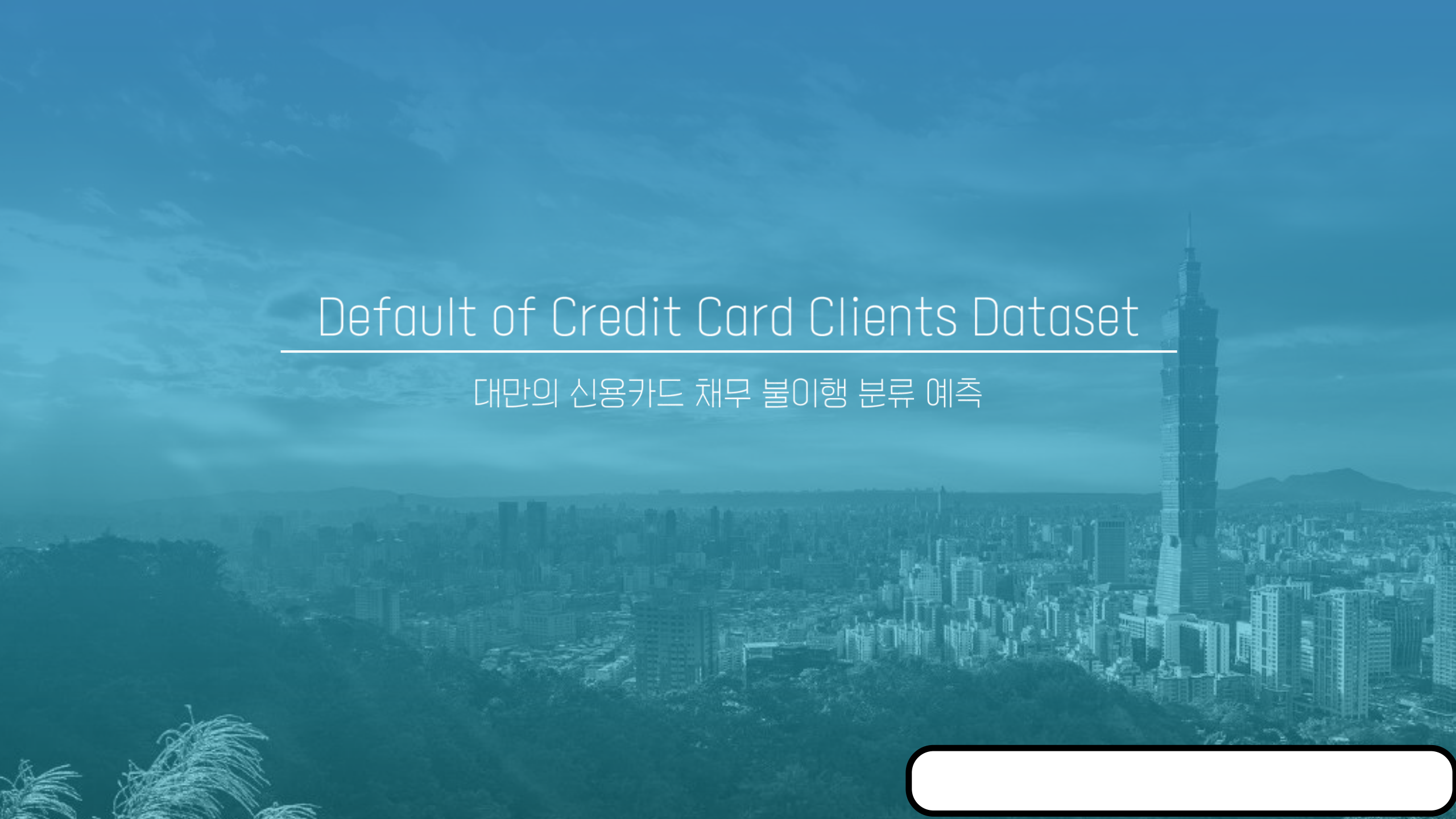


Default of Credit Card Clients Dataset

대만의 신용카드 채무 불이행 분류 예측

Context

01 데이터 소개

02 EDA(탐색적 자료 분석)

03 전처리

- 차원축소
- SMOTE / RUS

04 Modeling

04 결론

데이터 소개

Data explain

Default of Credit Card Clients Dataset

Default Payments of Credit Card Clients in Taiwan from 2005

대만의 고객 채무불이행 발생 사례 대상으로 채무불이행 발생 분류 예측 정확도 비교

2005년 4월부터 2005년 9월까지

대만의 신용카드 고객들의 채무불이행, 인구통계학적 요인, 신용 데이터, 결제이력, 청구서 등에 대한 정보

from

Name: I-Cheng Yeh
email addresses: (1) icyeh '@' chu.edu.tw (2) 140910 '@' mail.tku.edu.tw
institutions: (1) Department of Information Management, Chung Hua University, Taiwan.
(2) Department of Civil Engineering, Tamkang University, Taiwan.
other contact information: 886-2-26215656 ext. 3181

변수설명

Number of instances : 30000

변수명	데이터 타입	내용
ID	범주형	고객에게 부여한 번호
AGE	범주형	나이
SEX	범주형	성별 (1 – 남성 / 2 – 여성)
EDUCATION	범주형	학력수준(졸업), 1 - 대학원 / 2 – 대학 / 3 – 고등학교 / 4 – other
MARRIAGE	범주형	결혼여부, 1 – 결혼 / 2 – 미혼 / 3 - others
LIMIT_BAL	연속형	신용한도(개인 및 가족/보조금 포함), Limit Balance
PAY_(숫자)	범주형	상환상태, ex) PAY_1 = -1, 2005년 9월 제 때 상환한 상황 나타냄
BILL_AMT(숫자)	연속형	청구서 상에 청구된 금액, ex) BILL_AMT1 = 1000, 2005년 9월 청구된 금액이 1000NT$
PAY_AMT(숫자)	연속형	지급한 금액, ex) PAY_AMT1 = 1000, 2005년 9월 지불한 금액이 1000NT$
Default.paymet.next.month	범주형	다음달(2005년 10월)에 체납유무, 1 – 그렇다 / 0 - 아니다

NT$: New Taiwan dollar, NT dollar

변수설명(자세히)

상환상태

-2 : 카드 사용 X
-1과 0 : 상환
1 ~ : 연체된 개월 수

변수명	값	내용
PAY_1	-2	2005/09
PAY_2	-2	2005/08
PAY_3	-2	2005/07
PAY_4	-2	2005/06
PAY_5	-1	2005/05
PAY_6	-1	2005/04

청구서에 청구된 금액

변수명	값	내용
BILL_AMT1	0	2005/09
BILL_AMT2	0	2005/08
BILL_AMT3	0	2005/07
BILL_AMT4	0	2005/06
BILL_AMT5	13007	2005/05
BILL_AMT6	13912	2005/04

지불금액

변수명	값	내용
PAY_AMT1	0	2005/09
PAY_AMT2	0	2005/08
PAY_AMT3	0	2005/07
PAY_AMT4	13007	2005/06
PAY_AMT5	1122	2005/05
PAY_AMT6	0	2005/04

Example 1

2005년 5월 청구서 금액 : 13007
BILL_AMT5

▶

2005년 6월 상환 금액 : 13007
PAY_AMT4

▶

2005년 5월 상환상태 : -1
PAY_5

▶

성실히 잘 갚는군!

변수설명(자세히)

상환상태

▶ -2 : 카드 사용 X
-1과 0 : 상환
1 ~ : 연체된 개월 수

변수명	값	내용
PAY_1	-2	2005/09
PAY_2	-2	2005/08
PAY_3	-2	2005/07
PAY_4	-2	2005/06
PAY_5	-1	2005/05
PAY_6	-1	2005/04

청구서에 청구된 금액

변수명	값	내용
BILL_AMT1	0	2005/09
BILL_AMT2	0	2005/08
BILL_AMT3	0	2005/07
BILL_AMT4	0	2005/06
BILL_AMT5	13007	2005/05
BILL_AMT6	13912	2005/04

지불금액

변수명	값	내용
PAY_AMT1	0	2005/09
PAY_AMT2	0	2005/08
PAY_AMT3	0	2005/07
PAY_AMT4	13007	2005/06
PAY_AMT5	1122	2005/05
PAY_AMT6	0	2005/04

Example 2

2005년 6월 청구서 금액 : 0

BILL_AMT4

▶

2005년 7월 상환 금액 : 0

PAY_AMT3

▶

2005년 6월 상환상태 : -2

PAY_4

▶

사용한 금액이 없군!

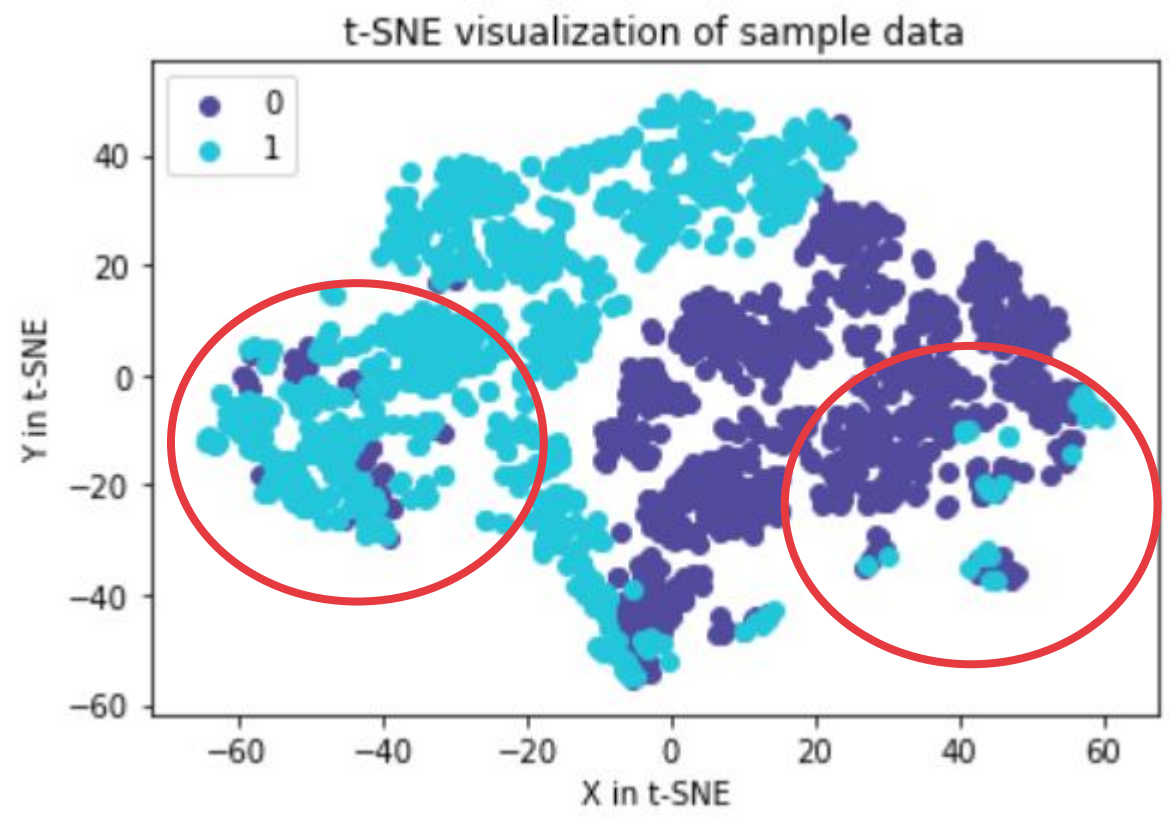
변수설명(자세히)

파생변수명	데이터 타입	내용	생성 방법
USE_AMT(숫자)	연속형	사용한 금액	이번 달 청구액 – (저번달 청구액 – 저번달 갚은 금액) Ex) BILL_AMT5 – (BILL_AMT6 – PAY_AMT5) = USE_AMT5
USE_RATE(숫자)	연속형	신용한도 대비 사용금액	사용금액 / 신용한도 의 비율

EDA(탐색적 자료 분석)

시각화

t-SNE를 이용한 시각화



Manifold 기법의 한 종류

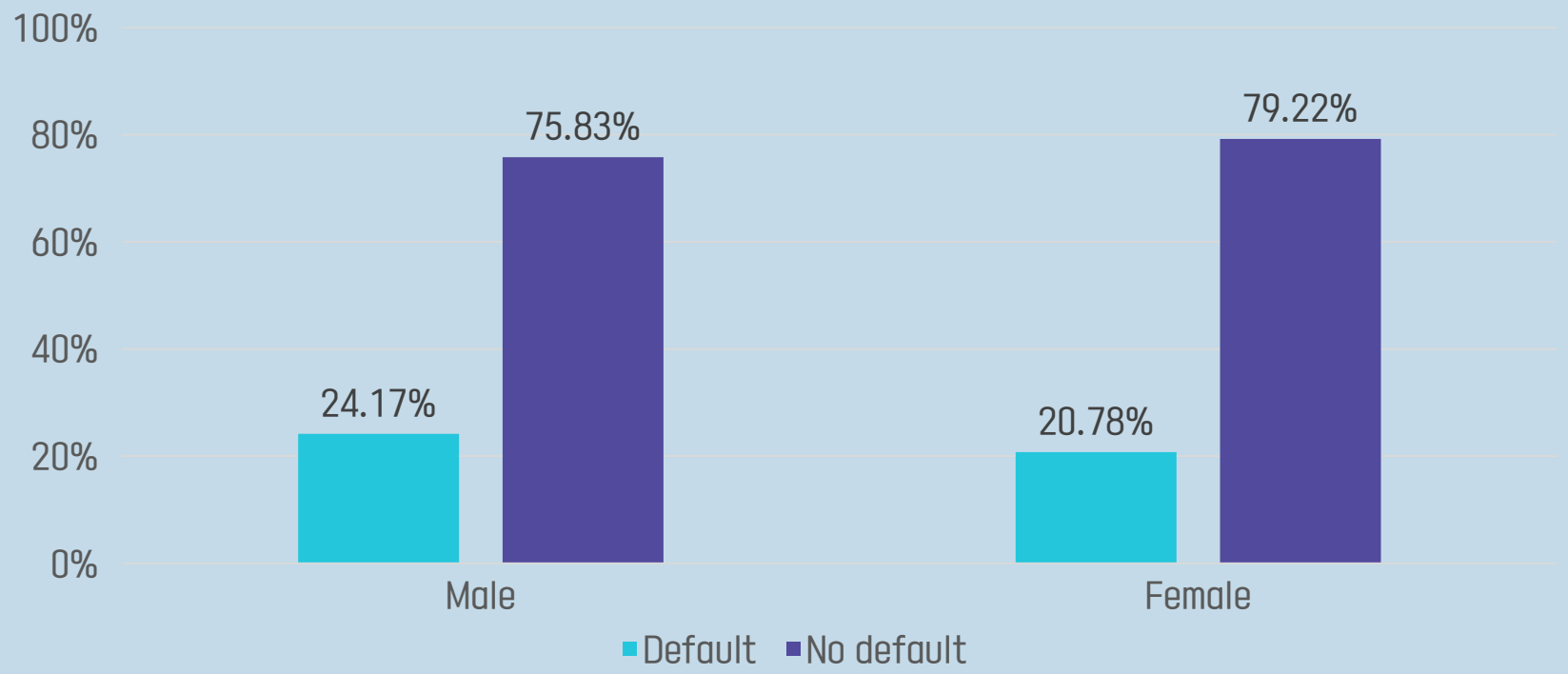
고차원의 데이터에 대해 이웃들 간의 거리를
보존하면서 저차원으로 축소하는 방법

개략적인 분포를 확인할 때 쓰인다.

