

이학 석사학위논문

# 클래스 불균형 데이터 분류를 위한 GEV 활성 함수에 관한 연구

전남대학교 대학원

수학 / 통계학과

홍주영

2022년 2월

# 클래스 불균형 데이터 분류를 위한 GEV 활성화 함수에 관한 연구

이 논문을 이학 석사학위 논문으로 제출함

전남대학교 대학원  
수 학 / 통 계 학 과

홍 주 영

지도교수 박 정 수

홍주영의 이학 석사 학위논문을 인준함

심사위원장

김 민 수

심 사 위 원

고 봉 균

심 사 위 원

박 정 수

2022년 2월

## <목 차>

|  |     |
|--|-----|
| 목 차 .....                              | I   |
| 표 차례 .....                             | III |
| 그림 차례 .....                            | IV  |
| 국문 초록 .....                            | V   |
| <br>                                   |     |
| 1. 서론 .....                            | 1   |
| 가. 연구의 배경 .....                        | 2   |
| 나. 연구의 목적 .....                        | 2   |
| 다. 논문의 구성 .....                        | 3   |
| 2. 선행 연구 .....                         | 4   |
| 가. 일반화 극단분포 (GEVD) .....               | 4   |
| 나. GEV 분포를 활용한 클래스 불균형 데이터 분류 .....    | 4   |
| 3. 연구 방법 .....                         | 5   |
| 가. 인공신경망을 활용한 이진 분류 모형 .....           | 6   |
| 나. GEV-활성함수 .....                      | 7   |
| 다. 비용 민감 학습 .....                      | 8   |
| 1) $\alpha$ -balanced Focal Loss ..... | 8   |
| 2) 임계값 이동 .....                        | 9   |
| 라. SMOTE .....                         | 10  |

|                           |    |
|---------------------------|----|
| 4. 실험 .....               | 12 |
| 가. 데이터 셋 .....            | 12 |
| 나. 실험 개요 .....            | 14 |
| 다. 실험 환경 .....            | 15 |
| 라. 모델 구조 .....            | 16 |
| 마. 평가지표 .....             | 17 |
| 5. 실험 결과 .....            | 18 |
| 가. 일원 배치 분산 분석 .....      | 18 |
| 나. 오버샘플링을 적용한 실험 결과 ..... | 21 |
| 다. SOTA 모델과의 비교 .....     | 22 |
| 6. 결론 및 제언 .....          | 23 |
| 가. 요약 .....               | 23 |
| 나. 한계 및 향후 연구 과제 .....    | 24 |
| 참고문헌(References) .....    | 25 |
| 영문초록(Abstract) .....      | 27 |
| 부록(Appendix) .....        | 29 |

## <표 차례>

|   |    |
|---|----|
| 표 3.1 SMOTE 알고리즘 .....                          | 11 |
| 표 4.1 KEEL imbalanced data sets .....           | 12 |
| 표 5.1 실험 방법에 따른 5가지 평가지표의 30회 결과 평균(표준편차) ..... | 18 |
| 표 5.2 분산분석표 .....                               | 19 |
| 표 5.3 Leven의 등분산 검정 결과 .....                    | 20 |
| 표 5.4 사후검정 결과 .....                             | 20 |
| 표 5.5 일원 배치 분산분석 및 사후분석 결과 우수한 방법의 개수 .....     | 21 |
| 표 5.6 오버샘플링 비율에 따른 실험 결과의 평균(표준편차) .....        | 21 |
| 표 5.7 제안된 방법과 GEV-NN 모델과의 비교 .....              | 22 |

## 〈그림 차례〉

|   |    |
|---|----|
| 그림 1.1 클래스 불균형 .....                                  | 2  |
| 그림 1.2 연구 목적 .....                                    | 3  |
| 그림 2.1 일반화 극단 분포 도식 .....                             | 4  |
| 그림 2.2 선행 연구 및 본 연구의 위치 (파란색 배경) .....                | 4  |
| 그림 3.1 퍼셉트론 .....                                     | 5  |
| 그림 3.2 인공신경망을 활용한 기존 이진 분류 모델 .....                   | 6  |
| 그림 3.3 GEV-활성함수 .....                                 | 7  |
| 그림 3.4 GEV-활성함수의 모수 추정 .....                          | 7  |
| 그림 3.5 데이터 수준의 접근 방법 .....                            | 10 |
| 그림 3.6 SMOTE .....                                    | 11 |
| 그림 4.1 KEEL imbalance data set 100종에 대한 불균형도와 샘플 수 분포 | 13 |
| 그림 4.2 실험 순서도 .....                                   | 14 |
| 그림 4.3 조기 종료 .....                                    | 16 |
| 그림 4.4 실험에 사용된 신경망 구조 .....                           | 16 |
| 그림 4.5 ROC 곡선 .....                                   | 17 |
| 그림 5.1 일원 배치 분산분석 및 사후검정 결과 우수한 방법의 개수 .....          | 21 |

# 클래스 불균형 데이터 분류를 위한 GEV 활성화 함수에 관한 연구

홍 주 영

전남대학교대학원 수학/통계학과

(지도교수 : 박정수)

(국문 초록)

클래스 불균형 문제는 다수 클래스(Majority class)가 소수 클래스(Minority class)에 비하여 매우 높은 빈도로 등장하는 경우를 의미하며, 의료 진단, 화재 발생 탐지, 사기 거래 감지 등 실생활 대부분에서 매우 자주 발생한다. 이러한 불균형 데이터에 전통적 분류 알고리즘을 적용하면, 정확도 향상을 위해 소수 클래스를 무시하는 경향이 있다. 그러나 대부분 소수 클래스가 우리의 관심이며 전체적인 정확도 보다는 이를 잘 분류하는 것이 중요하다. 따라서 본 연구에서는 위 문제를 해결하기 위한 방안으로 GEV 활성화 함수와  $\alpha$ -balanced Focal Loss, Thresholding 및 오버 샘플링(SMOTE)을 결합한 방법을 제안한다. 제안된 방법의 효용성을 진단하기 위해 100종의 데이터셋을 활용하여 5가지 평가지표를 비교하였으며, 5% 유의수준하에서 일원배치 분산분석 및 사후검정을 수행하였다. 데이터 셋과 평가지표에 대한 500개의 조합 중 162개, 즉 전체의 32.4%가 5% 유의수준 하에서 유의한 차이를 보였으며, 모든 비교 결과를 종합하였을 때 제안하는 방법이 시그모이드함수를 사용할때보다 우수하게 나타났다. 오버 샘플링 적용시, 최적의 샘플링 비율은 20:1로 판단되며, 샘플링 후 제안한 방법의 결과를 SOTA 모델과 비교한 결과 5가지 데이터 셋 모두에서 우수한 결과를 얻었다.

주요용어: 클래스 불균형, 일반화 극단분포, 활성화 함수 등

## 1. 서론

통상적으로 사용되는 데이터 수집 및 관리, 처리 소프트웨어의 수용 한계를 넘어서는 크기의 데이터를 빅데이터라고 칭한다. 2000년대 스마트폰, SNS, 사물인터넷(IOT)의 확산에 따라 데이터가 폭증하며 빅데이터는 정보통신기술(ICT) 분야의 새로운 패러다임이자 신성장동력으로 급부상하고 있다. 다양한 종류의 대규모 데이터를 생성, 수집하며 분석하는 특징을 가진 빅데이터 기술의 발전은 다변화된 현대사회를 정확하게 예측하며, 사회가 효율적으로 작동하게 한다. 이러한 빅데이터를 비즈니스에 활용함으로써, 비즈니스 비용 절감 및 효율성 개선을 이루어 낼 수 있고, 그 결과를 마케팅에 활용하거나 새로운 비즈니스 모델을 창출해 낼 수 있다.

이처럼 빅데이터는 정보통신기술의 발전에 따라 등장한 복잡하면서, 무한한 가능성을 가진 자원이다. 정보의 생산, 유통, 보유량이 기하급수적으로 증가함으로써 그 중요성이 점차 확대되고 있으며, 이제 빅데이터의 이용은 필수적이라고 할 수 있다. 또한, 이를 이용하여 비즈니스 인사이트(Insight)를 찾아내고 정책 또는 비즈니스 생산성 향상에 활용하려는 욕구가 점차 증가하는 추세이다.

한편 기계학습은 데이터로부터 지식을 추출하는 도구이다. 기계학습 알고리즘은 빅데이터로부터 인사이트를 추출하고, 이를 통해 우리가 신속, 정확하게 의사결정을 할 수 있도록 돕는다.[1] 이 때문에, 최근 많은 웹사이트와 기기들은 영화 추천에서부터 음식 주문, 쇼핑, 개인 맞춤형 추천 서비스 등 여러 가지 분야에서 기계학습 알고리즘을 핵심 기술로 채택하고 있다.

기계학습으로 해결할 수 있는 문제 중 가장 자주 접하는 문제는 분류(Classification)이다. 분류는 주어진 관측치가 미리 정의된, 가능성이 있는 여러 클래스 중 어디에 속하게 될지를 예측하는 문제이다.[2] 기존 통계학에서는 로지스틱 회귀분석, 선형판별분석, k-근접 이웃과 같은 알고리즘을 통해 분류문제를 해결하였다. 이러한 알고리즘들은 클래스 간 샘플 수가 거의 비슷한 경우를 가정하지만, 대부분의 실제 데이터는 한쪽 클래스에 편중되는 경향이 있다.



## 가. 연구의 배경

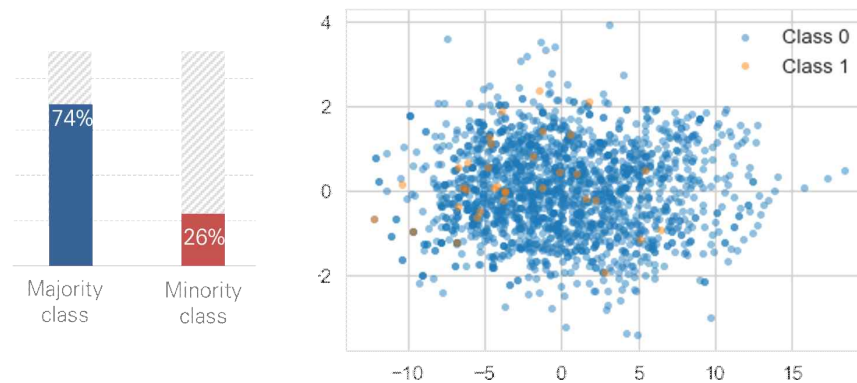


그림 1.1 클래스 불균형

클래스 불균형 문제는 주어진 데이터에서 각 클래스의 샘플 수가 현저하게 차이가 나는 문제이다. 이러한 문제는 의료 진단, 화재 발생 탐지, 사기 거래 감지 등 다양한 분류문제에서 매우 자주 발생한다.[3] 대부분 이런 경우에는 다수 클래스(Majority Class)보다는 소수 클래스(Minority Class)에 관심이 많으며, 전체적인 정확도 보다는 소수 클래스를 정확하게 예측하는 것이 중요하다.

기계학습 알고리즘은 대부분 각 클래스의 비율이 비슷함을 가정하기 때문에, 불균형 데이터를 그대로 기계학습 모형에 학습시키는 경우 다수 클래스에 편향되어 학습하게 된다.[4] 이는 전체적인 정확도를 높일 수는 있지만, 주로 관심 있는 소수 클래스에 대한 예측 정확도는 감소하게 된다. 또한, 정보통신 기술의 발전으로 데이터를 수집, 관리, 유통하는 환경이 복잡해짐에 따라 데이터의 불균형도가 높아지는 등 여러 가지 문제가 발생하고 있다. 따라서 클래스 불균형 문제에 대한 해결 방안이 반드시 필요하다.

## 나. 연구의 목적

본 연구의 목적을 요약하면 다음과 같다.

- (1) 클래스 불균형 문제를 극복하기 위한 알고리즘으로써 인공신경망에 최근 개발된 GEV-활성함수[5]를 적용하고, 여기에 기존 알고리즘(비용 민감 학습 및 오버 샘플링)을 결합하여 더 높은 분류 정확도를 갖는 모형을 제시한다.
- (2) 제안하는 방법에서의 최적 샘플링 비율을 탐색 후 제시한다.
- (3) 선행논문[5]에서 향후 연구과제로 남겨둔 data - efficient method 평가를 수행한다.

궁극적으로는 제안하는 방법을 활용하여 클래스 불균형 자료에 대한 분류 성능 향상을 목표로 한다.

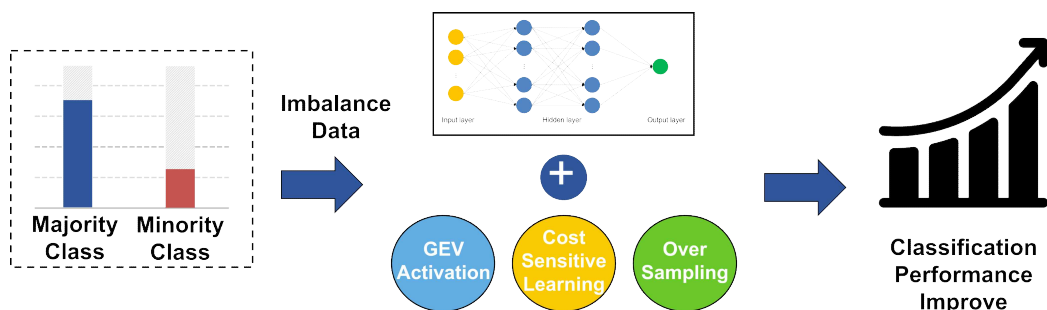


그림 1.2 연구 목적

## 다. 논문의 구성

본 논문의 구성은 다음과 같다: 2. 선행 연구에서는 클래스 불균형 문제를 해결하기 위한 관련 선행 연구에 대해 요약한다. 이 연구에서 제안하는 불균형 데이터를 효율적으로 분류하기 위한 방법은 3. 연구 방법에서 설명한다. 제안한 방법의 효용성을 검증하기 위한 실험의 설계 및 개요는 4. 실험에 제시한다. 5. 실험 결과에서는 실험의 결과를 제시하며, 본 연구에서 사용한 데이터에 대한 SOTA 모델의 결과와 비교하였다. 마지막으로 6. 결론 및 제언에서는 본 논문의 내용을 요약하며, 해당 연구의 한계점 및 향후 과제에 대해 기술하였다.

