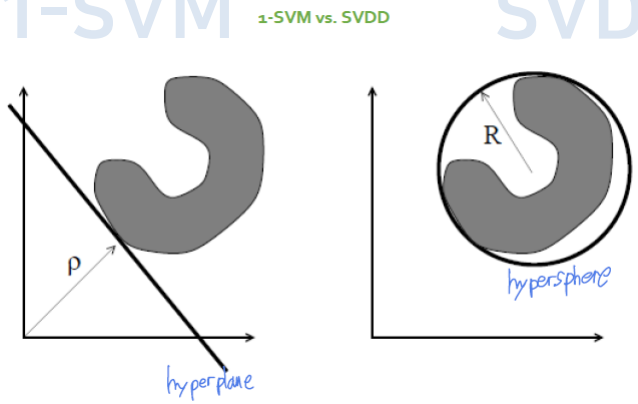
- 이상치 탐지(Outlier/Novelty/Anomaly Detection)의 관점에서 문제를 바라본 것도 채점하면서 재밌는 부분이었음
- 이상치 탐지는 극단적인 imbalance 상황에 어울리는 방법으로 비율만 보았을 때는 우리의 데이터에도 적용가능한 방법론!
- 이상치 탐지에는 다양한 방법들이 존재!
  Gaussian, Gaussian Mixture, Knn, SVDD, Isolation Forest…
- 아이디어는 좋았지만 성능이 좋지 않아서 아쉬웠음!

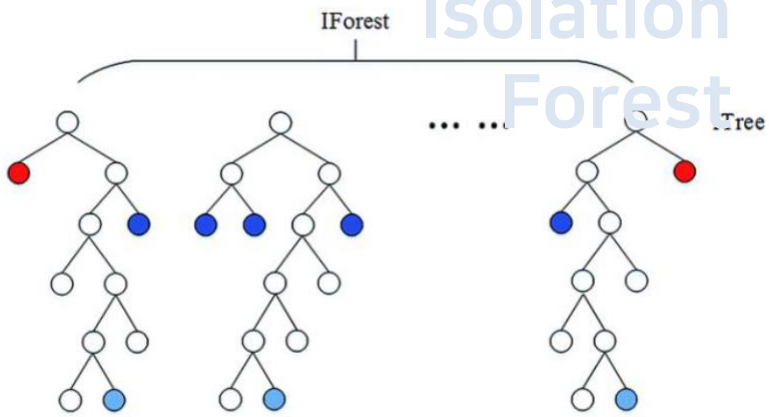
통계적 모델링과 머신러닝 실습
열혈 수강생이 많아서 흐뭇하실듯 ㅎㅎ

이상치 탐지

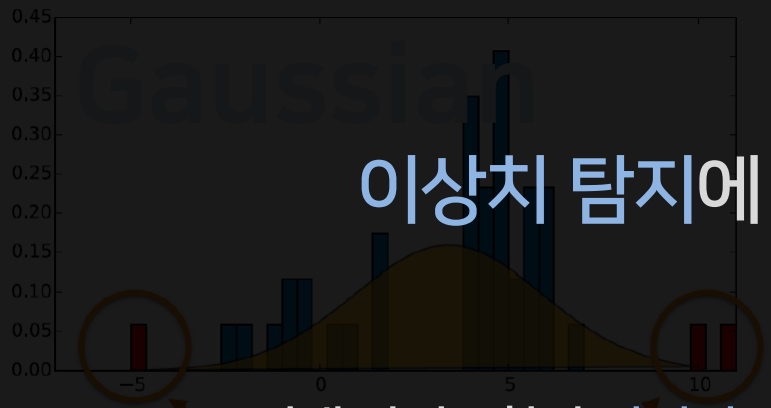

Gaussian
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−5
0
5
10
Low density regions