어느정도 클래스의 구분은 가능함
하지만, 부분적으로 구분이 불가능한 영역이 존재

시각화

성별 연체 분류

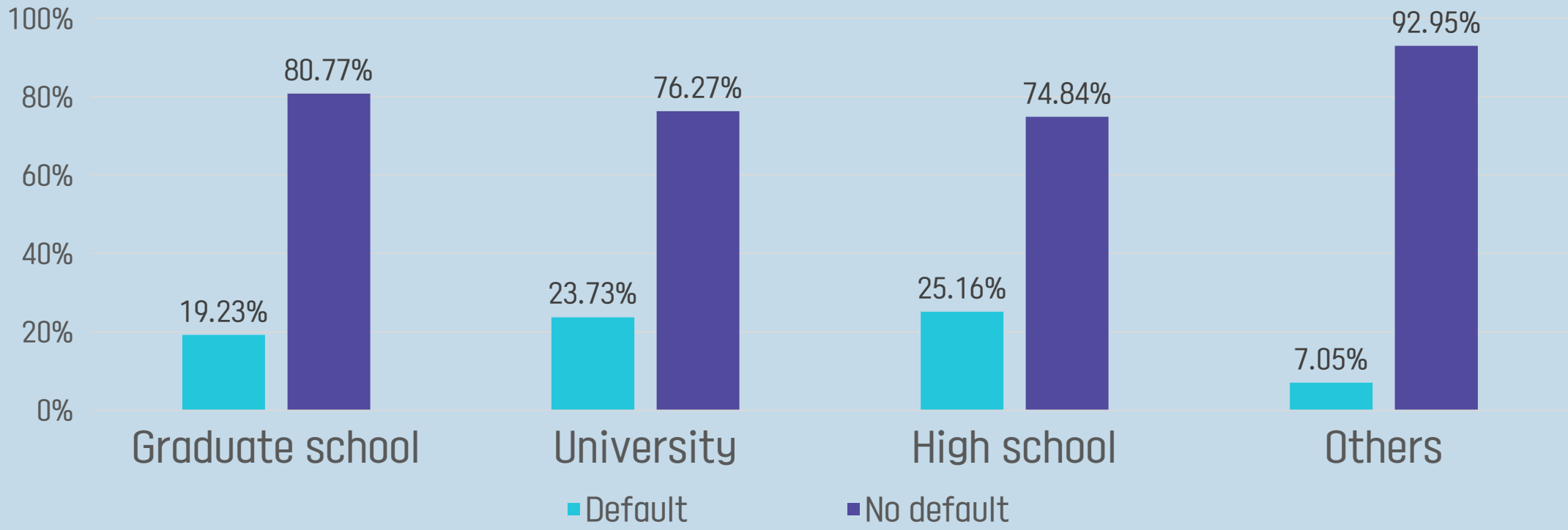


남녀의 데이터 수는 5 : 5

다소 차이가 존재하나 높은 상관성 X

시각화

교육수준별 연체 분포

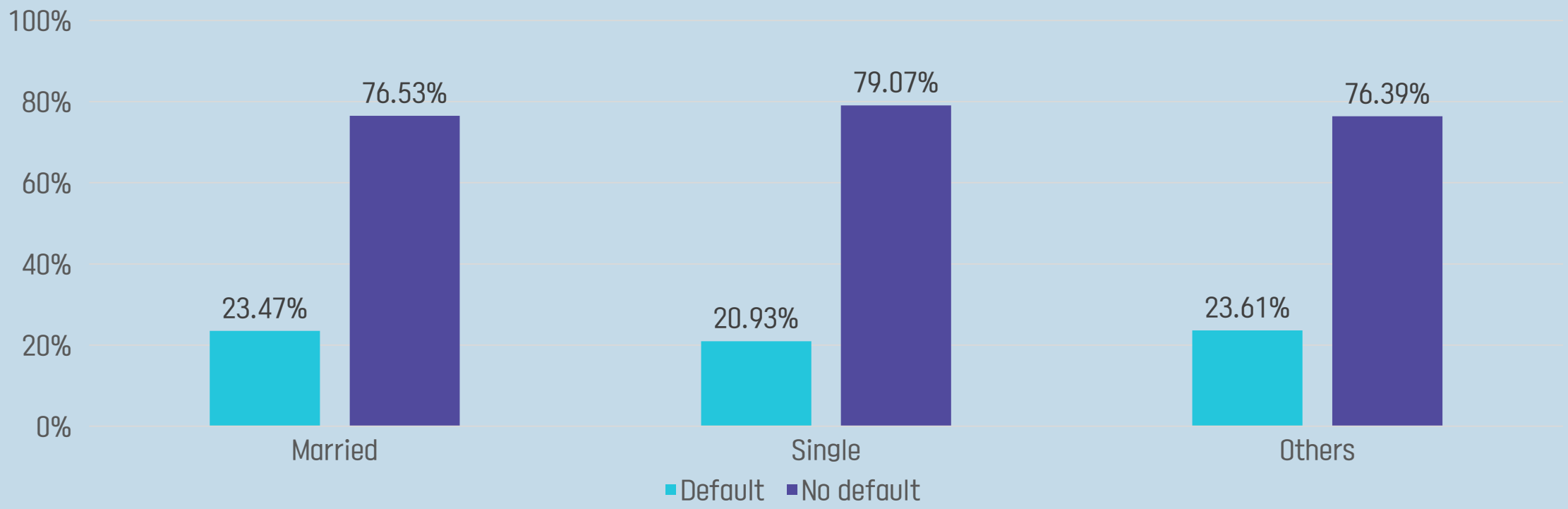


학력 수준이 낮을수록 연체비율이 다소 높아짐

Graduate school(대학원) / Others (결측치)