## 2. 선행 연구

### 가. 일반화 극단분포 (GEVD)

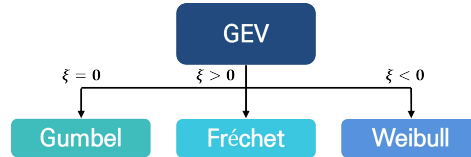


그림 2.1 일반화 극단 분포 도식

일반화 극단 분포(Generalized Extreme Value Distribution)[6]는 통계학에서 일변량 극단값들을 분석하는데 널리 사용된다. GEV 분포는 Gumbel, Fréchet, Weibull 분포로 알려진 3가지 극단 분포를 하나로 통합하는 분포이다. 일반화 극단분포의 누적 분포 함수는 다음과 같다.

$$G(x) = \exp \left\{ - \left[ 1 + \xi \left( \frac{x - \mu}{\sigma} \right) \right]^{-1/\xi} \right\} \quad (2.1)$$

일반화 극단분포는  $\{x : 1 + \xi(x - \mu)/\sigma > 0\}$ 에서 정의되며, 3가지 모수를 갖는다. 각 모수들은 다음과 같은 범위를 만족해야한다:  $-\infty < \mu < \infty$ ,  $\sigma > 0$ ,  $-\infty < \xi < \infty$ .  $\mu$ ,  $\sigma$ ,  $\xi$ 는 각각 위치(location) 모수, 규모(scale) 모수, 모양(shape) 모수이다. 특히, (2.1)의 등식에서  $\xi = 0$ 인 경우, 확률 변수  $X$ 는 Gumbel 분포,  $\xi > 0$ 인 경우, Fréchet,  $\xi < 0$ 인 경우, Weibull 분포를 따른다.

### 나. GEV 분포를 활용한 클래스 불균형 데이터 분류

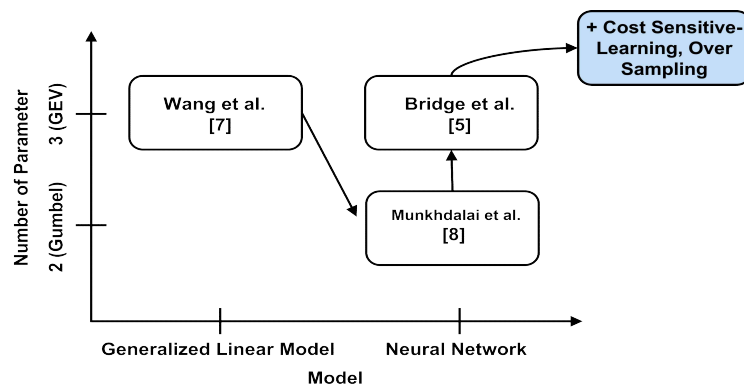


그림 2.2 선행 연구 및 본 연구(파란색 배경)의 위치

몇몇 연구자들은 클래스 불균형 데이터의 효과적인 분류를 위해 일반화 극단분포를 활용하였다. Wang & Dey.[7]는 일반화 선형 모형(GLM)에 비대칭 연결함수(Link-function)로써 GEV분포를 적용하여 이진 분류 상황에서 클래스 분류 확률을 추정하였으며, Munkhdalai et al.[8]은 이진 분류 상황에서 인공신경망 모형의 활성화함수로 GEV 분포의 한가지 클래스인 Gumbel 분포를 사용하여 예측 성능을 향상 시켰다. 가장 최근에는 Bridge et al.[5]이 CT와 X-ray 이미지를 활용하여 코로나 바이러스(COVID-19)를 진단하는 CNN 모형에 GEV-활성함수를 사용하여 분류 성능을 향상 시켰다. 그러나 이전의 연구들에서는 GEV-연결함수 또는 활성화함수에 비용 민감 학습 및 샘플링 방법을 결합하여 적용하지 않았으며, 앞선 방법들을 결합하였을 때 성능향상이 가능할 것으로 판단하였다. 따라서 본 연구에서는 GEV-활성함수를 이용한 인공 신경망 모형에 비용 민감 학습 및 오버 샘플링을 결합한 방법을 제안한다.

### 3. 연구 방법

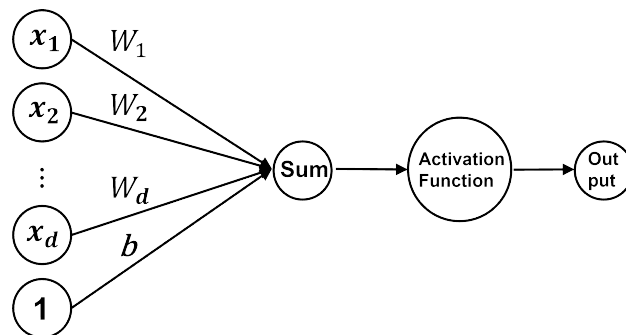


그림 3.1 퍼셉트론

그림 3.1은 인공신경망을 이루는 퍼셉트론의 도식이다.  $x_i$ 는 입력층의  $i$ 번째 노드의 입력이며,  $W_i$ 는  $i$ 번째 노드의 가중치이다.  $b$ 는 편의를 의미한다. 입력 변수들( $x_i$ ,  $i = 1, \dots, d$ )은 각각 가중치들( $W_i$ ,  $i = 1, \dots, d$ )과 곱해지며, 편 의( $b$ )와 함께 더해진다. 더해진 값은 활성화함수를 거치고 출력값이 결정되게 된다. 퍼셉트론을 여러 층으로 쌓으면 다층퍼셉트론(MLP)이 되며, 다층퍼셉

트론에서 은닉층을 여러 층으로 깊게 만든 신경망을 딥러닝으로 간주한다.

딥러닝은 다양한 분야에서 성공적으로 목적을 달성해 냈으며, 여러 분야에  
서 현재 활발하게 연구 중이다. 예를 들면, 이미지 분류, 자율주행, 음성인식  
등 수많은 직무를 높은 정확도로 성취해내고 있다.

#### 가. 인공신경망을 활용한 이진 분류 모형

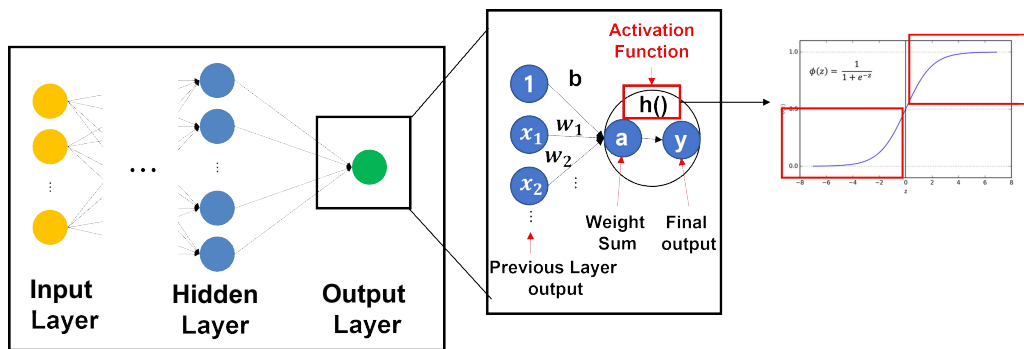


그림 3.2 인공신경망을 활용한 기존 이진 분류 모델

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (3.1)$$

다층퍼셉트론을 활용한 기존의 이진 분류 모델의 동작 방식은 그림 3.2와 같다. 입력층(Input Layer)을 통해 설명변수들이 들어오며, 이들은 몇 개의 은닉층(Hidden Layer)을 거치며 마지막 신호값( $a$ )이 출력층(Output Layer)에 도달하게 된다. 마지막 가중치 신호 조합( $a$ )은 최종적으로 활성화함수( $h$ )를 거쳐 최종 출력값( $\hat{y}$ )이 되는데, 이때 활성화함수( $h$ )로 주로 시그모이드(Sigmoid) 함수를 사용한다. 시그모이드 함수는 모든 실수 입력을 미분 가능한 0과 1사이 값으로 변환해주기 때문에, 인공신경망의 최종 출력값을 주어진 클래스에 분류될 확률로 이해할 수 있다. 이러한 성질 때문에 시그모이드 함수는 이진 분류상황에 사용하기 적합하다. 그러나 그림 3.2의 가장 오른쪽처럼 시그모이드 함수는 우측 상단과 좌측 하단의 그래프 개형이 대칭적인 구조이다. 따라

서 선행 연구[7, 8, 5]와 같이 비대칭한 구조를 갖는 함수를 시그모이드 대신 사용하면, 불균형 데이터에 대한 분류 성능이 향상될 것으로 기대한다.

#### 나. GEV-활성함수

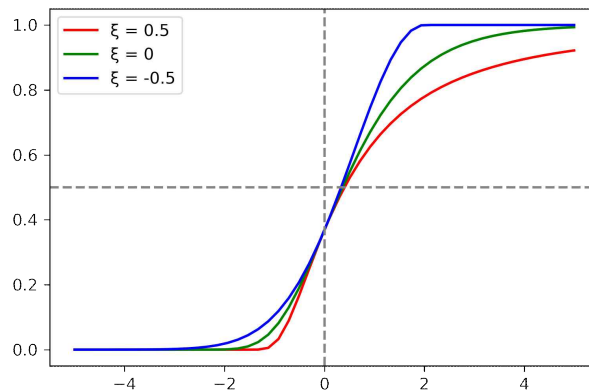


그림 3.3 GEV-활성함수

GEV-활성함수는 일반화 극단 분포의 CDF를 인공신경망의 활성화함수로 사용한 것이다. 이 함수는 GEV 분포의 CDF이므로 모든 실수 입력값을 0과 1 사이 값으로 변환해주며, 이진 분류에 유용하게 활용할 수 있다. 활성화함수의 그래프 개형은 그림 3.3과 같으며, 모양 모수  $\xi$ 의 값에 따라 활성화함수의 좌측 하단과 우측상단이 비대칭하게 변화한다. 이러한 성질 때문에 불균형 데이터에 알맞은 결정 경계를 찾아줄 것으로 기대할 수 있다.

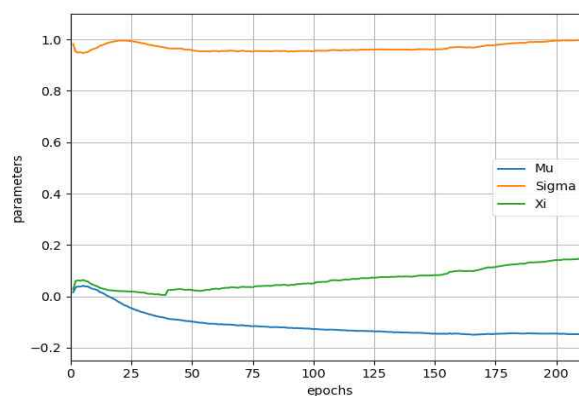


그림 3.4 GEV-활성함수의 모수 추정

(Data : abalone-20\_vs\_8-9-10)

활성함수가 갖는 3가지 모수 ( $\mu$ ,  $\sigma$ ,  $\xi$ )는 오차 역전파 방법을 통해, 다른 가중치 및 편의와 동시에 추정하였다. 세 모수의 초깃값은  $\mu = 0$ ,  $\sigma = 1$ ,  $\xi = 0.01$ 로 주었으며, 경사 하강법을 통해 매 에폭(Epoch)마다 추정값을 갱신한다.

활성함수 구현 시  $\xi \neq 0$ 인 경우, 2가지 문제점이 발생한다. (1) 함수가 미분 불가능하여, 모수를 갱신하기 위한 기울기를 구하기 어렵다. (2) 함수는  $1 + \xi(x - \mu)/\sigma > 0$ 에서만 정의되므로, 정의되는 구간을 벗어나지 않도록 모수가 갱신되어야 한다.

위의 두 가지 문제점은 다음과 같이 해결하였다. (1) tensorflow 패키지의 GradientTape API를 활용하여, 모든 연산을 테이프(Tape)에 기록한 뒤, 후진 방식 자동미분(Reverse mode differentiation)을 사용해 테이프에 기록된 연산의 그래디언트를 계산하여 사용하였다. (2) 함수의 정의 구간을 벗어나지 않도록 갱신하기 위해, 매 에폭마다  $\xi(x - \mu)/\sigma$  와  $-1$ 을 비교하여,  $\xi(x - \mu)/\sigma$ 이  $-1$  이하 일때는  $-1$ 에 epsilon을 더한 값을 대신 사용하였다.

#### 다. 비용 민감 학습

비용 민감 학습(Cost-Sensitive Learning)은 오분류 비용을 고려하는 학습 방식이다. 이것은 불균형 자료의 분류에 자주 사용되는 방법 중 하나이며, 전체 오차(예를 들면, MSE)를 최소화하는 것이 아닌, 오분류 비용을 정의하고, 총 오분류 비용을 최소화하도록 학습하는 방법이다. 대표적인 방법으로는 비용 행렬(Cost-matrix)을 이용하거나 임계값을 이동하는 방법(Threshold moving), 손실 함수(Loss function)를 변경하는 방법이 있다.

##### 1) $\alpha$ -balanced Focal Loss

$$p_t = \begin{cases} p & \text{if } y = 1 \text{ (positive)} \\ 1-p & \text{otherwise,} \end{cases} \quad (3.2)$$

$$FL(p_t) = -\alpha_t(1-p_t)^\gamma \log(p_t) \quad (3.3)$$

본 연구에서 사용된 손실함수는  $\alpha$ -balanced Focal Loss[9] 이며, 수식 3.3과 같이 정의된다. Focal Loss는 교차 엔트로피(Cross Entropy)의 변형으로, Dynamic하게 Scaling되는 교차 엔트로피로 이해할 수 있다. 이 함수는 객체 탐지(Object Detection)에서 전경(foreground)과 배경(background)의 클래스 사이의 극단적인 불균형을 해결하기 위해 제안되었으며, 모형이 분류하기 쉬운 샘플에 대해서는 비용을 낮추어 분류하기 어려운 샘플에 집중할 수 있게 한다.

