



# 아동·청소년 상담 데이터를 활용한 9가지 학대 및 폭력 유형 분류 연구

아동·청소년 대상 학대와 폭력 문제가 심각해지고 있습니다.  
이 연구는 상담 데이터를 활용해 9가지 유형을 분류하고 분석합니다. 데이터  
기반 접근으로 효과적인 대응 방안을 모색합니다.

201904008 곽재원  
202104216 백종민

# 주제 소개 및 데이터 소개

## 단원 목표

- 주제 선정 배경에 대해 알아본다.
- 데이터 수집 경위에 대한 소개
- 데이터 샘플을 소개한다.



# CHILD ABUSE

— on flesting nedabusis —

113

Zenley a now of child  
applingsted tof  
child abuse



Lrafts of Insegiaced  
arrgnrgly rising  
conultte recotal  
chid item.



Child te  
aredet  
imortin  
chil



UNTLENT ABUTS

Cplifized  
efteralces

Auring  
jehences

21496

Lowrored chick svant  
and milk of absies.

1336

Colley chnolt ascia  
etesrable renfer creaia  
chid abuse rer abus.

## 아동·청소년 학대 및 폭력의 심각성

아동학대 행위자의 86%는 부모, 전년 대비 3.2%p 증가

### 보건복지부, '2023년 아동학대 연차보고서' 발간

작성

"아동학대 행위자의 86%는 부모, 전년 대비 3.2%p 증가"

이용남 記者 입력 2024.08.30 15:53

- 아동학대 신고는 48,522건으로 지난해보다 2,419건 증가 -

### 재학대의 심각성

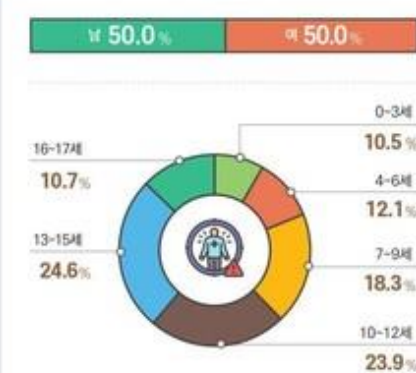
전체 아동학대 사례 중 재학대 비율이 15.7%에 달합니다. 이는 학대가 반복되고 만성화 될수 있음을 보여주며, 이에 대한 심층적인 연구와 대책이 필요합니다.

# 2023 아동학대 통계현황

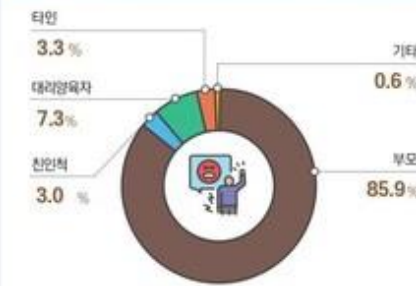
## 01 | 신고접수·판단 및 아동학대 유형



## 02 | 피해아동 특성



## 04 | 학대행위자와 피해아동의 관계



## 03 | 피해아동 상황



## 05 | 학대행위자 상황



## 06 | 서비스 제공 현황



## 07 | 아동학대 사망



# 주제 선정 배경

1

## 사회적 문제 인식

아동·청소년 학대와 폭력 사례가 증가하고 있습니다.  
이는 사회적 관심과 대책 마련이 시급한 문제입니다.

2

## 데이터 활용 가능성

상담 데이터를 통해 학대와 폭력의 패턴을 파악할 수 있습니다.  
이는 예방과 조기 개입에 큰 도움이 됩니다.

3

## 연구의 필요성

체계적인 분류와 분석으로 효과적인 대응 방안을 마련할 수 있습니다.  
이를 통해 아동·청소년 보호 정책 수립에 기여합니다.



