Fraud Detection

SOCAR

CONTENTS

본 데이터는 쏘카 측의 제공된 데이터로 일부 데이터에 대한 정보 공개가 불가하여 일부 수정되었음

“

계기

”

뒷쿵

ABOUT

보험금을 목적으로 **렌터카**를 이용한
고의로 인한 차량 사고

PURPOSE

사고 데이터
모델링

DETAILS

프로젝트 개요

· 쏘카의 사고 데이터 중 클래스(0 : 정상사고 , 1: 사기사고)를 머신러닝을 통해 테스트셋의 클래스를 **분류 예측**

· 사기 데이터의 경우, 예측 결과가 의사결정에 중대한 영향을 미칠 것으로 판단하여 성능의 지표는 **재현율**과 **정확도에** 우선

활용 솔루션

1. 클래스의 분포가 과도하게 **불균형**하여 **샘플링을 활용**하여 문제 해결 시도
2. 모델의 성능을 향상시키기 위한 다양한 **데이터 전처리** 진행
3. 데이터의 노이즈를 줄일 수 있는 **차원축소** 기법을 사용

역할

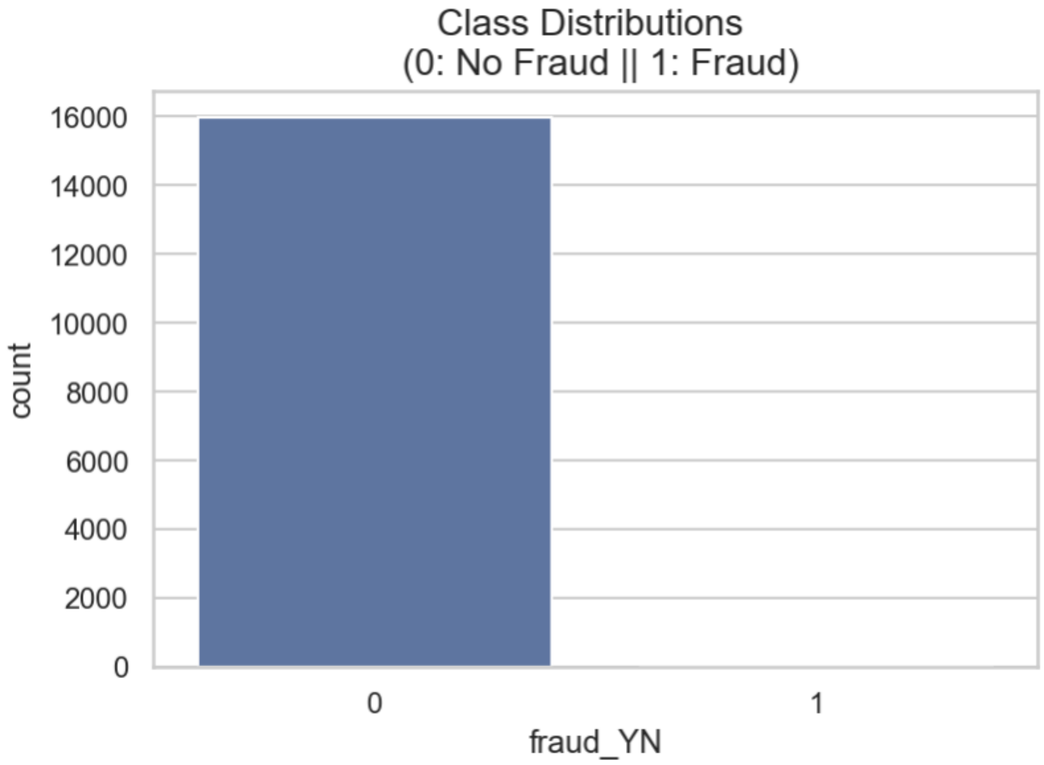
1. 다양한 샘플링 기법에 대한 활용을 위한 샘플링 적용
2. 데이터 전처리 중 KNN과 imputer 기능 활용
3. 하이퍼파라미터 튜닝과 모델 성능 활용에 대한 코드 작업

“EDA”

EDA

```
# 사기 데이터 비율
frauds_rate = round(raw_data["fraud_YN"].value_counts()[1]/len(raw_data)*100,2)
print("Frauds rate :", frauds_rate, "%")
```