Focal Loss는 다음 2가지 특성을 갖는다. (1) 한 개체가 잘못 분류되었을 때,  $p_t$  값이 작은 경우에는 조절인자  $(1-p_t)^\gamma$  의 값이 1 근처로 가며 손실함수는 영향을 받지 않게 된다. 그러나 올바르게 분류된 경우  $p_t$  값이 1에 가까워짐에 따라 조절인자의 값이 0 근처로 가며 손실함수의 값이 작아진다. (2) 집중 모수  $\gamma > 0$ 는 분류하기 쉬운 예의 down weighting 정도를 부드럽게 조절한다.  $\gamma = 0$ 이면 Focal Loss는 교차 엔트로피와 같으며,  $\gamma$ 의 값이 커질수록 조절인자의 효과도 함께 증가한다.

$\alpha$ -balanced Focal Loss는 Focal Loss의  $\alpha$ -balanced 변형이다. Focal Loss에 곱해진 모수  $\alpha_t \in [0, 1]$ 는 양성/음성 샘플의 중요도 균형을 맞추는 역할을 한다.  $\alpha_t$ 는 각 Class의 샘플 수의 역수로 정해지거나, 교차 검증을 통해 추정할 수 있다. 선행 논문에서,  $\alpha$ -balanced 변형이 기존 Focal Loss보다 더 개선되었다는 결과가 있으며, 선행논문에서는  $\alpha_t = 0.25$ ,  $\gamma = 2$  가 최적 모수 조합으로 선정되었다. 따라서 본 연구에서는 최적 모수 조합을 그대로 사용하였다.

## 2) 임계값 이동

비용 민감 학습 방법 중 가장 간단한 방법으로 임계값 이동(threshold moving)이 있다. 임계값 이동 방법은 기계학습 알고리즘이 오분류 비용을 최



소화하는 분류 경계값을 Line Search 하는 방법이다. 이 방법은 “thresholding” 또는 “threshold optimization”이라고 하며, 이진 분류상황에서 매우 일반적으로 사용된다. 본 연구에서는 불균형 자료에 대한 평가지표 중 하나인 기하 평균(Geometric Mean)을 최대화할 수 있는 임계값을 탐색한 뒤 분류 경계값으로 사용하였다.

#### 라. SMOTE

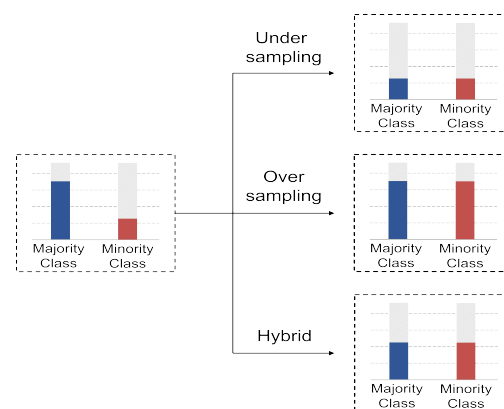


그림 3.5 데이터 수준의 접근 방법

클래스 불균형 문제를 해결하기 위한 데이터 수준의 접근 방법에는 크게 오버샘플링과 언더샘플링 및 두 가지를 결합한 방법이 있다. 이러한 방법들은 기계학습 모형의 훈련 과정에서 다수 클래스에 편향되지 않게끔 훈련 데이터의 불균형 정도를 조절하거나 잡음이 포함된 데이터를 줄인다. 즉, 기계학습 알고리즘은 그대로 사용되, 훈련 데이터 자체를 변경하는 접근법이며 이러한 방법만으로도 충분히 성능을 개선할 수 있다.

오버샘플링 방법은 훈련 데이터 중 다수 클래스에 속하는 샘플은 그대로 두고, 소수 클래스의 샘플들을 활용하여, 인위적으로 소수 클래스의 샘플을 생성하는 방법이다. 인공 소수 클래스의 데이터를 생성함으로써, 다수 클래스의 데이터와 샘플수를 소수 클래스의 샘플 수와 대략 비슷한 수로 맞추는 뒤에 기계학습 알고리즘을 적용한다.

오버샘플링 알고리즘으로는 Random Over Sampling(ROS), Synthetic Minority Over-sampling Technique(SMOTE), SMOTE for Nominal and Continuous(SMOTENC), SMOTE for Nomial(SMOTEN), Adaptive Synthetic(ADASYN), BoderlineSMOTE, KMeansSMOTE, SVMSMOTE 등이 있다. ROS는 임의복원추출을 활용하며, ROS를 제외한 오버샘플링 방법은 대부분 SMOTE[10]를 기반으로 하고 있다. 최근에는 Generative Adversarial Networks(GAN)와 같은 생성모델을 활용한 오버샘플링 방법을 활용하기도 한다.

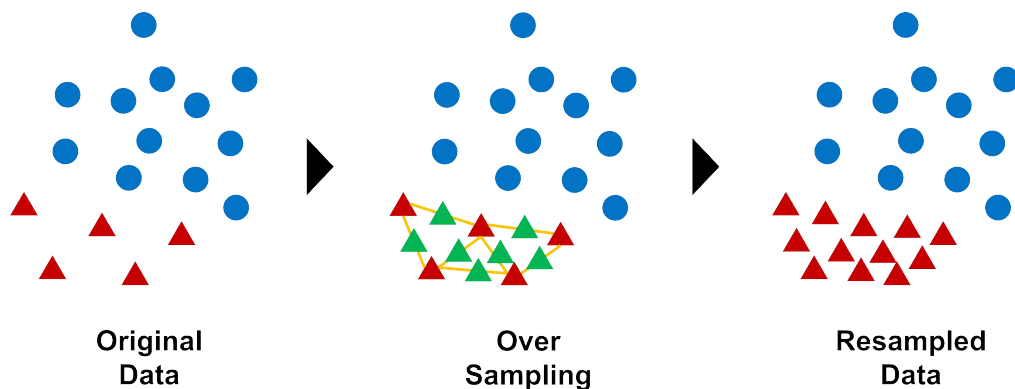


그림 3.6 SMOTE

오버샘플링의 대표적인 알고리즘이라고 할 수 있는 SMOTE는 다음과 같이 동작한다.

| SMOTE 알고리즘                               |  |
|--|--|
| SMOTE를 적용할 비율 $N$ 과, 최근접 이웃의 수 $k$ 를 선택, |  |
| $T$ = 소수 클래스의 샘플수, numattrs = 변수 개수      |  |
| 1:                                       | for $i$ in $1 \rightarrow T$                                   |
| 2:                                       | $i$ 번째 소수 클래스 데이터의 $k$ -최근접 이웃을 계산                             |
| 3:                                       | 1과 $k$ 사이의 정수를 임의로 선택 ( $t$ ) 후                                |
| 3:                                       | 2에서 계산한 $k$ 최근접 이웃 중 $t$ 번째 이웃을 선택                             |
| 4:                                       | for attr in $1 \rightarrow \text{numattrs}$                    |
| 5:                                       | 3에서 선택한 $t$ 번째 이웃과 $i$ 번째 샘플의 변수의 차이를 계산(diff) 후               |
| 5:                                       | 균등분포에서 난수 발생( $0 < \text{gap} < 1$ )                           |
| 6:                                       | $i$ 번째 샘플의 변수 값 + $\text{gap} * \text{diff}$ 로 인공 데이터의 변수 값 결정 |
| 1 ~ 6까지 원하는 비율만큼 반복하여 인공 데이터 생성          |  |

표 3.1 SMOTE 알고리즘

이처럼 오버샘플링은 소수클래스의 인공적인 데이터를 생성하는 방식으로 클래스의 균형을 맞추며, 언더샘플링과는 다르게 정보 손실이 없다는 장점이 있다. 그러나 데이터 수가 매우 많아지므로 소수 클래스에 모델이 과적합 될 가능성이 있으며, 학습시간이 더 오래걸릴 수 있다. 노이즈 또는 이상치에 민감하다는 단점도 있다.

#### 4. 실험

본 연구에서는 GEV-활성함수의 유효성을 검증하기 위하여, 100개의 데이터 셋을 활용한 실험을 진행하였다. 결과 비교를 위해, 실험에 사용된 데이터 셋은 선행논문[8]의 데이터와 같은 데이터 셋으로 진행하였다.

##### 가. 데이터 셋

본 연구에는 KEEL(Knowledge Extraction based on Evolutionary Learning) imbalanced data sets[11]가 사용되었다. 해당 데이터 셋은 4가지 부분으로 구분되어 있으며, 총 100종의 데이터 셋으로 구성된다.

| No.   | Title                                   | Number of dataset |
|-------|---|-------------------|
| 1     | Imbalanced ratio betwwen 1.5 and 9      | 22                |
| 2     | Imbalanced ratio higher than 9 – Part 1 | 22                |
| 3     | Imbalanced ratio higher than 9 – Part 2 | 22                |
| 4     | Imbalanced ratio higher than 9 – Part 3 | 34                |
| Total |   | 100               |

표 4.1 KEEL imbalanced data sets

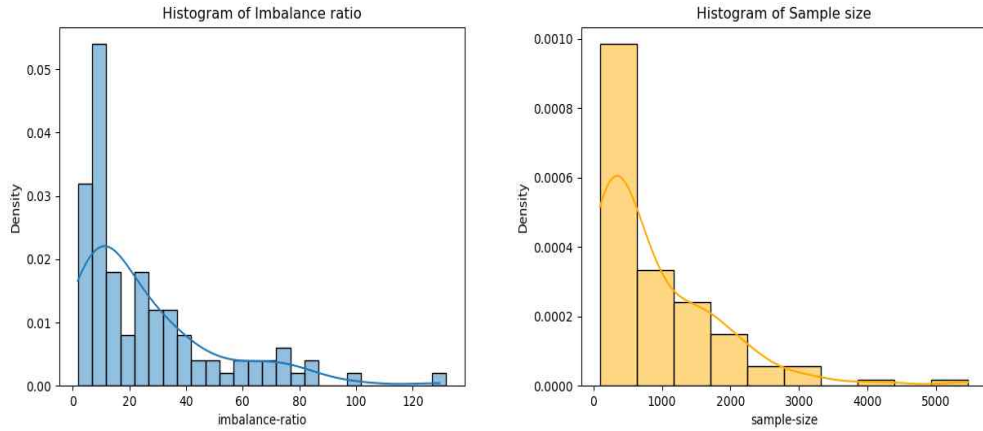


그림 4.1 KEEL imbalance data set 100종에 대한 불균형도(왼쪽)와 샘플 수(오른쪽) 분포

$$\rho = \frac{\max_i \{|C_i|\}}{\min_i \{|C_i|\}} \quad (4.1)$$

100종의 데이터 셋에 따른 불균형도( $\rho$ )와 샘플수의 분포는 그림 4.1과 같다. 불균형도는 수식 4.1과 같이 정의하였으며,  $|C_i|$ 는  $i$  번째 클래스의 샘플 수를 의미한다[12]. 이진 분류상황에서 불균형도가 1인 경우는 완벽히 균형 잡힌 데이터를 의미하며, 값이 커질수록 불균형이 심한 데이터이다. 예를 들어, 불균형도가 10인 경우 소수 클래스의 샘플 수와 다수 클래스의 샘플 수의 비는 1:10이 된다.

대부분의 데이터 셋은 불균형도가 20~30보다 작은 데이터이며, 샘플 사이즈는 대부분 2000보다 작다. 그러나 일반적으로 인공신경망 알고리즘을 사용하기 위해서는 샘플 사이즈가 큰 데이터 셋이 요구되기 때문에, 샘플링 방법 중에서도 오버샘플링을 적용하기로 결정하였다. 데이터 셋에 대한 더욱 자세한 정보는 부록1에 작성하였다.

## 나. 실험 개요

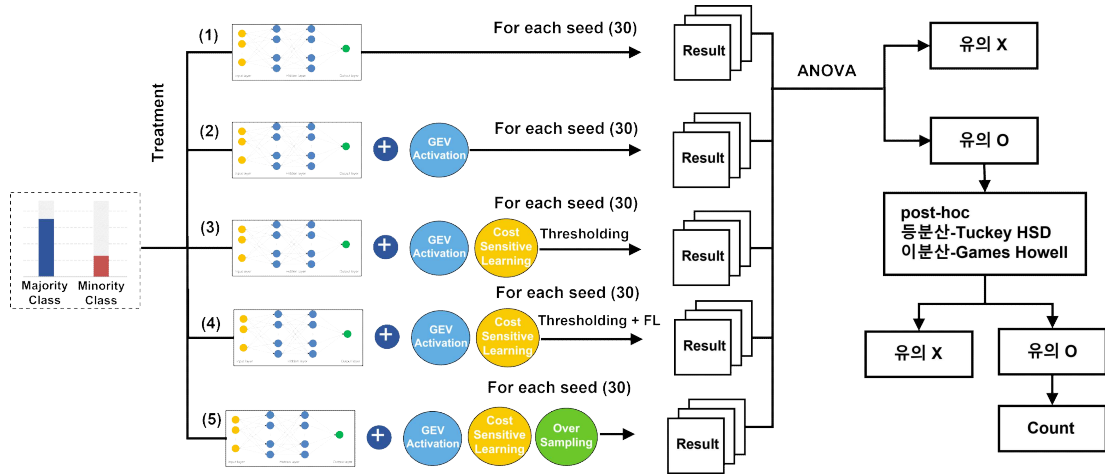


그림 4.2 실험 순서도

GEV 활성화함수의 성능 비교를 위해, 다음 5가지 경우를 적용해보고 결과를 비교하였다. 그러나 오버샘플링의 경우, 불균형도가 매우 심한 경우에만 비율을 바꿔가며 샘플링 방법을 적용할 수 있으며, 샘플 수가 매우 적은 경우에는 적용할 수 없기 때문에 모든 데이터 셋에 적용하기 어렵다. 따라서 불균형도가 70 이상이며, 샘플 수가 1000개 이상인 8개 데이터 셋에 대해서만 (5)번 방법을 적용하고 100개 데이터 셋에 대한 성능비교는 주로 (1) ~ (4)의 방법을 비교하였다.

- (1) Simoid 함수를 활용한 MLP (baseline)
- (2) GEV 함수를 활용한 MLP
- (3) GEV 함수, 임계값 이동을 활용한 MLP
- (4) GEV 함수, 임계값 이동,  $\alpha$ -balanced Focal Loss를 활용한 MLP
- (5) GEV 함수, 임계값 이동,  $\alpha$ -balanced Focal Loss, 오버샘플링(SMOTE)을 활용한 MLP

결과의 신뢰도를 위해, seed 값을 바꿔가며 (20210905 ~ 20210934) 동

일한 조건에서 30회 실험 이후, 불균형 자료에 적합한 총 5가지 평가지표 (F1-score, Geometric Mean, Area Under ROC Curve, Balanced Accuracy, Brider Inaccuracy)에 대한 실험 결과를 얻었다.