Gaussian
Mixture
1
2
3
4
5


1-SVM
1-SVM vs. SVDD
SVDD
ρ
hyperplane
R
hypersphere


IForest
Isolation
Forest
Tree

이상치 탐지

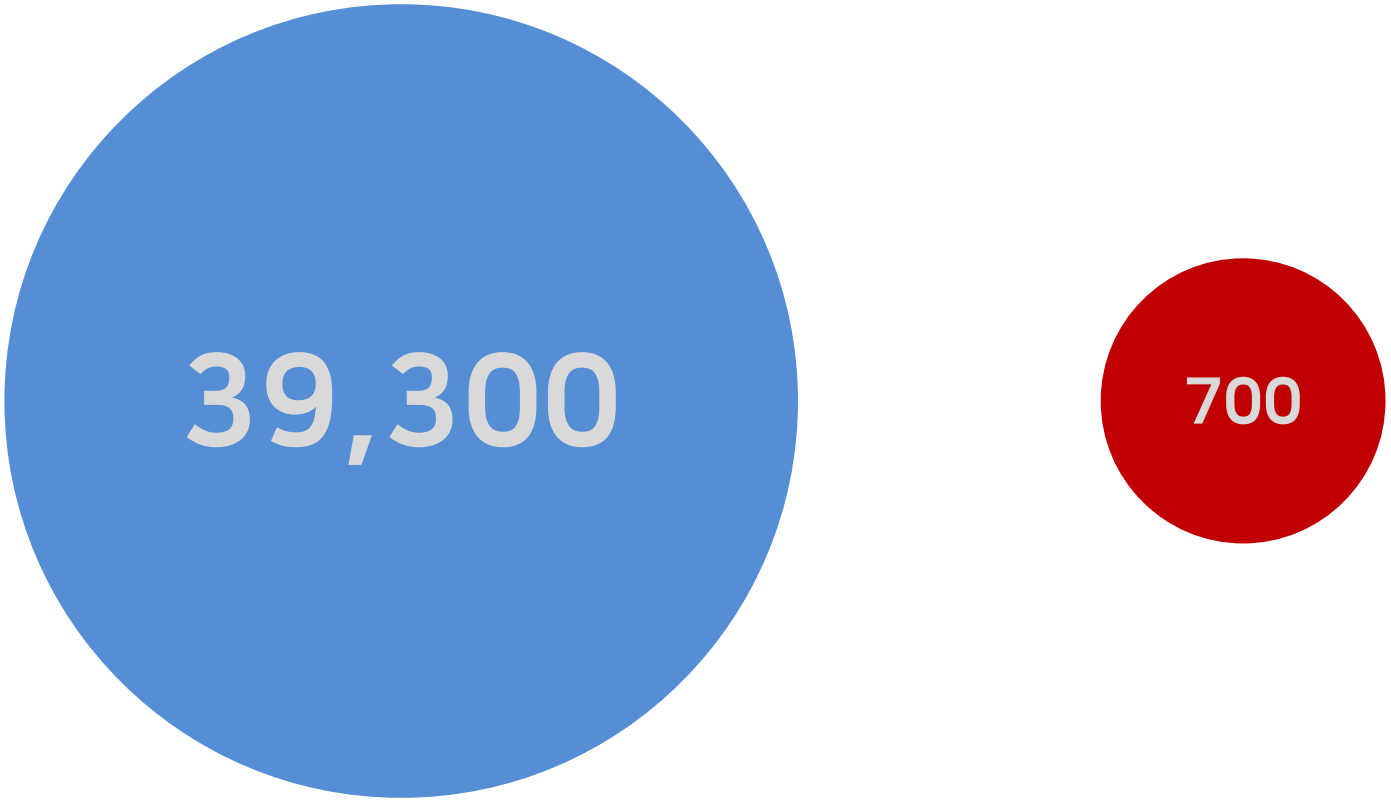


이상치 탐지에 대해 궁금하면

고려대 산업공학과 강필성 교수님 youtube 참고 추천…!