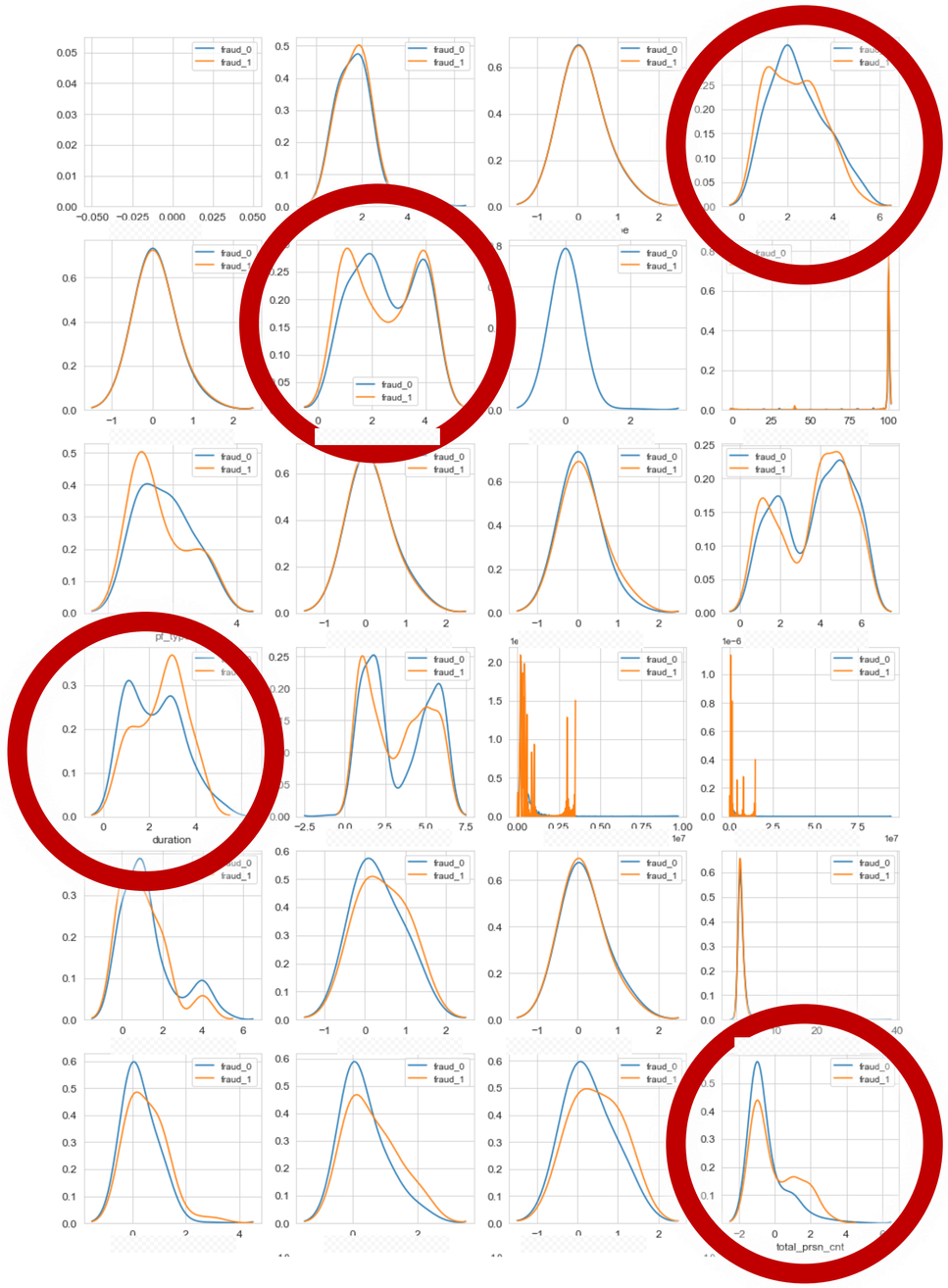
```
Frauds rate : 0.26 %
```



매우 **불균형**한
데이터 분포

전체 16000건 중
Fraud 데이터는
단, 41건

EDA

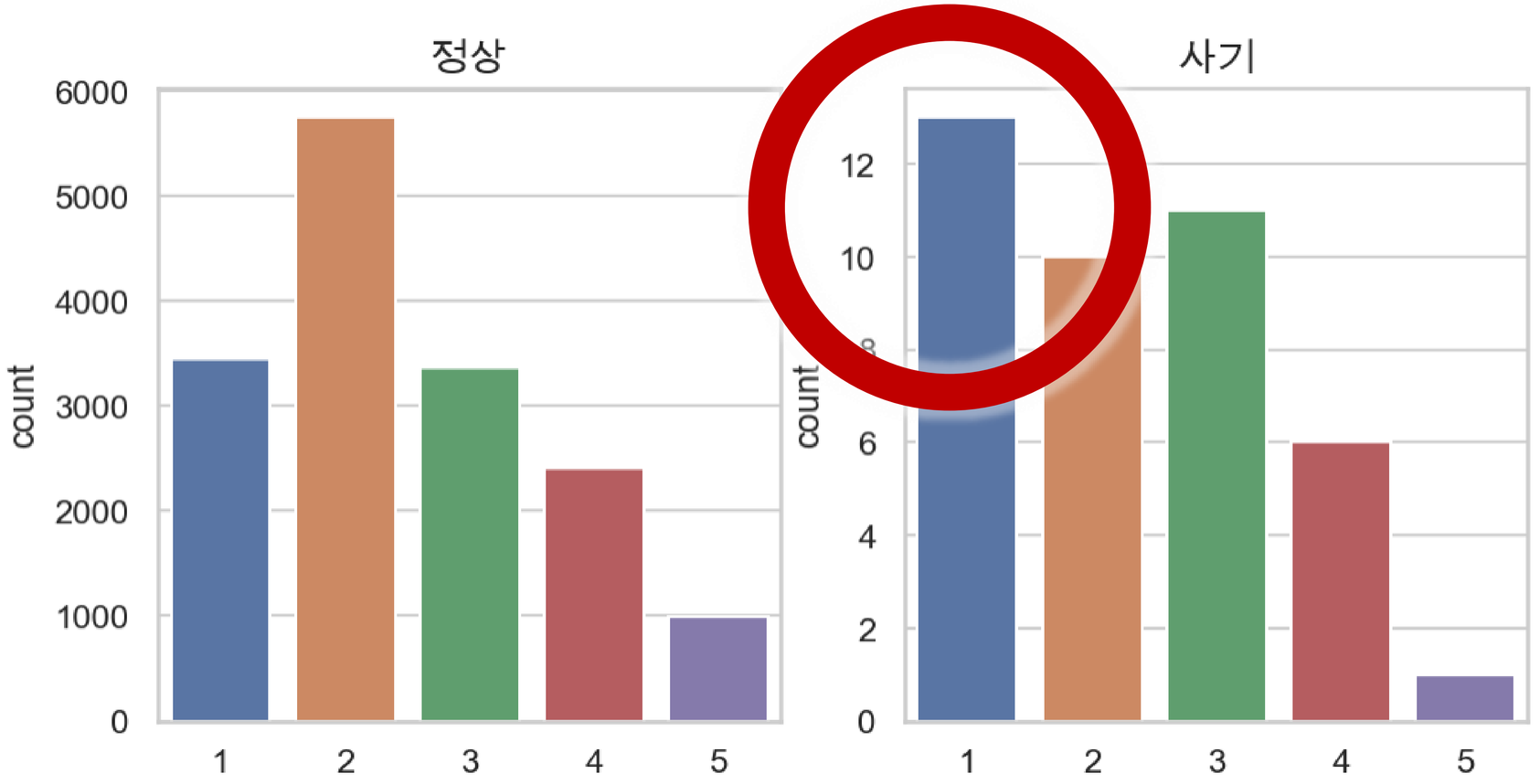


특정 컬럼에 대한
다른 분포의 모습을 확인

Fraud 유형에 대한
포커스에 맞추어
EDA 진행

EDA

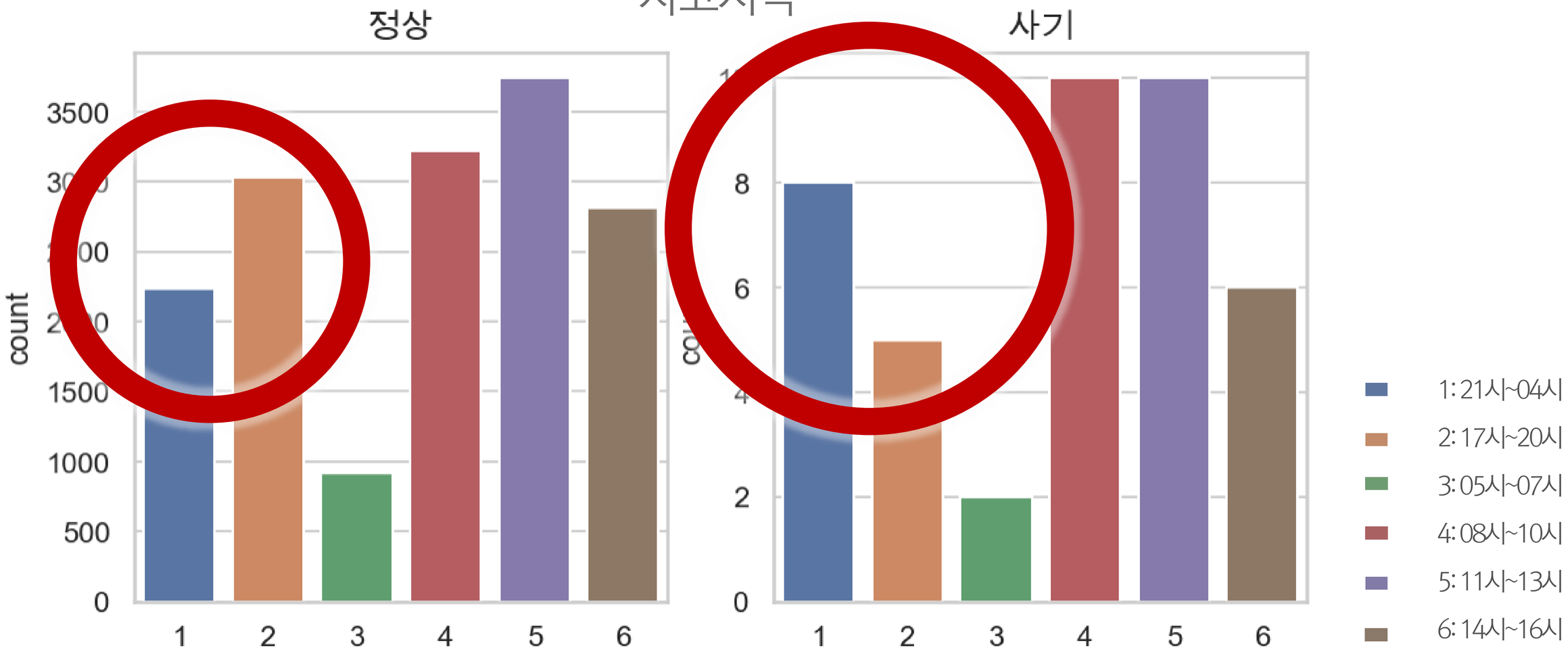
연령대



20대 위주의 사고 비중이 높지만
유독 **20대 초반**의 Fraud 비중이 높음

EDA

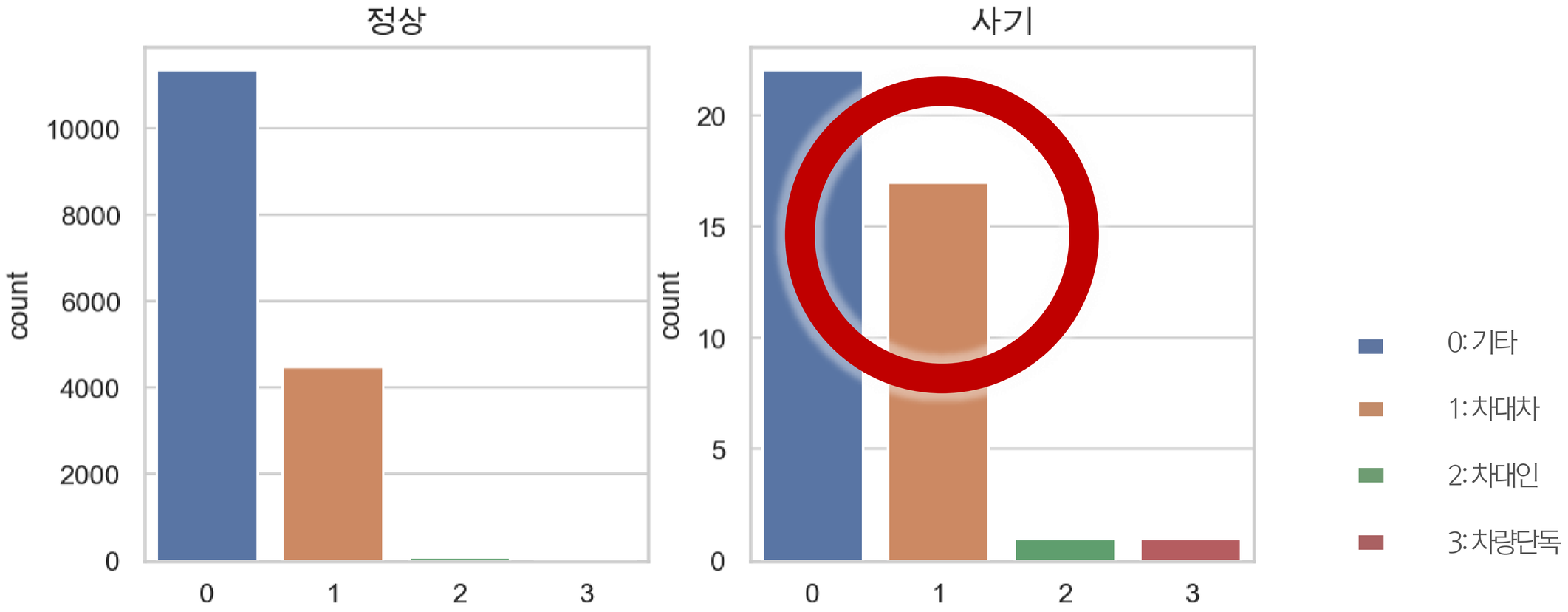
사고시각



일반 사고 사건 대비
저녁 시간대의 사고 비중이 높아짐

EDA

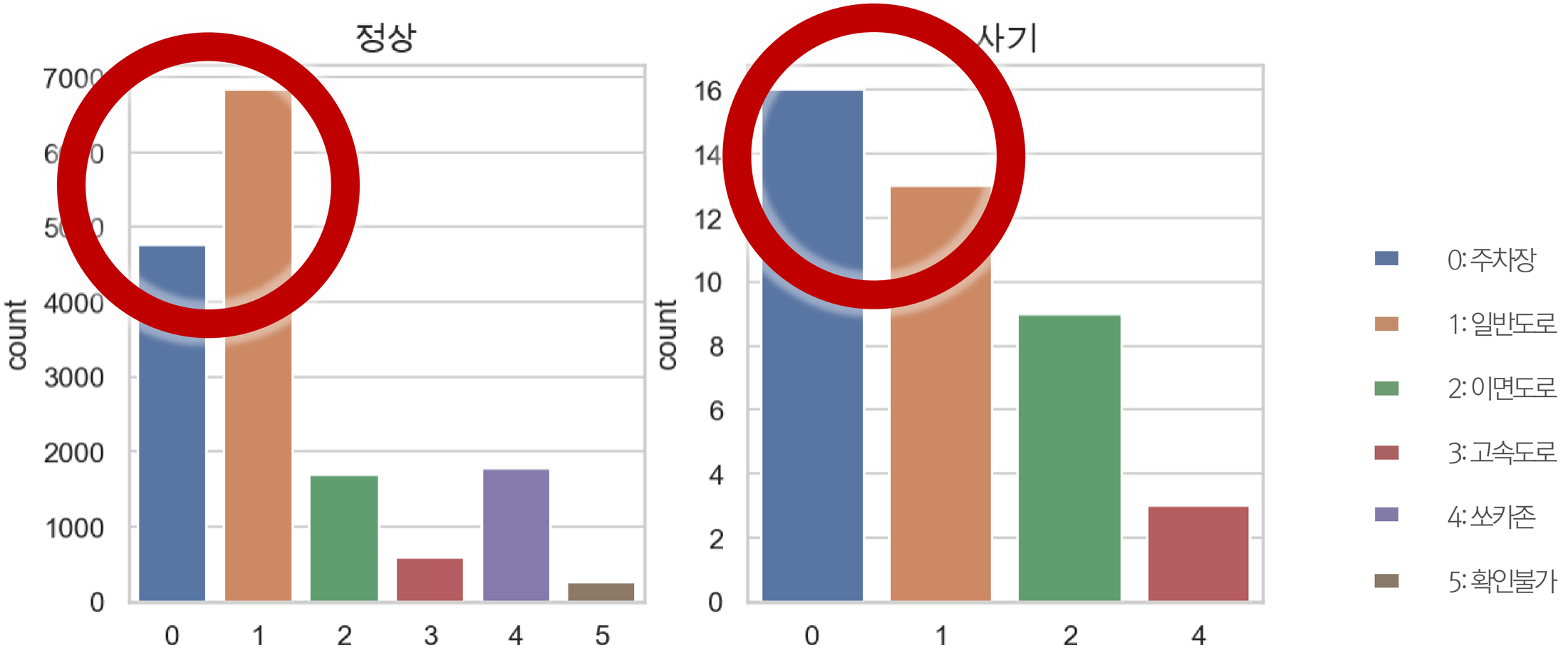
사고유형



일반 사고 사건 대비
차대차 사고 비중이 높아짐

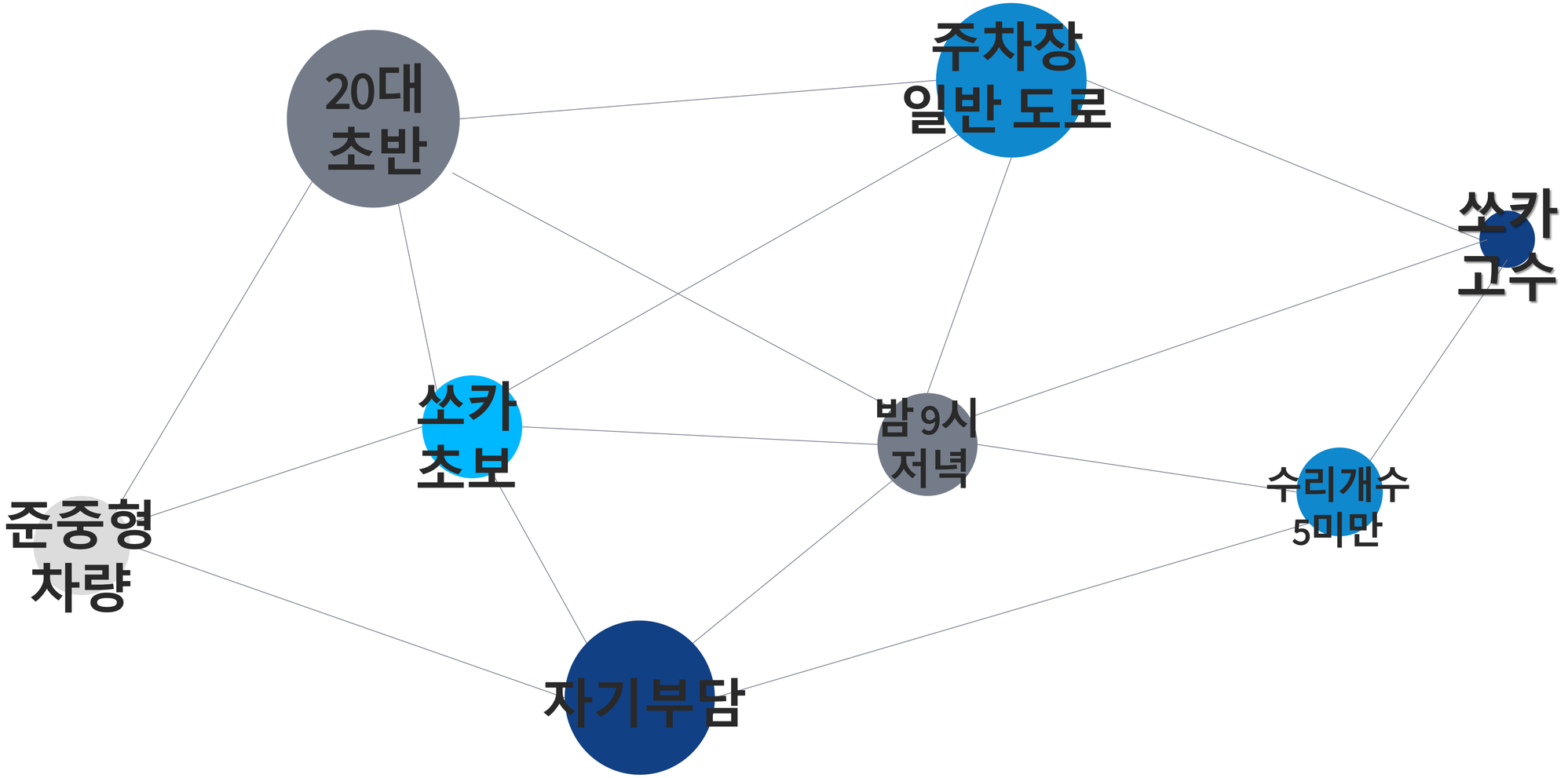
EDA

사고장소



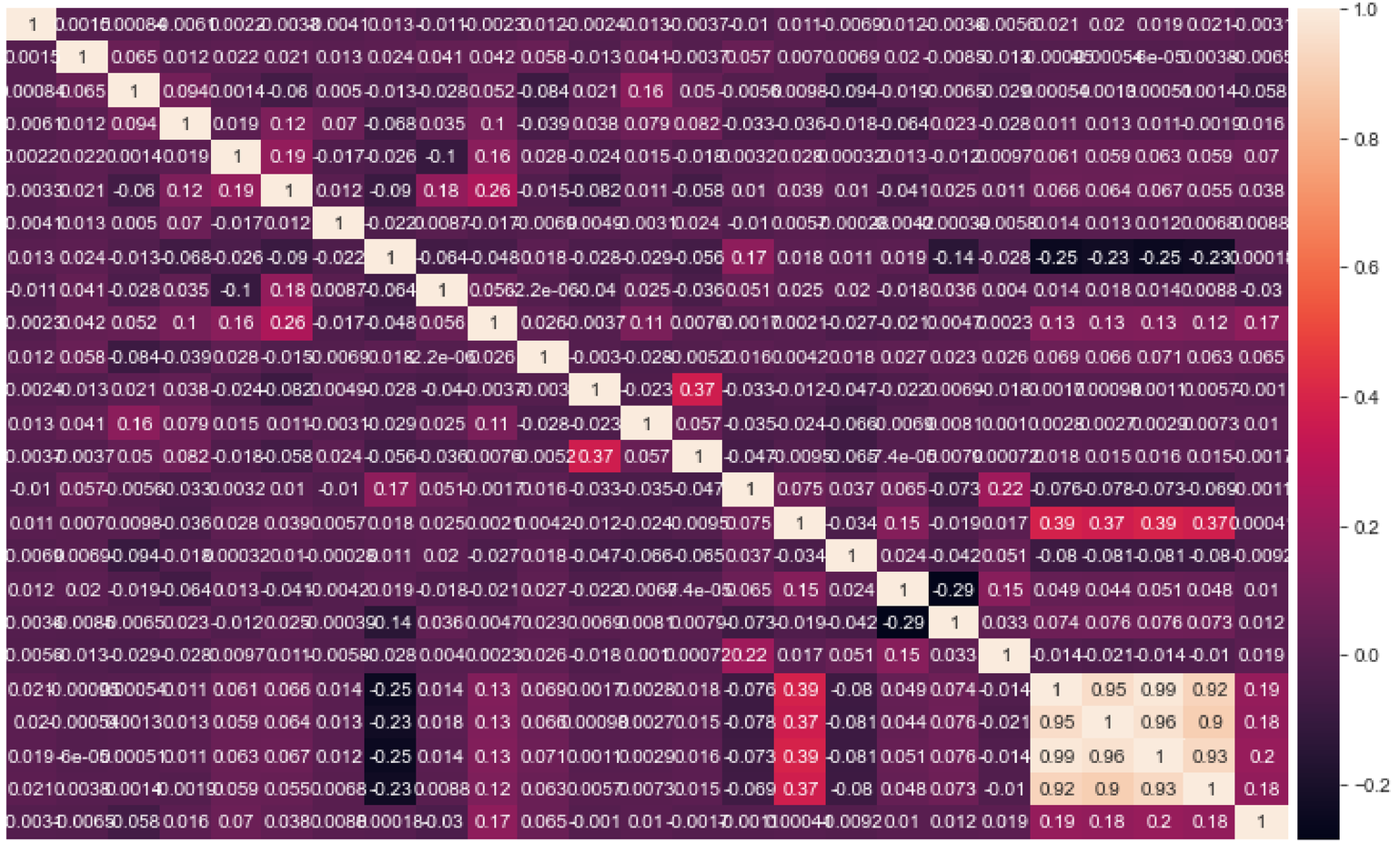
도로 및 주차장에서의 사고 비중이 높음

EDA



Fraud 유형의 전반적인 사고 패턴

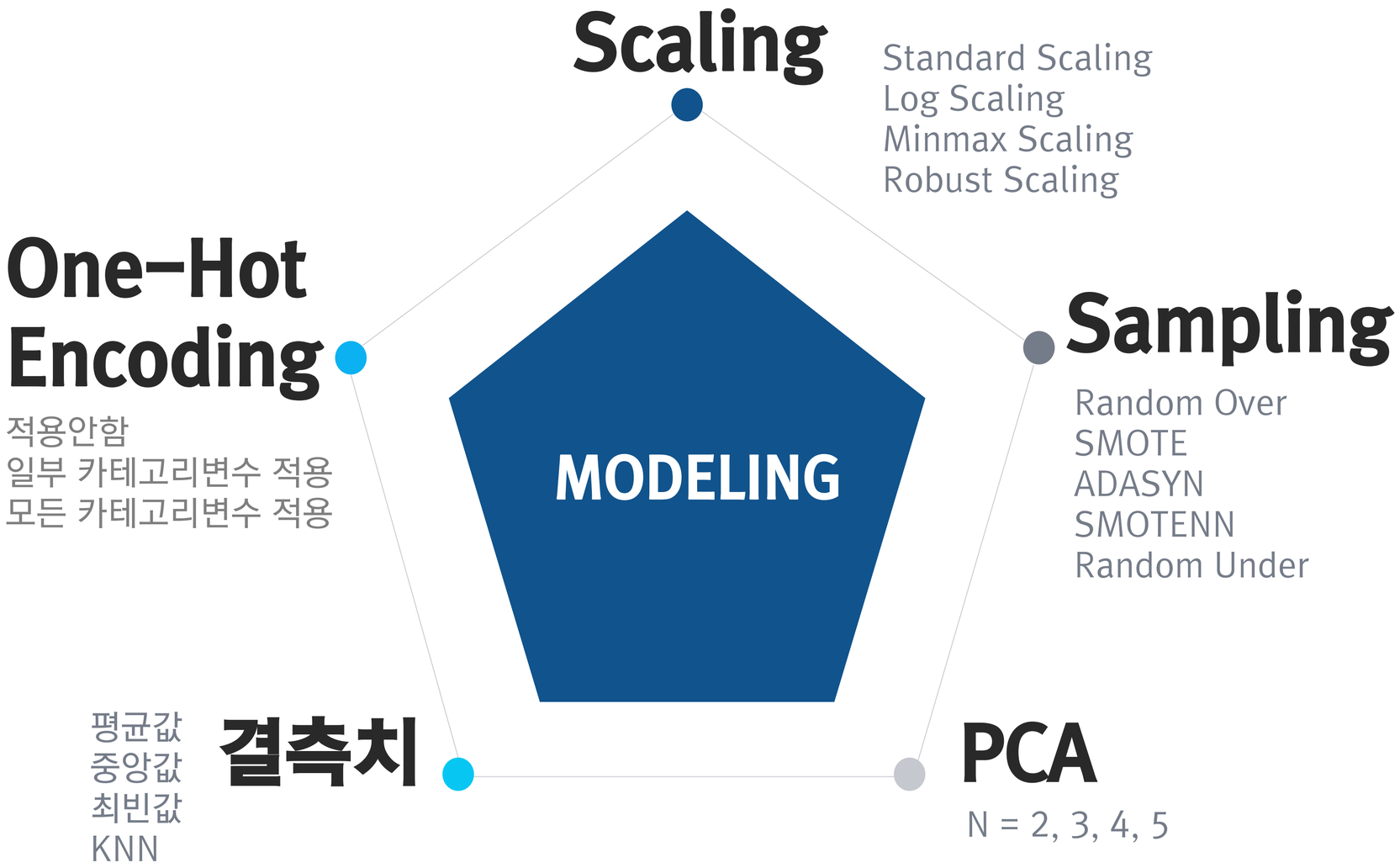
EDA



컬럼 간의
상관관계가 크게
두드러지지 않음

“문제 해결”

다양한 데이터 가공


Scaling
Standard Scaling
Log Scaling
Minmax Scaling
Robust Scaling