이후, 실험 결과에 일원 배치 분산분석을 적용하여, 5% 유의수준하에서 통계적으로 유의한 차이가 있는 경우를 식별하였다. 통계적으로 유의한 차이가 있는 경우에는 Levene's 등분산 검정을 통해 등분산, 이분산인 경우를 식별하고, 등분산인 경우 Tukey의 방법, 이분산인 경우 Games-Howell 방법을 적용하여 다시 5% 유의수준하에서 사후검정을 실시하였다. 최종적으로 통계적으로 유의한 차이가 있는 경우에서 평균 및 분산을 비교하였으며, 평균값이 더 큰 방법의 개수를 세어 다수가 우수하다고 판단하였다.

#### 다. 실험 환경

먼저 데이터 셋의 전체에서 훈련자료와 테스트 자료를 7:3 비율로 분할하였다. 이후 나눠진 훈련자료에서 다시 8:2 비율로 훈련자료와 검증 자료를 분할하였다. 즉, 전체 데이터를 56:14:30 의 비율로 분할하여 각각 훈련, 검증, 테스트 자료로 사용하였다. 데이터 분할시에는 층화추출 방법으로, 불균형 비율이 유지되도록 하였다. 모든 설명변수( $X$ )는 MinMaxScaling 방법을 적용하여 0과 1 사이의 값으로 표준화하였으며, 훈련에 사용된 batch size는 32로 적용하였다.

(1) ~ (3) 까지의 실험에서는 손실함수로 이진 교차 엔트로피 (Binary Cross Entropy)를 사용하였으며, (4)와 (5)의 실험에서는  $\alpha$ -balanced Focal Loss를 손실함수로 사용하였다. 최적화 알고리즘으로는 Adam(Adaptive Moment Estimation)을 활용하였으며, 학습률(Learning rate)은 0.001로 설정하였다. 인공신경망 학습을 위한 에폭은 2000으로 설정하였으며, 과적합을 방지하기 위하여 조기 종료(Early Stopping) 방법을 적용하였다.

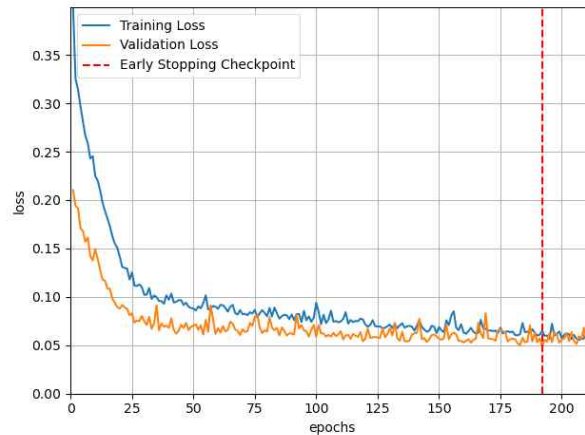


그림 4.3 조기 종료

(Data : abalone-20\_vs\_8-9-10)

조기 종료 방법으로는 가장 간단한 방법을 적용하였다. 매 에폭마다 검증 자료에 훈련된 모델을 적용하고, 손실함수의 값이 더 이상 감소하지 않으면 종료하는 방법을 적용하였다. 이때, patience는 20으로 설정하여, 연속으로 20회 이상 검증 자료의 손실 값이 감소하지 않으면 모델의 훈련을 조기 종료하였다.

오버샘플링 방법으로는 SMOTE를 적용하였으며, 샘플링 비율은 10:1부터 70:1까지 10씩 증가시키며 실험하였다.

라. 모델 구조

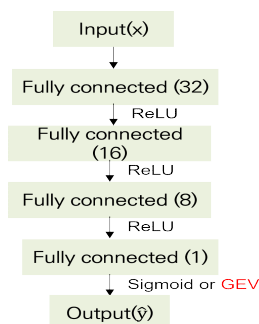


그림 4.4 실험에 사용된 신경망 구조

모델 구조로는 그림4.4와 같은 비교적 단순한 구조의 신경망을 활용하였다. 입력자료( $x$ )는 32개의 뉴런으로 구성된 완전 연결 층을 지나며, ReLU(Rectified Linear Unit)활성함수를 거친다. 위와 같은 과정을 몇 번 더 거친 뒤 가장 마지막에는 Sigmoid 함수 또는 GEV 함수를 거쳐 최종 예측값 ( $\hat{y}$ )을 생성한다. (1)의 실험에서는 시그모이드 함수를 활용하였으며, 그 밖의 실험에서는 GEV 함수를 활용하였다.

#### 마. 평가지표

최종 훈련된 모형의 평가지표로는 불균형 자료에 적합한 평가지표 5가지를 선정하였다. 선택된 평가지표는 수식 4.2 ~ 4.6과 같다.

$$F1 - score = \frac{2 \times (Recall \times Precision)}{Recall + Precision} \quad (4.2)$$

$$Geometric - Mean = \sqrt{Recall \times Specificity} \quad (4.3)$$

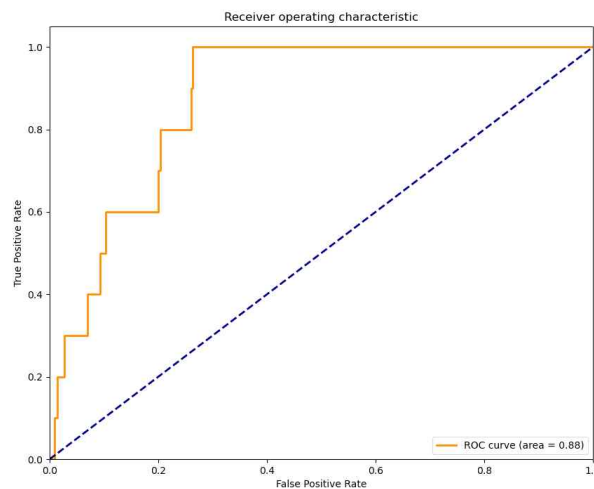


그림 4.5 ROC 곡선 (Data : abalone19)



$$Area\ Under\ the\ ROC\ Curve(AUC) = ROC\ 곡선의\ 아래\ 면적 \quad (4.4)$$

$$Balanced - Accuracy = \frac{1}{2} \times (Recall + Specificity) \quad (4.5)$$

$$Brier - Inaccuracy = \frac{1}{N} \sum_{i=1}^N \sum_{j=0}^1 [\hat{p}(c=j, x^i) - p(c=j, x^i)]^2 \quad (4.6)$$

수식 4.2 ~ 4.5까지의 평가지표들은  $[0, 1]$ 의 값을 가지며, 값이 클수록 좋음을 의미한다. AUC는  $[0.5, 1]$ 의 값을 가지며, 역시 값이 클수록 좋음을 의미한다. Brier Inaccuracy는  $[0, 2]$ 의 값을 가지며 이는 위의 다른 지표들과는 다르게 값이 작을수록 좋다. Brier Inaccuracy에서  $x^i$ 는  $i$ 번째 입력 벡터를 의미하며,  $\hat{p}$ 와  $p$ 는 모델의 클래스 예측 확률과 실제 클래스 확률을 의미한다.  $c \in \{0, 1\}$ 는 클래스의 label을 의미하며,  $N$ 은 전체 샘플 수이다.

## 5. 실험 결과

### 가. 일원 배치 분산 분석

| Data      | Method | F1-score                       | Geometric Mean                 | Area Under ROC Curve           | Balanced Accuracy              | Brier Inaccuracy             |
|-----------|--------|--------------------------------|--------------------------------|--------------------------------|--------------------------------|------------------------------|
| abalone19 | (1)    | 0.0<br>(0.0)                   | 0.0<br>(0.0)                   | <b>0.794</b><br><b>(0.084)</b> | 0.5<br>(0.0)                   | <b>0.016</b><br><b>(0.0)</b> |
|           | (2)    | 0.0<br>(0.0)                   | 0.0<br>(0.0)                   | 0.659<br>(0.143)               | 0.5<br>(0.0)                   | 0.019<br>(0.013)             |
|           | (3)    | 0.033<br>(0.021)               | 0.662<br>(0.16)                | 0.659<br>(0.143)               | 0.687<br>(0.105)               | 0.019<br>(0.013)             |
|           | (4)    | <b>0.045</b><br><b>(0.023)</b> | 0.733<br>(0.165)               | 0.76<br>(0.119)                | 0.757<br>(0.095)               | 0.037<br>(0.012)             |
|           | (5)    | 0.044<br>(0.015)               | <b>0.762</b><br><b>(0.057)</b> | 0.781<br>(0.068)               | <b>0.770</b><br><b>(0.056)</b> | 0.039<br>(0.008)             |

표 5.1 실험 방법에 따른 5가지 평가지표의 30회 결과 평균(표준편차)

(Data : abalone19, 소수점 넷째자리 반올림) 가장 높은 평균값은 굵은 글씨로 표기함

표 5.1은 abalone19 자료에 (1) ~ (5)까지 실험방법과 seed 값을 달리하며 얻은 평가지표의 30회 결과의 그룹별 평균 및 표준편차의 예시이다. 시그모이드 활성화함수를 사용한 (1)의 경우보다 최종 제안하는 방법인 (5)에서 F1-score, Geometric Mean, Balanced Accuracy 등이 값이 우수하게 나타남을 확인할 수 있다. AUC나 Brier Inaccuracy에서 (1)의 결과가 더 좋은 것으로 나타났으나, AUC의 경우 통계적으로 유의한 차이를 보이지 않으며, 특히 Geometric Mean에서 상당한 개선이 있었으므로, (5)의 결과가 (1)의 결과보다 우수하다고 판단하였다. 또한, (2)에서 (5)까지 불균형 자료를 위한 처리 방법을 추가할수록 제안하는 방법의 성능이 점차 증가하는 것이 흥미로운 부분이다. 이후 부록 2에 중요한 5가지 데이터 셋의 5가지 평가지표의 30회 결과를 제시하였다.

| Data          | metric                              | TRT      | df  | SS     | MS    | F       | PR(>F) |
|---------------|-------------------------------------|----------|-----|--------|-------|---------|--------|
| abalone<br>19 | <b>F1-score</b>                     | method   | 4   | 0.062  | 0.016 | 67.184  | 0      |
|               |                                     | Residual | 145 | 0.034  | 0     |         |        |
|               | <b>Geometric<br/>Mean</b>           | method   | 4   | 18.763 | 4.691 | 416.808 | 0      |
|               |                                     | Residual | 145 | 1.632  | 0.011 |         |        |
|               | <b>Area<br/>Under<br/>ROC Curve</b> | method   | 4   | 0.529  | 0.132 | 9.885   | 0      |
|               |                                     | Residual | 145 | 1.941  | 0.013 |         |        |
|               | <b>Balanced<br/>Accuracy</b>        | method   | 4   | 2.159  | 0.54  | 116.54  | 0      |
|               |                                     | Residual | 145 | 0.672  | 0.005 |         |        |
|               | <b>Brier<br/>Inaccuracy</b>         | method   | 4   | 0.015  | 0.004 | 35.148  | 0      |
|               |                                     | Residual | 145 | 0.015  | 0     |         |        |

표 5.2 분산분석표 (Data : abalone19, 소수점 넷째자리 반올림)

통계적으로 유의한 차이가 있는 경우는 굵은 글씨로 표기함

전체 100개의 데이터 셋에서 5개의 평가지표에 대한 (1) ~ (5)의 실험 결과에 대해, 일원 배치 분산분석을 수행하였다. 데이터 셋과 평가지표에 따른 전체 500개 조합 중 309개, 즉 전체의 61.8%가 5% 유의수준하에서 유의한 차이가 존재하였다. 표 5.2는 abalone19 데이터를 활용한 분산분석표의 예시이며, 나머지 중요한 분산분석표는 부록 3에 포함하였다.

| Data      | Metric                      | Statistic     | p-value      |
|-----------|-----------------------------|---------------|--------------|
| abalone19 | <b>F1-score</b>             | <b>20.17</b>  | <b>0</b>     |
|           | <b>Geometric Mean</b>       | <b>8.157</b>  | <b>0</b>     |
|           | <b>Area Under ROC Curve</b> | <b>4.782</b>  | <b>0.001</b> |
|           | <b>Balanced Accuracy</b>    | <b>17.296</b> | <b>0</b>     |
|           | Brier Inaccuracy            | 1.85          | 0.122        |

표 5.3 Levene의 등분산 검정 결과 (Data : abalone19, 소수점 넷째자리 반올림)  
귀무가설을 기각하고 이분산으로 판단되는 경우는 굵은 글씨로 표기함

| Data      | Group1 | Group2 | Metric                   | p-adj        | Method       |
|-----------|--------|--------|--------------------------|--------------|--------------|
| abalone19 | (5)    | (1)    | Area Under the ROC Curve | 0.900        | Games-Howell |
|           | (5)    | (1)    | <b>Brier Inaccuracy</b>  | <b>0.001</b> | tukeyHSD     |

표 5.4 사후검정 결과 (Data : abalone19, 소수점 넷째자리 반올림)  
통계적으로 유의한 차이가 있는 경우는 굵은 글씨로 표기함

유의한 차이가 존재하는 조합에 5% 유의수준하에서 Levene의 등분산 검정을 수행하였다. 중요한 5개 데이터 셋에 대한 등분산 검정 결과는 부록 4에 포함하였다. 등분산인 경우 Tukey의 HSD(honestly significant difference) 검정을, 이분산인 경우 Games-Howell의 사후검정을 수행하였다. 같은 데이터에 대한 사후검정 결과는 부록 5에 포함하였다. 결과의 분산이 0이므로 p-value를 구하지 못하여, 사후검정을 수행하지 못 하는 경우에는 단순히 평균값을 비교하여 평균이 높은 쪽을 우수하다고 판단하였다. 사후검정 결과 전체 500개 조합 중 162개 즉, 전체의 32.4%가 5% 유의수준하에서 유의한 차이가 존재하였다.