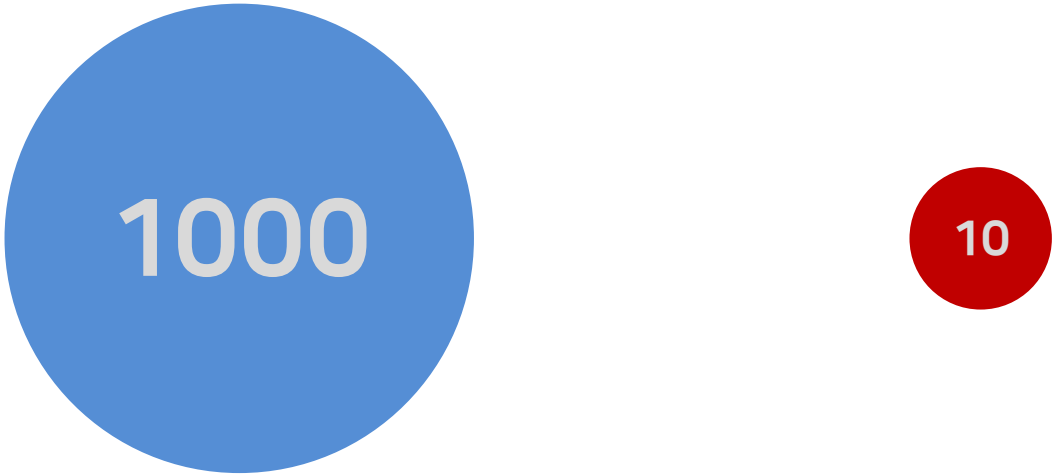
https://www.youtube.com/playlist?list=PLetSlH8YjIfWMdw9AuLR5ybkVvGcoG2EW

이상치 탐지



아이디어는 좋았지만 성능이 좋지 않았던 이유는?

