

2020-2 기계학습 프로젝트

신용카드 사기거래 탐지

: Anomaly Detection with Extremely Imbalanced Data

B511140 이송희 B611121 오현지

# 목차

---

## 1. 프로젝트 개요

- 사용된 모델

## 2. 데이터 분석 및 전처리

## 3. Under Sampling & Over Sampling

- Under Sampling
- 이상치 제거
- Raw Data vs Under Sampling vs Over Sampling

## 4. 하이브리드 모델

- 앙상블
- 하이브리드 모델

## 5. 실험 결과

## 6. 결론

# 프로젝트 개요

## Motivation

- 현실에서는 데이터의 구성이 불균형한 경우가 많음
- Anomaly Detection의 많은 경우 정상 데이터의 비율이 비정상적인 데이터에 비해 매우 큼
- 불균형 데이터를 효과적으로 학습할 방법 모색



최적의 인풋 전략과 모델 제시

## Experiment

- 1 Raw Data, Under Sampling Data, Over Sampling Data를 정의한 후, 각 데이터를 이용해 기본 모델로 학습
- 2 실험1의 결과에 따라 사용할 Dataset과 모델을 선별하여 하이브리드 모델 구현 후 다양한 조합으로 성능 평가

## 사용된 모델

### Logistic Regression

종속변수가 범주형 데이터일 때, 독립변수의 선형결합을 이용해 데이터가 어떤 범주에 속할 확률을 예측

### RandomForest

훈련 과정에서 구성한 다수의 결정 트리로부터 다수결 또는 평균을 계산하여 예측치를 출력

### SVM

서포트 벡터를 사용해서 결정 경계를 정의하고, 분류되지 않은 점을 해당 결정 경계와 비교해서 분류

### KNN

k개의 최근접 이웃 사이에서 다른 데이터의 레이블을 참조하여 가장 공통적인 분류에 할당

### AdaBoost

이전 단계에서 잘못 분류된 데이터에 가중치를 주어 다음 단계에 연속적으로 적용하는 방식으로 분류

### ExtraTrees

결정 트리의 각 후보 특성을 무작위로 분할하는 방식으로 랜덤포레스트의 무작위성을 증가시킨 분류기

### AutoEncoder

입력데이터에 데이터의 노이즈를 추가하거나 제거한 후 복원하여 출력하는 네트워크

## 데이터 분석

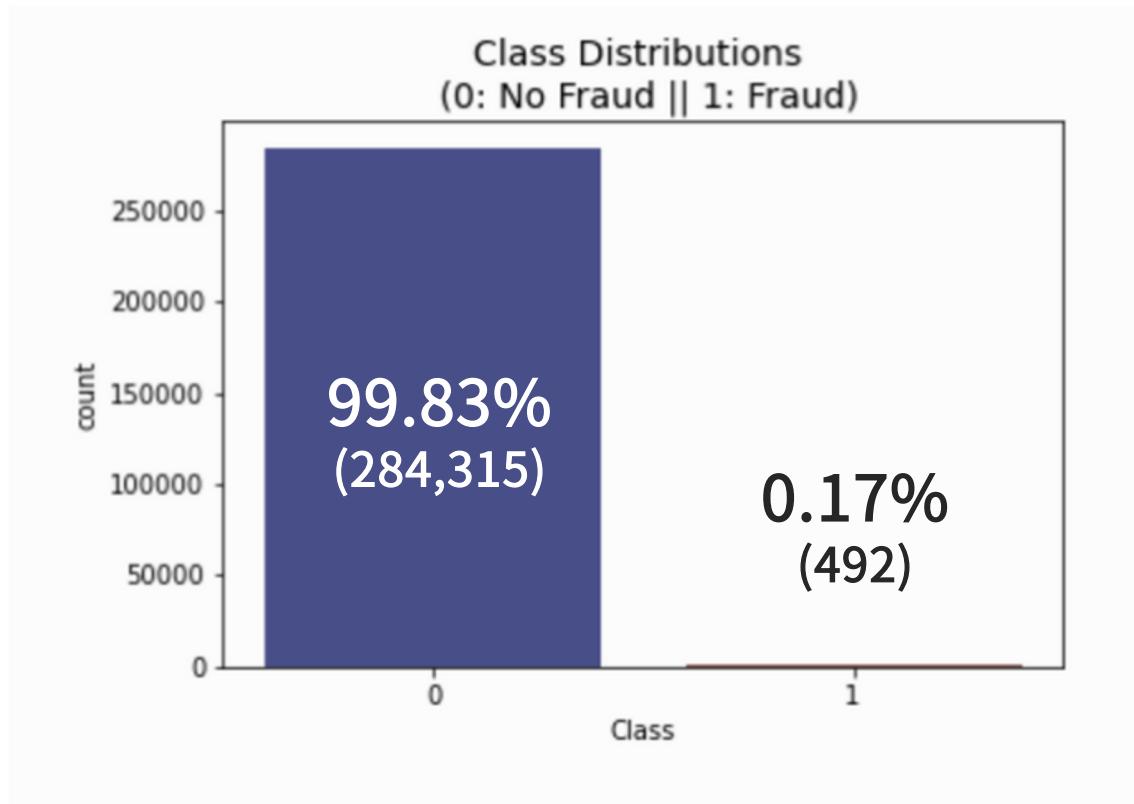
Description	Instances	Format	Update	Size	Origin
European Cardholders가 만든 2일 동안 발생한 카드 거래 데이터	284,807	Number	2013	150.8MB	Kaggle

Time	V1	V2	V3	V4	V5	V6
0	-1.3598071	-0.0727812	2.53634674	1.37815522	-0.3383208	0.46238778
0	1.19185711	0.26615071	0.16648011	0.44815408	0.06001765	-0.0823608
1	-1.3583541	-1.3401631	1.77320934	0.37977959	-0.5031981	1.80049938
1	-0.9662717	-0.185226	1.79299334	-0.8632913	-0.0103089	1.24720317
2	-1.1582331	0.87773675	1.54871785	0.40303393	-0.4071934	0.09592146
2	-0.4259659	0.96052304	1.14110934	-0.1682521	0.42098688	-0.0297276
4	1.22965763	0.14100351	0.04537077	1.20261274	0.19188099	0.27270812
7	-0.6442694	1.41796355	1.07438038	-0.492199	0.94893409	0.42811846
7	-0.8942861	0.2861572	-0.1131922	-0.2715261	2.66959866	3.72181806
9	-0.3382618	1.11959338	1.04436655	-0.2221873	0.49936081	-0.2467611
10	1.44904378	-1.1763388	0.91385983	-1.3756667	-1.9713832	-0.6291521
10	0.38497822	0.61610946	-0.8742997	-0.0940186	2.92458438	3.31702717