시각화

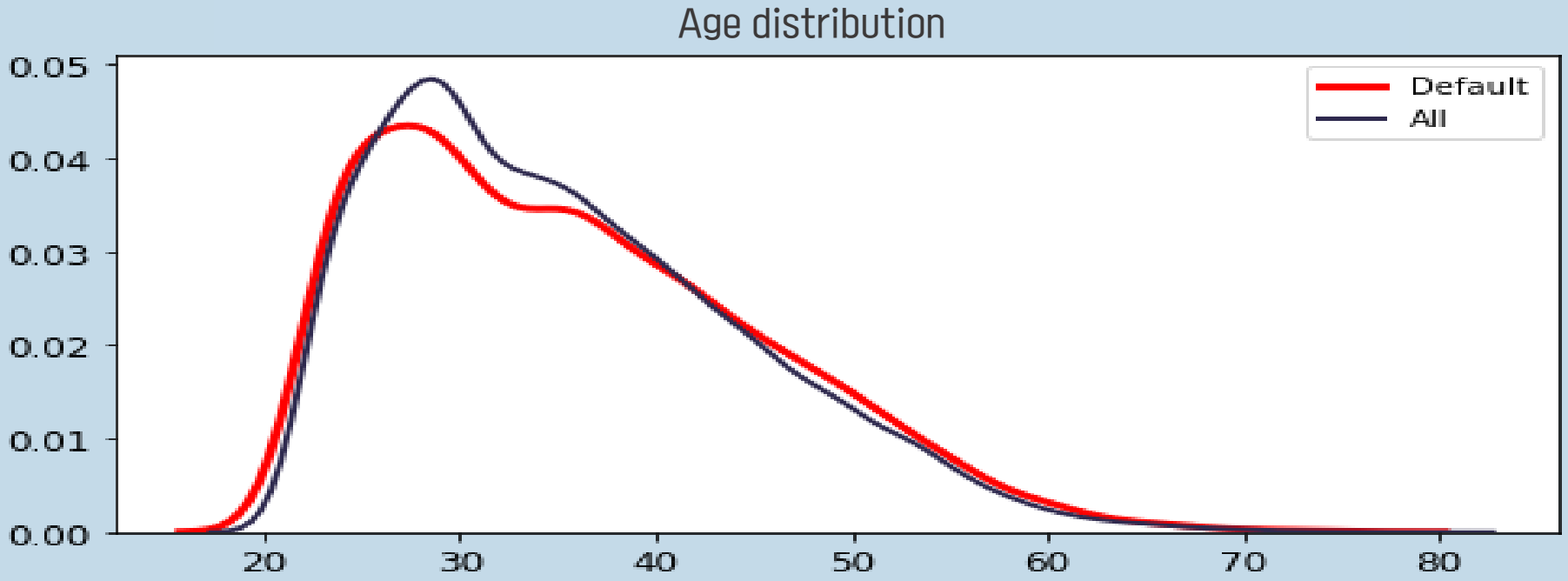
혼인상태별 연체 분포



싱글일 경우 타 그룹에 비해 연체율이 낮음

시각화

나이별 연체 분포

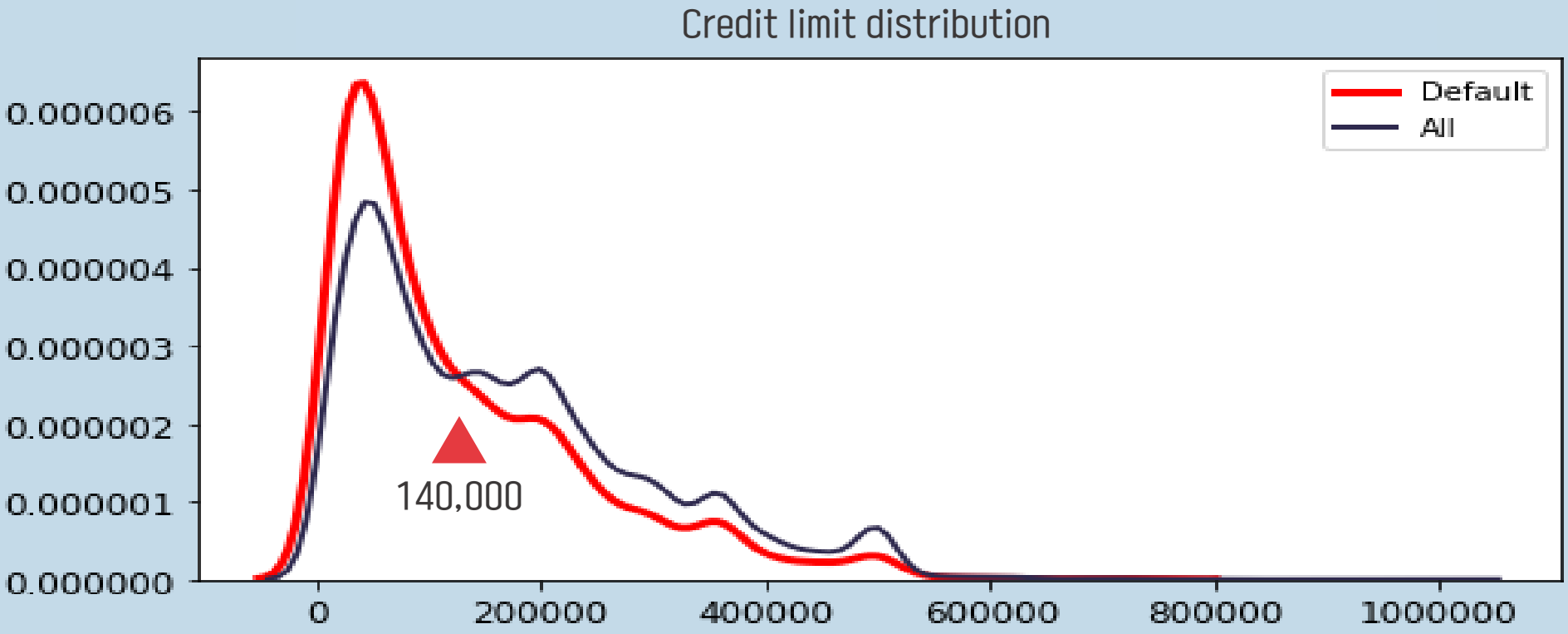


나이에 따른 데이터 수 분포와 Default 비율 분포는 유사함

그러나, 25~40세 구간의 경우 상환율이 높음

시각화

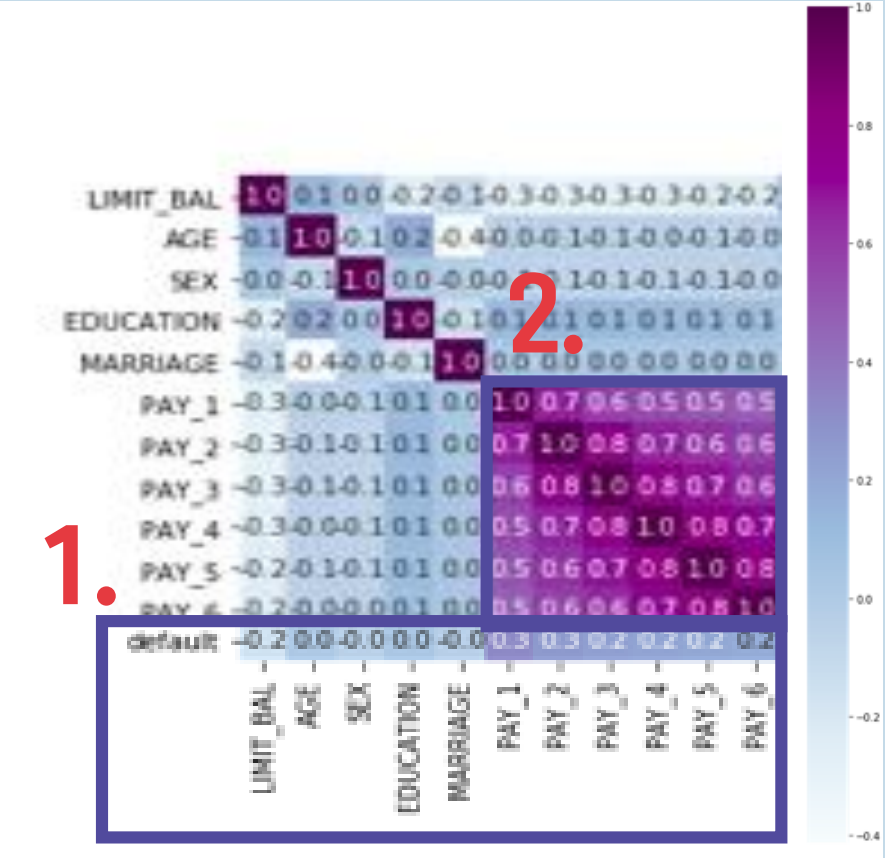
신용한도별 연체 분류



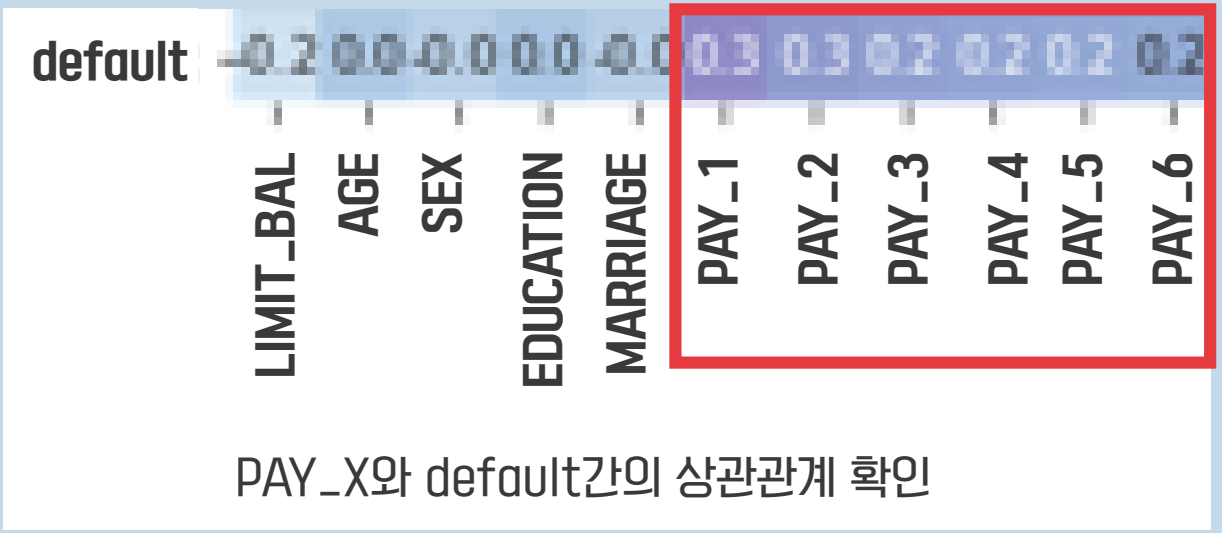
신용한도 낮을수록 연체율 높음

시각화

상관관계 분석



1. Default와 PAY_X 간의 상관관계



PAY_X와 default간의 상관관계 확인

2. PAY_X(1~6) 간의 상관관계

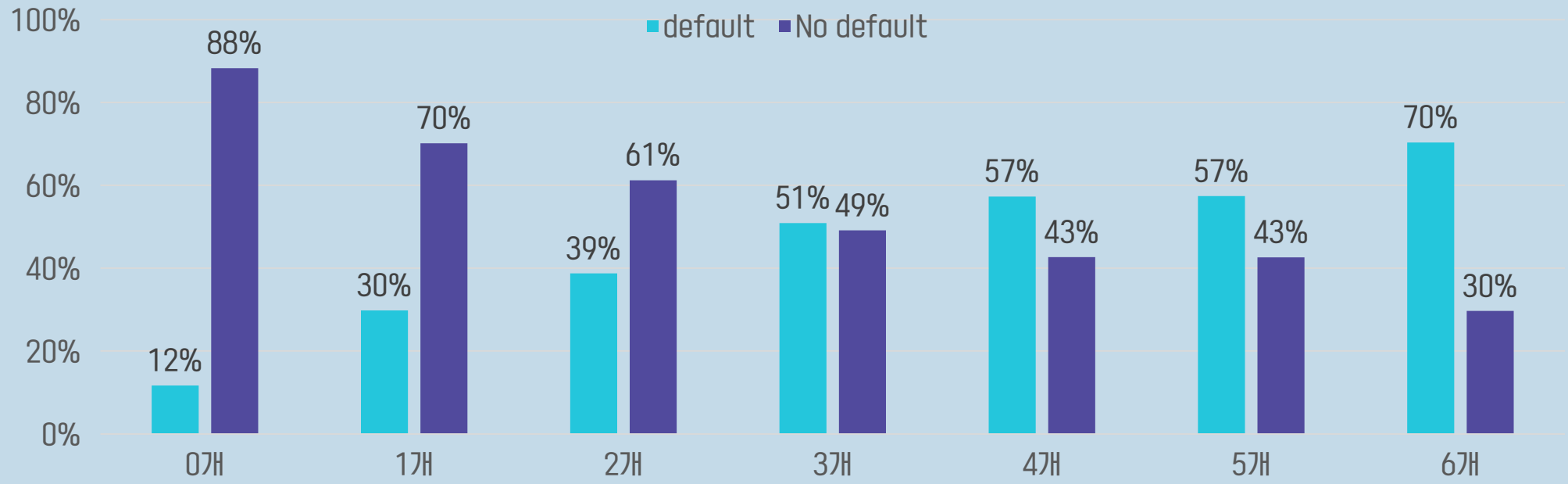
3. 2번에 대한 정보를 조합하여 연체에 대한 파생변수 생성

파생변수설명(자세히)

파생변수명	데이터 타입	내용	생성 방법
PAY_CNT(숫자)	연속형	연체를 한 개월의 개수	PAY_(숫자) >= 1 인 수들의 개수 (PAY_1부터 PAY_6까지)
PAY_SUM(숫자)	연속형	연체를 한 총 기간	PAY_(숫자) >= 1 인 수들의 합 (PAY_1부터 PAY_6까지)

시각화

파생변수 1. PAY_CNT

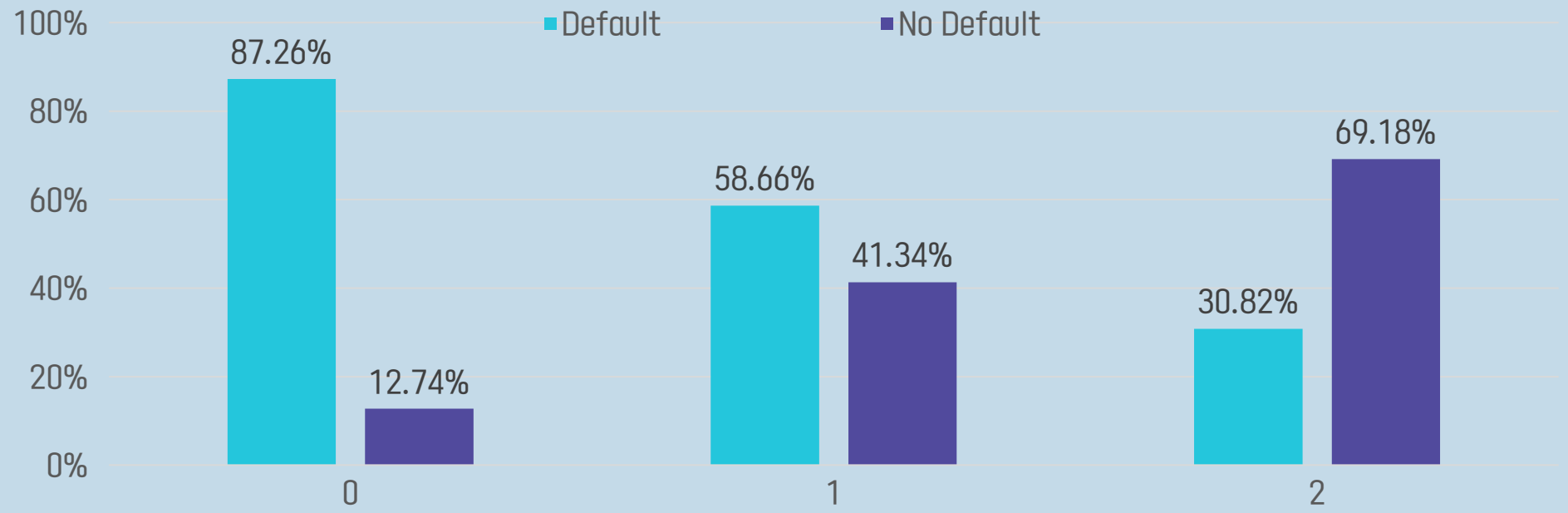


PAY_CNT(숫자)	연체를 한 개월의 개수	PAY_(숫자) >= 1 인 수들의 개수 (PAY_1부터 PAY_6까지)

PAY_CNT와 default가 높은 상관관계를 보임

시각화

파생변수 2. PAY_SUM



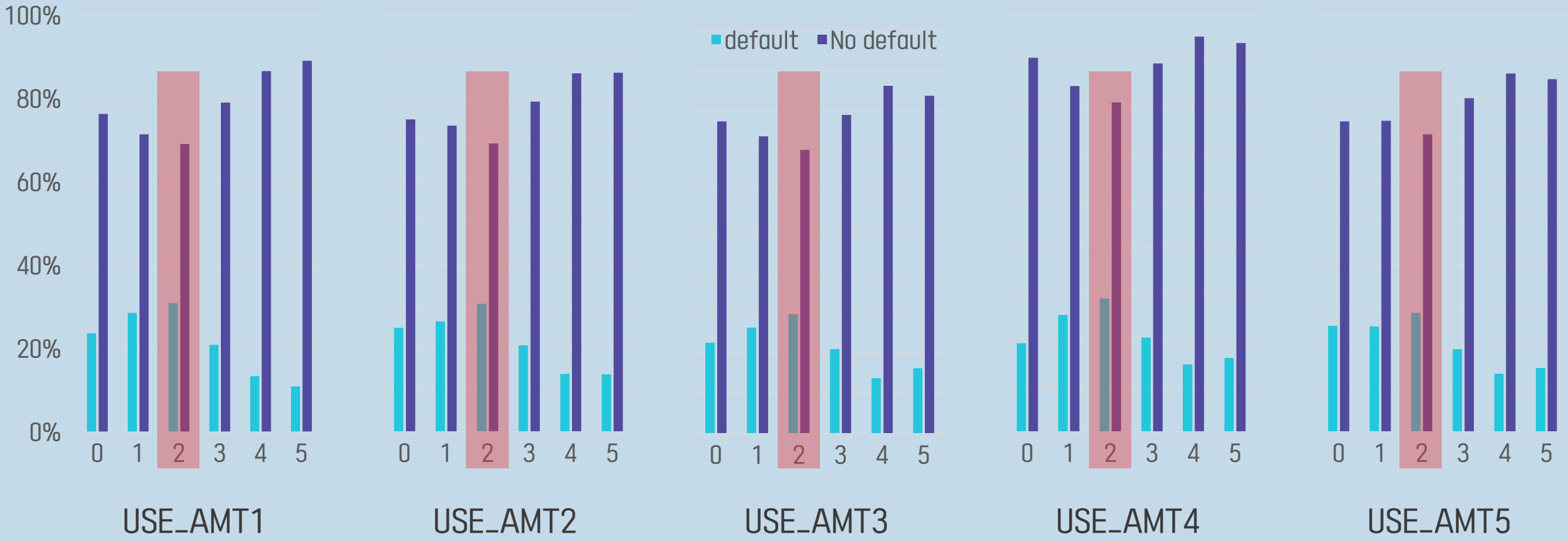
PAY_SUM(숫자)	연체를 한 총 개월 기간	PAY_(숫자) >= 1 인 수들의 합 (PAY_0부터 PAY_6까지)