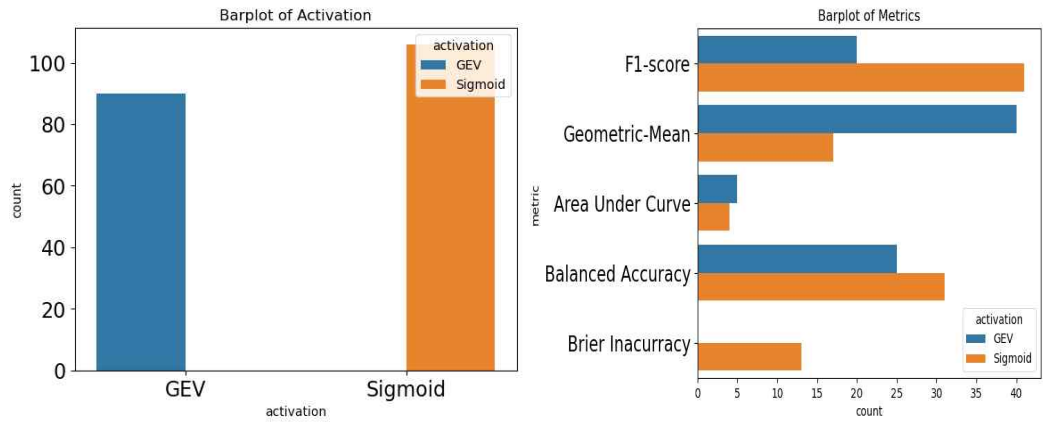


그림 5.1 일원 배치 분산분석 및 사후검정 결과 우수한 방법의 개수  
(왼쪽: 전체, 오른쪽: 지표별)

| 활성함수    | F1-score | Geometric Mean | Area Under the ROC Curve | Balanced Accuracy | Brier Inaccuracy | 합계  |
|---------|----------|----------------|--------------------------|-------------------|------------------|-----|
| Sigmoid | 32       | 0              | 3                        | 1                 | 24               | 60  |
| GEV     | 7        | 44             | 0                        | 50                | 1                | 102 |

표 5.5 일원 배치 분산분석 및 사후분석 결과 우수한 방법의 개수

전체적으로는 제안하는 방법이 시그모이드 함수(실험 (1))보다 우수하게 나타났으며, 특히 Geometric-Mean과 Balanced-Accuracy에서 제안하는 방법이 우수하게 나타났다.

#### 나. 오버샘플링을 적용한 실험 결과

| 불균형도   | F1-score            | Geometric-Mean      | Area Under Curve    | Balanced Accuracy   | Brier Inaccuracy    |
|--------|---------------------|---------------------|---------------------|---------------------|---------------------|
| 10.035 | 0.041(0.016)        | 0.729(0.150)        | 0.769(0.081)        | 0.753(0.073)        | 0.049(0.023)        |
| 20.157 | 0.044(0.015)        | <b>0.762(0.057)</b> | <b>0.781(0.068)</b> | <b>0.770(0.056)</b> | <b>0.039(0.008)</b> |
| 30.104 | 0.044(0.017)        | 0.734(0.154)        | 0.767(0.094)        | 0.757(0.083)        | 0.042(0.014)        |
| 40.667 | 0.043(0.019)        | 0.723(0.164)        | 0.751(0.126)        | 0.744(0.101)        | 0.042(0.014)        |
| 50.391 | 0.041(0.017)        | 0.722(0.157)        | 0.751(0.107)        | 0.746(0.088)        | 0.042(0.014)        |
| 61.000 | <b>0.046(0.018)</b> | 0.731(0.155)        | 0.767(0.099)        | 0.756(0.084)        | 0.041(0.015)        |
| 70.242 | 0.042(0.019)        | 0.722(0.160)        | 0.743(0.127)        | 0.747(0.089)        | 0.039(0.013)        |

표 5.6 오버샘플링 비율에 따른 실험 결과의 평균(표준편차) (소수점 넷째자리 반올림)

(Data : abalone19), 가장 좋은 결과는 굵은 글씨로 표기

불균형도가 70 이상이며, 샘플 사이즈가 1000이상인 8개 데이터 셋에 대해서는 오버 샘플링의 비율(10:1 ~ 70:1)과 seed 값(20210905 ~ 20210934)을 바꿔가며 5가지 평가지표에 대한 결과를 기록하였다. 이후, 30회 결과의 평균과 표준편차를 비교하였다. 데이터 셋에 따라 최적 샘플링비율은 다르게 나타났으나, 대부분의 데이터 셋에서 불균형도가 20~30일 때, 제안하는 방법의 성능이 가장 좋은 것으로 나타났다. 8개 데이터 셋 중 중요한 5개 데이터 셋에 대한 결과는 부록 6에 기록하였다.

#### 다. SOTA 모델과의 비교

Munkhdalai et al.[8]은 그들이 개발한 GEV-NN이라는 인공신경망 모형과 KEEL imbalance data set을 사용했던 선행 연구의 결과들을 비교하였다. 먼저 Fernández et al.[13]의 유전 알고리즘 기반의 기계학습 알고리즘들의 성능과 비교하였으며, 33개의 벤치마크 데이터셋에서 Geometric mean의 값이 22개 base line 모델보다 우수한 성능을 보였다. 또한 최근 제안된 contrast pattern based classifiers[14, 15]와 fuzzy rule-based oversampling technique[16]과 비교하였다. 그들이 제안한 GEV-NN은 contrast pattern-based classifiers(PBC4cip, etc), 샘플링 방법들의 조합, 다른 SOTA(State-of-the-art) 모델보다 우수하다고 주장했다. 따라서 GEV-NN을 SOTA 모델로 간주하고, 같은 데이터 셋에 대한 GEV-NN의 결과(Geometric-Mean, AUC)와 본 연구에서 제안한 모델의 결과의 평균을 직접 비교한 결과를 표 5.7에 제시하였다. 제안한 모델은 실험(5)에 해당하는 결과이며, 불균형도가 20일때를 기준으로 결과를 제시하였다.

| Data                    | GM(Proposed) | GM(GEV-NN) | AUC(Proposed) | AUC(GEV-NN) |
|-------------------------|--------------|------------|---------------|-------------|
| abalone19               | <b>0.762</b> | 0.7247     | <b>0.781</b>  | 0.7419      |
| abalone-20_vs_8-9-10    | <b>0.908</b> | 0.884      | <b>0.935</b>  | 0.9009      |
| kr-vs-k-zero_vs_fifteen | <b>1</b>     | <b>1</b>   | <b>1</b>      | <b>1</b>    |
| poker-8_vs_6            | <b>0.998</b> | 0.9714     | <b>0.999</b>  | 0.966       |
| poker-8-9_vs_5          | <b>0.752</b> | 0.5165     | <b>0.719</b>  | 0.408       |

표 5.7 제안된 방법(소수점 넷째자리 반올림)과 GEV-NN(SOTA) 모델과의 비교

## 6. 결론 및 제언

### 가. 요약

본 연구는 클래스 불균형 문제를 해결하고, 분류 성능 개선 방안을 제시하였다. GEV-활성함수와 비용 민감 학습( $\alpha$ -balanced Focal Loss, Thresholding) 및 오버샘플링(SMOTE)을 결합한 인공신경망을 최적 모델로 제시하였다.

모델의 유효성을 검증하기 위해 불균형도가 각기 다른 100종의 데이터 셋(KEEL-imbalance data set)을 사용하여 5가지 평가지표의 실험 결과에 대해 비교하였으며, 결과의 신뢰도를 위해 seed값을 바꿔가며 불균형 자료에 적합한 5가지 평가지표에 대한 결과를 30회 획득하였다. 30회 실험 결과에 일원 배치 분산분석을 수행하여, 5% 유의수준하에서 통계적으로 평균에 차이가 있는 경우를 식별하였으며, 차이가 있는 경우에는 5% 유의수준하에서 Leven의 등분산 검정을 수행하고, 등분산인 경우, Tukey's HSD를, 이분산인 경우 Games-Howell 방법을 사용하여 (1)과 (4) 또는 (5)를 사후검정하였다. 시그모이드 함수와 비교하였을 때 전반적으로 제안한 방법이 효과적인 것으로 나타났다. 특히 Geometric-Mean과 Balanced Accuracy 평가지표에서 효과적인 것으로 나타났다.

또한, 최적 샘플링 비율을 제시하기 위해 불균형도가 70 이상이며, 샘플 사이즈가 1000 이상인 8개 데이터 셋에 대해 (5)번 방법을 적용하여 5가지 평가지표의 평균 및 표준편차를 제시하였으며, 불균형도가 20:1이 될 수 있도록 샘플링 하는 방법이 효과적인 것으로 나타났다.

결과적으로 5가지 지표를 종합적으로 고려하였을 때, 제안하는 방법이 기존 방법(Sigmoid) 또는 선행 논문의 방법(Gumbel) 보다 우수하게 나타났다.

## 나. 한계 및 향후 연구 과제

본 연구는 KEEL 불균형 데이터 셋을 활용한 toy model 실험을 거쳤으나, 강수량 자료, 금융 자료 등 실제 자료에 적용하지 않았다. 따라서 실제 자료에 적용하여 성능을 검증하는 과정이 필요할 것으로 보인다.

또한, 평가지표에 따라 제안하는 방법의 우수성이 다르게 나타났다. 현재는 통계적으로 유의한 차이가 있는 경우, 단순히 평균이 더 높은 경우의 개수를 세는 방식이었으나, 평가지표들의 가중평균과 같은 모델과 데이터의 특성을 고려한 비교 방법이 필요할 것으로 생각된다.

$\alpha$ -balanced Focal Loss의 경우,  $\alpha$ 와  $\gamma$ 를 선택해야 하는 문제점이 있다. 그러나 이 연구에서는 2개의 모수를 선택하지 않고, 선행논문에서 최적값으로 제안한  $\alpha = 0.25$ ,  $\gamma = 2.0$ 을 그대로 사용하였다. 위의 값을 교차검증 등을 통해 최적값으로 교체한다면, 현재보다 더 성능을 높일 수 있을 것으로 기대한다.

오버샘플링의 경우에는 최적 비율이 20:1 정도로 나타났으나, (1) 제시한 비율보다 데이터가 더 불균형해야만 오버샘플링을 적용할 수 있으며, (2) 소수 클래스의 샘플 수가 매우 작아 k-최근접 이웃도 선택하지 못한다면, 적용이 어렵다. (3) 선행논문[10]에 따르면, 오버샘플링과 언더샘플링을 함께 적용할 때 결과가 더 좋다고 한다. 따라서 SMOTE와 다른 언더샘플링 방법을 함께 적용하면 결과가 더 좋을 수 있을 것으로 기대한다. Van Hulse et al.[17]은 7개의 샘플링 방법과 11개의 기계학습 방법의 조합을 이용하여 결과를 비교하였으며, 어떤 샘플링 방법도 모든 데이터 셋에 대해서 일관되게 성능이 우수한 경우는 없었다. 따라서 SMOTE가 아닌 다른 샘플링 방법을 적용했을 때, 더 성능을 향상시킬 수 있을 것으로 기대한다.

마지막으로 현재는 이진 분류의 경우에만 위 방법을 적용하였으나, 향후에는 다항 분류문제까지로도 쉽게 확장이 가능할 것으로 보인다.

## 참고문헌 (References)

1. Zhou, L., Pan, S., Wang, J., & Vasilakos, A. V. (2017). Machine learning on big data: Opportunities and challenges. *Neurocomputing*, 237, 350-361.
2. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 127). New York: springer.
3. Sun, Y., Wong, A. K., & Kamel, M. S. (2009). Classification of imbalanced data: A review. *International journal of pattern recognition and artificial intelligence*, 23(04), 687-719.
4. Johnson, J. M., & Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1), 1-54.
5. Bridge, J., Meng, Y., Zhao, Y., Du, Y., Zhao, M., Sun, R., & Zheng, Y. (2020). Introducing the GEV activation function for highly unbalanced data to develop COVID-19 diagnostic models. *IEEE journal of Biomedical and Health Informatics*, 24(10), 2776-2786.
6. Coles, S., Bawa, J., Trenner, L., & Dorazio, P. (2001). An introduction to statistical modeling of extreme values (Vol. 208, p. 208). London: Springer.
7. Wang, X., & Dey, D. K. (2010). Generalized extreme value regression for binary response data: An application to B2B electronic payments system adoption. *The Annals of Applied Statistics*, 4(4), 2000-2023.
8. Munkhdalai, L., Munkhdalai, T., & Ryu, K. H. (2020). GEV-NN: A deep neural network architecture for class imbalance problem in binary classification. *Knowledge-Based Systems*, 194, 105534.
9. Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 2980-2988).



10. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16, 321-357.
11. Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., & Herrera, F. (2011). Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic & Soft Computing*, 17.
12. Buda, M., Maki, A., & Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106, 249-259.
13. Fernández, A., García, S., Luengo, J., Bernadó-Mansilla, E., & Herrera, F. (2010). Genetics-based machine learning for rule induction: state of the art, taxonomy, and comparative study. *IEEE Transactions on Evolutionary Computation*, 14(6), 913-941.
14. Loyola-González, O., Martínez-Trinidad, J. F., Carrasco-Ochoa, J. A., & García-Borroto, M. (2016). Study of the impact of resampling methods for contrast pattern based classifiers in imbalanced databases. *Neurocomputing*, 175, 935-947.
15. Loyola-González, O., Medina-Pérez, M. A., Martínez-Trinidad, J. F., Carrasco-Ochoa, J. A., Monroy, R., & García-Borroto, M. (2017). PBC4cip: A new contrast pattern-based classifier for class imbalance problems. *Knowledge-Based Systems*, 115, 100-109.
16. Liu, G., Yang, Y., & Li, B. (2018). Fuzzy rule-based oversampling technique for imbalanced and incomplete data learning. *Knowledge-Based Systems*, 158, 154-174.
17. Van Hulse, J., Khoshgoftaar, T. M., & Napolitano, A. (2007, June). Experimental perspectives on learning from imbalanced data. In *Proceedings of the 24th international conference on Machine learning* (pp. 935-942).