~

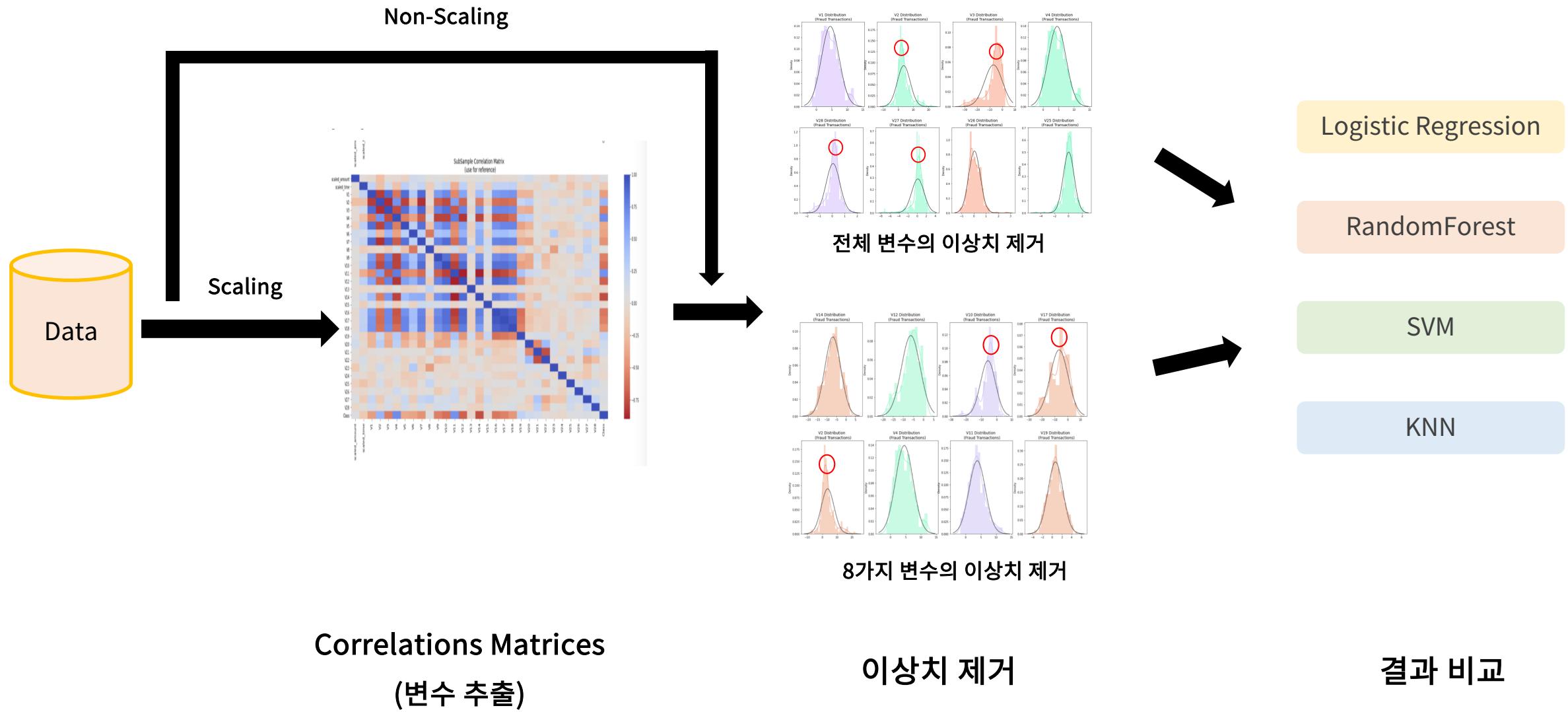
V27	V28	Amount	Class
0.13355838	-0.0210531	149.62	0
-0.0089831	0.01472417	2.69	0
-0.0553528	-0.0597518	378.66	0
0.06272285	0.06145763	123.5	0
0.21942223	0.21515315	69.99	0
0.25384422	0.08108026	3.67	0
0.03450743	0.00516777	4.99	0
-1.2069211	-1.0853392	40.8	0
0.01174736	0.14240433	93.2	0
0.2462193	0.08307565	3.68	0
0.04284987	0.01625326	7.8	0
0.04247244	-0.0543374	9.99	0

## 데이터 분석



- 284,807 건의 거래 내역 중 492건이 사기거래이며 정상거래의 비율은 99.83%, 사기거래의 비율은 0.17%
- 아무런 예측 없이 모든 거래가 정상거래라고 판단해도 99.83%의 정확도를 달성할 수 있으므로 Recall, Precision 등 별도의 평가 지표 필요

# 데이터 전처리



# 데이터 전처리

## 스케일링

Scaled Data

Aa 분류기	Logistic Reg	KNN	SVC	Random For
Accuracy	1.00	1.00	1.00	1.00
Recall	0.55	0.69	0.62	0.67
Precision	0.96	0.94	1.00	0.82
F1 Score	0.70	0.79	0.76	0.74
ROC AUC score	0.77	0.85	0.81	0.83

Non-Scaled Data

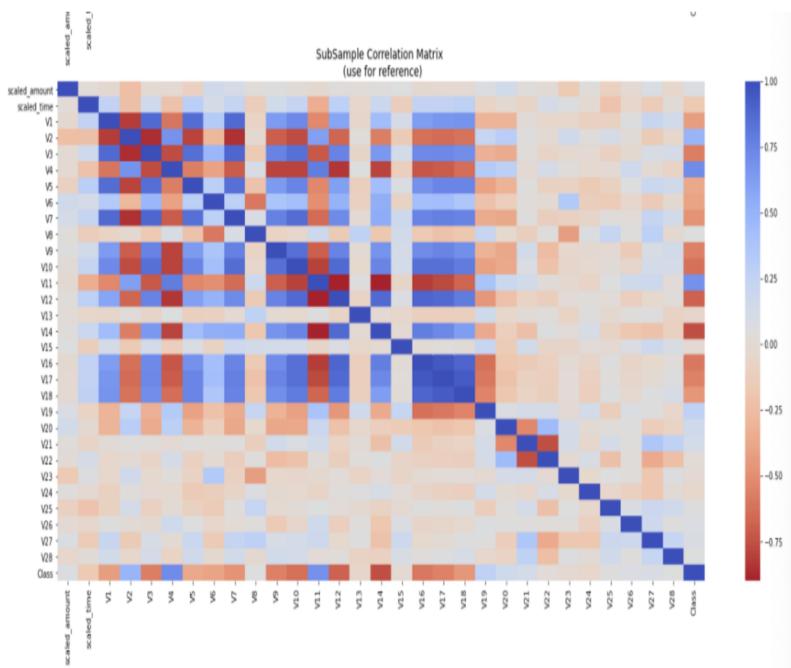
Aa 분류기	Logistic Reg	KNN	SVC	Random Forest
Accuracy	1.00	0.02	1.00	1.00
Recall	0.45	1.00	0.00	0.69
Precision	1.00	0.00	0.00	0.78
F1 Score	0.62	0.00	0.00	0.73
ROC AUC score	0.73	0.51	0.50	0.85