[0, 36] 사이의 값을 3개 구간으로 범주화

PAY_SUM과 default 높은 상관관계를 보임

시각화

파생변수 3. USE_AMT

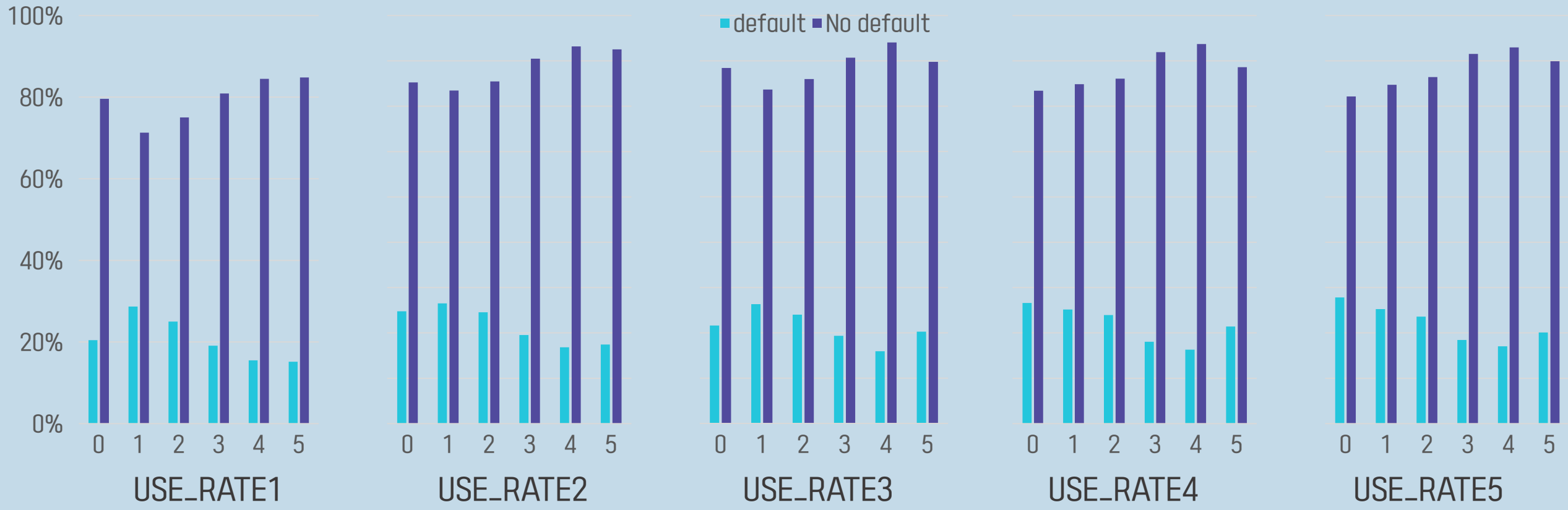


소비금액 6개 구간으로 범주화

사용금액 규모가 중간(2)인 그룹이 연체 위험이 가장 높음

시각화

파생변수 4. USE_RATE

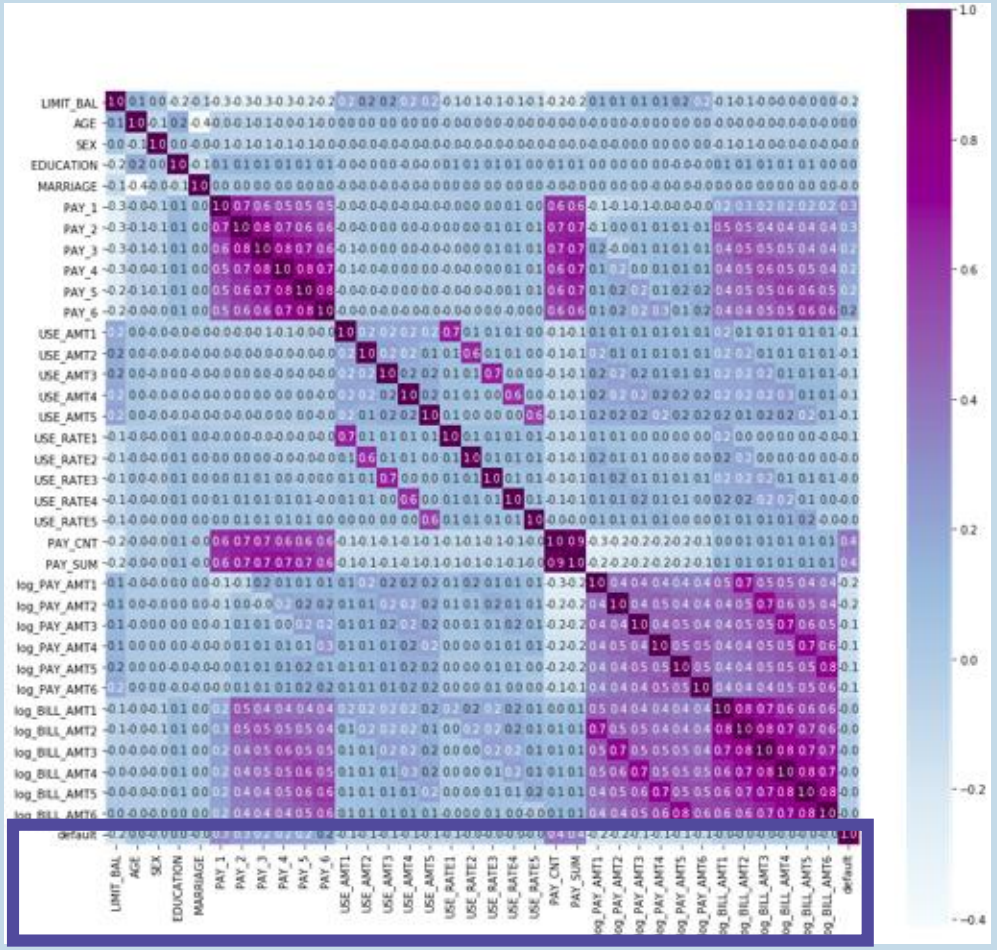


신용한도 대비 사용금액을 6개 구간으로 범주화

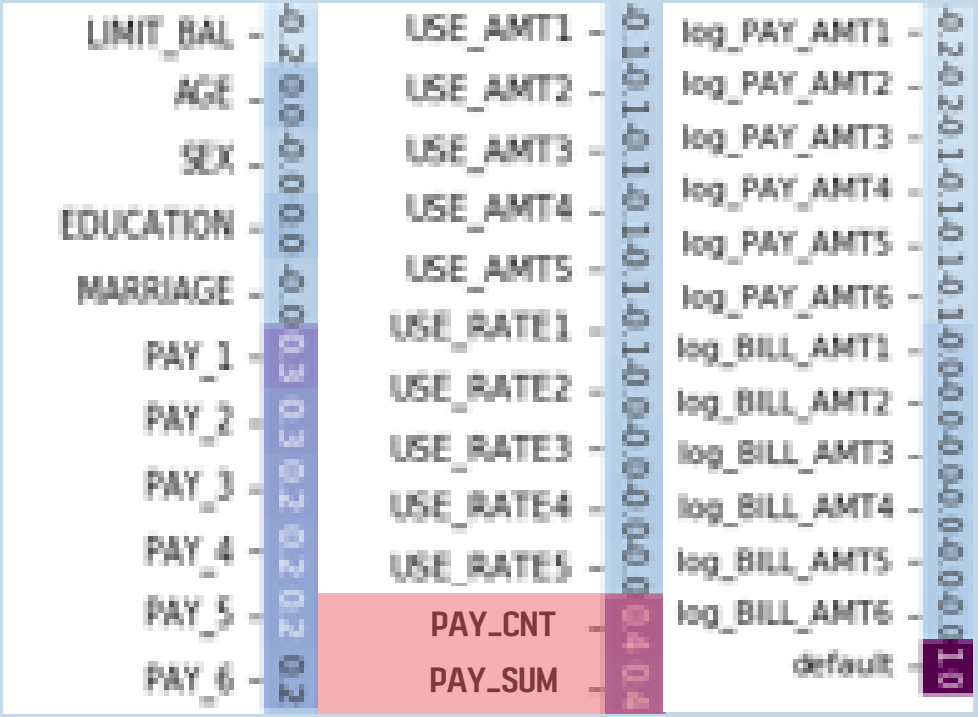
상관성이 없음

시각화

상관관계 분석



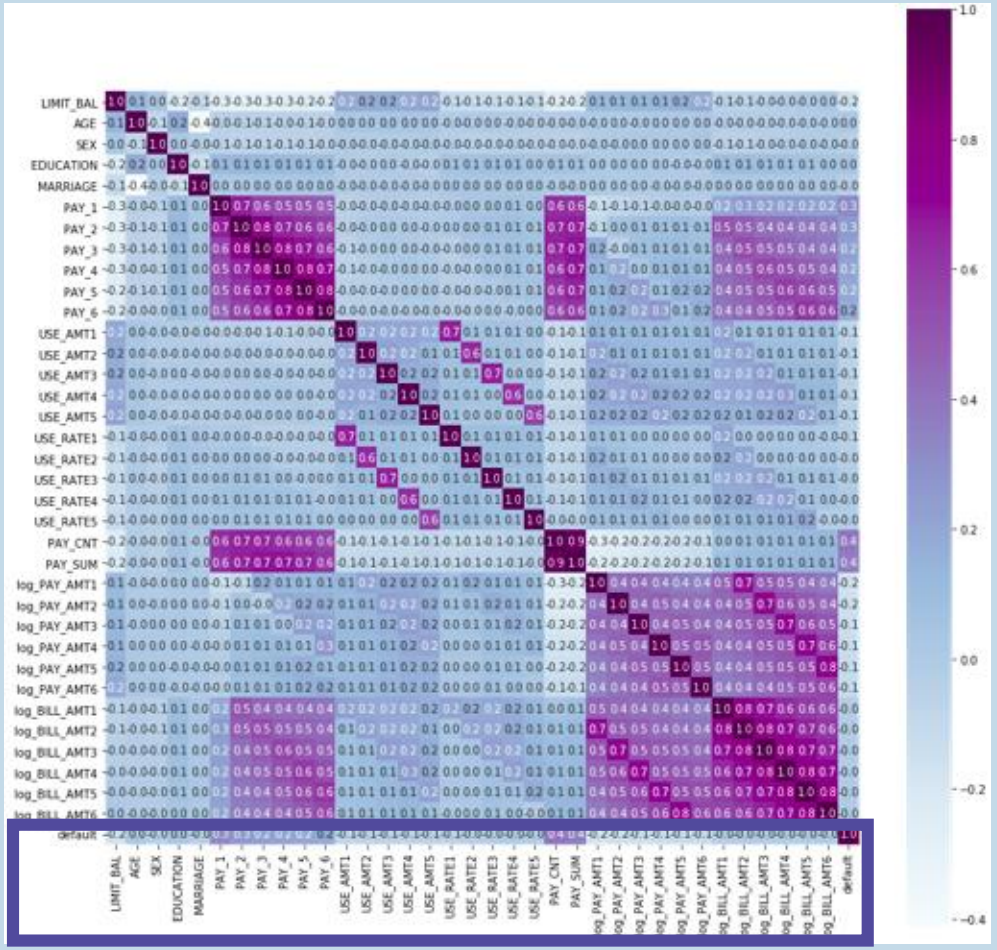
Default 부분 확대



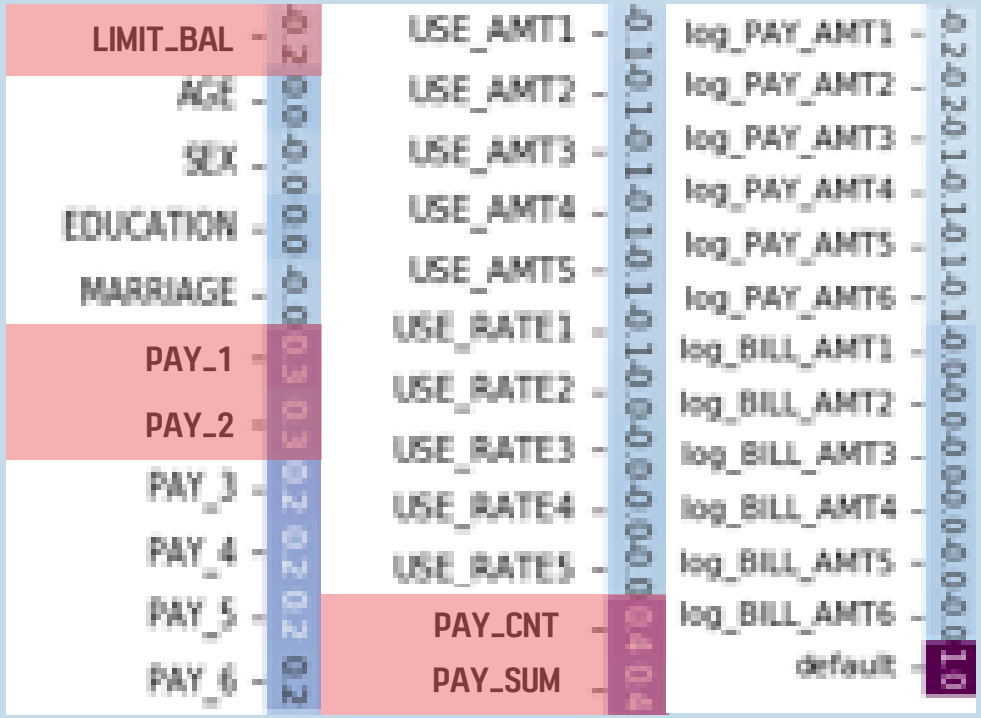
PAY_CNT, PAY_SUM (파생변수) : 0.4

시각화

상관관계 분석



Default 부분 확대



주요 변수

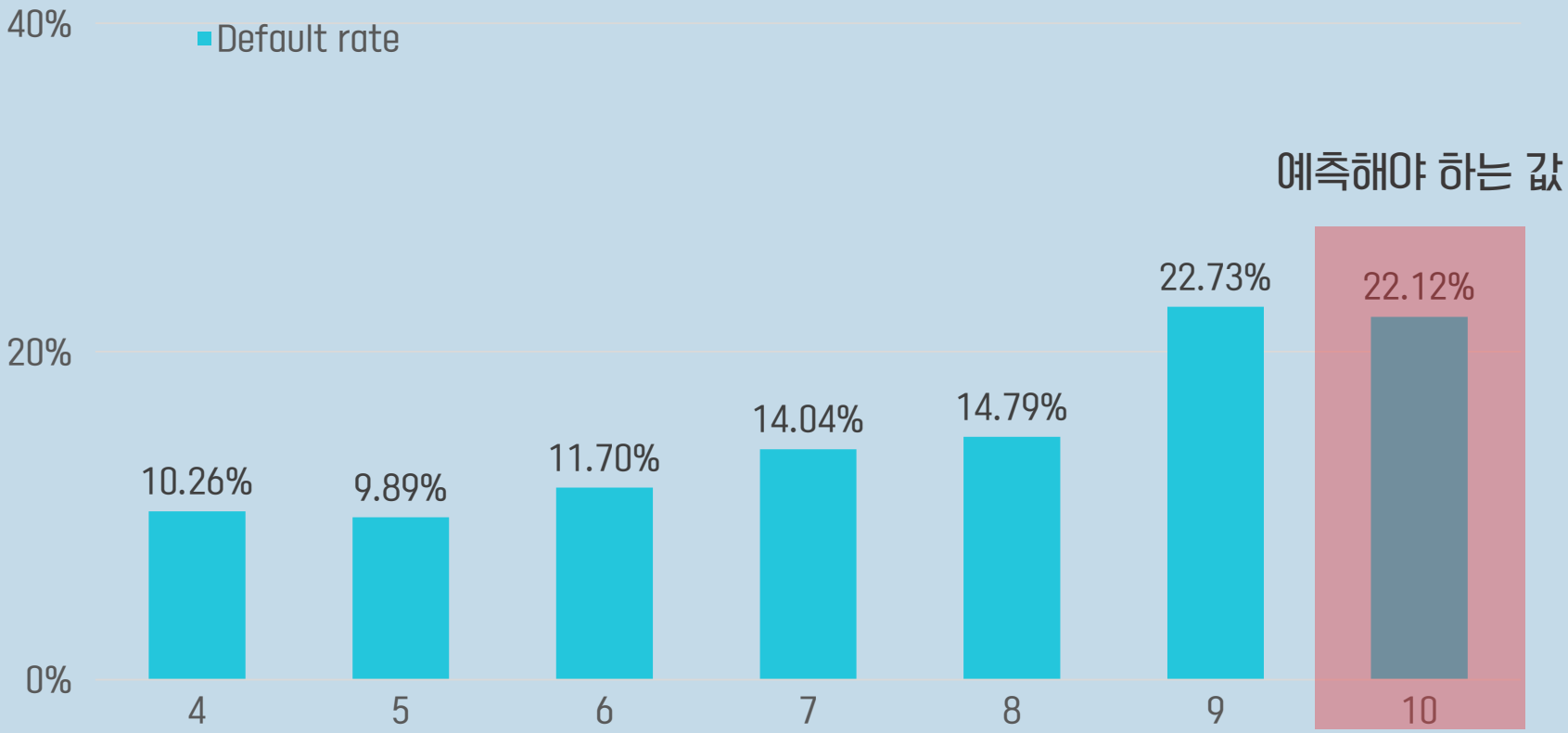
PAY_1, PAY_2 (8월, 9월 상환상태) : 0.3

LIMIT_BAL (신용한도) : -0.2

PAY_CNT, PAY_SUM (파생변수) : 0.4

시각화

월별 연체율



9월(PAY_1)부터 연체율 급격히 증가

전처리

차원축소

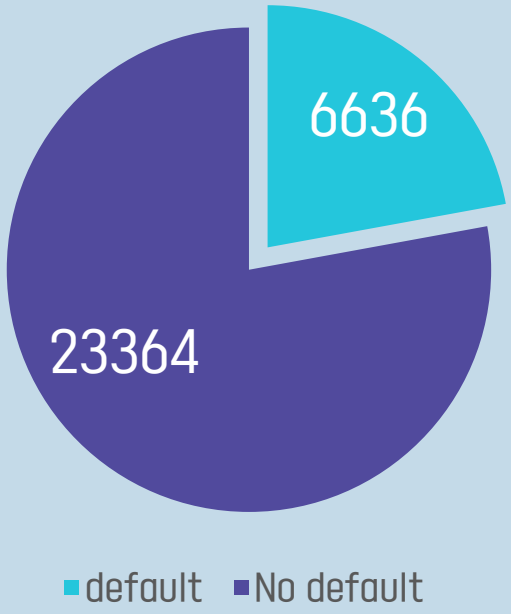
차원 축소 고차원 데이터 특성을 저차원 데이터의 특징으로 표현하는 작업