One-Hot Encoding
적용안함
일부 카테고리변수 적용
모든 카테고리변수 적용
MODELING
Sampling
Random Over
SMOTE
ADASYN
SMOTENN
Random Under
결측치
평균값
중앙값
최빈값
KNN
PCA
N = 2, 3, 4, 5

결측치 처리

```
socar.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16000 entries, 0 to 15999
Data columns (total 25 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   f                       16000 non-null  int64
 1   c                       16000 non-null  int64
 2   s                       16000 non-null  int64
 3   a                       16000 non-null  int64
 4   h                       16000 non-null  int64
 5   c                       16000 non-null  int64
 6   b                       16000 non-null  int64
 7   a                       16000 non-null  int64
 8   p                       16000 non-null  int64
 9   s                       16000 non-null  int64
 10  s                       16000 non-null  int64
 11  s                       16000 non-null  int64
 12  d                       16000 non-null  int64
 13  a                       16000 non-null  int64
 14  r                       16000 non-null  float64
 15  i                       16000 non-null  int64
 16  a                       16000 non-null  int64
 17  c                       16000 non-null  int64
 18  c                       16000 non-null  int64
```

데이터 상 Nan 데이터는
확인되지 아니함

세부적인 데이터의
skimming 작업을 진행

결측치 처리

지정값	설명	필드
ca		1 : 2: ' 3: ' 4: ' 5: '
sl		0 : 1 :
ag		1: 2 2: 2 3: 2 4: 3 5: 4
ha	유무	
cu	횟수	0: 0 1: 1 2: 2 3: 6 4: >
bi		0: 7 1: 붙 2: 붙
ac		
pl		1: 지 2: 지 3: 지
sc		0: 업 1: 있
sc	로	0: 일 1: 소
st	시간	1 : 2 2 : 1 3 : 4 : 5 : 6 :

지정값	설명	필드