# 사용된 데이터셋: AI허브의 아동 청소년 상담데이터

 AI Hub

AI 데이터찾기

AI 허브소개

참여하기

커뮤니티

AI 개발지원

고객지원

로그인

회원가입

데이터 찾기

🏠 | AI 데이터찾기 &gt; 데이터 찾기



#상담(아동청소년) #자연어 #음성

**NEW** 아동·청소년 상담데이터

분야 한국어 유형 오디오, 텍스트

구축년도: 2023 갱신년월: 2024-10 조회수: 5,449 다운로드: 344 용량: 11.54 GB

다운로드

↓ 샘플 데이터 ?

관심데이터 등록

👍 30

## 소개

만7~12세 아동·청소년 **3,596건**의 신체적, 정신적 문제와 상황을 포괄한 **7문항 18항목 이상의 상담데이터**

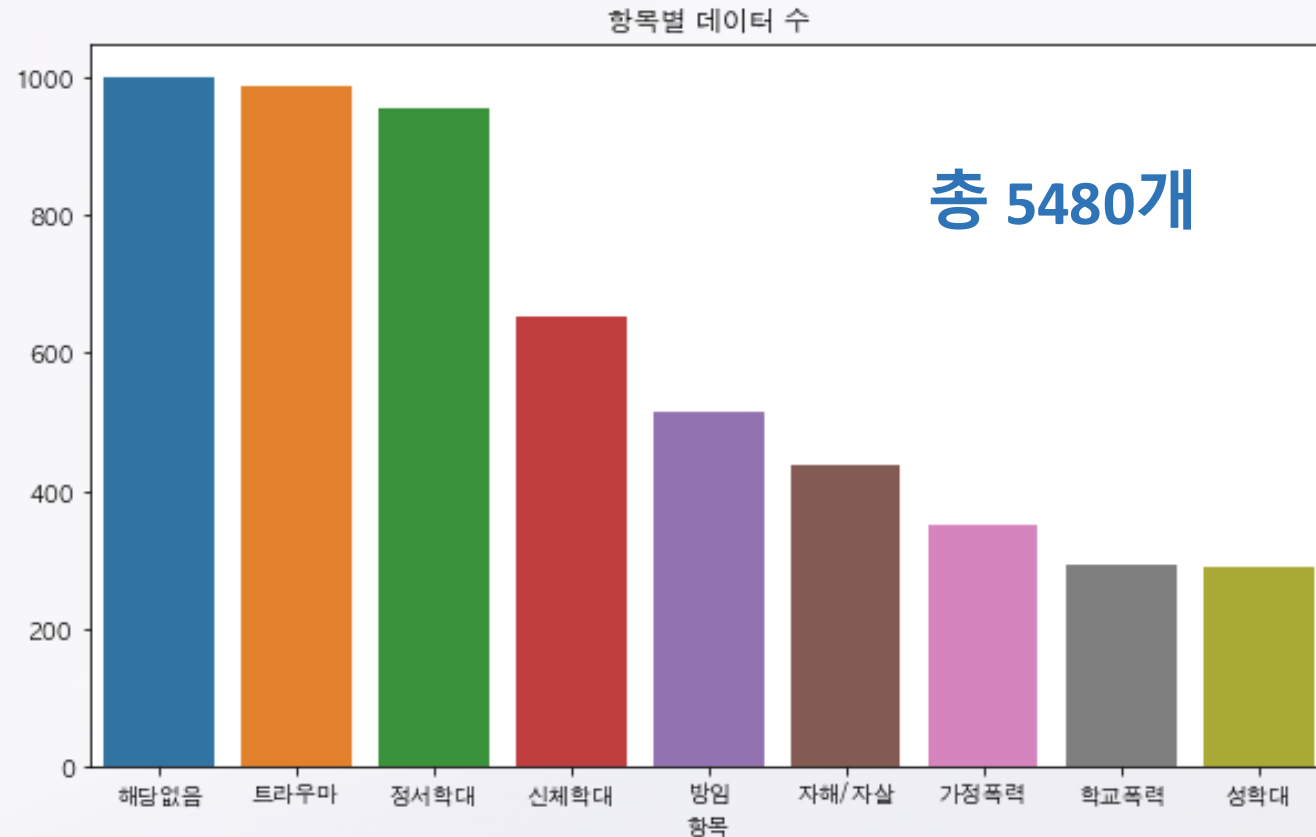
## 구축목적

**아동·청소년의 신체적, 정신적 건강 상태를 평가**하고 다양한 문제를 이해하는데 도움을 주기 위한 목적으로 상담가 및 전문가들이 효과적으로 **아동·청소년을 상담하고 지원**하기 위함.

# 사용된 데이터셋: AI허브의 아동 청소년 상담데이터

```
{
  "문항": "대인관계",
  "위기단계": "관찰필요",
  "list": [
    {
      "항목": "아버지",
      "임상가코멘트": {
        "val": "㉓ 무서움, 두려움(공포)을 경험하는 것으로 보입니다."
      },
      "점수": 6,
      "아버지": {
        "val": "자기중심성(분노)"
      },
      "audio": [
        {
          "type": "Q",
          "text": "아빠가 화 많이 내는 표정을 짓는다고 했는데 그런 이유가 있는 거 같아요?",
          "wave": "all.wav",
          "start": "01:00.190",
