1. **초록**

선정한 데이터는 Kaggle에서 가져온 데이터로, 카드 사기에 대한 문제를 담고 있는 데이터이다.   
<https://www.kaggle.com/datasets/dhanushnarayananr/credit-card-fraud>

본 실험에서는 이미 Labelling 되어 있는 Y값이 존재한다. (정상: 0, 사기: 1)   
이를 활용하여 이상탐지 알고리즘을 통해 해당 학습 모델을 구축한다.   
그 후 해당 데이터의 정상과 비정상을 탐지하고, 이를 실제 값과 비교하여 이들의 효율성을 측정하려 한다.   
본 실험에서는 SVM과 SVDD 알고리즘을 적용하여 정확도 측면의 비교 실험을 수행하였다.

1. **문제 설명**

점점 많은 거래가 디지털로 이루어짐에 따라, 사이버 사기도 이에 따라 급속히 늘어나기 시작했다. 일 조가 넘는 거래가 하루 안에 이루어지고 있는 현 상황에서, 이 중 사기 거래를 걸러내어 (novelty detection) 더욱 성숙하고 건강한 거래 문화를 만들어야 한다.

1. **이상탐지를 적용하려는 데이터의 EDA**

이 데이터는 7개의 독립변수와 (=x) 1개의 종속변수(= y)가 존재한다.   
이는 다음과 같다:

Distance\_from\_home: 거래가 일어난 장소와 집 사이의 거리를 나타낸다.

Distance\_from\_last\_transaction: 마지막 거래 장소와 집 사이의 거리를 나타낸다

Ratio\_to\_median\_purchase\_price: 거래 구매 값/전체 거래의 중앙값의 비율을 나타낸다.

Repeat\_retailer: 같은 소매인에게 반복적으로 거래가 일어났는지를 암시한다.

Used\_chip: 카드를 긁어서 계산했는지 확인한다.

Used\_pin\_number: 핀 번호를 입력하여 계산했는지 확인한다.

Online\_order: 거래가 온라인으로 이루어졌는지 확인한다.

Fraud: 거래가 사기였는 지 확인한다.

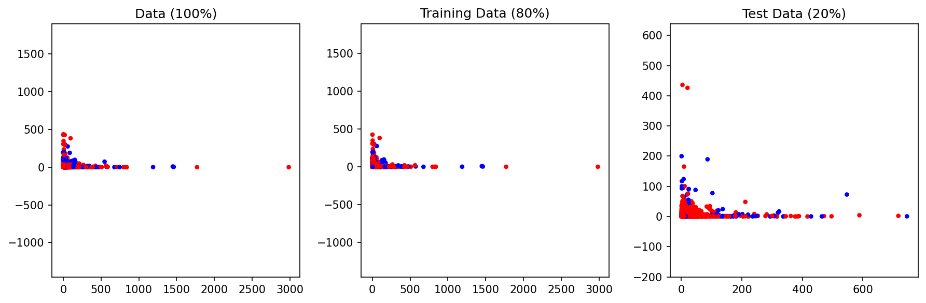
전체 샘플의 개수는 1,000,000개가 존재하였고, 그중 정상 데이터는 912597개, 그리고 비정상 데이터는 87043개였다. (약 9.57%)  
하지만 RAM의 물리적 한계로 인하여, 전체 데이터를 사용할 수는 없었고, 이 중 2000개를 층화 선택하여 모델 구축에 사용하였다.

모든 continuous변수는 왼쪽에 치우친 분포를 보여주었고, binary 변수는 특정한 분포를 나타내지는 않았다.

대부분의 correlation은 낮은 편이였고, 유일하게 ratio\_to\_median\_purchase\_price와 fraud만이 유의미한 0.46 상관성을 지니고 있었다.

데이터를 가공하기 위해 SMOTE 기법을 사용하였지만, 이 기법의 결과가 원치 않은 결과를 나타냈기 때문에,  
이를 철회하였다. 자세한 사항은 하단에 후술하도록 하겠다.

A picture containing chart

Description automatically generated

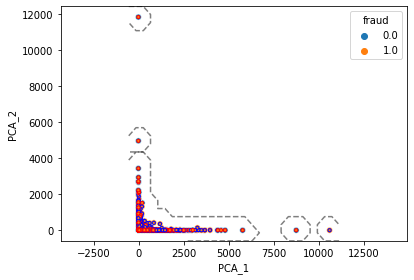
Fig 1, 2. SMOTE를 적용한 데이터(위)와 적용하지 않은 데이터 (아래).