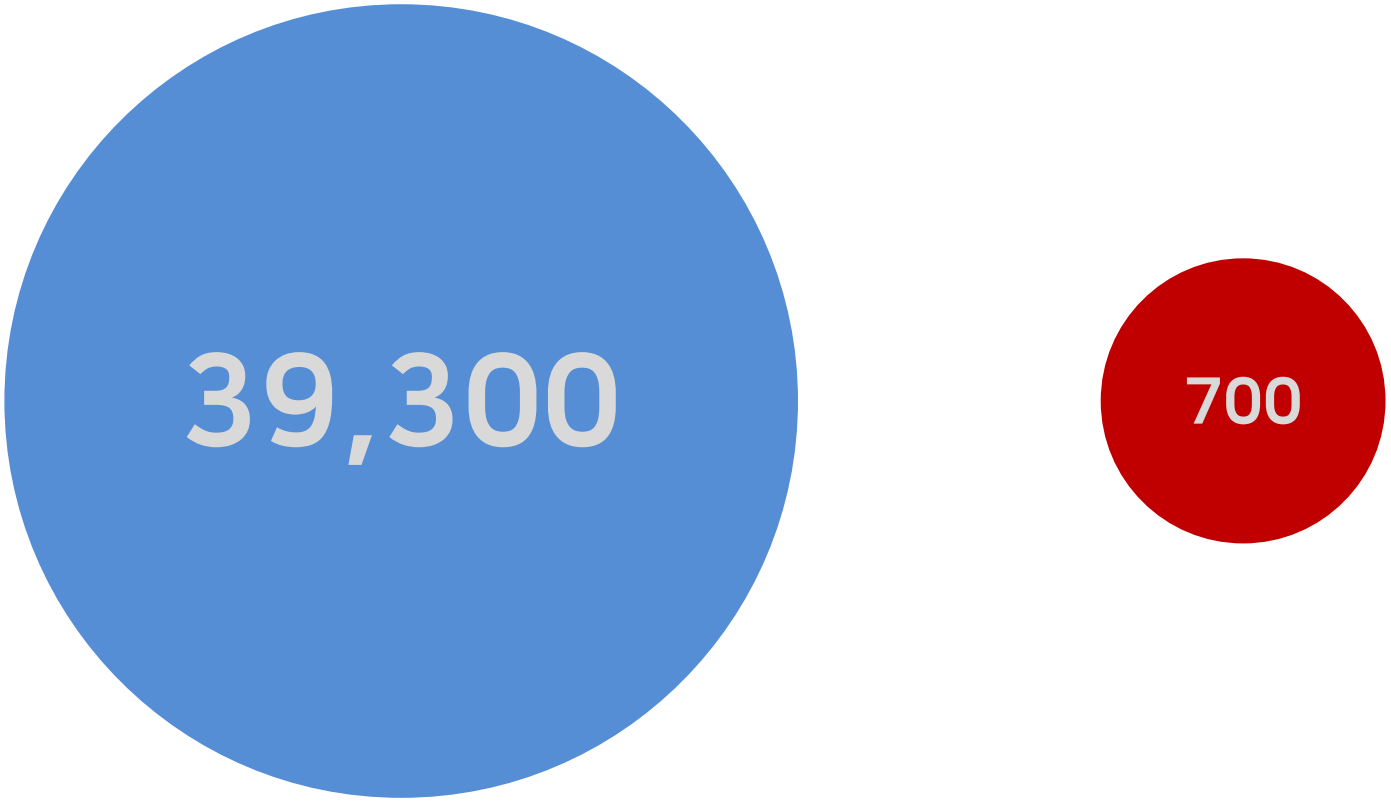
이상치 탐지



비율만큼이나 중요한 것이 작은 범주의 개수!

동일한 비율이더라도 위와 같은 상황에선 이상치 탐지 방법이 잘 작동할 수 있음

이상치 탐지



저범주가 어느정도 충분한 관측치를 가질 경우,
오버샘플링 등의 방법을 이용해 분류문제로 접근하는것이 성능이 더 좋다!

변수 변환

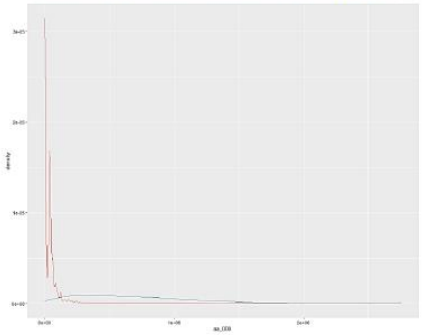
Yeo-Johnson Power Transformations

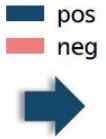
: 분산을 안정화 시키기 위한 방법의 일종으로 실수전체를 정규화 시키는 방법

데이터가 한쪽으로 쏠린 데이터가 많아 이를 해결하기 위해 Yeo-Johnson변환을 적용

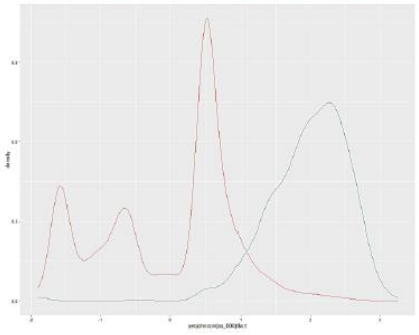
$$\psi(\lambda, y) = \begin{cases} ((y+1)^{\lambda} - 1)/\lambda & \text{if } \lambda \neq 0, y \geq 0 \\ \log(y+1) & \text{if } \lambda = 0, y \geq 0 \\ -[(-y+1)^{2-\lambda} - 1)]/(2-\lambda) & \text{if } \lambda \neq 2, y < 0 \\ -\log(-y+1) & \text{if } \lambda = 2, y < 0 \end{cases}$$