          "end": "01:06.410"
        },
        {
          "type": "A",
          "text": "그냥 자기가 화나면 저희한테 화풀이해요.",
          "wave": "all.wav",
          "start": "01:06.710",
          "end": "01:11.980"
        },
        {
          "type": "Q",
          "text": "그 표정을 지을 때 기분이 어때요?",
          "wave": "all.wav",
          "start": "01:12.270",
          "end": "01:15.880"
        },
        {
          "type": "A",
          "text": "슬프기도 하고 무섭기도 해요.",
          "wave": "all.wav",
          "start": "01:16.180",
          "end": "01:20.590"
        }
      ]
    }
  ]
},
```

원본 데이터 샘플



항목: 가정폭력, 대화:

Q: 가족들끼리 소리를 지르면서 싸우는 걸 듣거나 본 적이 있나요?

A: 엄마 아빠가 둘이 싸우진 않고 기분이 나빠지면 우리를 때려서 그런 일은 없어요.

Q: 평소 어른들이 술을 자주 마시나요? 술 마시고 무섭게 행동하는 일이 있나요?

A: 네. 엄마 아빠 둘 다 거의 매일 술을 마셔서 취해 있어요. 저희를 때릴 때도 ...

# Weka를 통한 분석

## 단원 목표

- 나이브 베이즈 텍스트를 통해 데이터를 분석한다.
- `StringToWordVector` 전처리 방법에 대해 알아본다.
- SMO를 통해 데이터를 분석한다.
- 시각화를 통해 데이터의 분포를 간단히 확인한다.