# 데이터 전처리

## 상관 분석

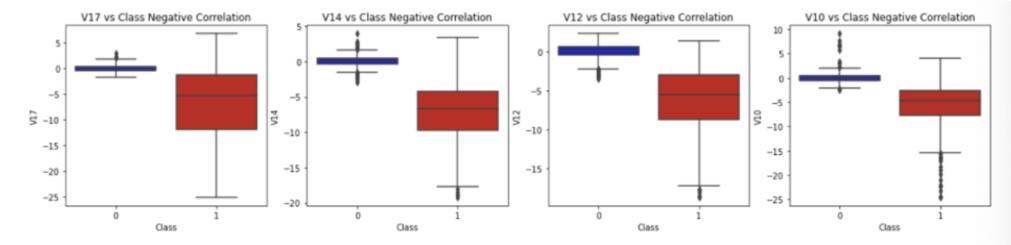
### Correlations Matrices

어떤 변수가 사기 거래 판단에 큰 영향을 미치는지 확인



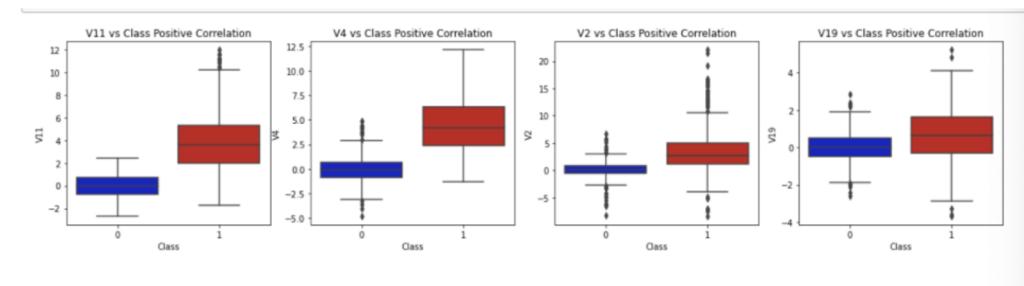
1. 음의 상관 관계 : V17, V14, V12, V10.

해당 변수의 값이 낮을 수록 사기 거래가 될 가능성이 높다.



2. 양의 상관 관계 : V2, V4, V11, V19.

해당 변수의 값이 높을 수록 사기 거래가 될 가능성이 높다.

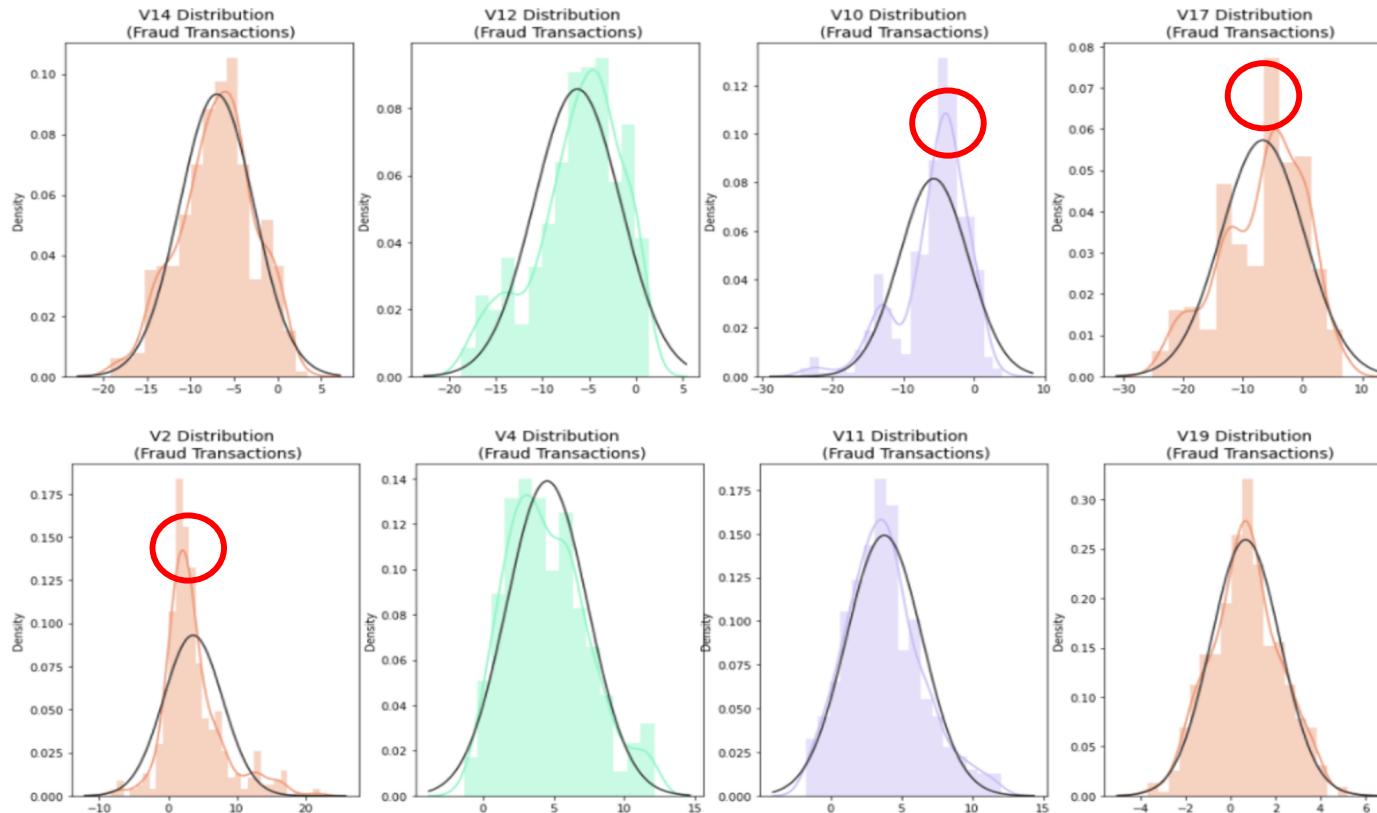


# 데이터 전처리

## 이상치 제거

# Outlier 제거

IQR 방법 사용



# 데이터 전처리

## 결과 비교

원본 데이터 | Class 0 : 284315, Class 1 : 492

Aa 분류기	Logistic Reg	KNN	SVC	Random For
Accuracy	1.00	1.00	1.00	1.00
Recall	0.67	0.77	0.73	0.83
Precision	0.88	0.99	0.99	0.99
F1 Score	0.76	0.86	0.84	0.90
ROC AUC score	0.84	0.88	0.87	0.91

Cut\_off = iqr \* 1.5

Cut\_off = iqr \* 2.0

전체 변수 이상치 제거 | Class 0 : 217230, Class 1 : 213

Aa 분류기	Logistic Reg	KNN	SVC	Random For
Accuracy	1.00	1.00	1.00	1.00
Recall	0.55	0.69	0.62	0.67
Precision	0.96	0.94	1.00	0.82
F1 Score	0.70	0.79	0.76	0.74
ROC AUC score	0.77	0.85	0.81	0.83