- PCA (주성분 분석)
  - 설명력
    - 0.9
    - 0.95
    - 0.99
- Kernel PCA
  - Kernel function
    - RBF
    - Linear
- Manifold
  - LLE
  - t-SNE
- Random Forest feature selection
- XGBoost feature selection
- SelectKBest
- LGBM

Imbalaced Sampling

불균형

고차원 데이터 특성을 저차원 데이터의 특징으로 표현하는 작업



default(22%) vs No default(78%) 로 샘플의 불균형 발생

이를 해결하기 위해 SMOTE / RUS 실시

Imbalaced Sampling

불균형

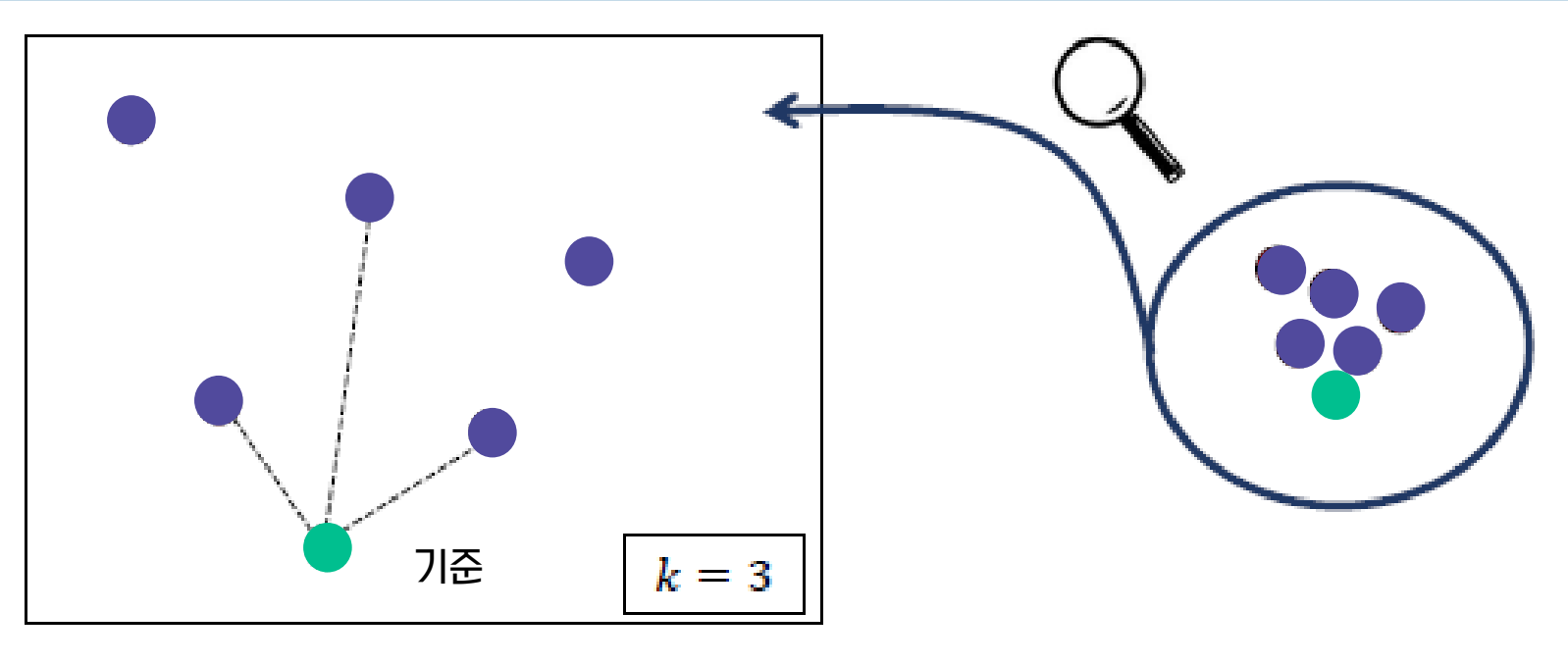
	작동원리	장점	단점
오버 샘플링	큰 쪽에 맞추는 것	- 샘플 삭제가 없기 때문에 정보 손실 없음	- 언더 샘플링에 비해 연산속도 느림 - 오버샘플링 발생 가능성
언더 샘플링	작은 쪽에 맞추는 것	- 데이터 셋의 사이즈를 축소시킴 - 오버 샘플링에 비해 연산 속도 낮음	- 샘플을 지우기 때문에 데이터에 대한 정보손실

SMOTE : synthetic minority Over Sampling

RUS : Random Under Sampling

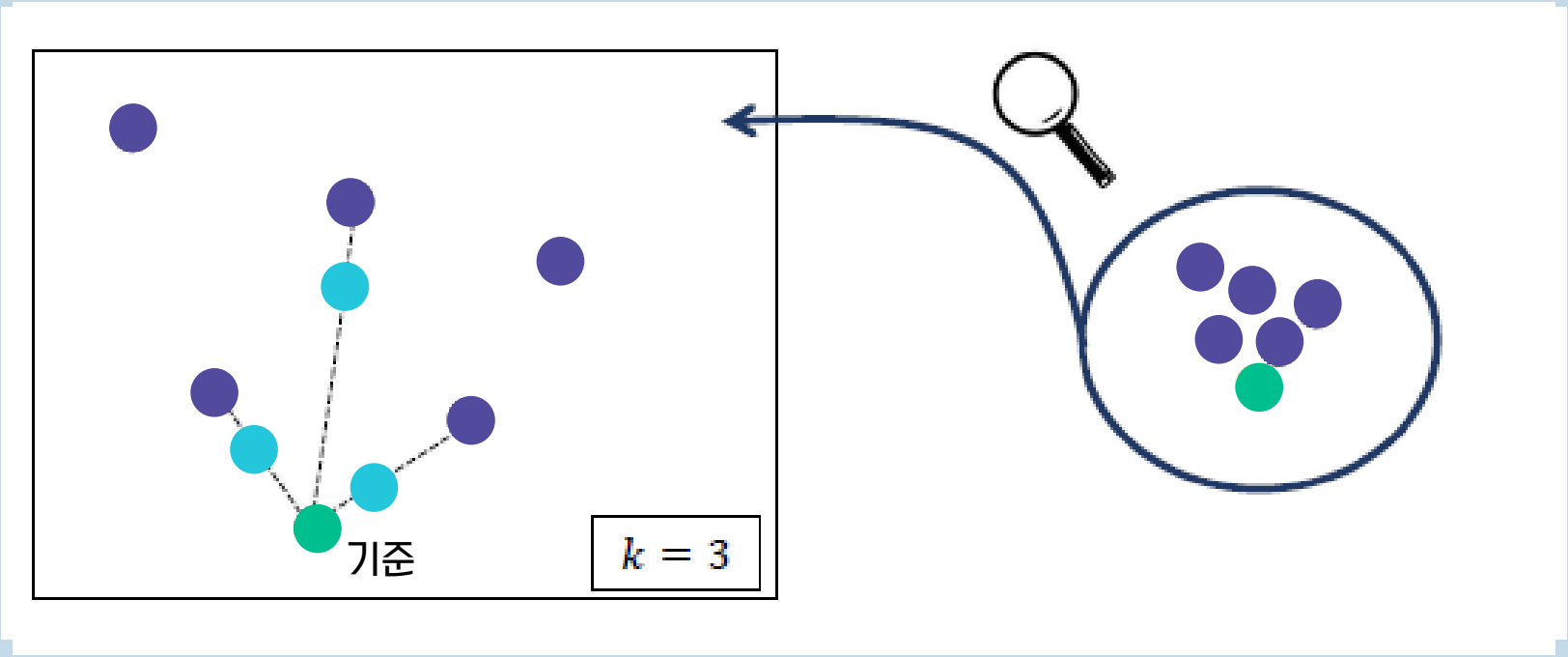
SMOTE

K nearest neighbors 고르기 (K = hyperparameter)