# Weka를 통한 나이브 베이즈 텍스트 분석

## Lowercase Tokens (소문자 변환 여부)

▶ 소문자 변환을 통해 단어의 일관성을 유지하고, 어휘 사전의 크기를 줄여 모델의 일반화 능력을 향상

## stemmer (어간추출기)

▶ 관련 있는 단어들을 하나의 특징으로 묶어 모델의 성능과 일반화 능력을 향상

## stopwordsHandler (불용어 처리기)

▶ 불용어를 제거하면 노이즈를 줄이고 어휘 사전의 크기를 감소시켜 모델의 효율성과 성능을 향상

## useWordFrequencies (단어 빈도 사용 여부)

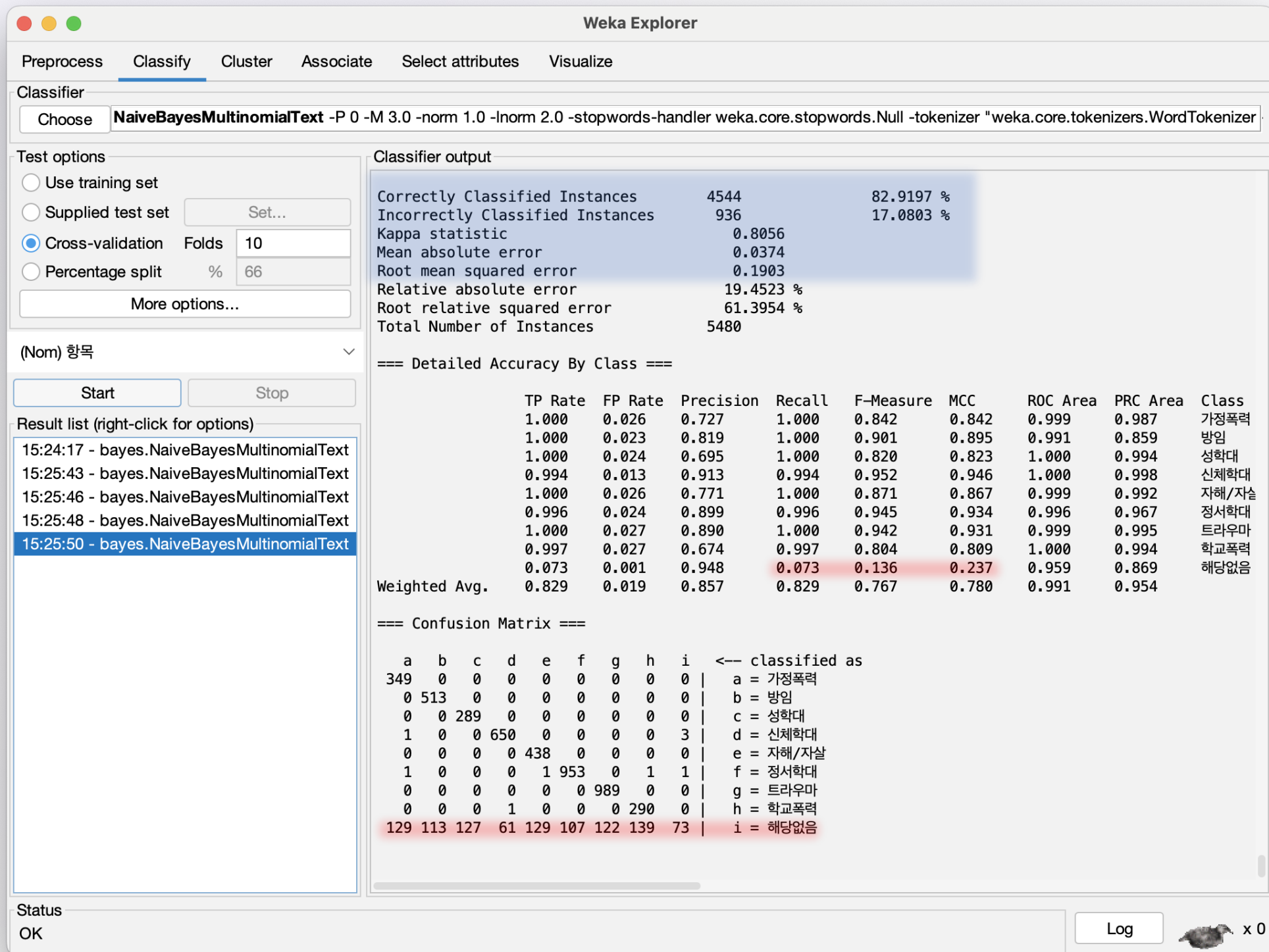
▶ 빈도를 고려하지 않으면 단어가 한 번 등장했는지 여러 번 등장했는지 구분하지 못해 중요한 정보를 놓칠 수 있음

The screenshot shows the 'weka.gui.GenericObjectEditor' window for the 'weka.classifiers.bayes.NaiveBayesMultinomialText' classifier. The 'About' section states: 'Multinomial naive bayes for text data.' with 'More' and 'Capabilities' buttons. The configuration parameters are as follows:

Parameter	Value
LNorm	2.0
batchSize	100
debug	False
doNotCheckCapabilities	False
lowercaseTokens	False
minWordFrequency	3.0
norm	1.0
normalizeDocLength	False
numDecimalPlaces	2
periodicPruning	0
stemmer	Choose <b>NullStemmer</b>
stopwordsHandler	Choose <b>Null</b>
tokenizer	Choose <b>WordTokenizer -delimiters " \r\n\t.,;:\\"0</b>
useWordFrequencies	False

At the bottom, there are four buttons: 'Open...', 'Save...', 'OK', and 'Cancel'.