전체 변수 이상치 제거 | Class 0 : 246376, Class 1 : 261

Aa 분류기	Logistic Reg	KNN	SVC	Random Forest
Accuracy	1.00	1.00	1.00	1.00
Recall	0.63	0.71	0.63	0.71
Precision	0.97	0.95	1.00	0.90
F1 Score	0.77	0.81	0.78	0.80
ROC AUC score	0.82	0.86	0.82	0.86

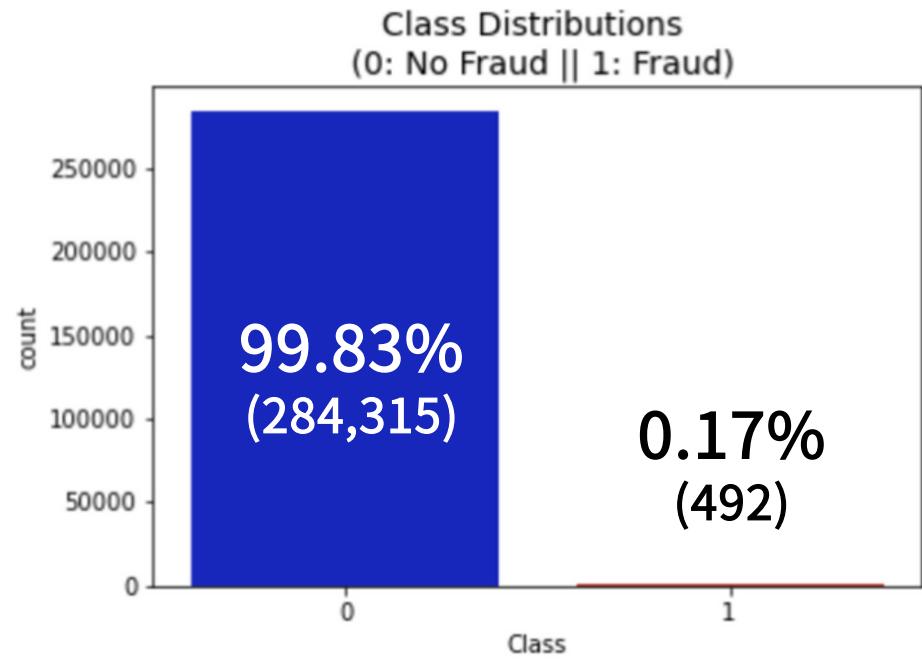
8가지 변수 이상치 제거 | Class 0 : 276110, Class 1 : 390

Aa 분류기	Logistic Reg	KNN	SVC	Random For
Accuracy	1.00	1.00	1.00	1.00
Recall	0.63	0.74	0.60	0.71
Precision	0.94	0.98	1.00	1.00
F1 Score	0.75	0.85	0.75	0.83
ROC AUC score	0.81	0.87	0.80	0.85

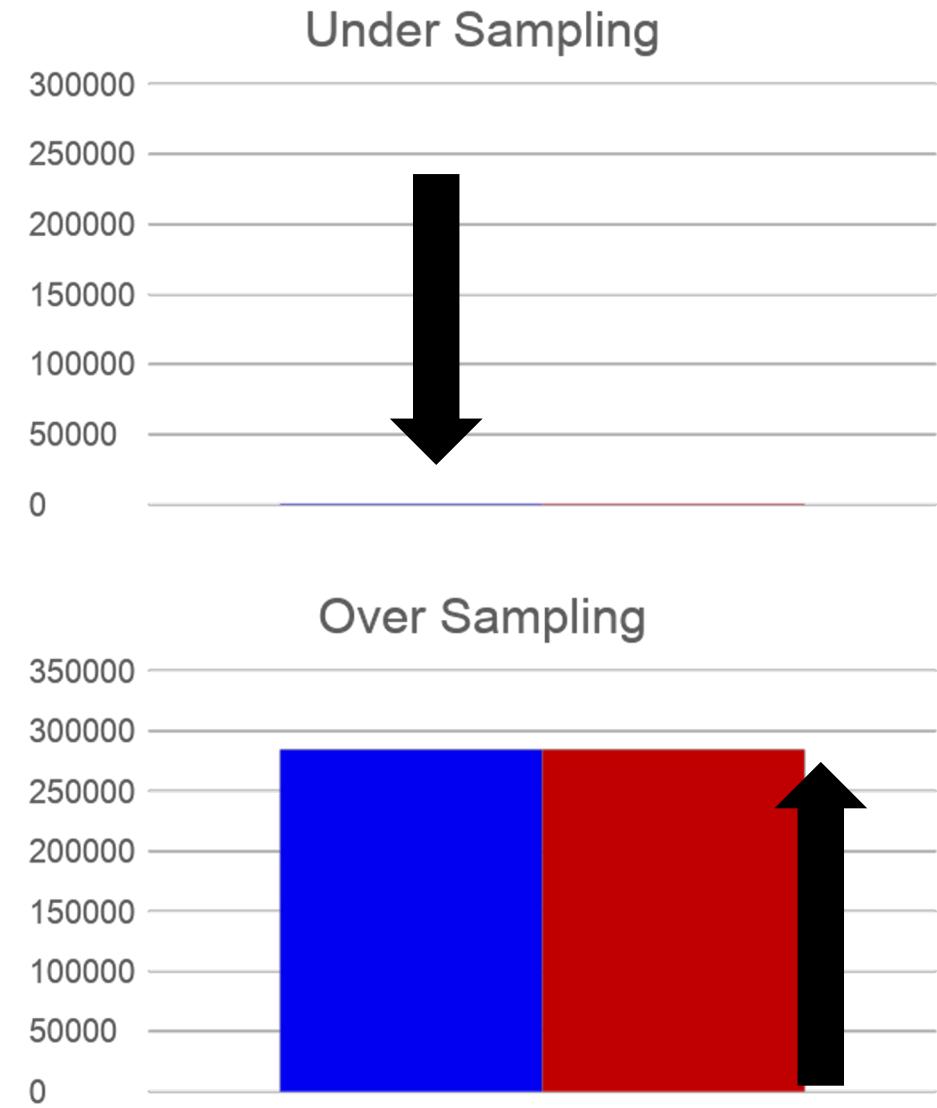
8가지 변수 이상치 제거 | Class 0 : 281341, Class 1 : 443

Aa 분류기	Logistic Reg	KNN	SVC	Random Forest
Accuracy	1.00	1.00	1.00	1.00
Recall	0.57	0.72	0.59	0.68
Precision	0.94	0.98	1.00	1.00
F1 Score	0.71	0.83	0.74	0.81
ROC AUC score	0.78	0.86	0.80	0.84