du		-1: null 1: 2 2: 6 3: 1 4: 3 5: 0
ac		1 : 2 2 : 1 3 : 5 4 : 8 5 : 1 6 : 1 **-1 : 설명없음**
re		**0 : null**
ins		**0 : null**
ac		0 : 주차장 1 : 일반도로 2 : 이면도로 3 : 고속도로 4 : 쏘카존 **5 : 확인불가**
ca		0 : 앞 1 : 앞
ca		0 : 뒤 1 : 뒤
re		0~1
ac		0: 7 1: 차 2: 차 3: 차 4: 차대사진기
ins	로	**0: 알수없음** 1: 신고 2: 미신고
po		**0: 알수없음** 1: 신고 2: 미신고

결측치를 대체한 **데이터의 불분명한 값** 발견

결측치 처리

accident_hour

- 사고시각 중 '-1 : 알수없음' 에 해당하는 값을 결측치로 정의함
- 범주형 데이터로 **최빈값**을 통해 결측치를 처리

repair_cost

- 차량 수리 비용 '0 : null' 에 해당하는 값을 결측치로 정의함
- 대부분의 데이터의 값이 0에 해당하여 **평균값, 중앙값** 등 1개의 값으로 대체하는 것 외,
  사고 차량/사고 부위 개수 등 영향이 있을 것으로 판단하여 **KNN imputer** 를 사용하여 보완하기로 함

insure_cost

- 보험 비용 '0 : null' 에 해당하는 값을 결측치로 정의함
- 대부분의 데이터의 값이 0에 해당하여 **평균값, 중앙값** 등 1개의 값으로 대체하는 것 외,
  사고 차량/사고 부위 개수 등 영향이 있을 것으로 판단하여 **KNN imputer** 를 사용하여 보완하기로 함