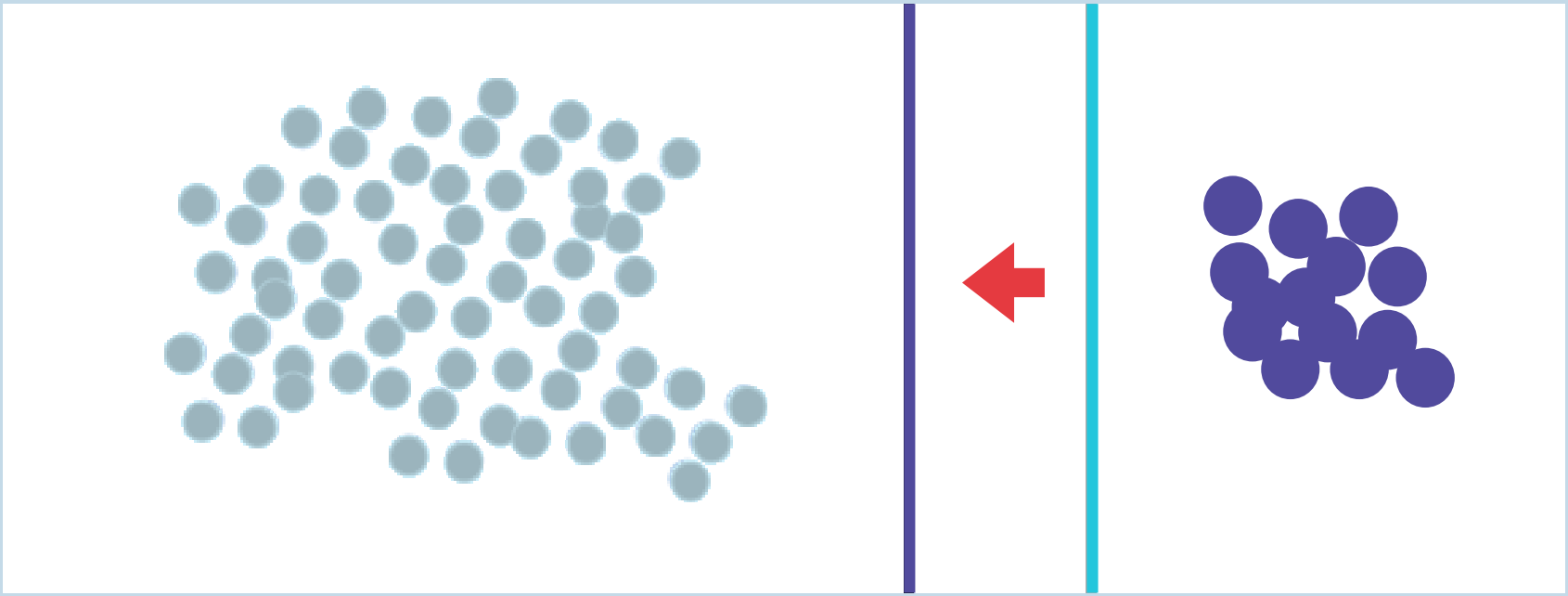
SMOTE

두 점 사이를 일적선으로 하여 임의의 minority class를 생성



SMOTE

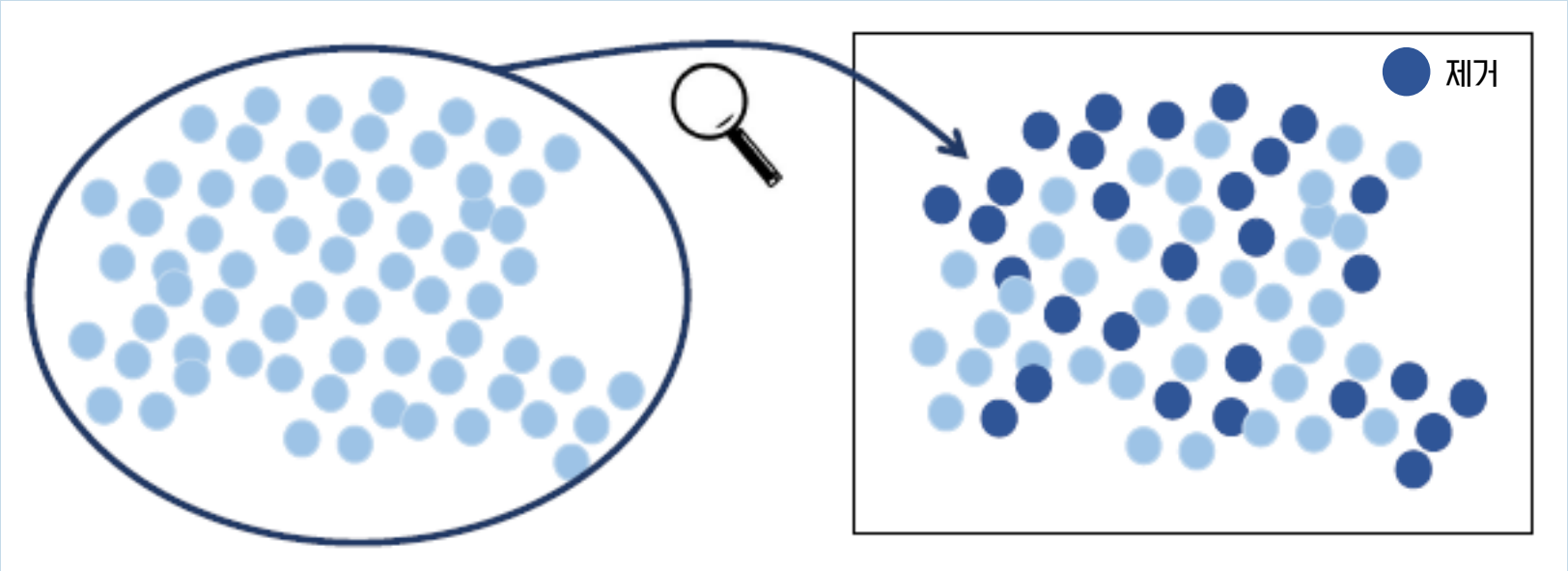
Imbalaced 문제의 해결 방법



Decision Boundary의 치우침 문제 해소

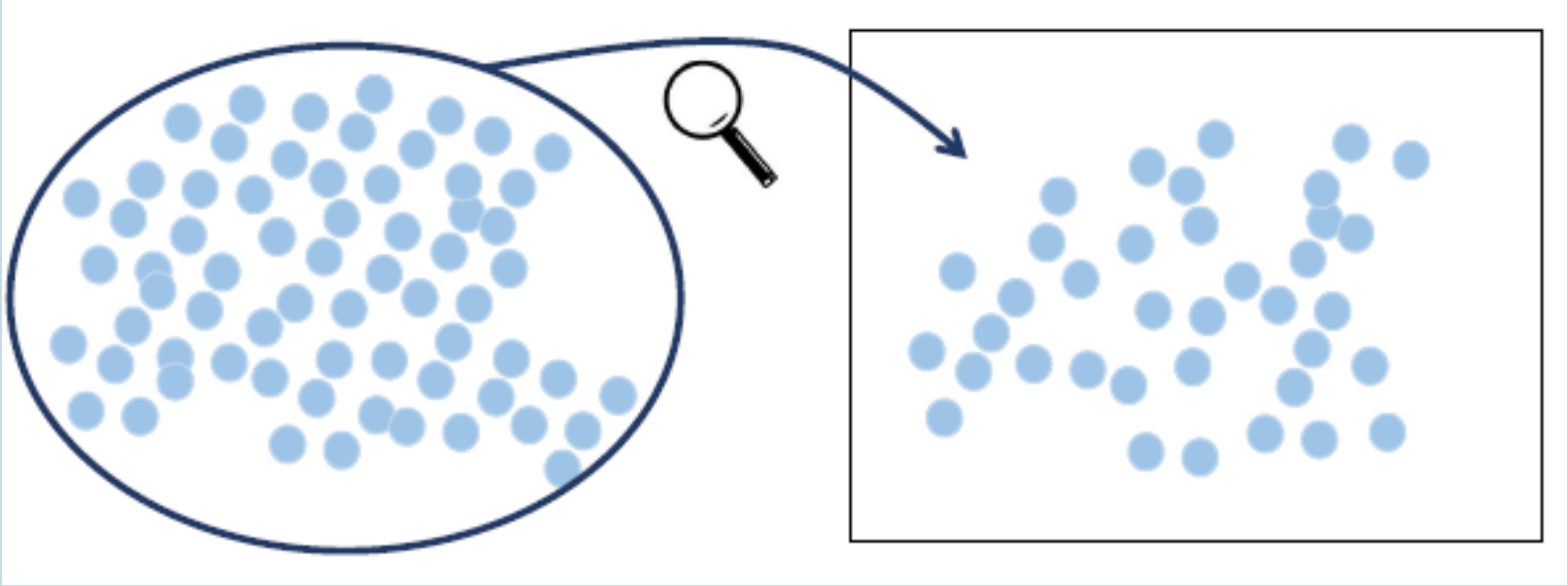
Rus

Majority Data point를 Randomly 제거



Rus

Majority Data point를 Randomly 제거




Modeling

Modeling

차원축소

- Raw data
- PCA
  - 0.9
  - 0.95
  - 0.99
- Kernel PCA
  - RBF
- Manifold
  - Linear
  - LLE
  - t-SNE
- SelectKBest
- XGBoost feature selection
- Random Forest feature selection
- LGBM

Sampling

- 기본
- SMOTE
- RUS

12 X 3 X 8 : 288회 실시

Modeling

- Logistic regression
- Decision Tree
- Random Forest
- Gaussian Naïve Bayesian
- KNN. K-Nearest Neighbor
- SVC, Support Vector Classifier
- Light GBM
- XGBoost

Random Forest

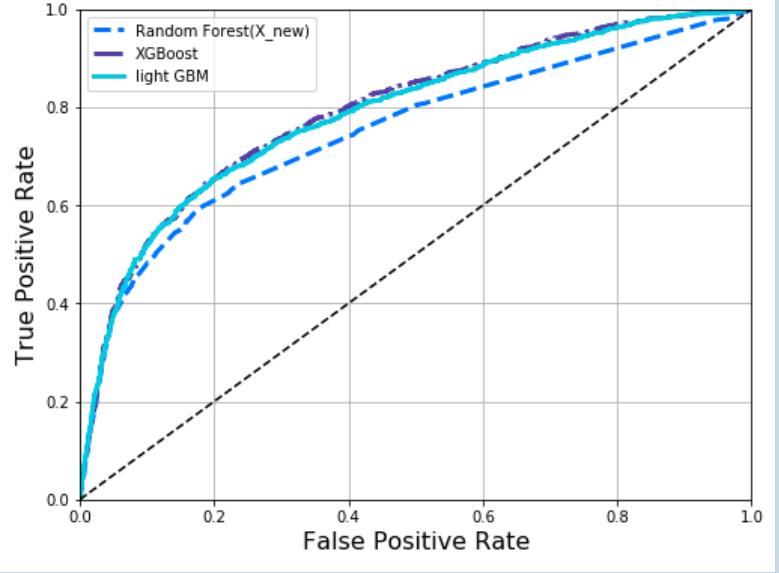
example

차원축소 방법	기본	Kselect	Kernel Linear	Kernel RBF	LGBM	LLE	T-SNE	PCA (0.9)	PCA (0,95)	PCA (0.99)	Random forest	XGBoost
Accuracy	0.813	0.816	0.799	0.793	0.788	0.783	0.777	0.776	0.776	0.777	0.781	0.784
F1-score	0.457	0.456	0.375	0.293	0.281	0.262	0.262	0.257	0.253	0.251	0.269	0.285
auROC	0.65	0.649	0.613	0.587	0.579	0.571	0.568	0.565	0.563	0.562	0.57	0.577

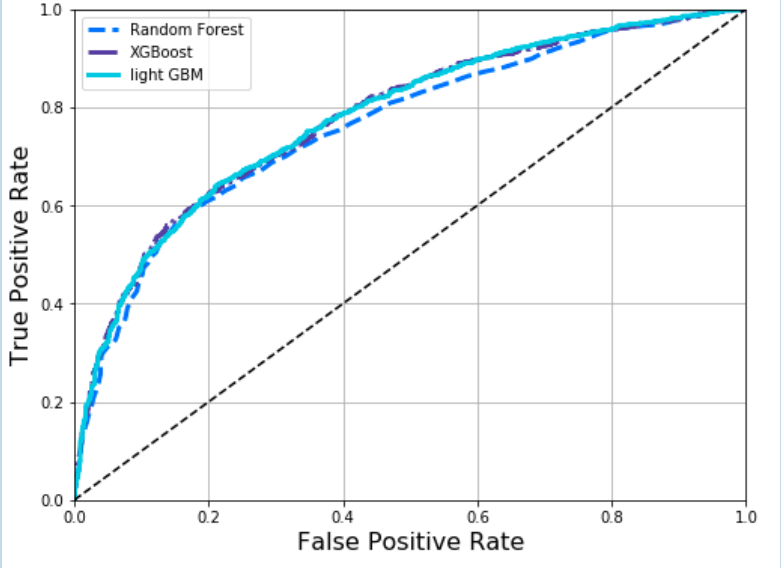
Summary

Sampling	Model	차원축소	Accuracy	F1-score	auROC
Raw data	XGB	Original Data	0.82	0.46	0.65
RUS	XGB	Original Data	0.715	0.696	0.715
SMOTE	LGBM	Original Data	0.862	0.853	0.862

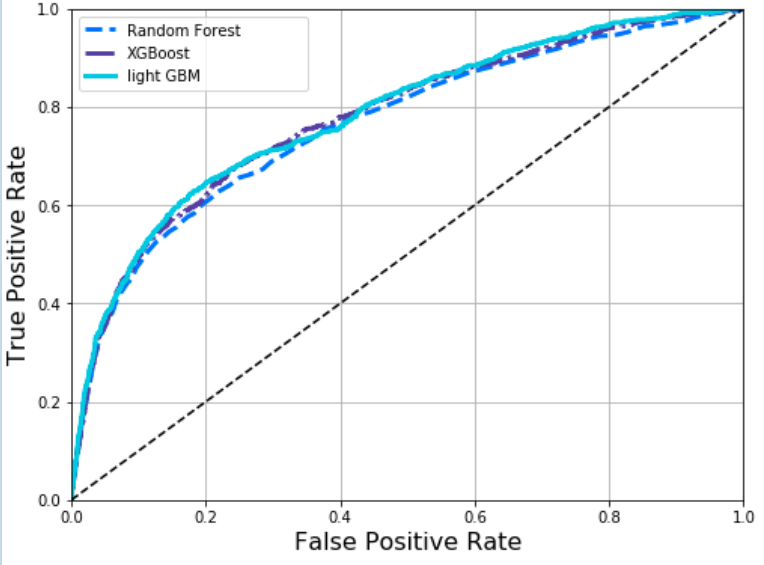
Raw data



RUS



SMOTE



Result

Result

성능 평가

지급불이행에 대한 예측모형 분류

SMOTE – LGBM : 0.862
Raw Data – LGBM : 0.819
RUS – LGBM : 0.71

지급불이행에 대한 예측모형 분류에서는
기본 데이터에 SMOTE – LGBM을 적용하여 분류한 성능이 가장 좋았음

RUS의 경우 Imbalance 문제의 해결보다 데이터의 정보손실이 미치는 영향이 컸음

적절한 차원축소를 하지 않는 경우 외에 모두 score 손실을 가져옴

기본 데이터 : 차원 축소를 하지 않은 data

Raw Data	SMOTE
new_X _SVC	new_X _SVC
0.813	0.7
0.625	0.7
0.398	0.672
0.625	0.7

But,

SMOTE 방법을 사용하는 것이 무조건적인 결과 향상을 가져오지 않음

대다수의 경우 SMOTE로 Oversampling하여 분석한 결과
Accuracy가 하락됨
Imbalanced 문제를 해결하기 때문에
F1 score / auROC는 상승

Result

LR	kernel linear	Kernel RBF	LightGBM	Manifold LLE	Manifold TSNE	PCA (0.9)	PCA (0.95)	PCA (0.99)	Random Forest	XGboost
Accuracy	0.779	0.779	0.779	0.779	0.779	0.779	0.779	0.779	0.783	0.786
Balanced accuracy	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.513	0.527
F1score	0	0.001	0	0	0	0	0	0	0.044	0.085
auROC	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.513	0.527
DT	kernel linear	Kernel RBF	LightGBM	Manifold LLE	Manifold TSNE	PCA (0.9)	PCA (0.95)	PCA (0.99)	Random Forest	XGboost
Accuracy	0.775	0.777	0.777	0.776	0.776	0.776	0.776	0.776	0.781	0.784
Balanced accuracy	0.512	0.509	0.511	0.509	0.512	0.512	0.512	0.513	0.528	0.54
F1score	0.072	0.051	0.06	0.056	0.067	0.068	0.071	0.076	0.119	0.151
auROC	0.512	0.509	0.511	0.509	0.512	0.512	0.512	0.513	0.528	0.54

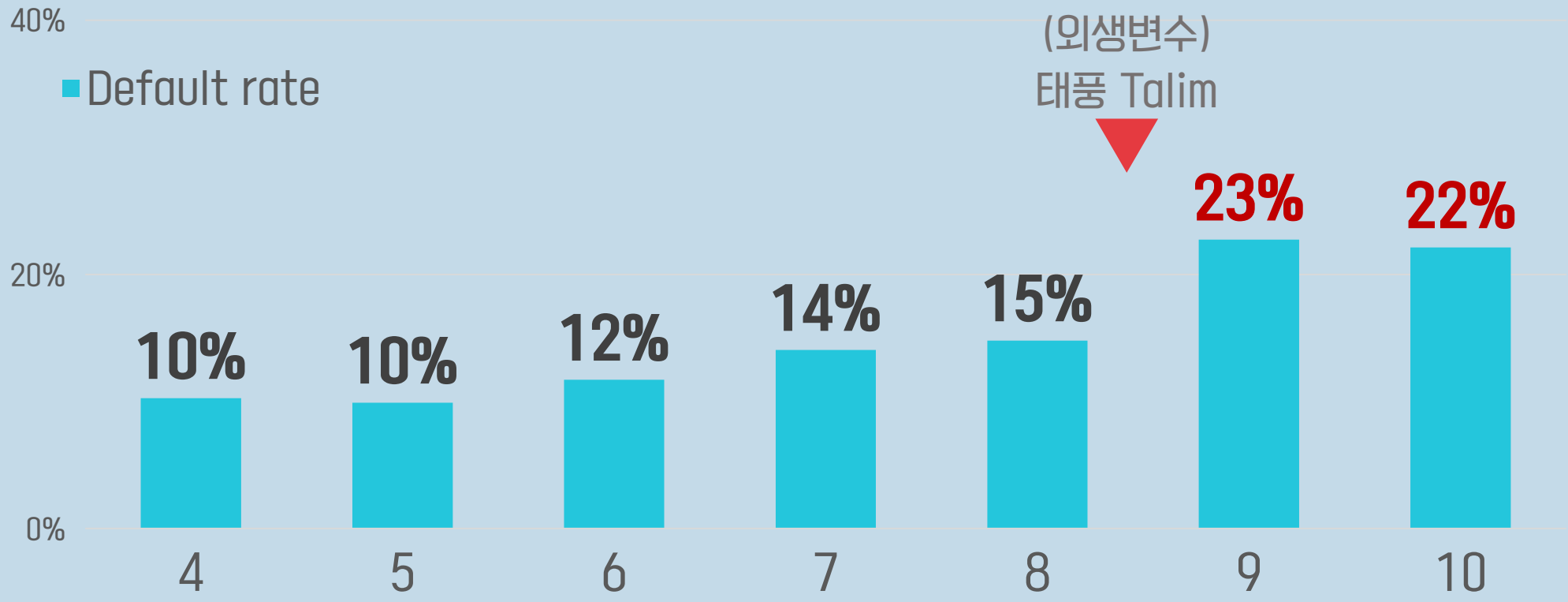
Decision Tree와 LogisticRegression의 경우 특히 F1 score 값이 낮게 나타남.