# Raw Data vs Under Sampling vs Over Sampling



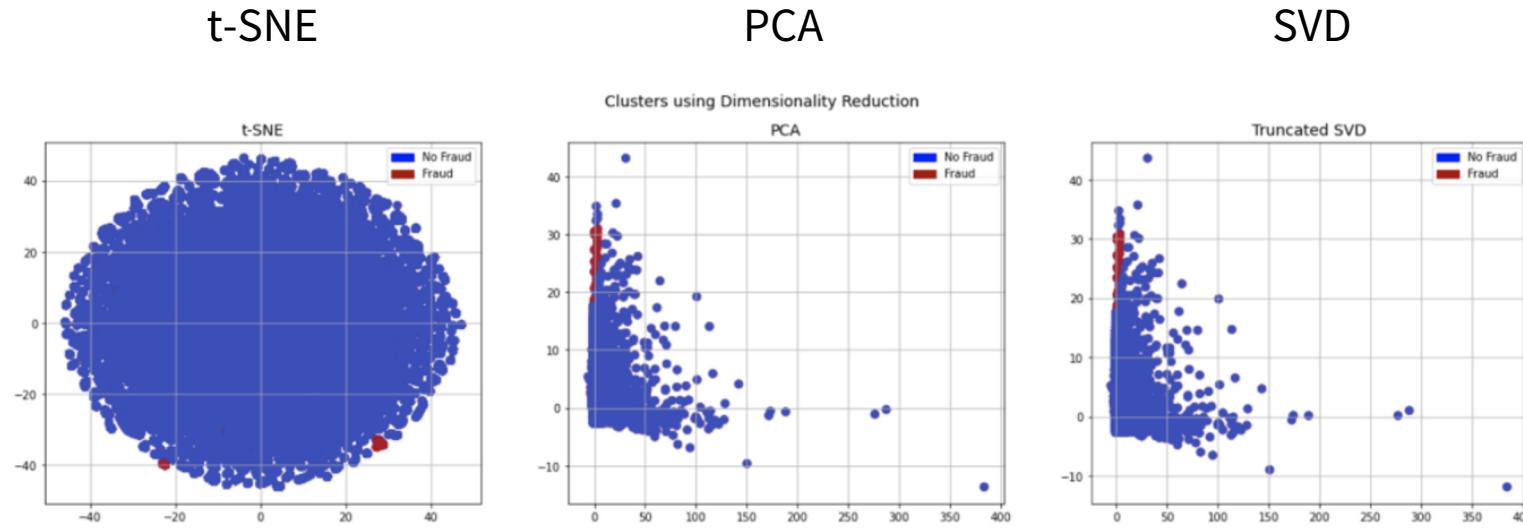
Raw Data  
(= Scaled Data)



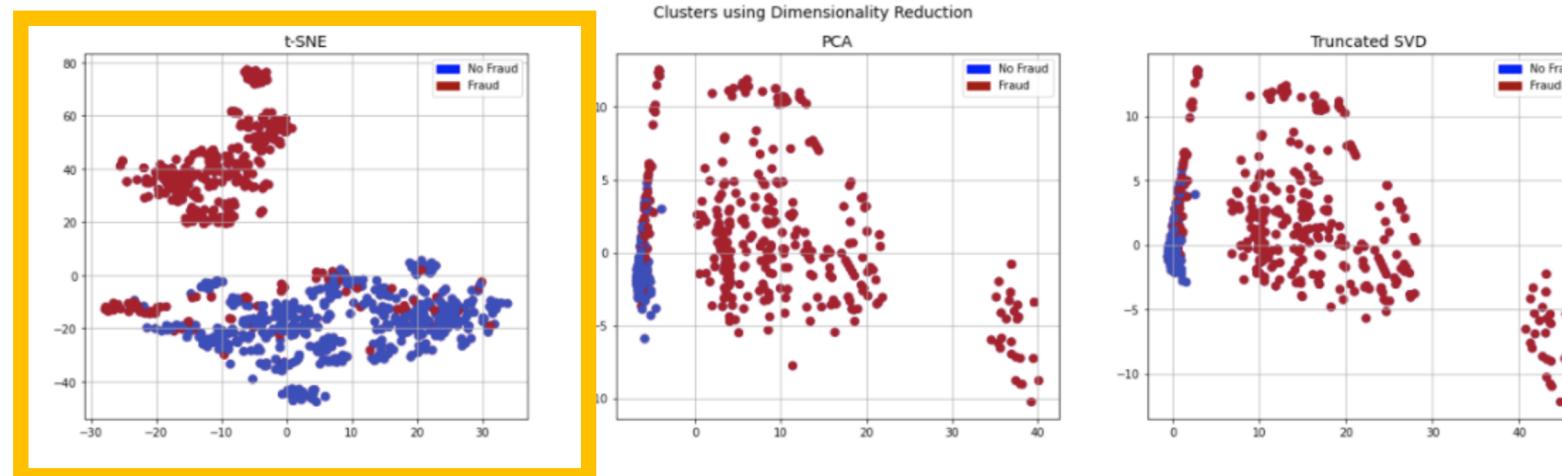
# Raw Data vs Under Sampling vs Over Sampling