accident_location

- 사고 장소 '5 : 확인불가' 데이터 값에 대해 결측치로 정의함
- 범주형 데이터로 **최빈값**을 통해 결측치를 처리

insurance_site_aid_YN

- 보험사 출동 유무 중 '0 : 알수없음' 에 해당하는 값을 결측치로 정의함
- 범주형 데이터로 **최빈값**을 통해 결측치를 처리

police_site_aid_YN

- 경찰 출동 유무 중 '0 : 알수없음' 에 해당하는 값을 결측치로 정의함
- 범주형 데이터로 **최빈값**을 통해 결측치를 처리

PCA

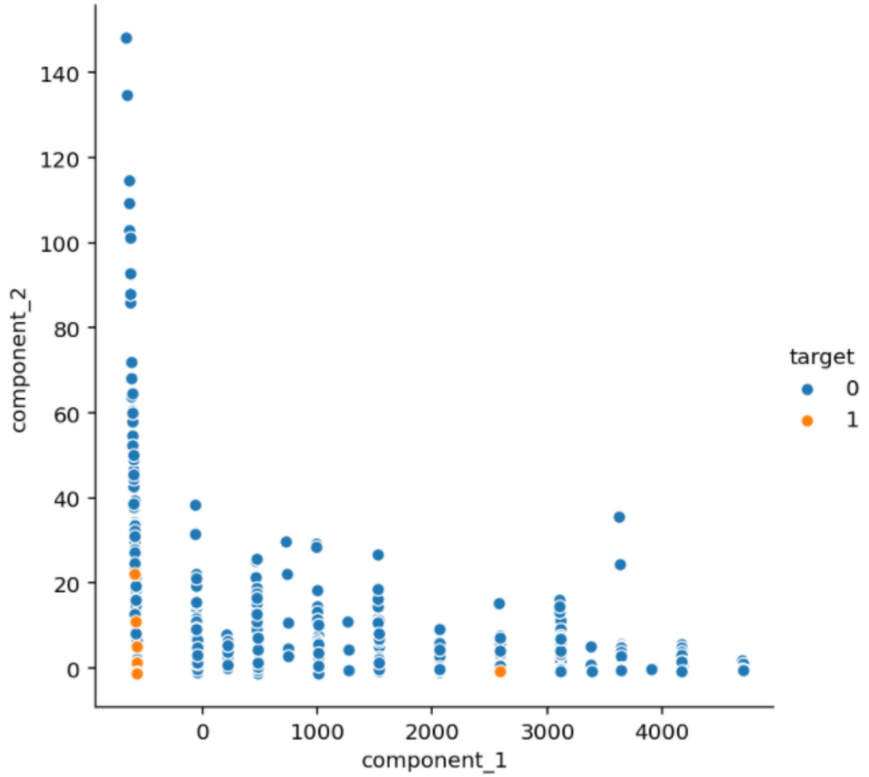
데이터 값 중 '알수없음', '확인불가' 데이터 노이즈로 판단

PCA 차원축소를 통해 노이즈 영향 제거

	component_1	component_2	target
12874	-574.604341	-1.387728	0
12875	-574.375247	-1.476862	0
12876	-574.788079	-1.382804	0
12877	-574.392880	-1.481290	0
12878	-575.759410	0.437704	0

```
1 print (np.sum(pca.explained_variance_ratio_))
```

```
0.9999986587449938
```



PCA

데이터 값 중 ‘알수없음’, ‘확인불가’ 데이터 노이즈로 판단

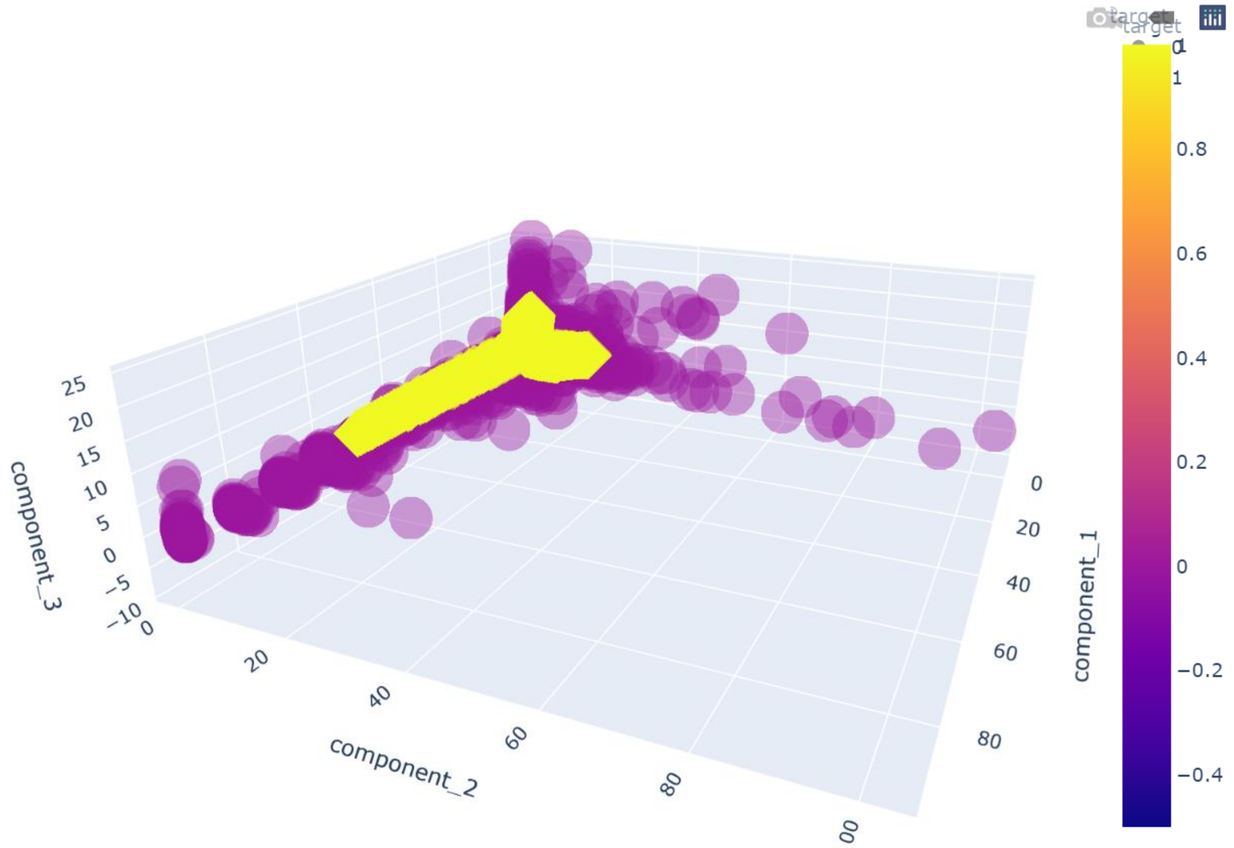
PCA 차원축소를 통해 노이즈 영향 제거

	component_1	component_2	component_3	target
0	-576.356894	-1.492348	-0.873226	0
1	-589.670395	20.317932	-0.240312	0
2	4701.702936	-0.558775	-0.198595	0
3	-576.925943	-0.810789	-0.186182	1
4	-577.163608	-1.484178	-0.757452	0
...	...	...	...	...
10298	-578.796602	3.305350	-1.274367	0
10299	-576.862631	-1.466437	-0.254853	0
10300	-576.985163	-1.459670	-0.260107	0
10301	-577.039303	-1.477947	-0.771680	0
10302	-577.013970	-1.472519	-0.676101	0

10303 rows × 4 columns

```
1 print (np.sum(pca.explained_variance_ratio_))
```

0.9999994952197623



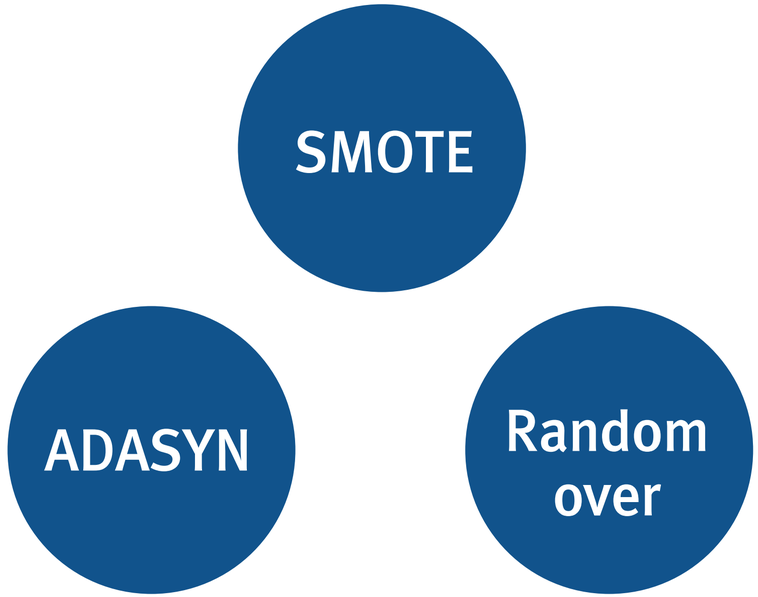
샘플링 – 언더샘플링

Random under

	Accuracy	precision	recall	f1	roc
LogiReg	0.803183	0.005929	0.428571	0.011696	0.616388
DecisionTree	0.568711	0.005376	0.857143	0.010686	0.712534
RandomFore	0.559006	0.005259	0.857143	0.010453	0.707668
LGBM	0.571817	0.004521	0.714286	0.008985	0.642857
SVC	0.817547	0.006397	0.428571	0.012605	0.623589