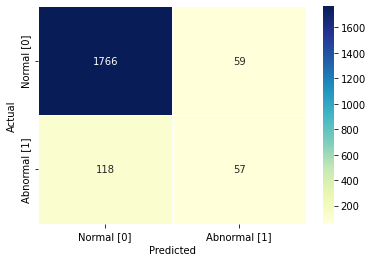
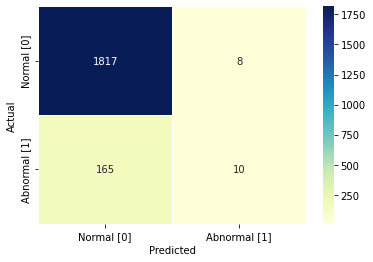
Fig 3. SVM이 적용된 이미지.

1. **본인이 선정한 새 이상탐지 알고리즘 간략 소개**

이번에 선정된 새로운 이상탐지 알고리즘은 SVDD(Support Vector Data Description)로,  
 Boundary-based Novelty Detection의 일종이다.   
이 알고리즘은 전 처리된 데이터를 가지고 라그랑지 듀얼 함수를 구한 후, 도출된 알파 값에 따라 데이터를 분류한다. 그 후 결정된 support vector를 이용해 radius를 계산, 이를 사용하여 Classification을 구함으로써 결과를 도출한다.

SVM이 d-1차원의 hyperplane boundary를 찾아 이상치를 찾는 것과는 달리   
SVDD는 최소 체적의 구를 찾는 것을 목적함수로 두고, 비선형 SVM을 응용하여 이상치 탐지를 위한 boundary를 제공한다. (참고로, 만일 모든 데이터들이 unit vector로 normalize되어 있다면 SVDD와 1 SVM은 동일하다.)   
이 알고리즘은 또한 Patch-SVDD, Deep SVDD 등 다양한 변칙적 하위 알고리즘이 존재한다.

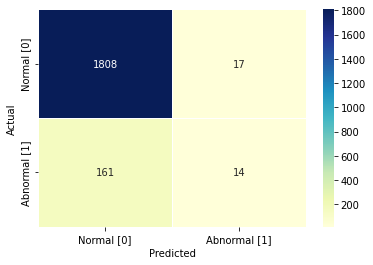
1. **실험결과 및 결론**

다음 결과는 SMOTE를 적용하지 않은 결과이다. (각각 SVM, SVDD)  


(Accuracy, Precision, Recall, F1은 아래 두 자리 수에서 반올림 처리하였다.)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Accuracy | Precision | Recall | F1 | FRR | FAR |
| SVM | 0.91 | 0.06 | 0.56 | 0.10 | 0.00438 | 0.94286 |
| SVDD | 0.91 | 0.33 | 0.49 | 0.39 | 0.03233 | 0.67429 |

SMOTE를 적용한 결과는 다음과 같다.

Chart, treemap chart

Description automatically generated

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Accuracy | Precision | Recall | F1 | FRR | FAR |
| SVM | 0.92 | 0.06 | 0.67 | 0.11 | 0.00274 | 0.94286 |
| SVDD | 0.91 | 0.08 | 0.45 | 0.14 | 0.00932 | 0.92000 |

SMOTE를 적용하지 않았을 때, 두 알고리즘의 차이는 크게 벌어지지만, SMOTE를 적용하면 두 알고리즘이 비슷해지는 경향을 나타낸다.

사기를 잡아내야 하는 특성 상, type I error 보다는 type II error가 더 치명적이라고 판단하였다.   
즉, 이는 낮은 FAR(False Acceptance Rate)를 가지고 있을수록 좋은 모델이라 가정하였고, 이에 따라 SMOTE를 적용하지 않은 SVDD를 가장 효율적이라고 가정하였다. (또한 F1 역시 가장 높은 값을 가지고 있다.)

물론, TP 부분이 상대적으로 밀리기 때문에, 이를 중요시 한다면 SMOTE를 사용한 SVDD를 사용하는 것도 한 방법이 될 수 있다.