# GEV activation function for classification of imbalanced data

Juyoung Hong

Department of Mathematics and Statistics  
Graduate School Chonnam National University  
(Supervised by Professor Jeong-Soo Park)

(Abstract)

The class imbalance problem refers to a case in which a specific class(Majority class) appears very frequently compared to other classes(Minority class), and occurs very frequently in most real life situations such as medical diagnosis, fire detection, and fraudulent transaction detection. When a traditional classification algorithm is applied to such imbalanced data, it tends to ignore the minority class to improve accuracy. However, in most cases, minority classes are our interest. So, it is important to classify them well rather than their overall accuracy. Therefore, this study proposes a method that combines the GEV-activation function with  $\alpha$ -balanced Focal Loss, Thresholding, and Over-sampling(SMOTE) as a solution to the above problem. In order to check the effectiveness of the proposed method, 100 data sets were used and 5 evaluation metrics were compared, and one-way ANOVA and post-hoc tests were performed under the 5% significance level. 162 out of 500 combinations of data sets and evaluation metrics, that is, 32.4% of total showed a significant difference under the 5% significance level, and

Over all, the proposed method was superior to that using the sigmoid function. (102:60) The optimal sampling ratio is judged to be 20:1, and when we compared the results of proposed method after sampling and SOTA model, excellent results were obtained in all five data sets.

Key words : Class imbalance, Generalized Extreme Value Distribution, Activation Function, etc

부록(Appendix)

부록 1. KEEL imbalance data sets

| Data                      | 불균형도<br>( $\rho$ ) | Sample<br>Size | 설명<br>변수<br>개수 | 비고   |
|---------------------------|--------------------|----------------|----------------|--|
| abalone-17_vs_7-8-9-10    | 39.31              | 2,338          | 8              | Abalone data set<br>(Pos : 17, Neg : 7-8-9-10)       |
| abalone19                 | 129.44             | 4,174          | 8              | Abalone data set<br>(Pos : 19, Neg : rest)           |
| abalone-19_vs_10-11-12-13 | 49.69              | 1,622          | 8              | Abalone data set<br>(Pos : 19, Neg : 10-11-12-13)    |
| abalone-20_vs_8-9-10      | 72.69              | 1,916          | 8              | Abalone data set<br>(Pos : 20, Neg : 8-9-10)         |
| abalone-21_vs_8           | 40.50              | 581            | 8              | Abalone data set<br>(Pos : 21, Neg : 8)              |
| abalone-3_vs_11           | 32.47              | 502            | 8              | Abalone data set<br>(Pos : 3, Neg : 11)              |
| abalone9-18               | 16.40              | 731            | 8              | Abalone data set<br>(Pos : 9, Neg : 18)              |
| car-good                  | 24.04              | 1,728          | 14             | Car Evaluation data set<br>(Pos : good, Neg : rest)  |
| car-vgood                 | 25.58              | 1,728          | 14             | Car Evaluation data set<br>(Pos : vgood, Neg : rest) |
| cleveland-0_vs_4          | 12.31              | 173            | 12             | Cleveland data set<br>(Pos : 0, Neg : 4)             |
| dermatology-6             | 16.90              | 358            | 33             | Dermatology data set<br>(Pos : 6, Neg : rest)        |
| ecoli-0_vs_1              | 34.75              | 220            | 6              | Ecoli data set<br>(Pos : im, Neg : cp)               |
| ecoli-0-1_vs_2-3-5        | 9.17               | 244            | 6              | Ecoli data set<br>(Pos:cp,im, Neg:imS,imL,om)        |
| ecoli-0-1_vs_5            | 11.00              | 240            | 5              | Ecoli data set<br>(Pos : cp, Neg : om)               |
| ecoli-0-1-3-7_vs_2-6      | 39.14              | 281            | 6              | Ecoli data set<br>(Pos:pp,imL, Neg:cp,im,imU,imS)    |
| ecoli-0-1-4-6_vs_5        | 13.00              | 280            | 5              | Ecoli data set<br>(Pos : cp, im, imU, Neg : om)      |

|                          |       |       |    |  |
|--------------------------|-------|-------|----|--|
| ecoli-0-1-4-7_vs_2-3-5-6 | 10.59 | 336   | 6  | Ecoli data set<br>(Pos:cp,im,imU,pp<br>Neg:imS,imL,om,omL) |
| ecoli-0-1-4-7_vs_5-6     | 12.28 | 332   | 5  | Ecoli data set<br>(Pos:cp,im,imU, Neg:om,omL)              |
| ecoli-0-2-3-4_vs_5       | 9.10  | 202   | 6  | Ecoli data set<br>(Pos:cp,imS,imL,imU, Neg:om)             |
| ecoli-0-2-6-7_vs_3-5     | 9.18  | 224   | 6  | Ecoli data set<br>(Pos:cp,inS,omL,pp, Neg:imL,om)          |
| ecoli-0-3-4_vs_5         | 9.00  | 200   | 6  | Ecoli data set<br>(Pos:cp,imL,imU, Neg:om)                 |
| ecoli-0-3-4-6_vs_5       | 9.25  | 205   | 6  | Ecoli data set<br>(Pos:cp,imL,imU,omL, Neg:om)             |
| ecoli-0-3-4-7_vs_5-6     | 9.28  | 257   | 6  | Ecoli data set<br>(Pos:cp,imL,imU,pp, Neg:om,omL)          |
| ecoli-0-4-6_vs_5         | 9.15  | 203   | 5  | Ecoli data set<br>(Pos:cp,imU,omL, Neg:om)                 |
| ecoli-0-6-7_vs_3-5       | 9.09  | 222   | 6  | Ecoli data set<br>(Pos:cp,omL,pp, Neg:imL,om)              |
| ecoli-0-6-7_vs_5         | 10.00 | 220   | 5  | Ecoli data set<br>(Pos:cp,omL,pp, Neg:om)                  |
| ecoli1                   | 63.75 | 336   | 6  | Ecoli data set<br>(Pos : im, Neg : rest)                   |
| ecoli2                   | 70.00 | 336   | 6  | Ecoli data set<br>(Pos : pp, Neg : rest)                   |
| ecoli3                   | 8.60  | 336   | 6  | Ecoli data set<br>(Pos : imU, Neg : rest)                  |
| ecoli4                   | 15.80 | 336   | 6  | Ecoli data set<br>(Pos : om, Neg : rest)                   |
| flare-F                  | 23.79 | 1,066 | 16 | Solar Flare data set<br>(Pos : F, Neg : rest)              |
| glass0                   | 2.06  | 214   | 8  | Glass data set<br>(Pos : 0, Neg : rest)                    |
| glass-0-1-2-3_vs_4-5-6   | 3.20  | 214   | 8  | Glass data set<br>(Pos : 0,1,2,3, Neg : 4,5,6)             |
| glass-0-1-4-6_vs_2       | 11.06 | 205   | 8  | Glass data set<br>(Pos : 1,2,5,7, Neg : 3)                 |
| glass-0-1-5_vs_2         | 9.12  | 172   | 8  | Glass data set<br>(Pos : 1,2,6, Neg : 3)                   |

|                                 |        |       |    |  |
|---------------------------------|--------|-------|----|--|
| glass-0-1-6_vs_2                | 10.29  | 192   | 8  | Glass data set<br>(Pos : 2, Neg : 0,1,6)                 |
| glass-0-1-6_vs_5                | 19.44  | 184   | 8  | Glass data set<br>(Pos : 5, Neg : 0,1,6)                 |
| glass-0-4_vs_5                  | 9.22   | 92    | 8  | Glass data set<br>(Pos : 1,5, Neg : 6)                   |
| glass-0-6_vs_5                  | 11.00  | 108   | 8  | Glass data set<br>(Pos : 1,7, Neg : 6)                   |
| glass1                          | 1.82   | 214   | 8  | Glass data set<br>(Pos : 1, Neg : rest)                  |
| glass2                          | 11.59  | 214   | 8  | Glass data set<br>(Pos : 2, Neg : rest)                  |
| glass4                          | 15.46  | 214   | 8  | Glass data set<br>(Pos : 4, Neg : rest)                  |
| glass5                          | 22.78  | 214   | 8  | Glass data set<br>(Pos : 5, Neg : rest)                  |
| glass6                          | 6.38   | 214   | 8  | Glass data set<br>(Pos : 6, Neg : rest)                  |
| haberman                        | 2.78   | 306   | 2  | Haberman data set<br>(Pos, Neg)                          |
| iris0                           | 2.00   | 150   | 3  | Iris data set<br>(Pos:setosa, Neg:versicolor,virginica)  |
| kddcup-buffer_overflow_vs_back  | 73.43  | 2,233 | 44 | KDD Cup data set<br>(Pos:buffer_overflow, Neg:back)      |
| kddcup-guess_passwd_vs_satan    | 29.98  | 1,642 | 67 | KDD Cup data set<br>(Pos:guess_passwd, Neg:satan)        |
| kddcup-land_vs_portsweep        | 49.52  | 1,061 | 78 | KDD Cup data set<br>(Pos:land, Neg:portsweep)            |
| kddcup-land_vs_satan            | 75.67  | 1,610 | 64 | KDD Cup data set<br>(Pos:land, Neg:satan)                |
| kddcup-rootkit-ima<br>p_vs_back | 100.14 | 2,225 | 48 | KDD Cup data set<br>(Pos:rootkit-ima<br>p, Neg:back)     |
| kr-vs-k-one_vs_fif<br>teen      | 27.77  | 2,244 | 19 | Chess - King-Rook vs King<br>(Pos : one, Neg : fifteen)  |
| kr-vs-k-three_vs_e<br>leven     | 35.23  | 2,935 | 19 | Chess - King-Rook vs King<br>(Pos : three, Neg : eleven) |
| kr-vs-k-zero_vs_ei<br>ght       | 53.07  | 1,460 | 19 | Chess - King-Rook vs King<br>(Pos : zero, Neg : eight)   |

|                                |       |       |    |   |
|--------------------------------|-------|-------|----|---|
| kr-vs-k-zero_vs_fifteen        | 80.22 | 2,193 | 19 | Chess - King-Rook vs King<br>(Pos : zero, Neg : fifteen)              |
| kr-vs-k-zero-one_vs_draw       | 26.63 | 2,901 | 19 | Chess - King-Rook vs King<br>(Pos : zero-one, Neg : draw)             |
| led7digit-0-2-4-5-6-7-8-9_vs_1 | 10.97 | 443   | 6  | LED Display Domain data set<br>(Pos : 0,2,4,5,6,7,8,9, Neg : 1)       |
| lymphography-normal-fibrosis   | 23.67 | 148   | 31 | Lymphography data set<br>(Pos:normal-fibrosis, Neg:rest)              |
| new-thyroid1                   | 5.14  | 215   | 4  | Thyroid Disease<br>(Pos, Neg)   |
| newthyroid2                    | 5.14  | 215   | 4  | Thyroid Disease<br>(Pos:hypothyroidism, Neg:rest)                     |
| page-blocks0                   | 8.79  | 5,472 | 9  | Page Blocks data set<br>(Pos : rest, Neg : text)                      |
| page-blocks-1-3_vs_4           | 15.86 | 472   | 9  | Page Blocks data set<br>(Pos : graphic, Neg :<br>horiz.line, picture) |
| pima                           | 1.87  | 768   | 7  | Pima data set<br>(Pos, Neg)   |
| poker-8_vs_6                   | 85.88 | 1,477 | 9  | Poker Hand data set<br>(Pos : 8, Neg : 6)                             |
| poker-8-9_vs_5                 | 82.00 | 2,075 | 9  | Poker Hand data set<br>(Pos : 8-9, Neg : 5)                           |
| poker-8-9_vs_6                 | 58.40 | 1,485 | 9  | Poker Hand data set<br>(Pos : 8-9, Neg : 6)                           |
| poker-9_vs_7                   | 29.50 | 244   | 9  | Poker Hand data set<br>(Pos : 9, Neg : 7)                             |
| segment0                       | 6.02  | 2,308 | 18 | Image Segmentation data set<br>(Pos : 1, Neg : rest)                  |
| shuttle-2_vs_5                 | 66.67 | 3,316 | 8  | Statlog - Shuttle data set<br>(Pos : 2, Neg : 5)                      |
| shuttle-6_vs_2-3               | 22.00 | 230   | 8  | Statlog - Shuttle data set<br>(Pos : 6, Neg : 2-3)                    |
| shuttle-c0-vs-c4               | 13.87 | 1,829 | 8  | Statlog - Shuttle data set<br>(Pos : 0, Neg : 4)                      |
| shuttle-c2-vs-c4               | 20.50 | 129   | 8  | Statlog - Shuttle data set<br>(Pos : 2, Neg : 4)                      |
| vehicle0                       | 3.25  | 846   | 17 | Vehicle Silhouettes data set<br>(Pos : Van, Neg : rest)               |

|                                |       |       |    |  |
|--------------------------------|-------|-------|----|--|
| vehicle1                       | 1.97  | 846   | 17 | Vehicle Silhouettes data set<br>(Pos : Saab, Neg : rest)         |
| vehicle2                       | 2.17  | 846   | 17 | Vehicle Silhouettes data set<br>(Pos : Bus, Neg : rest)          |
| vehicle3                       | 1.97  | 846   | 17 | Vehicle Silhouettes data set<br>(Pos : Opel, Neg : rest)         |
| vowel0                         | 9.98  | 988   | 12 | Vowel data set<br>(Pos : 0, Neg : rest)                          |
| winequality-red-3_<br>vs_5     | 68.10 | 691   | 10 | Red Wine Quality data set<br>(Pos : 3, Neg : 5)                  |
| winequality-red-4              | 29.17 | 1,599 | 10 | Red Wine Quality data set<br>(Pos : 4, Neg : rest)               |
| winequality-red-8_<br>vs_6     | 35.44 | 656   | 10 | Red Wine Quality data set<br>(Pos : 8, Neg : 6)                  |
| winequality-red-8_<br>vs_6-7   | 46.50 | 855   | 10 | Red Wine Quality data set<br>(Pos : 8, Neg : 6-7)                |
| winequality-white-<br>3_vs_7   | 44.00 | 900   | 10 | White Wine Quality data set<br>(Pos : 3, Neg : 7)                |
| winequality-white-<br>3-9_vs_5 | 58.28 | 1,482 | 10 | White Wine Quality data set<br>(Pos : 3-9, Neg : 5)              |
| winequality-white-<br>9_vs_4   | 32.60 | 168   | 10 | White Wine Quality data set<br>(Pos : 9, Neg : 4)                |
| wisconsin                      | 1.86  | 683   | 8  | Wisconsin data set<br>(Pos, Neg)                                 |
| yeast-0-2-5-6_vs_<br>3-7-8-9   | 9.14  | 1,004 | 7  | Yeast data set<br>(Pos:MIT,CYT,ME3,EXC<br>Neg:ME1,VAC,POX,ERL)   |
| yeast-0-2-5-7-9_<br>vs_3-6-8   | 9.14  | 1,004 | 7  | Yeast data set<br>(Pos:MIT,CYT,ME3,VAC,<br>ERL, Neg:ME1,EXC,POX) |
| yeast-0-3-5-9_vs_<br>7-8       | 9.12  | 506   | 7  | Yeast data set<br>(Pos:MIT,ME1,ME3,ERL,<br>Neg:VAC,POX)          |
| yeast-0-5-6-7-9_<br>vs_4       | 9.35  | 528   | 7  | Yeast data set<br>(Pos:ME2,<br>Neg:MIT,ME3,EXC,VAC,ERL)          |
| yeast1                         | 2.46  | 1,484 | 7  | Yeast data set<br>(Pos:NUC, Neg:rest)                            |