클래스 간의 오버랩을 방지하지만 **데이터의 유실**이 발생
본 데이터의 데이터 양이 많지 않아 지양하기로 함

샘플링 – 오버샘플링



SMOTE

	Accuracy	precision	recall	f1	roc
LogiReg	0.685559	0.004932	0.571429	0.009780	0.628649
DecisionTree	0.604425	0.002947	0.428571	0.005854	0.516738
RandomFore	0.974379	0.000000	0.000000	0.000000	0.488517
LGBM	0.974767	0.000000	0.000000	0.000000	0.488712
SVC	0.721661	0.005571	0.571429	0.011034	0.646750

ADASYN

LogiReg	0.687888	0.004969	0.571429	0.009852	0.629817
DecisionTree	0.668478	0.003517	0.428571	0.006977	0.548852
RandomFore	0.975155	0.000000	0.000000	0.000000	0.488906
LGBM	0.977096	0.000000	0.000000	0.000000	0.489879
SVC	0.688276	0.004975	0.571429	0.009864	0.630012

Random over

LogiReg	0.140916	0.002705	0.857143	0.005393	0.498054
DecisionTree	0.736025	0.004418	0.428571	0.008746	0.582717
RandomFore	0.997283	0.000000	0.000000	0.000000	0.500000
LGBM	0.997283	0.000000	0.000000	0.000000	0.500000
SVC	0.139752	0.002701	0.857143	0.005386	0.497470

데이터의 유실이 발생하지 않지만
클래스 간의 **오버랩**되거나 **과적합의 발생** 가능성
시간이 오래 걸림

샘플링 – 복합샘플링

SMOTE
-ENN

SMOTE
-Tomek

SMOTE
-ENN

	Accuracy	precision	recall	f1	roc
LogiReg	0.67314	0.00238	0.28571	0.00473	0.47995
DecisionTree	0.83929	0.00244	0.14286	0.00481	0.49202
RandomFore	0.99379	0.00000	0.00000	0.00000	0.49825
LGBM	0.98913	0.00000	0.00000	0.00000	0.49591

SMOTE
-Tomek

	Accuracy	precision	recall	f1	roc
LogiReg	0.80551	0.00202	0.14286	0.00398	0.47509
DecisionTree	0.82415	0.00000	0.00000	0.00000	0.41320
RandomFore	0.99495	0.00000	0.00000	0.00000	0.49883
LGBM	0.99301	0.00000	0.00000	0.00000	0.49786

언더 샘플링과 오버샘플링의 결합 방식

스케일링

Standard Scailing

아웃라이어의 존재의 경우,
민감하게 반응

Minmax Scailing

아웃라이어의 존재의 경우,
민감하게 반응

Log Scailing

큰 이상치에 대해 민감할 수 있음

Robust Scailing

이상치에 대한 영향을 적게 받음

모델링

Logistic Regression

Decision Tree

Random Forest

Light GBM

SVC

다양한 모델 사용을 사용하여
적합한 모델링 선정

모델링

#116	MinMax	삭제안함	적용안함	처리 안함	-	2	SMOTE	LogiReg	0.4142080745	0.002649006623	0.5714285714	0.005273566249
		동일분포	적용함	accident_hour / a	최빈값	3	ADASYN	DecisionTree	0.3559782609	0.003008423586	0.7142857143	0.005991611744
		경찰/보험출동유무				4	random under	RandomFore	0.9906832298	0	0	0
						5	SMOTENN	LGBM	0.9072204969	0.004273504274	0.1428571429	0.008298755187
							랜덤오버샘플링	SVC	0.4114906832	0.002636783125	0.5714285714	0.005249343832
#117	MinMax	삭제안함	적용안함	처리 안함	-	2	SMOTE	LogiReg	0.5135869565	0.002396166134	0.4285714286	0.004765687053
		동일분포	적용함	accident_hour / a	최빈값	3	ADASYN	DecisionTree	0.514363354	0.001602564103	0.2857142857	0.003187250996
		경찰/보험출동유무				4	random under	RandomFore	0.9930124224	0	0	0
						5	SMOTENN	LGBM	0.9250776398	0	0	0
							랜덤오버샘플링	SVC	0.5093167702	0.002375296912	0.4285714286	0.004724409449
#118	MinMax	삭제안함	적용안함	처리 안함	-	2	SMOTE	LogiReg	0.4596273292	0.003584229391	0.7142857143	0.007132667618
		동일분포	적용함	accident_hour / a	최빈값	3	ADASYN	DecisionTree	0.5163043478	0.004003202562	0.7142857143	0.007961783439
		경찰/보험출동유무				4	random under	RandomFore	0.9972826087	0	0	0
						5	SMOTENN	LGBM	0.9961180124	0	0	0
							랜덤오버샘플링	SVC	0.4611801242	0.003594536305	0.7142857143	0.007153075823
#119	MinMax	삭제안함	적용안함	처리 안함	-	2	SMOTE	LogiReg	0.4526397516	0.003538570418	0.7142857143	0.007042253521
		동일분포	적용함	accident_hour / a	최빈값	3	ADASYN	DecisionTree	0.3788819876	0.003738317757	0.8571428571	0.007444168734
		경찰/보험출동유무				4	random under	RandomFore	0.9972826087	0	0	0
						5	SMOTENN	LGBM	0.9968944099	0	0	0
							랜덤오버샘플링	SVC	0.4561335404	0.003561253561	0.7142857143	0.007087172218
#120	MinMax	삭제안함	적용안함	처리 안함	-	2	SMOTE	LogiReg	0.3940217391	0.002560819462	0.5714285714	0.005098789038
		동일분포	적용함	accident_hour / a	최빈값	3	ADASYN	DecisionTree	0.4371118012	0.00412371134	0.8571428571	0.008207934337
		경찰/보험출동유무				4	random under	RandomFore	0.9972826087	0	0	0
						5	SMOTENN	LGBM	0.9965062112	0	0	0
							랜덤오버샘플링	SVC	0.3947981366	0.002564102564	0.5714285714	0.005105296745