고차원의 데이터에 대해 분류하는 능력이 현저히 떨어짐.

Boosting 기법이 고차원 데이터에서는 성과가 좋음.

Thinking about

더 잘 예측할 순 없을까?



2005년 8월 태풍 Talim : 62명의 사상자와 약 591억 원의 피해

피해 복구를 위한 비용 ▶ 연체 비율 상승

Over Sampling의 잘못된 예

"Random forest yields the best accuracy that is 91 percent…"

In [243]:
```
from sklearn.utils import resample

# Upsample minority class
df_minority_oversampling = resample(df_minority,
                                    replace=True,     # sample with replacement
                                    n_samples=22677,    # to match majority class
                                    random_state=587) # reproducible results
# Combine majority class with upsampled minority class
df_oversample = pd.concat([df_majority, df_minority_oversampling])
# Display new class counts
print("Now the distribution of non default and default are almost close")
df_oversample['default'].value_counts()
```

```
Now the distribution of non default and default are almost close
```

Out[243]:
```
0.0    23364
1.0    22677
Name: default, dtype: int64
```

Splitting into train and test 80 percent train, 20 percent test

In [244]:
```
#using the new data frame - oversampled dataframe --- oversampling of minority class

X = df_oversample.drop(["default"], axis=1).values #Setting the X to do the split
y = df_oversample["default"].values # transforming the values in array
X_train, X_test, y_train, y_test=train_test_split(X, y, random_state=2, test_size=0.20)
```

Varsha Waingankar, "Predict credit card default of clients in Taiwan

잘못된 실행

기본 데이터 ▶ SMOTE ▶ Train and Test Split 실시 ▶ 91% (X)

올바른 실행

기본 데이터 ▶ Train and Test Split 실시 ▶ Train Set에서 SMOTE 실행 ▶ 77%

Kaggle

Kernel 최고 점수

지급불이행에 대한 예측모형 분류

Tensor flow on Credit Card Clients Data, 82% Acc

Python notebook using data from Default of Credit Card Clients Dataset · 4,260 views · 3y ago

finance

성능이 가장 좋은 모델

기본 데이터에 SMOTE – LGBM을 적용

SMOTE – LGBM : 0.862


Thank you

QnA

대만의 신용카드 채무 불이행 분류 예측

Light GBM

example

1. Raw Data

차원축소 방법	기본	Kselect	Kernel Linear	Kernel RBF	LGBM	LLE	T-SNE	PCA (0.9)	PCA (0,95)	PCA (0.99)	Random forest	XGBoost
Accuracy	0.819	0.819	0.806	0.799	0.795	0.793	0.791	0.789	0.788	0.788	0.791	0.793
Balanced accuracy	0.653	0.651	0.608	0.583	0.571	0.561	0.556	0.551	0.549	0.549	0.558	0.565
F1-score	0.465	0.46	0.342	0.268	0.239	0.207	0.194	0.183	0.177	0.179	0.204	0.225
auROC	0.653	0.651	0.608	0.583	0.571	0.561	0.556	0.551	0.549	0.549	0.558	0.565

Light GBM

example

2. SMOTE

차원축소 방법	기본	Kselect	Kernel Linear	Kernel RBF	LGBM	LLE	T-SNE	PCA (0.9)	PCA (0,95)	PCA (0.99)	Random forest	XGBoost
Accuracy	0.862	0.784	0.746	0.689	0.7	0.692	0.687	0.685	0.685	0.686	0.688	0.689
Balanced accuracy	0.862	0.784	0.746	0.689	0.7	0.692	0.687	0.685	0.685	0.686	0.688	0.689
F1-score	0.853	0.759	0.734	0.58	0.612	0.622	0.628	0.635	0.642	0.648	0.65	0.651
auROC	0.862	0.784	0.746	0.689	0.7	0.692	0.687	0.685	0.685	0.686	0.688	0.689

Light GBM

example

3. RUS

차원축소 방법	기본	Kselect	Kernel Linear	Kernel RBF	LGBM	LLE	T-SNE	PCA (0.9)	PCA (0,95)	PCA (0.99)	Random forest	XGBoost
Accuracy	0.71	0.705	0.675	0.635	0.637	0.628	0.627	0.626	0.626	0.627	0.633	0.639
Balanced accuracy	0.71	0.705	0.675	0.636	0.637	0.629	0.627	0.626	0.626	0.627	0.633	0.639
F1-score	0.695	0.675	0.658	0.6	0.611	0.608	0.609	0.611	0.614	0.616	0.62	0.624
auROC	0.71	0.705	0.675	0.636	0.637	0.629	0.627	0.626	0.626	0.627	0.633	0.639

XGBoost

example

1. Raw Data

차원축소 방법	기본	Kselect	Kernel Linear	Kernel RBF	LGBM	LLE	T-SNE	PCA (0.9)	PCA (0,95)	PCA (0.99)	Random forest	XGBoost
Accuracy	0.82	0.82	0.806	0.8	0.796	0.793	0.791	0.79	0.789	0.788	0.791	0.793
Balanced accuracy	0.651	0.649	0.6	0.576	0.564	0.554	0.547	0.541	0.537	0.535	0.546	0.555
F1-score	0.46	0.455	0.308	0.239	0.206	0.172	0.152	0.134	0.123	0.116	0.148	0.174
auROC	0.651	0.649	0.6	0.576	0.564	0.554	0.547	0.541	0.537	0.535	0.546	0.555

XGBoost

example

2. SMOTE

차원축소 방법	기본	Kselect	Kernel Linear	Kernel RBF	LGBM	LLE	T-SNE	PCA (0.9)	PCA (0,95)	PCA (0.99)	Random forest	XGBoost
Accuracy	0.796	0.751	0.715	0.665	0.669	0.658	0.654	0.652	0.653	0.654	0.658	0.661
Balanced accuracy	0.796	0.751	0.715	0.665	0.669	0.658	0.654	0.652	0.653	0.654	0.658	0.661
F1-score	0.777	0.722	0.702	0.552	0.58	0.588	0.595	0.603	0.611	0.618	0.623	0.627
auROC	0.796	0.751	0.715	0.665	0.669	0.658	0.654	0.652	0.653	0.654	0.658	0.661

Light GBM

example

3. RUS

차원축소 방법	기본	Kselect	Kernel Linear	Kernel RBF	LGBM	LLE	T-SNE	PCA (0.9)	PCA (0,95)	PCA (0.99)	Random forest	XGBoost
Accuracy	0.715	0.709	0.681	0.641	0.641	0.632	0.63	0.629	0.629	0.63	0.637	0.642
Balanced accuracy	0.715	0.709	0.681	0.641	0.642	0.632	0.63	0.629	0.63	0.63	0.637	0.642
F1-score	0.696	0.679	0.664	0.526	0.551	0.561	0.569	0.577	0.585	0.591	0.598	0.603
auROC	0.715	0.709	0.681	0.641	0.642	0.632	0.63	0.629	0.63	0.63	0.637	0.642

Gaussian Naive Bayes

example

1. Raw Data

차원축소 방법	기본	Kselect	Kernel Linear	Kernel RBF	LGBM	LLE	T-SNE	PCA (0.9)	PCA (0,95)	PCA (0.99)	Random forest	XGBoost
Accuracy	0.757	0.779	0.779	0.779	0.778	0.778	0.778	0.778	0.757	0.721	0.729	0.735
Balanced accuracy	0.692	0.677	0.618	0.589	0.572	0.56	0.552	0.545	0.551	0.551	0.564	0.574
F1-score	0.511	0.495	0.33	0.253	0.211	0.176	0.151	0.132	0.161	0.183	0.214	0.239
auROC	0.692	0.677	0.618	0.589	0.572	0.56	0.552	0.545	0.551	0.551	0.564	0.574

Gaussian Naive Bayes

example

2. SMOTE

차원축소 방법	기본	Kselect	Kernel Linear	Kernel RBF	LGBM	LLE	T-SNE	PCA (0.9)	PCA (0,95)	PCA (0.99)	Random forest	XGBoost
Accuracy	0.592	0.629	0.604	0.579	0.575	0.567	0.57	0.568	0.566	0.564	0.576	0.586
Balanced accuracy	0.592	0.629	0.604	0.579	0.575	0.567	0.57	0.568	0.566	0.564	0.576	0.586
F1-score	0.681	0.622	0.635	0.481	0.519	0.542	0.547	0.561	0.573	0.582	0.587	0.589
auROC	0.592	0.629	0.604	0.579	0.575	0.567	0.57	0.568	0.566	0.564	0.576	0.586

Gaussian Naive Bayes

example

3. RUS

차원축소 방법	기본	Kselect	Kernel Linear	Kernel RBF	LGBM	LLE	T-SNE	PCA (0.9)	PCA (0,95)	PCA (0.99)	Random forest	XGBoost
Accuracy	0.556	0.611	0.593	0.571	0.567	0.561	0.566	0.565	0.563	0.561	0.573	0.583
Balanced accuracy	0.556	0.611	0.593	0.571	0.567	0.561	0.566	0.565	0.563	0.561	0.573	0.583
F1-score	0.67	0.618	0.63	0.479	0.515	0.54	0.546	0.56	0.57	0.58	0.585	0.586
auROC	0.556	0.611	0.593	0.571	0.567	0.561	0.566	0.565	0.563	0.561	0.573	0.583

Random Forest

example

1. Raw data

차원축소 방법	기본	Kselect	Kernel Linear	Kernel RBF	LGBM	LLE	T-SNE	PCA (0.9)	PCA (0,95)	PCA (0.99)	Random forest	XGBoost
Accuracy	0.813	0.816	0.799	0.793	0.788	0.783	0.777	0.776	0.776	0.777	0.781	0.784
Balanced accuracy	0.65	0.649	0.613	0.587	0.579	0.571	0.568	0.565	0.563	0.562	0.57	0.577
F1-score	0.457	0.456	0.375	0.293	0.281	0.262	0.262	0.257	0.253	0.251	0.269	0.285
auROC	0.65	0.649	0.613	0.587	0.579	0.571	0.568	0.565	0.563	0.562	0.57	0.577

Random Forest

example

2. SMOTE

차원축소 방법	기본	Kselect	Kernel Linear	Kernel RBF	LGBM	LLE	T-SNE	PCA (0.9)	PCA (0,95)	PCA (0.99)	Random forest	XGBoost
Accuracy	0.85	0.777	0.774	0.71	0.722	0.73	0.729	0.734	0.74	0.747	0.742	0.739
Balanced accuracy	0.85	0.777	0.774	0.71	0.722	0.73	0.729	0.734	0.74	0.747	0.742	0.739
F1-score	0.845	0.754	0.761	0.593	0.628	0.653	0.664	0.677	0.69	0.702	0.699	0.696
auROC	0.85	0.777	0.774	0.71	0.722	0.73	0.729	0.734	0.74	0.747	0.742	0.739

Random Forest

example

3. RUS

차원축소 방법	기본	Kselect	Kernel Linear	Kernel RBF	LGBM	LLE	T-SNE	PCA (0.9)	PCA (0,95)	PCA (0.99)	Random forest	XGBoost
Accuracy	0.706	0.702	0.669	0.63	0.629	0.619	0.613	0.612	0.613	0.615	0.622	0.628
Balanced accuracy	0.706	0.702	0.669	0.63	0.63	0.619	0.613	0.612	0.613	0.615	0.622	0.628
F1-score	0.687	0.67	0.646	0.507	0.53	0.534	0.54	0.548	0.555	0.562	0.571	0.578
auROC	0.706	0.702	0.669	0.63	0.63	0.619	0.613	0.612	0.613	0.615	0.622	0.628

Logistic Regression

example

1. Raw data

차원축소 방법	기본	Kselect	Kernel Linear	Kernel RBF	LGBM	LLE	T-SNE	PCA (0.9)	PCA (0,95)	PCA (0.99)	Random forest	XGBoost
Accuracy	0.809	0808	0.779	0.779	0.779	0.779	0.779	0.779	0.779	0.779	0.783	0.786
Balanced accuracy	0.619	0.613	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.513	0.527
F1-score	0.391	0.377	0	0.001	0	0	0	0	0	0	0.044	0.085
auROC	0.619	0.613	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.513	0.527

Logistic Regression

example

2. SMOTE

차원축소 방법	기본	Kselect	Kernel Linear	Kernel RBF	LGBM	LLE	T-SNE	PCA (0.9)	PCA (0,95)	PCA (0.99)	Random forest	XGBoost
Accuracy	0.699	0.697	0.598	0.55	0.567	0.56	0.566	0.571	0.576	0.579	0.592	0.602
Balanced accuracy	0.699	0.697	0.598	0.55	0.567	0.56	0.566	0.571	0.576	0.579	0.592	0.602
F1-score	0.659	0.654	0.635	0.327	0.43	0.455	0.484	0.509	0.527	0.541	0.551	0.559
auROC	0.699	0.697	0.598	0.55	0.567	0.56	0.566	0.571	0.576	0.579	0.592	0.602