## 1. Clustering

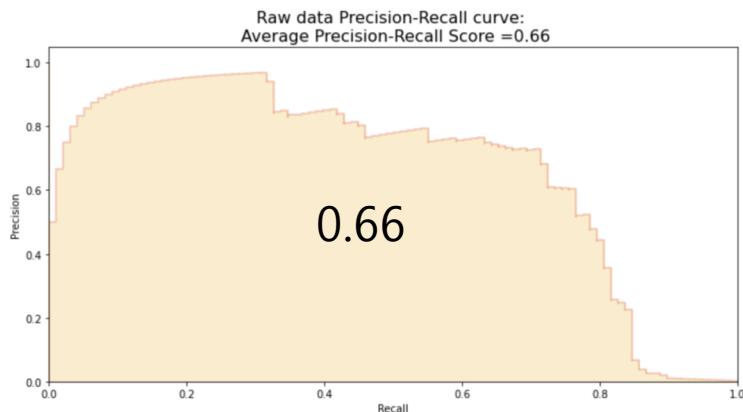
Raw Data



Under Sampling

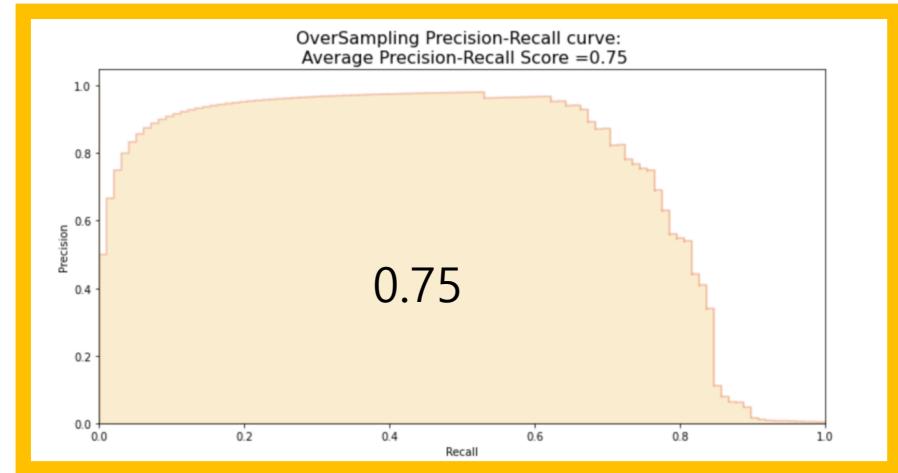


# Raw Data vs Under Sampling vs Over Sampling

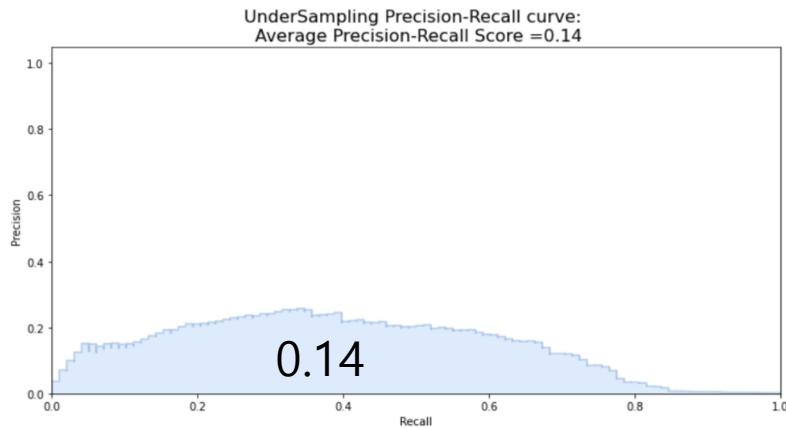


Raw Data

## 2. Classifier



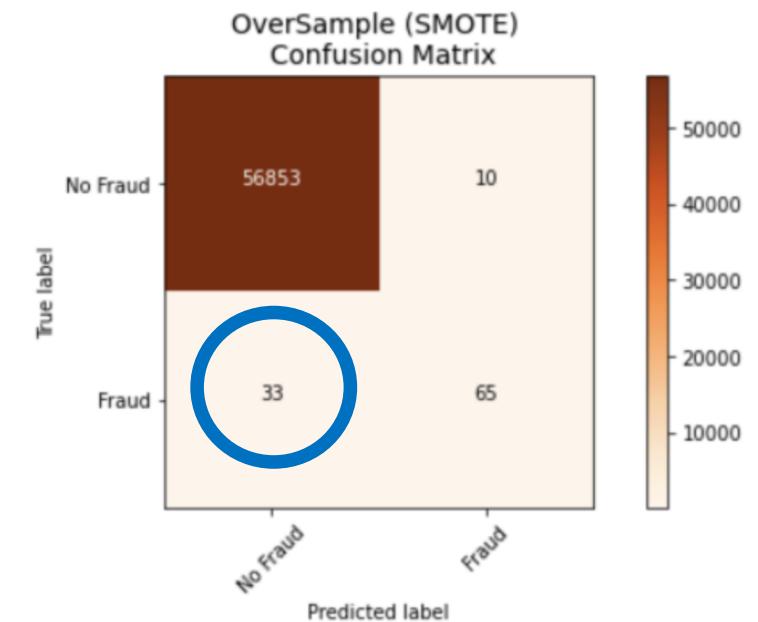
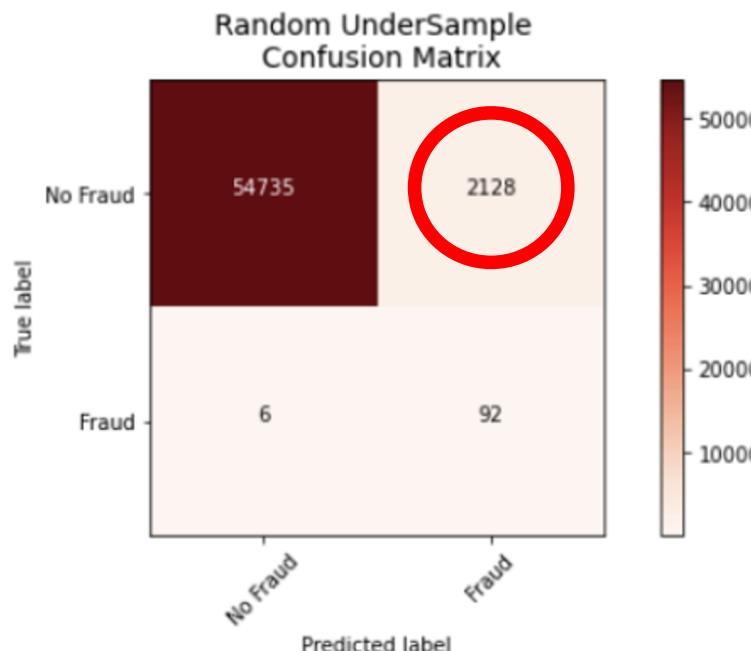
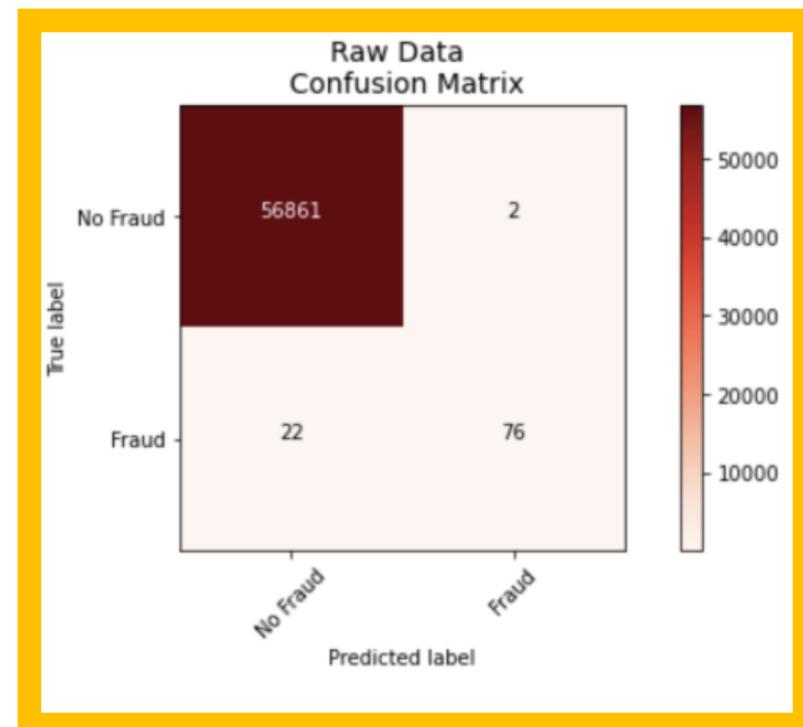
Over Sampling



Under Sampling

# Raw Data vs Under Sampling vs Over Sampling

## 3. Neural Network



사기 거래(True)인데  
정상거래(False)라고 하는 비율  
⇒ FN 가 높음 ( Recall 낮을 것)

정상 거래(False)인데  
사기 거래(True)라고 하는 비율  
⇒ FP 가 높음 ( Precision 낮을 것)