aa_000변수 Yeo-Johnson변환 적용 전





aa_000변수 Yeo-Johnson변환 적용 후

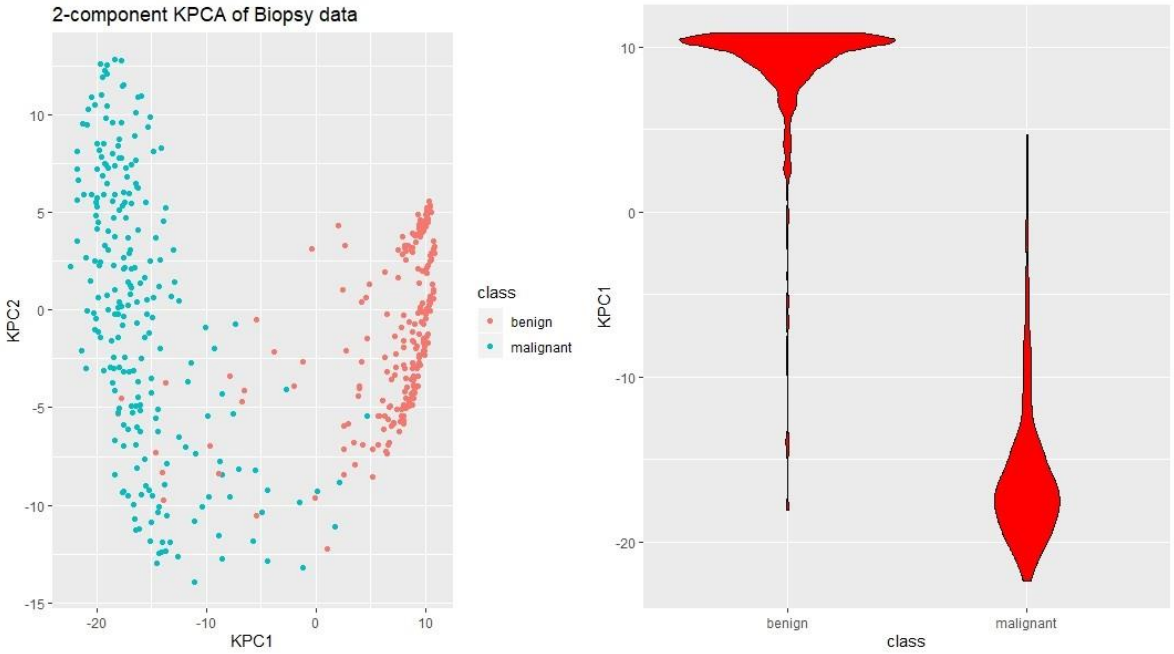


변수 변환을 거쳤을때 두 클래스가
상대적으로 명확하게 나눠지는 경향을 파악 가능!

변수 변환

- 단순히 변수 변환을 한 것을 넘어서 시각적인 차이를 제시해줘서 와닿았음

- 다른 범주의 Y는 다른 underlying distribution을 가질텐데,
  이를 명확하게 구분해주면서 성능이 크게 높아졌을듯!



3

공통 피드백

공동피드백

1. 시드 고정

R에서는 set.seed()로 간단히 시드를 고정할 수 있지만,
Python에서는 총 3가지의 시드 고정이 필요

1
```
>>> import random
>>> random.seed(100)
```

2
```
import numpy as np
np.random.seed(0)
```

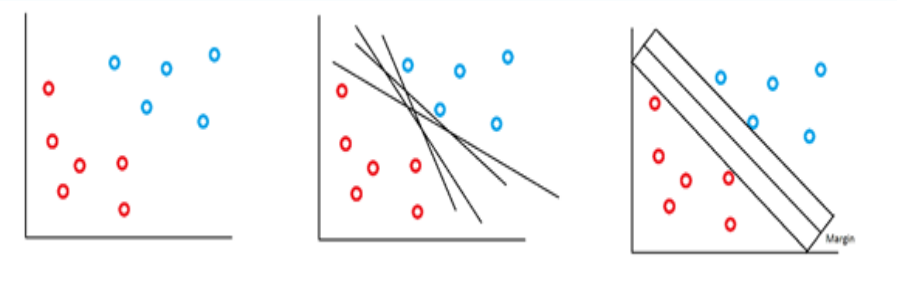
3
```
import lightgbm as lgb
train_ds = lgb.Dataset(X_train, label = y_train)
test_ds = lgb.Dataset(X_val, label = y_val)
params = {'learning_rate': 0.01,
          'max_depth': 16,
          'boosting': 'gbdt',
          'objective': 'regression',
          'metric': 'mse',
          'is_training_metric': True,
          'num_leaves': 144,
          'feature_fraction': 0.9,
          'bagging_fraction': 0.7,
          'bagging_freq': 5,
          'seed':2020}

model = lgb.train(params, train_ds, 1000, test_ds, verbose_eval=100, early_stopping_rounds=100)
y_pred=model.predict(X_val)
```

시드 미고정으로
Cost 가 달라진 경우가 발생했으니 다음부턴
시드 고정해주세용!