Logistic Regression

example

3. RUS

차원축소 방법	기본	Kselect	Kernel Linear	Kernel RBF	LGBM	LLE	T-SNE	PCA (0.9)	PCA (0,95)	PCA (0.99)	Random forest	XGBoost
Accuracy	0.618	0.656	0.59	0.544	0.564	0.552	0.56	0.565	0.569	0.573.	0.586	0.597
Balanced accuracy	0.618	0.656	0.589	0.546	0.565	0.554	0.562	0.566	0.571	0.574	0.587	0.598
F1-score	0.657	0.653	0.645	0.397	0.477	0.479	0.502	0.526	0.545	0.559	0.567	0.573
auROC	0.618	0.656	0.589	0.546	0.565	0.554	03562	0.566	0.571	0.574	0.587	0.598

Decision Tree

example

1. Raw data

차원축소 방법	기본	Kselect	Kernel Linear	Kernel RBF	LGBM	LLE	T-SNE	PCA (0.9)	PCA (0,95)	PCA (0.99)	Random forest	XGBoost
Accuracy	0.813	0.817	0.775	0.777	0.777	0.776	0.776	0.776	0.776	0.776	0.781	0.784
Balanced accuracy	0.649	0.65	0.512	0.509	0.511	0.509	0.512	0.512	0.512	0.513	0.528	0.54
F1-score	0.455	0.459	0.072	0.051	0.06	0.056	0.067	0.068	0.071	0.076	0.119	0.151
auROC	0.649	0.65	0.512	0.509	0.511	0.509	0.512	0.512	0.512	0.513	0.528	0.54

Decision Tree

example

2. SMOTE

차원축소 방법	기본	Kselect	Kernel Linear	Kernel RBF	LGBM	LLE	T-SNE	PCA (0.9)	PCA (0,95)	PCA (0.99)	Random forest	XGBoost
Accuracy	0.723	0.713	0.638	0.577	0.602	0.602	0.605	0.61	0.615	0.619	0.629	0.636
Balanced accuracy	0.723	0.713	0.638	0.577	0.602	0.602	0.605	0.611	0.615	0.619	0.629	0.636
F1-score	0.699	0.679	0.676	0.386	0.479	0.514	0.536	0.559	0.575	0.588	0.597	0.604
auROC	0.723	0.713	0.638	0.577	0.602	0.602	0.605	0.611	0.615	0.619	0.629	0.636

Decision Tree

example

3. RUS

차원축소 방법	기본	Kselect	Kernel Linear	Kernel RBF	LGBM	LLE	T-SNE	PCA (0.9)	PCA (0,95)	PCA (0.99)	Random forest	XGBoost
Accuracy	0.693	0.697	0.614	0.565	0.588	0.586	0.587	0.592	0.595	0.596	0.607	0.616
Balanced accuracy	0.693	0.697	0.614	0.565	0.588	0.586	0.587	0.592	0.595	0.596	0.607	0.617
F1-score	0.668	0.663	0.639	0.368	0.46	0.491	0.511	0.533	0.546	0.555	0.567	0.576
auROC	0.693	0.697	0.614	0.565	0.588	0.586	0.587	0.592	0.595	0.596	0.607	0.617

KNN

example

1. Raw data

차원축소 방법	기본	Kselect	Kernel Linear	Kernel RBF	LGBM	LLE	T-SNE	PCA (0.9)	PCA (0,95)	PCA (0.99)	Random forest	XGBoost
Accuracy	0.79	0.794	0.779	0.77	0.764	0.76	0.759	0.758	0.757	0.757	0.761	0.765
Balanced accuracy	0.63	0.63	0.604	0.581	0.57	0.564	0.563	0.561	0.56	0.559	0.567	0.573
F1-score	0.42	0.419	0.366	0.308	0.286	0.275	0.273	0.271	0.27	0.269	0.283	0.296
auROC	0.63	0.63	0.604	0.581	0.57	0.564	0.563	0.561	0.56	0.559	0.567	0.573

KNN

example

2. SMOTE

차원축소 방법	기본	Kselect	Kernel Linear	Kernel RBF	LGBM	LLE	T-SNE	PCA (0.9)	PCA (0,95)	PCA (0.99)	Random forest	XGBoost
Accuracy	0.776	0.729	0.729	0.676	0.684	0.69	0.694	0.698	0.702	0.706	0.704	0.703
Balanced accuracy	0.776	0.729	0.729	0.676	0.684	0.69	0.694	0.698	0.702	0.706	0.704	0.703
F1-score	0.798	0.705	0.722	0.629	0.652	0.668	0.678	0.687	0.696	0.703	0.694	0.689
auROC	0.776	0.729	0.729	0.676	0.684	0.69	0.694	0.698	0.702	0.706	0.704	0.703

KNN

example

3. RUS

차원축소 방법	기본	Kselect	Kernel Linear	Kernel RBF	LGBM	LLE	T-SNE	PCA (0.9)	PCA (0,95)	PCA (0.99)	Random forest	XGBoost
Accuracy	0.594	0.641	0.626	0.596	0.594	0.59	0.59	0.591	0.59	0.591	0.6	0.607
Balanced accuracy	0.594	0.642	0.626	0.596	0.594	0.59	0.59	0.59	0.59	0.591	0.6	0.607
F1-score	0.6	0.614	0.61	0.578	0.581	0.581	0.582	0.585	0.586	0.587	0.591	0.595
auROC	0.594	0.642	0.626	0.596	0.594	0.59	0.59	0.59	0.59	0.591	0.6	0.607

SVM

example

1. Raw data

차원축소 방법	기본	Kselect	Kernel Linear	Kernel RBF	LGBM	LLE	T-SNE	PCA (0.9)	PCA (0,95)	PCA (0.99)	Random forest	XGBoost
Accuracy	0.806	0.813	0.801	0.796	0.793	0.79	0.789	0.787	0.787	0.786	0.789	0.791
Balanced accuracy	0.594	0.625	0.583	0.563	0.55	0.542	0.536	0.531	0.528	0.525	0.536	0.545
F1-score	0.329	0.398	0.266	0.201	0.16	0.134	0.115	0.1	0.089	0.08	0.114	0.142
auROC	0.594	0.625	0.583	0.563	0.55	0.542	0.536	0.531	0.528	0.525	0.536	0.545

SVM

example

2. SMOTE

차원축소 방법	기본	Kselect	Kernel Linear	Kernel RBF	LGBM	LLE	T-SNE	PCA (0.9)	PCA (0,95)	PCA (0.99)	Random forest	XGBoost
Accuracy	0.699	0.7	0.668	0.626	0.622	0.603	0.603	0.603	0.604	0.605	0.613	0.621
Balanced accuracy	0.699	0.7	0.668	0.626	0.622	0.603	0.603	0.603	0.604	0.605	0.613	0.621
F1-score	0.661	0.672	0.657	0.496	0.524	0.548	0.561	0.569	0.576	0.583	0.59	0.597
auROC	0.699	0.7	0.668	0.626	0.622	0.603	0.603	0.603	0.604	0.605	0.613	0.621

SVM

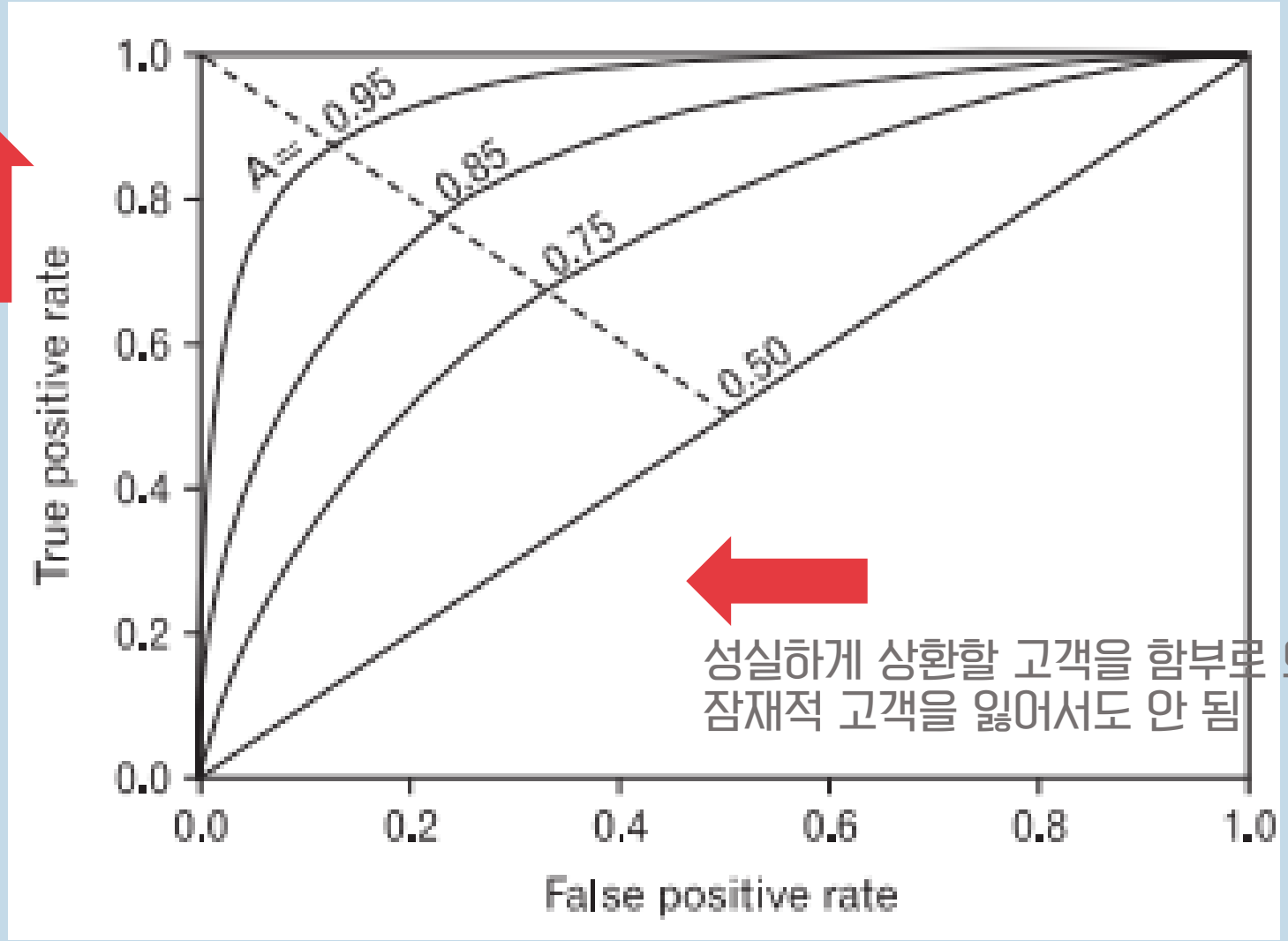
example

3. RUS

차원축소 방법	기본	Kselect	Kernel Linear	Kernel RBF	LGBM	LLE	T-SNE	PCA (0.9)	PCA (0,95)	PCA (0.99)	Random forest	XGBoost
Accuracy	0.513	0.606	0.575	0.556	0.548	0.539	0.543	0.539	0.537	0.535	0.549	0.562
Balanced accuracy	0.513	0.606	0.576	0.557	0.548	0.54	0.544	0.54	0.538	0.535	0.55	0.562
F1-score	0.076	0.364	0.273	0.241	0.213	0.244	0.292	0.267	0.247	0.231	0.271	0.303
auROC	0.513	0.606	0.576	0.557	0.548	0.54	0.544	0.54	0.538	0.535	0.55	0.562

AUC score를 보아야 하는 이유

연체할 사람을 가능한
많이 잡아내는 동시에



성실하게 상환할 고객을 함부로 의심해서
잠재적 고객을 잃어서도 안 됨

Kernel PCA

SVM에서의 커널기법처럼 원래의 d차원 데이터를 k차원으로 매핑하는 하나의함수를 생각할 수 있다. 이 때,커널을 활용하여 데이터를 고차원으로 변환하는 비선형 매핑을 수행한 뒤, 일반적인 PCA를 통해 데이터들이 선형 분리가 가능한 저차원 공간으로 다시 투영하는 아이디어가 커널 PCA의 기본적인 생각

이전 PCA에서 피처간의 관계를 공분산 행렬로 표현하였는데 공분산은 피처간의 내적을 통해 표현되며, 두 데이터의 유사도를 의미하였다.

두 벡터 사이를 내적하는 부분의 함수를, 데이터를 고차원으로 매핑하는 커널 함수로 대체하면 두 데이터의 상관관계를 우리가 정한 Kernel function으로써 나타내는 것이 가능하고, 유사도라는 개념을 더욱 범용적이고 고차원적인 방법으로 사용이 가능해진다.

벡터의 내적을 이용해 상관관계를 구하면 그 관계의 모양은 linear한 선형적 의미를 가지지만 커널을 사용한다면, 그 커널과 맞는 선형적 의미를 가지게 된다. (가우시안 커널을 사용한다면 방사형의 관계를 가짐)

다시말해, 기존 pca와 달리 linear projection이 nonlinear한 projection으로 바뀌게 되어 우리가 분류하고자 하는 데이터셋은 커널의 성질을 띤 채로 non-linear하게 구분이 된다.

고차원으로 매핑하는 함수로는 (RBF, Linear, Sigmoid) 함수가 있으며 각 데이터의 성질에 따라 효율이 달라진다

Manifold LLE

LLE는 비선형 차원 축소(NonLinear Dimensionality Reduction, NLDR) 기법으로 '차원 PCA와 달리 투영(projection)이 아닌 매니폴드 학습(manifold learning) 이다.

LLE는 머신러닝 알고리즘 중 Unsupervised Learning에 해당하며, 서로 인접한 데이터들을 보존하면서 고차원인 데이터셋을 저차원으로 축소하는 방법이다. 즉, LLE는 입력 데이터셋을 낮은 차원의 단일 글로벌 좌표계(single global coordinate system)으로 매핑하는 알고리즘이다

Step 1: Select Neighbors

Step 2: Reconstruct with linear weights

Step 3: Map to Embedded Coordinates

자세한 수학식은(https://excelsior-cjh.tistory.com/168) 참조