# 양상블

## 1. 단독 모델

Aa 분류기	Logistic Reg	KNN	SVC	Random Forest	AdaBoost	ExtraTrees
Accuracy	1.00	1.00	1.00	1.00	1.00	1.00
Precision	0.88	0.99	0.99	0.99	0.86	0.95
Recall	0.67	0.77	0.73	0.83	0.72	0.78
F1 Score	0.76	0.86	0.84	0.90	0.78	0.85
ROC AUC score	0.84	0.88	0.87	0.91	0.86	0.89

## 2. Voting

Raw Data

Aa 분류기	Soft Voting 1	Soft Voting 2	Soft Voting3
Accuracy	1.00	1.00	1.00
Precision	0.97	0.97	0.97
Recall	0.71	0.71	0.79
F1 score	0.82	0.82	0.87
AUC	0.86	0.86	0.89
시간	느림	매우 느림	빠름

Over Sampling

Aa 분류기	Soft Voting 1	Soft Voting 2	Soft Voting3
Accuracy	1.00	1.00	1.00
Precision	0.50	0.69	0.93
Recall	0.89	0.89	0.84
F1 score	0.64	0.78	0.88
AUC	0.94	0.94	0.92
시간	느림	매우 느림	빠름

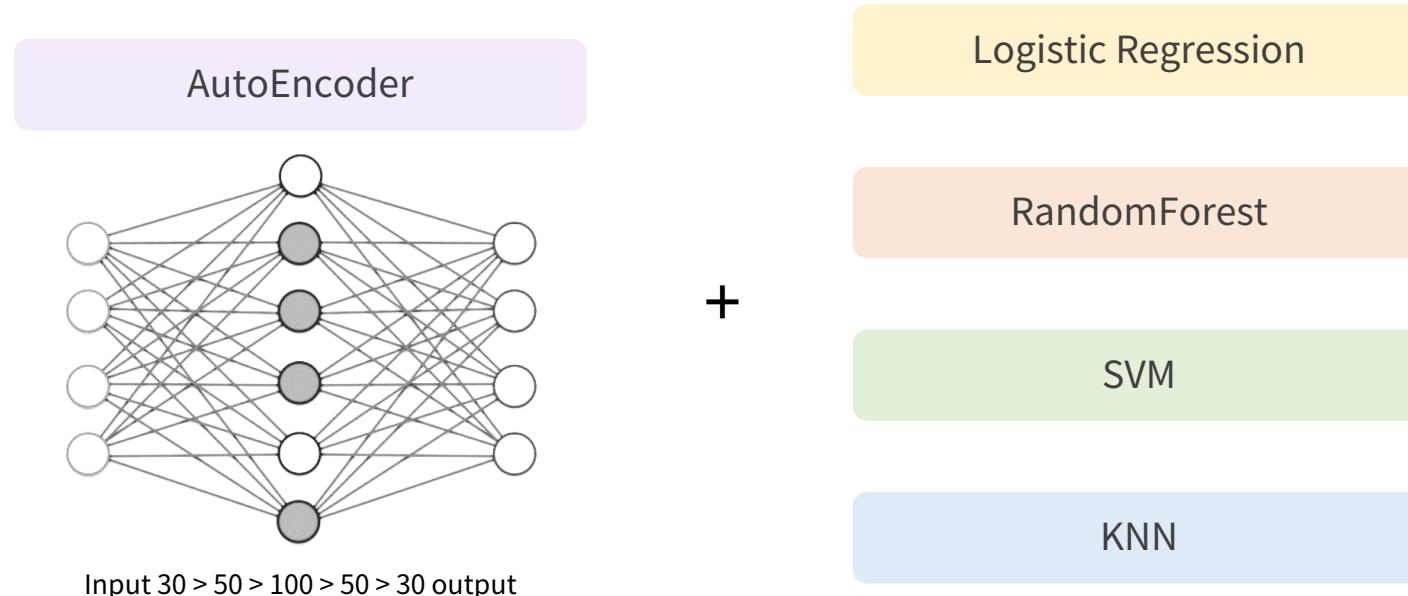
- Soft Voting 1 : Logistic Regression, Random Forest, KNN, SVC
- Soft Voting 2 : Logistic Regression, Random Forest, KNN, SVC, Ada Boost, Extra Trees
- Soft Voting 3 : Random Forest, Extra Trees, Ada Boost

## 3. Stacking

Aa 분류기	Stacking Vo...	Stacking Voti...	Stacking Vo...
Accuracy	1.00	1.00	1.00
Precision	0.97	0.99	0.99
Recall	0.73	0.76	0.76
F1 score	0.84	0.86	0.86
AUC	0.87	0.88	0.88
시간	느림	매우 느림	빠름

- Stacking 1 : Logistic Regression, Random Forest, KNN, SVC
- Stacking 2 : Logistic Regression, Random Forest, KNN, SVC, Ada Boost, Extra Trees
- Stacking 3 : Random Forest, Extra Trees, Ada Boost

## 하이브리드 모델



- 오토인코더로 정상 거래 데이터만 학습 시켜 정상 거래만 원복하는 패턴을 인지하는 모델을 생성
- 해당 모델에 input으로 정상 거래를 넣으면 잘 원복될 것이고, 사기 거래를 넣으면 잘 원복되지 않을 것
- 오토인코더의 output인 Latent Vector를 4개의 분류기에 넣어 분류 진행

# 하이브리드 모델

## 1. Without Oversampling

AutoEncoder + Classifier				
Aa 분류기	Logistic Reg	KNN	SVC	Random Forest
Accuracy	1.00	1.00	1.00	1.00
Precision	0.91	0.96	0.92	0.97
Recall	0.79	0.80	0.82	0.80
F1 Score	0.85	0.87	0.87	0.88
ROC AUC score	0.89	0.90	0.91	0.90

높아진 Recall

사기거래를 정상거래라고  
잘못 예측하는 비율 감소

Classifier				
Aa 분류기	Logistic Reg	KNN	SVC	Random Forest
Accuracy	1.00	1.00	1.00	1.00
Recall	0.67	0.77	0.73	0.83
Precision	0.88	0.99	0.99	0.99
F1 Score	0.76	0.86	0.84	0.90
ROC AUC score	0.84	0.88	0.87	0.91

낮아진 Precision

기본 모델은 거의 모든 데이터를  
정상 거래라고 분류한 것이라고 추론

전반적으로 좋아진 성능

RandomForest 제외 성능 향상

# 하이브리드 모델

## 2. With Oversampling

AutoEncoder + Classifier				
Aa 분류기	Logistic Reg	KNN	SVC	Random Forest
Accuracy	0.97	0.83	1.00	1.00
Precision	0.05	0.42	0.41	0.88
Recall	0.90	0.83	0.85	0.82
F1 Score	0.09	0.56	0.55	0.85
ROC AUC score	0.94	0.91	0.92	0.91

높아진 Recall

사기거래를 정상거래라고  
잘못 예측하는 비율 감소

Classifier				
Aa 분류기	Logistic Reg	KNN	SVC	Random Forest
Accuracy	1.00	1.00	1.00	1.00
Recall	0.67	0.77	0.73	0.83
Precision	0.88	0.99	0.99	0.99
F1 Score	0.76	0.86	0.84	0.90
ROC AUC score	0.84	0.88	0.87	0.91