|                    |       |       |    |   |
|--------------------|-------|-------|----|---|
| yeast-1_vs_7       | 14.30 | 459   | 6  | Yeast data set<br>(Pos:VAC, Neg:NUC)                |
| yeast-1-2-8-9_vs_7 | 30.57 | 947   | 7  | Yeast data set<br>(Pos:VAC,<br>Neg:NUC,CYT,POX,ERL) |
| yeast-1-4-5-8_vs_7 | 22.10 | 693   | 7  | Yeast data set<br>(Pos:VAC,<br>Neg:NUC,ME2,ME3,POX) |
| yeast-2_vs_4       | 9.08  | 514   | 7  | Yeast data set<br>(Pos:CYT, Neg:ME2)                |
| yeast-2_vs_8       | 23.10 | 482   | 7  | Yeast data set<br>(Pos:CYT, Neg:ME2)                |
| yeast3             | 8.10  | 1,484 | 7  | Yeast data set<br>(Pos:ME3, Neg:rest)               |
| yeast4             | 28.10 | 1,484 | 7  | Yeast data set<br>(Pos:ME2, Neg:rest)               |
| yeast5             | 32.73 | 1,484 | 7  | Yeast data set<br>(Pos:ME1, Neg:rest)               |
| yeast6             | 41.40 | 1,484 | 7  | Yeast data set<br>(Pos:EXC, Neg:rest)               |
| zoo-3              | 19.20 | 101   | 15 | Zoo data set<br>(Pos : 3, Neg : rest)               |

## 부록2. 평균 및 분산

가장 높은 평균값을 갖는 경우는 굵은 글씨로 표기함

| Data      | Method | F1-score                       | Geometric<br>Mean              | Area<br>Under<br>ROC Curve     | Balanced<br>Accuracy           | Brier<br>Inaccuracy          |
|-----------|--------|--------------------------------|--------------------------------|--------------------------------|--------------------------------|------------------------------|
| abalone19 | (1)    | 0.0<br>(0.0)                   | 0.0<br>(0.0)                   | <b>0.794</b><br><b>(0.084)</b> | 0.5<br>(0.0)                   | <b>0.016</b><br><b>(0.0)</b> |
|           | (2)    | 0.0<br>(0.0)                   | 0.0<br>(0.0)                   | 0.659<br>(0.143)               | 0.5<br>(0.0)                   | 0.019<br>(0.013)             |
|           | (3)    | 0.033<br>(0.021)               | 0.662<br>(0.16)                | 0.659<br>(0.143)               | 0.687<br>(0.105)               | 0.019<br>(0.013)             |
|           | (4)    | <b>0.045</b><br><b>(0.023)</b> | 0.733<br>(0.165)               | 0.76<br>(0.119)                | 0.757<br>(0.095)               | 0.037<br>(0.012)             |
|           | (5)    | 0.044<br>(0.015)               | <b>0.762</b><br><b>(0.057)</b> | 0.781<br>(0.068)               | <b>0.770</b><br><b>(0.056)</b> | 0.039<br>(0.008)             |

|                                 |     |                                |                                |                                |                                |                                |
|---------------------------------|-----|--------------------------------|--------------------------------|--------------------------------|--------------------------------|--------------------------------|
| abalone-<br>20_vs_8<br>-9-10    | (1) | 0.313<br>(0.223)               | 0.437<br>(0.278)               | 0.929<br>(0.09)                | 0.632<br>(0.103)               | <b>0.022</b><br><b>(0.004)</b> |
|                                 | (2) | 0.007<br>(0.041)               | 0.012<br>(0.065)               | 0.792<br>(0.196)               | 0.502<br>(0.011)               | 0.028<br>(0.012)               |
|                                 | (3) | 0.175<br>(0.157)               | 0.776<br>(0.21)                | 0.792<br>(0.196)               | 0.796<br>(0.159)               | 0.028<br>(0.012)               |
|                                 | (4) | <b>0.328</b><br><b>(0.119)</b> | 0.898<br>(0.047)               | 0.925<br>(0.058)               | 0.901<br>(0.044)               | 0.036<br>(0.016)               |
|                                 | (5) | 0.303<br>(0.12)                | <b>0.908</b><br><b>(0.04)</b>  | <b>0.935</b><br><b>(0.042)</b> | <b>0.91</b><br><b>(0.039)</b>  | 0.033<br>(0.008)               |
| kr-vs-k<br>-zero_vs<br>_fifteen | (1) | <b>0.962</b><br><b>(0.183)</b> | 0.962<br>(0.182)               | 0.983<br>(0.091)               | 0.979<br>(0.092)               | <b>0.001</b><br><b>(0.004)</b> |
|                                 | (2) | 0.785<br>(0.401)               | 0.791<br>(0.404)               | 0.968<br>(0.097)               | 0.892<br>(0.202)               | 0.015<br>(0.054)               |
|                                 | (3) | 0.05<br>(0.162)                | 0.945<br>(0.187)               | 0.968<br>(0.097)               | 0.963<br>(0.102)               | 0.015<br>(0.054)               |
|                                 | (4) | 0.666<br>(0.314)               | 0.962<br>(0.183)               | 0.981<br>(0.092)               | 0.979<br>(0.093)               | 0.012<br>(0.054)               |
|                                 | (5) | 0.784<br>(0.279)               | <b>1.0</b><br><b>(0.002)</b>   | <b>1.0</b><br><b>(0.0)</b>     | <b>1.0</b><br><b>(0.002)</b>   | 0.011<br>(0.054)               |
| poker-8<br>_vs_6                | (1) | <b>0.86</b><br><b>(0.314)</b>  | 0.868<br>(0.311)               | 0.94<br>(0.181)                | 0.923<br>(0.165)               | 0.004<br>(0.007)               |
|                                 | (2) | 0.6<br>(0.401)                 | 0.669<br>(0.421)               | 0.908<br>(0.184)               | 0.809<br>(0.223)               | 0.011<br>(0.009)               |
|                                 | (3) | 0.137<br>(0.246)               | 0.916<br>(0.151)               | 0.908<br>(0.184)               | 0.92<br>(0.145)                | 0.011<br>(0.009)               |
|                                 | (4) | 0.291<br>(0.272)               | 0.992<br>(0.042)               | 0.993<br>(0.036)               | 0.993<br>(0.037)               | 0.006<br>(0.014)               |
|                                 | (5) | 0.4<br>(0.237)                 | <b>0.998</b><br><b>(0.011)</b> | <b>0.999</b><br><b>(0.005)</b> | <b>0.998</b><br><b>(0.011)</b> | <b>0.003</b><br><b>(0.006)</b> |
| poker-8<br>-9_vs_5              | (1) | 0.055<br>(0.088)               | 0.111<br>(0.174)               | 0.609<br>(0.115)               | 0.519<br>(0.033)               | 0.028<br>(0.004)               |
|                                 | (2) | 0.0<br>(0.0)                   | 0.0<br>(0.0)                   | 0.53<br>(0.109)                | 0.5<br>(0.0)                   | <b>0.027</b><br><b>(0.003)</b> |
|                                 | (3) | 0.039<br>(0.04)                | 0.57<br>(0.138)                | 0.53<br>(0.109)                | 0.598<br>(0.088)               | <b>0.027</b><br><b>(0.003)</b> |
|                                 | (4) | 0.045<br>(0.077)               | 0.675<br>(0.091)               | 0.643<br>(0.115)               | 0.689<br>(0.087)               | 0.037<br>(0.015)               |
|                                 | (5) | <b>0.141</b><br><b>(0.145)</b> | <b>0.752</b><br><b>(0.072)</b> | <b>0.719</b><br><b>(0.096)</b> | <b>0.761</b><br><b>(0.068)</b> | 0.03<br>(0.006)                |

부록 3. 일원배치 분산분석표

귀무가설이 기각되어, 유의한 차이가 있다고 판단되는 경우는 굵은 글씨로 표기함

| Data                                | metric                      | TRT      | df  | SS     | MS    | F       | PR(>F)       |
|-------------------------------------|-----------------------------|----------|-----|--------|-------|---------|--------------|
| abalone<br>19                       | <b>F1-score</b>             | method   | 4   | 0.062  | 0.016 | 67.184  | <b>0</b>     |
|                                     |                             | Residual | 145 | 0.034  | 0     |         |              |
|                                     | <b>Geometric Mean</b>       | method   | 4   | 18.763 | 4.691 | 416.808 | <b>0</b>     |
|                                     |                             | Residual | 145 | 1.632  | 0.011 |         |              |
|                                     | <b>Area Under ROC Curve</b> | method   | 4   | 0.529  | 0.132 | 9.885   | <b>0</b>     |
|                                     |                             | Residual | 145 | 1.941  | 0.013 |         |              |
|                                     | <b>Balanced Accuracy</b>    | method   | 4   | 2.159  | 0.54  | 116.54  | <b>0</b>     |
|                                     |                             | Residual | 145 | 0.672  | 0.005 |         |              |
|                                     | <b>Brier Inaccuracy</b>     | method   | 4   | 0.015  | 0.004 | 35.148  | <b>0</b>     |
|                                     |                             | Residual | 145 | 0.015  | 0     |         |              |
| abalone<br>-20_vs<br>_8-9-1<br>0    | <b>F1-score</b>             | method   | 4   | 2.227  | 0.557 | 26.668  | <b>0</b>     |
|                                     |                             | Residual | 145 | 3.027  | 0.021 |         |              |
|                                     | <b>Geometric Mean</b>       | method   | 4   | 17.613 | 4.403 | 170.269 | <b>0</b>     |
|                                     |                             | Residual | 145 | 3.75   | 0.026 |         |              |
|                                     | <b>Area Under ROC Curve</b> | method   | 4   | 0.683  | 0.171 | 9.445   | <b>0</b>     |
|                                     |                             | Residual | 145 | 2.623  | 0.018 |         |              |
|                                     | <b>Balanced Accuracy</b>    | method   | 4   | 3.779  | 0.945 | 120.038 | <b>0</b>     |
|                                     |                             | Residual | 145 | 1.141  | 0.008 |         |              |
|                                     | <b>Brier Inaccuracy</b>     | method   | 4   | 0.004  | 0.001 | 6.936   | <b>0</b>     |
|                                     |                             | Residual | 145 | 0.019  | 0     |         |              |
| kr-vs-<br>k-zero<br>_vs_fift<br>een | <b>F1-score</b>             | method   | 4   | 14.814 | 3.703 | 46.708  | <b>0</b>     |
|                                     |                             | Residual | 145 | 11.497 | 0.079 |         |              |
|                                     | <b>Geometric Mean</b>       | method   | 4   | 0.794  | 0.198 | 3.747   | <b>0.006</b> |
|                                     |                             | Residual | 145 | 7.681  | 0.053 |         |              |
|                                     | <b>Area Under ROC Curve</b> | method   | 4   | 0.021  | 0.005 | 0.718   | 0.581        |
|                                     |                             | Residual | 145 | 1.035  | 0.007 |         |              |
|                                     | <b>Balanced Accuracy</b>    | method   | 4   | 0.209  | 0.052 | 3.831   | <b>0.005</b> |
|                                     |                             | Residual | 145 | 1.978  | 0.014 |         |              |
|                                     | <b>Brier Inaccuracy</b>     | method   | 4   | 0.004  | 0.001 | 0.439   | 0.78         |
|                                     |                             | Residual | 145 | 0.335  | 0.002 |         |              |

|                        |                   |          |     |        |       |                    |       |
|------------------------|-------------------|----------|-----|--------|-------|--------------------|-------|
| poker-<br>8_vs_6       | <b>F1-score</b>   | method   | 4   | 9.486  | 2.371 | 26.366             | 0     |
|                        |                   | Residual | 145 | 13.042 | 0.09  |                    |       |
|                        | <b>Geometric</b>  | method   | 4   | 2.16   | 0.54  | 9.055              | 0     |
|                        | <b>Mean</b>       | Residual | 145 | 8.648  | 0.06  |                    |       |
|                        | <b>Area</b>       | method   | 4   | 0.237  | 0.059 | 2.916              | 0.023 |
|                        | <b>Under</b>      | Residual | 145 | 2.949  | 0.02  |                    |       |
|                        | <b>ROC Curve</b>  |          |     |        |       |                    |       |
|                        | <b>Balanced</b>   | method   | 4   | 0.698  | 0.174 | 8.77               | 0     |
|                        | <b>Accuracy</b>   | Residual | 145 | 2.883  | 0.02  |                    |       |
|                        | <b>Brier</b>      | method   | 4   | 0.002  | 0     | 4.724              | 0.001 |
| poker-<br>8-9_vs<br>_5 | <b>Inaccuracy</b> | Residual | 145 | 0.012  | 0     |                    |       |
|                        | <b>F1-score</b>   | method   | 4   | 0.325  | 0.081 | 11.15              | 0     |
|                        |                   | Residual | 145 | 1.056  | 0.007 |                    |       |
|                        | <b>Geometric</b>  | method   | 4   | 14.104 | 3.526 | $\frac{280.33}{2}$ | 0     |
|                        | <b>Mean</b>       | Residual | 145 | 1.824  | 0.013 |                    |       |
|                        | <b>Area</b>       | method   | 4   | 0.772  | 0.193 | 16.233             | 0     |
|                        | <b>Under</b>      | Residual | 145 | 1.724  | 0.012 |                    |       |
|                        | <b>ROC Curve</b>  |          |     |        |       |                    |       |
|                        | <b>Balanced</b>   | method   | 4   | 1.477  | 0.369 | 87.674             | 0     |
|                        | <b>Accuracy</b>   | Residual | 145 | 0.611  | 0.004 |                    |       |
|                        | <b>Brier</b>      | method   | 4   | 0.002  | 0.001 | 9.838              | 0     |
|                        | <b>Inaccuracy</b> | Residual | 145 | 0.008  | 0     |                    |       |