2. 모델 선택

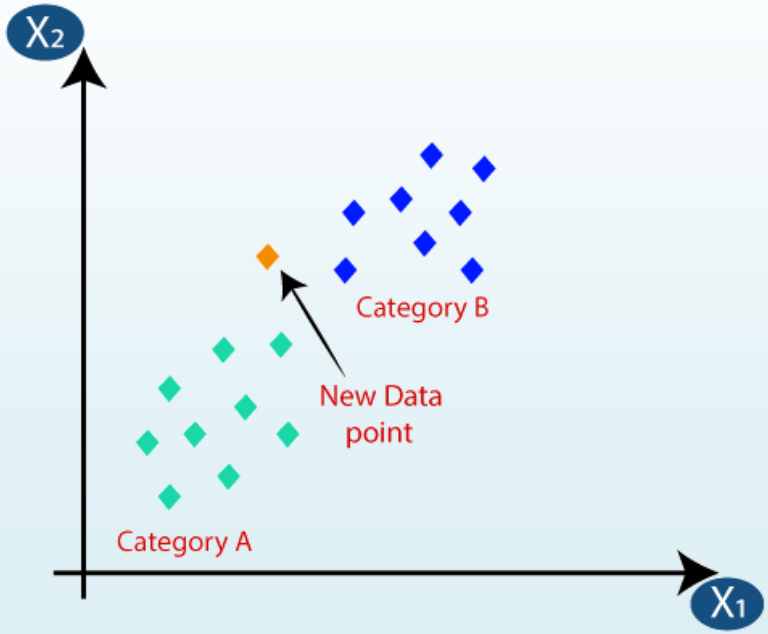
다양한 시도도 좋지만, 시간과 특성을 고려한 적절한 모델 선택 필요.
계산량이 많아 시간이 오래 걸리거나, 데이터 특성에 맞지 않는 모델은
후보군에서 배제 후 모델을 선택하는 것이 효율적



SVM, SVDD : 계산량 많음, 저범주의 양이 극단적으로 작을경우 이상치 탐지 용으로 사용

로지스틱 회귀 : 다중공선성이 높은 고차원 데이터에 패널티 부여 필요

KNN : 저차원 데이터에 적절



공동피드백

3. 모델링 FLOW, CODE

전처리, 모델링 , prediction 까지 모델링의 흐름을 잘 지킴
코드 또한 알아보기 쉽게 잘 정리 함

코드 채점이 어렵지 않았어요! 감사합니당
앞으로 모델링할 때
방세 경험과 제출코드가
많은 도움이 되었으면
좋겠습니당~~

4. 짧은 시간 내 다양한 모델링

- '불균형 처리, NA처리, 고차원 데이터의 다중공선성 해결'이라는 큰 과제를 해결하기 위해 많은 고민을 하고 시도를 하신 점이 인상깊었습니다!! 모두 노력 점 수 만 점 ㅠ

- NA의 비율을 고려한 데이터 필터링, 그리고 PCA와 상관관계를 고려한 featue engineering 및 selection 부분에 노력을 기울이신 게 보이네요…
  특히 불균형 처리, PCA, 모델 세가지를 기준으로 경우의 수를 나누어 COST를 낮추기 위한 다양한 시도를 해서 고생했다고 말해주고 싶습니다!

- 정말 다양한 모델링을 하셨는데, 데이터마이닝, 통머실, 피셋에서 배운 모든 모델링 기법을 마스터하신 것 같네요
  축하드립니다!!!!

방세 뿌셨다!!!!
(노력 천재들인27기를 표현하기 위해
패기 넘치는 어린정현과 지연의
사진을 넣어보았습니다.)

4

대망의 1등 팀 발표!

1등팀 발표

1등은 과연 어느팀?!?!

1등팀 발표

2팀 중 갖고 있는 사진이 정현이 밖에 없어서
어린 정현으로 대표사진을 넣어보았습니다
2팀 모두모두 정말 축하 드립니다!!!!!

1등팀 발표

특히 test set COST 가 가장 낮았던 2팀 !!!!
5만원으로 맛있는 거 먹으러 가시길~~~

모두 최고오오

등수 상관없이 모두 너무 고생 많으셨습니다
(총 점수 차이가 작다는 사실…><)

모두 최고오오

설날에 R/주피터 절대 켜지 말고
잘 쉬고
개강하고 보아용!
(팀장들은 교안 만듭시다^^)