매우 낮아진 Precision

잘 원복하지 못한 벡터의 비율이 커져  
분류 기준이 모호해진 것으로 추론

전반적으로 나빠진 성능

오히려 Oversampling Data에서는  
성능 저하

# 실험 결과

		Accuracy	Recall	Precision	F1 score	ROC AUC score
Raw data	Logistic Regression	1	0.67	0.88	0.76	0.84
	KNN	1	0.77	0.99	0.86	0.88
	SVM	1	0.73	0.99	0.84	0.87
	RandomForest	1	0.83	0.99	0.9	0.91
	AdaBoost	1	0.72	0.86	0.78	0.86
	ExtraTrees	1	0.78	0.95	0.85	0.89
	Soft Voting1	1	0.71	0.97	0.82	0.86
	Soft Voting2	1	0.71	0.97	0.82	0.86
	Soft Voting3	1	0.79	0.97	0.87	0.89
	Stacking Voting1	1	0.73	0.97	0.84	0.87
	Stacking Voting2	1	0.76	0.99	0.86	0.88
	Stacking Voting3	1	0.76	0.99	0.86	0.88
	AE + LR	1	0.79	0.91	0.85	0.89
	AE + KNN	1	0.8	0.96	0.87	0.9
	AE + SVM	1	0.82	0.92	0.87	0.91
Under Sampling	AE + RF	1	0.8	0.97	0.88	0.9
	Logistic Regression	0.96	0.93	0.03	0.07	0.94
	KNN	0.97	0.91	0.04	0.08	0.94
	SVM	0.98	0.91	0.07	0.13	0.94
	RandomForest	0.98	0.98	0.06	0.12	0.98
Over Sampling	Logistic Regression	0.98	0.93	0.06	0.11	0.95
	KNN	1	0.88	0.48	0.62	0.94
	SVM	0.98	0.88	0.09	0.16	0.93
	RandomForest	1	0.84	0.9	0.87	0.92
	Soft Voting1	1	0.89	0.5	0.64	0.94
	Soft Voting2	1	0.89	0.69	0.78	0.94
	Soft Voting3	1	0.84	0.93	0.88	0.92
	AE + LR	0.97	0.9	0.05	0.09	0.94
	AE + KNN	0.83	0.83	0.42	0.56	0.91
	AE + SVM	1	0.85	0.41	0.55	0.92
	AE + RF	1	0.82	0.88	0.85	0.91

Soft Voting 1 : Logistic Regression, Random Forest, KNN, SVC  
 Soft Voting 2 : Logistic Regression, Random Forest, KNN, SVC, Ada Boost, Extra Trees  
 Soft Voting 3 : Random Forest, Extra Trees, Ada Boost  
 Stacking 1 : Logistic Regression, Random Forest, KNN, SVC  
 Stacking 2 : Logistic Regression, Random Forest, KNN, SVC, Ada Boost, Extra Trees  
 Stacking 3 : Random Forest, Extra Trees, Ada Boost

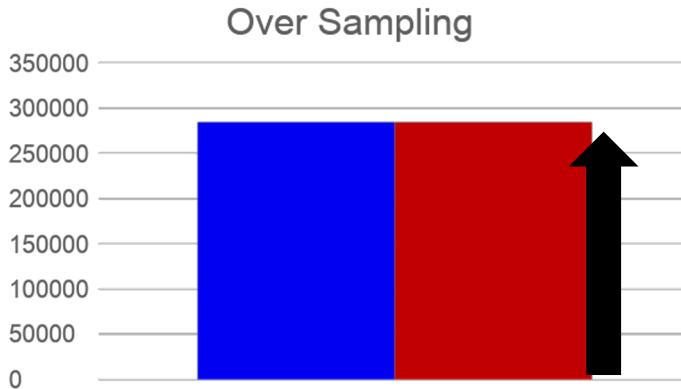
신경망은 Raw Data를  
이용할 때 성능 향상

Oversampling data는  
전반적으로 신경망 적용 시 성능 저하

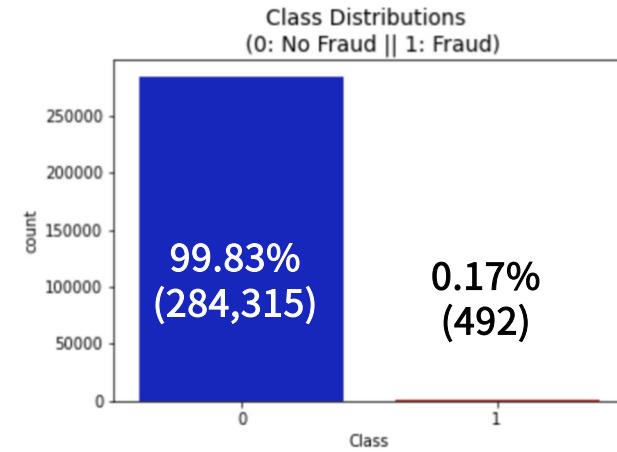
RandomForest 가  
모든 데이터셋에서 최고 성능 달성

# 결론

## Classifier



RandomForest



## Neural Network

AutoEncoder

+

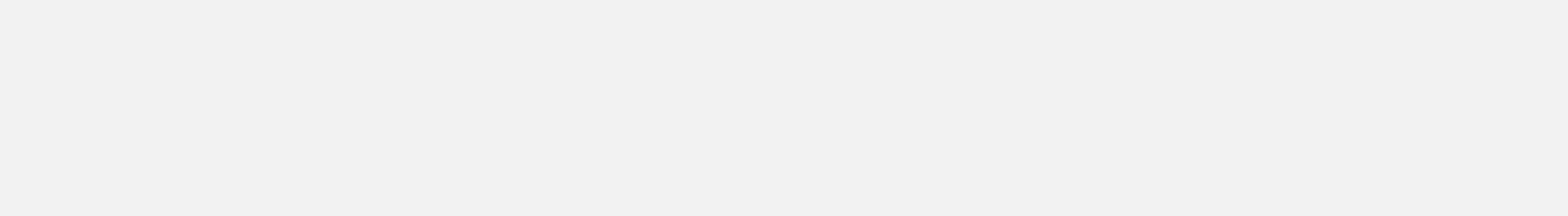
Logistic Regression

RandomForest

SVM

KNN

- Oversampling의 경우 학습은 Oversampling data로 하지만 결국 Test는 실제 데이터로 하므로 Precision이 좋지 않았다.
- 반면, 매우 불균형한 Raw data 데이터로도 Neural Network는 평균 0.9의 AUC Score를 달성했다.
- 실제 데이터의 세부적인 정보를 모르기 때문에 실사용이 불가능한 모델이라는 한계점을 가진다.
- 전반적으로 RandomForest의 성능이 좋았다는 점에서 Forest 모델을 이용한 이상 탐지 기법 연구가 더 이루어질 수 있을 것이다.



2020-2 기계학습 프로젝트

감사합니다

B511140 이송희 B611121 오현지