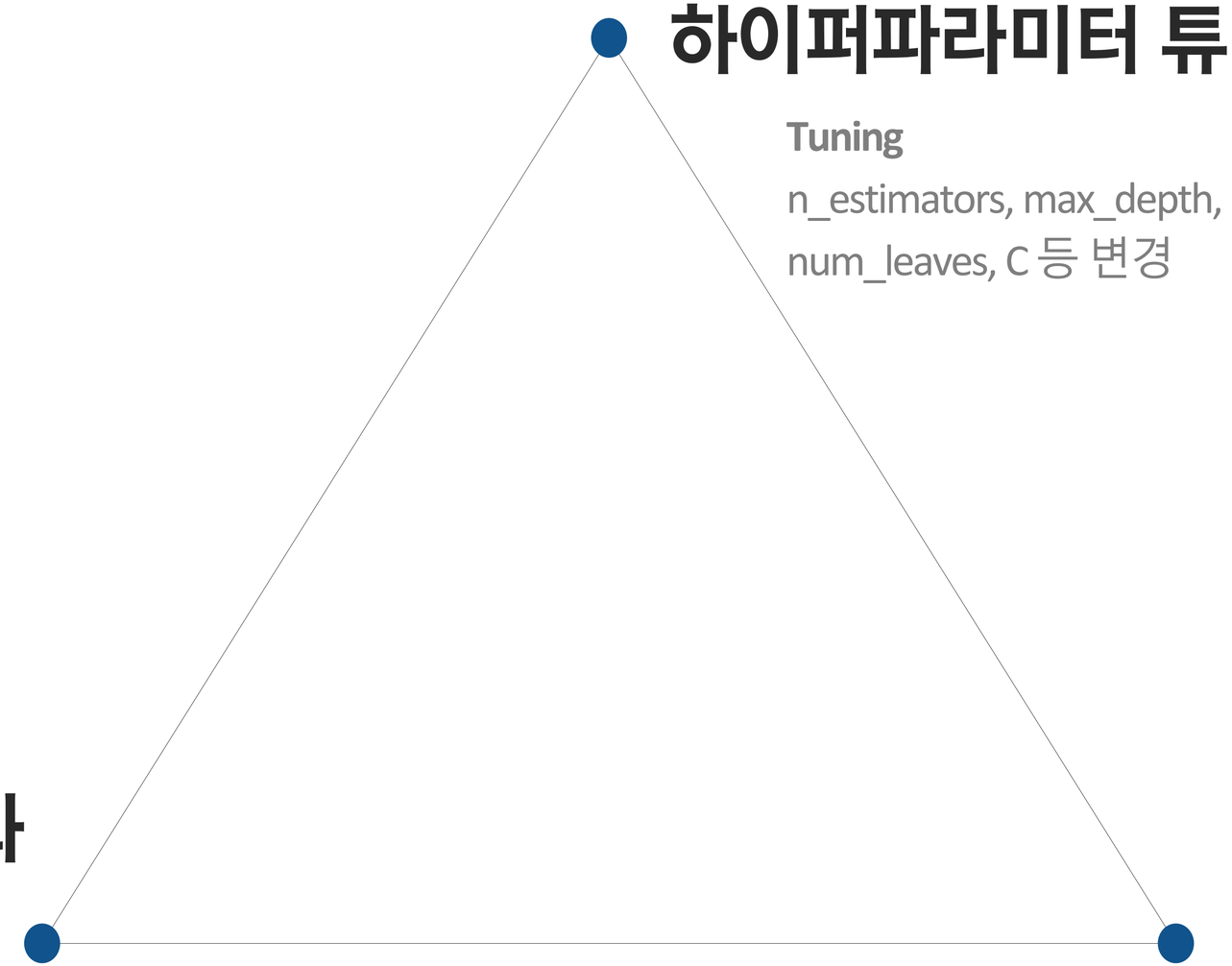
수 많은 알고리즘 시도와 다양한 전처리 시도

성능 평가

불필요 변수 삭제 → Standard Scaling → PCA (n=4) → ADASYN

	Accuracy	Precision	Recall	f1	ROC
LogiReg	0.716227	0.006812	0.714286	0.013495	0.715259
DecisionTree	0.701087	0.001307	0.142857	0.002591	0.422733
Random Fore	0.970109	0	0	0	0.486376
LGBM	0.968556	0	0	0	0.485598
SVC	**0.719332**	**0.006887**	**0.714286**	**0.013643**	**0.716816**

성능 개선



하이퍼파라미터 튜닝

Tuning

n_estimators, max_depth,
num_leaves, C 등 변경

결측치처리 다양화

평균값, 중앙값, KNN

StratifiedKFold

cv=3, cv=5

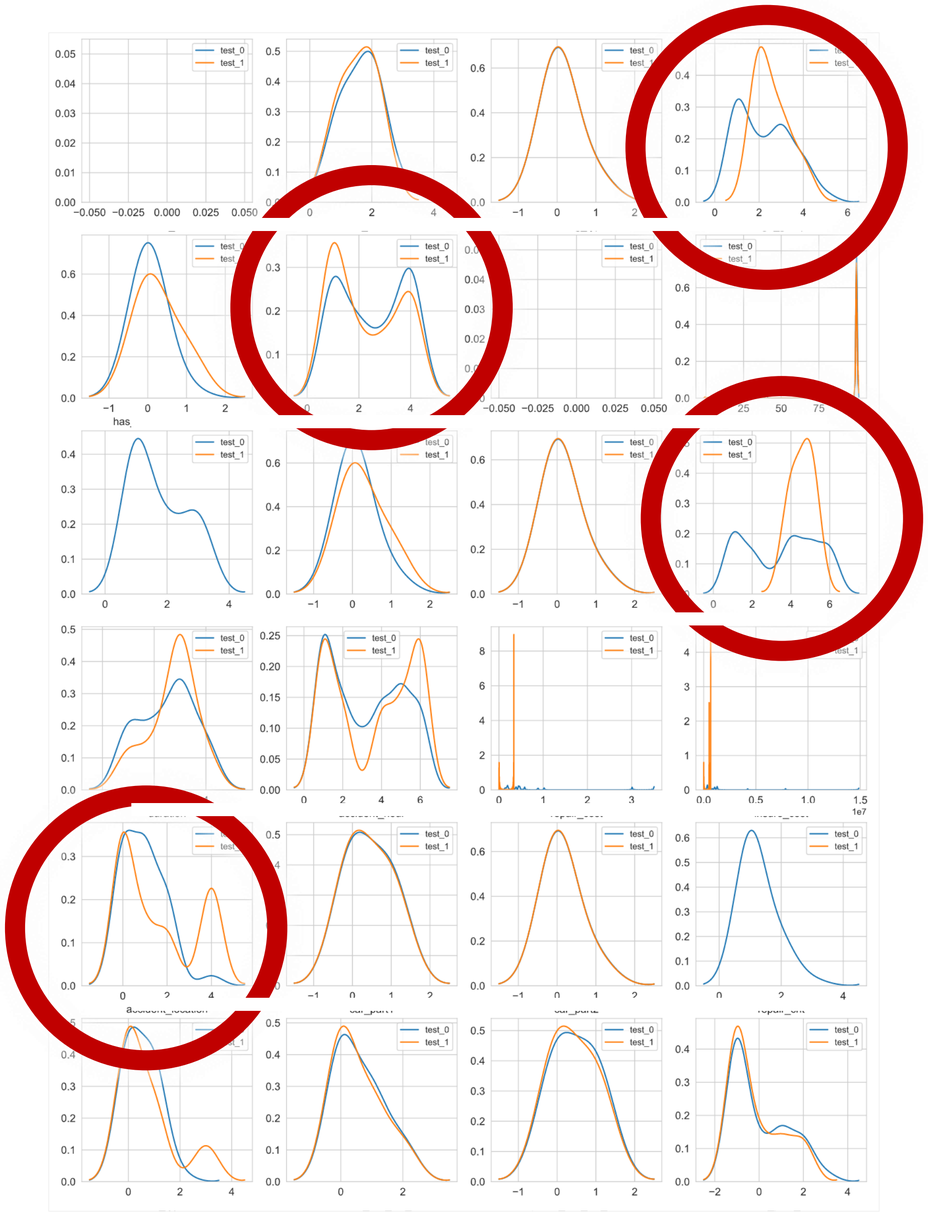
Cross Validation

성능 평가

불필요 변수 삭제 → 특정 컬럼 카테고리 축소 → One-Hot Encoding → Randoma OverSampling → 하이퍼 파라미터 변경

	Valid					Test				
	Accuracy	Precision	Recall	f1	ROC	Accuracy	Precision	Recall	f1	ROC
LogiReg	**0.733696**	**0.007257**	**0.714286**	**0.014368**	**0.724017**	**0.507850**	**0.002602**	**0.571429**	**0.005181**	**0.539568**
DecisionTree	0.932453	0.000000	0.000000	0.000000	0.467497	0.882409	0.002762	0.142857	0.005420	0.513465
Random Fore	0.996894	0.000000	0.000000	0.000000	0.499805	0.997116	0.000000	0.000000	0.000000	0.499679
LGBM	0.997283	0.000000	0.000000	0.000000	0.500000	0.997116	0.000000	0.000000	0.000000	0.499679
SVC	0.739130	0.005944	0.571429	0.011765	0.655508	0.509452	0.002611	0.571429	0.005198	0.540371

Test data에 대한 고민



- 다양한 알고리즘 모델링의 Test-set 성능 저하 결과
- 과적합이 아닌 Train-Test 의 데이터 **분포가 크게 상이**

Review

- **실무 데이터**를 통한
  문제 해결에 대해
  좀 더 고민을 해보는 기회

- **불균형 데이터**에 대한
  다양한 해결 방법

https://github.com/Yenabeam/dss_prjt_socar