#### 부록 4. Leven의 등분산 검정 결과

귀무가설이 기각되어 이분산으로 판단되는 경우는 굵은 글씨로 표기함

| Data      | Metric           | Statistic     | p-value      |
|-----------|------------------|---------------|--------------|
| abalone19 | <b>F1-score</b>  | <b>20.17</b>  | <b>0</b>     |
|           | <b>Geometric</b> | <b>8.157</b>  | <b>0</b>     |
|           | <b>Mean</b>      |               |              |
|           | <b>Area</b>      |               |              |
|           | <b>Under</b>     | <b>4.782</b>  | <b>0.001</b> |
|           | <b>ROC Curve</b> |               |              |
|           | <b>Balanced</b>  | <b>17.296</b> | <b>0</b>     |
|           | <b>Accuracy</b>  |               |              |
|           | Brier            | 1.85          | 0.122        |
|           | Inaccuracy       |               |              |

|                             |            |        |       |
|-----------------------------|------------|--------|-------|
| abalone-20_vs_8<br>-9-10    | F1-score   | 16.952 | 0     |
|                             | Geometric  |        |       |
|                             | Mean       | 17.214 | 0     |
|                             | Area       |        |       |
|                             | Under      | 5.797  | 0     |
|                             | ROC Curve  |        |       |
|                             | Balanced   |        |       |
|                             | Accuracy   | 17.302 | 0     |
| kr-vs-k-zero_vs<br>_fifteen | Brier      |        |       |
|                             | Inaccuracy | 1.543  | 0.193 |
|                             | F1-score   | 3.013  | 0.02  |
|                             | Geometric  |        |       |
|                             | Mean       | 3.747  | 0.006 |
|                             | Area       |        |       |
|                             | Under      | 0.718  | 0.581 |
|                             | ROC Curve  |        |       |
| poker-8_vs_6                | Balanced   |        |       |
|                             | Accuracy   | 3.831  | 0.005 |
|                             | Brier      |        |       |
|                             | Inaccuracy | 0.439  | 0.78  |
|                             | F1-score   | 3.717  | 0.007 |
|                             | Geometric  |        |       |
|                             | Mean       | 9.08   | 0     |
|                             | Area       |        |       |
|                             | Under      | 2.916  | 0.023 |
|                             | ROC Curve  |        |       |
|                             | Balanced   |        |       |
|                             | Accuracy   | 8.814  | 0     |
|                             | Brier      |        |       |
|                             | Inaccuracy | 2.658  | 0.035 |
|                             |            |        |       |
|                             |            |        |       |

|                |                          |        |       |
|----------------|--------------------------|--------|-------|
| poker-8-9_vs_5 | <b>F1-score</b>          | 9.608  | 0     |
|                | <b>Geometric Mean</b>    | 5.427  | 0     |
|                | Area Under ROC Curve     | 0.416  | 0.797 |
|                | <b>Balanced Accuracy</b> | 14.487 | 0     |
|                | <b>Brier Inaccuracy</b>  | 5.176  | 0.001 |

부록 5. 사후검정 결과

귀무가설이 기각되어, 유의한 차이가 있다고 판단되는 경우는 굵은 글씨로 표기함

| Data                    | Group1 | Group2 | Metric                   | p-adj        | Method       |
|-------------------------|--------|--------|--------------------------|--------------|--------------|
| abalone19               | (5)    | (1)    | Area Under the ROC Curve | 0.900        | Games-Howell |
|                         | (5)    | (1)    | <b>Brier Inaccuracy</b>  | <b>0.001</b> | tukeyHSD     |
| abalone-20_vs_8-9-10    | (5)    | (1)    | F1-score                 | 0.900        | Games-Howell |
|                         | (5)    | (1)    | <b>Geometric Mean</b>    | <b>0.001</b> | Games-Howell |
|                         | (5)    | (1)    | Area Under the ROC Curve | 0.900        | Games-Howell |
|                         | (5)    | (1)    | <b>Balanced Accuracy</b> | <b>0.001</b> | Games-Howell |
|                         | (5)    | (1)    | <b>Brier Inaccuracy</b>  | <b>0.002</b> | tukeyHSD     |
| kr-vs-k-zero_vs_fifteen | (5)    | (1)    | <b>F1-score</b>          | <b>0.039</b> | Games-Howell |
|                         | (5)    | (1)    | Geometric Mean           | 0.771        | Games-Howell |
|                         | (5)    | (1)    | Balanced Accuracy        | 0.714        | Games-Howell |

|                |     |     |                                 |              |              |
|----------------|-----|-----|---------------------------------|--------------|--------------|
| poker-8_vs_6   | (5) | (1) | <b>F1-score</b>                 | <b>0.001</b> | Games-Howell |
|                | (5) | (1) | Geometric-Mean                  | 0.178        | Games-Howell |
|                | (5) | (1) | Area Under the ROC Curve        | 0.401        | Games-Howell |
|                | (5) | (1) | Balanced Accuracy               | 0.128        | Games-Howell |
|                | (5) | (1) | Brier Inaccuracy                | 0.900        | Games-Howell |
| poker-8-9_vs_5 | (5) | (1) | <b>Area Under the ROC Curve</b> | <b>0.001</b> | tukeyHSD     |
|                | (5) | (1) | <b>Balanced Accuracy</b>        | <b>0.001</b> | Games-Howell |
|                | (5) | (1) | Brier Inaccuracy                | 0.652        | Games-Howell |

#### 부록 6. 오버샘플링 결과 (평균, 표준편차)

가장 좋은 결과는 굵은 글씨로 표기함

| Data      | 불균형도   | F1-score                 | Geometric Mean           | Area Under ROC Curve     | Balanced Accuracy        | Brier Inaccuracy         |
|-----------|--------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| abalone19 | 10.035 | 0.041<br>(0.016)         | 0.729<br>(0.150)         | 0.769<br>(0.081)         | 0.753<br>(0.073)         | 0.049<br>(0.023)         |
|           | 20.157 | 0.044<br>(0.015)         | <b>0.762<br/>(0.057)</b> | <b>0.781<br/>(0.068)</b> | <b>0.770<br/>(0.056)</b> | <b>0.039<br/>(0.008)</b> |
|           | 30.104 | 0.044<br>(0.017)         | 0.734<br>(0.154)         | 0.767<br>(0.094)         | 0.757<br>(0.083)         | 0.042<br>(0.014)         |
|           | 40.667 | 0.043<br>(0.019)         | 0.723<br>(0.164)         | 0.751<br>(0.126)         | 0.744<br>(0.101)         | 0.042<br>(0.014)         |
|           | 50.391 | 0.041<br>(0.017)         | 0.722<br>(0.157)         | 0.751<br>(0.107)         | 0.746<br>(0.088)         | 0.042<br>(0.014)         |
|           | 61.000 | <b>0.046<br/>(0.018)</b> | 0.731<br>(0.155)         | 0.767<br>(0.099)         | 0.756<br>(0.084)         | 0.041<br>(0.015)         |
|           | 70.242 | 0.042<br>(0.019)         | 0.722<br>(0.160)         | 0.743<br>(0.127)         | 0.747<br>(0.089)         | <b>0.039<br/>(0.013)</b> |

|                                 |        |                                |                                |                                |                               |                                |
|---------------------------------|--------|--------------------------------|--------------------------------|--------------------------------|-------------------------------|--------------------------------|
| abalone-<br>20_vs_8<br>-9-10    | 10.076 | 0.286<br>(0.122)               | 0.898<br>(0.043)               | 0.922<br>(0.049)               | 0.9<br>(0.042)                | 0.037<br>(0.014)               |
|                                 | 20.346 | 0.303<br>(0.12)                | <b>0.908</b><br><b>(0.04)</b>  | <b>0.935</b><br><b>(0.042)</b> | <b>0.91</b><br><b>(0.039)</b> | <b>0.033</b><br><b>(0.008)</b> |
|                                 | 30.229 | 0.32<br>(0.14)                 | <b>0.908</b><br><b>(0.046)</b> | 0.932<br>(0.047)               | 0.909<br>(0.045)              | <b>0.033</b><br><b>(0.012)</b> |
|                                 | 40.692 | <b>0.329</b><br><b>(0.101)</b> | 0.905<br>(0.053)               | 0.932<br>(0.049)               | 0.907<br>(0.053)              | 0.035<br>(0.015)               |
|                                 | 50.381 | 0.311<br>(0.125)               | 0.864<br>(0.18)                | 0.902<br>(0.128)               | 0.884<br>(0.1)                | 0.036<br>(0.016)               |
|                                 | 62.235 | <b>0.329</b><br><b>(0.12)</b>  | 0.862<br>(0.184)               | 0.901<br>(0.13)                | 0.88<br>(0.112)               | 0.034<br>(0.016)               |
|                                 | 70.533 | 0.317<br>(0.124)               | 0.857<br>(0.196)               | 0.898<br>(0.151)               | 0.877<br>(0.128)              | 0.037<br>(0.018)               |
|                                 | 10.025 | 0.735<br>(0.326)               | 0.999<br>(0.005)               | <b>1.0</b><br><b>(0.001)</b>   | 0.999<br>(0.005)              | 0.011<br>(0.055)               |
| kr-vs-k<br>-zero_vs<br>_fifteen | 20.217 | <b>0.784</b><br><b>(0.279)</b> | <b>1.0</b><br><b>(0.002)</b>   | <b>1.0</b><br><b>(0.0)</b>     | <b>1.0</b><br><b>(0.002)</b>  | 0.011<br>(0.054)               |
|                                 | 30.325 | 0.694<br>(0.336)               | 0.998<br>(0.012)               | 0.998<br>(0.012)               | 0.998<br>(0.011)              | <b>0.01</b><br><b>(0.054)</b>  |
|                                 | 40.433 | 0.646<br>(0.361)               | 0.964<br>(0.183)               | 0.982<br>(0.091)               | 0.981<br>(0.092)              | 0.012<br>(0.054)               |
|                                 | 50.542 | 0.678<br>(0.324)               | 0.994<br>(0.022)               | 0.996<br>(0.014)               | 0.995<br>(0.021)              | 0.012<br>(0.054)               |
|                                 | 60.650 | 0.676<br>(0.309)               | 0.998<br>(0.012)               | 0.998<br>(0.014)               | 0.998<br>(0.011)              | <b>0.01</b><br><b>(0.053)</b>  |
|                                 | 71.353 | 0.649<br>(0.334)               | 0.964<br>(0.183)               | 0.982<br>(0.091)               | 0.981<br>(0.092)              | 0.012<br>(0.056)               |
|                                 | 10.074 | <b>0.45</b><br><b>(0.302)</b>  | 0.997<br>(0.014)               | 0.999<br>(0.006)               | 0.997<br>(0.014)              | 0.003<br>(0.005)               |
| poker-8<br>_vs_6                | 20.400 | 0.4<br>(0.237)                 | 0.998<br>(0.011)               | 0.999<br>(0.005)               | 0.998<br>(0.011)              | 0.003<br>(0.006)               |
|                                 | 30.222 | 0.421<br>(0.232)               | <b>1.0</b><br><b>(0.001)</b>   | <b>1.0</b><br><b>(0.001)</b>   | <b>1.0</b><br><b>(0.001)</b>  | 0.003<br>(0.004)               |
|                                 | 40.800 | 0.379<br>(0.262)               | 0.996<br>(0.019)               | 0.997<br>(0.018)               | 0.997<br>(0.018)              | <b>0.002</b><br><b>(0.003)</b> |
|                                 | 51.000 | 0.355<br>(0.254)               | 0.999<br>(0.004)               | <b>1.0</b><br><b>(0.002)</b>   | 0.999<br>(0.004)              | 0.003<br>(0.004)               |
|                                 | 62.769 | 0.296<br>(0.28)                | 0.992<br>(0.046)               | 0.993<br>(0.037)               | 0.992<br>(0.046)              | 0.005<br>(0.012)               |
|                                 | 74.182 | 0.294<br>(0.29)                | 0.993<br>(0.035)               | 0.995<br>(0.024)               | 0.993<br>(0.035)              | 0.006<br>(0.012)               |
|                                 |        |                                |                                |                                |                               |                                |



|                    |        |                                |                                |                                |                                |                               |
|--------------------|--------|--------------------------------|--------------------------------|--------------------------------|--------------------------------|-------------------------------|
| poker-8<br>-9_vs_5 | 10.061 | <b>0.153</b><br><b>(0.112)</b> | 0.744<br>(0.076)               | 0.707<br>(0.088)               | 0.754<br>(0.071)               | 0.032<br>(0.011)              |
|                    | 20.123 | 0.141<br>(0.145)               | <b>0.752</b><br><b>(0.072)</b> | <b>0.719</b><br><b>(0.096)</b> | <b>0.761</b><br><b>(0.068)</b> | <b>0.03</b><br><b>(0.006)</b> |
|                    | 30.184 | 0.129<br>(0.126)               | 0.736<br>(0.066)               | 0.701<br>(0.09)                | 0.747<br>(0.059)               | 0.031<br>(0.006)              |
|                    | 40.964 | 0.079<br>(0.111)               | 0.721<br>(0.084)               | 0.678<br>(0.115)               | 0.73<br>(0.08)                 | 0.033<br>(0.011)              |
|                    | 52.136 | 0.064<br>(0.099)               | 0.708<br>(0.089)               | 0.676<br>(0.111)               | 0.719<br>(0.086)               | 0.034<br>(0.013)              |
|                    | 60.368 | 0.062<br>(0.081)               | 0.706<br>(0.095)               | 0.664<br>(0.121)               | 0.714<br>(0.091)               | 0.035<br>(0.014)              |
|                    | 71.688 | 0.062<br>(0.101)               | 0.717<br>(0.09)                | 0.678<br>(0.11)                | 0.727<br>(0.087)               | 0.034<br>(0.012)              |