## 우수한 모델의 성능

정확도 : 82.9197%

RMSE : 0.1903

Kappa Statistic: 0.8056

우연에 의한 일치를 배제한 상태에서 실제로 평가자 간의 일치가 얼마나 높은지를 -1과 1 사이의 값으로 표현함.

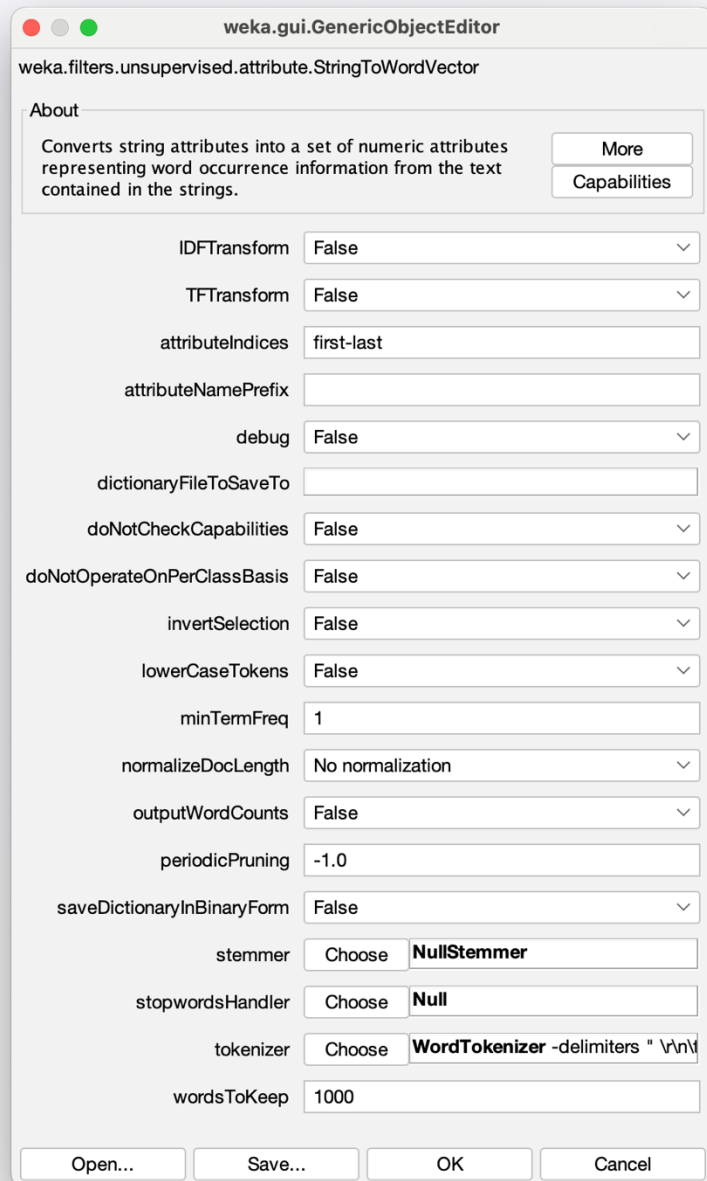
## 개선이 필요한 오분류율

해당없음 오분류율: 0.073

학대 받지 않는 학생을 학대 받고 있고 있다고 오해 할 수 있음.

데이터 불균형의 영향을 많이 받음.

# Weka - StringToWordVector



## IDFTransform (Inverse Document Frequency 변환 적용 여부)

- ▶ IDF 변환을 적용하지 않으면 흔한 단어가 높은 중요도를 가져 모델이 불필요한 특성에 초점을 맞추고 성능이 저하될 수 있음.
- ▶ IDF 적용은 중요한 단어에 가중치를 부여해 분류 성능을 향상시킴.

## TFTransform (Term Frequency 변환 적용 여부)

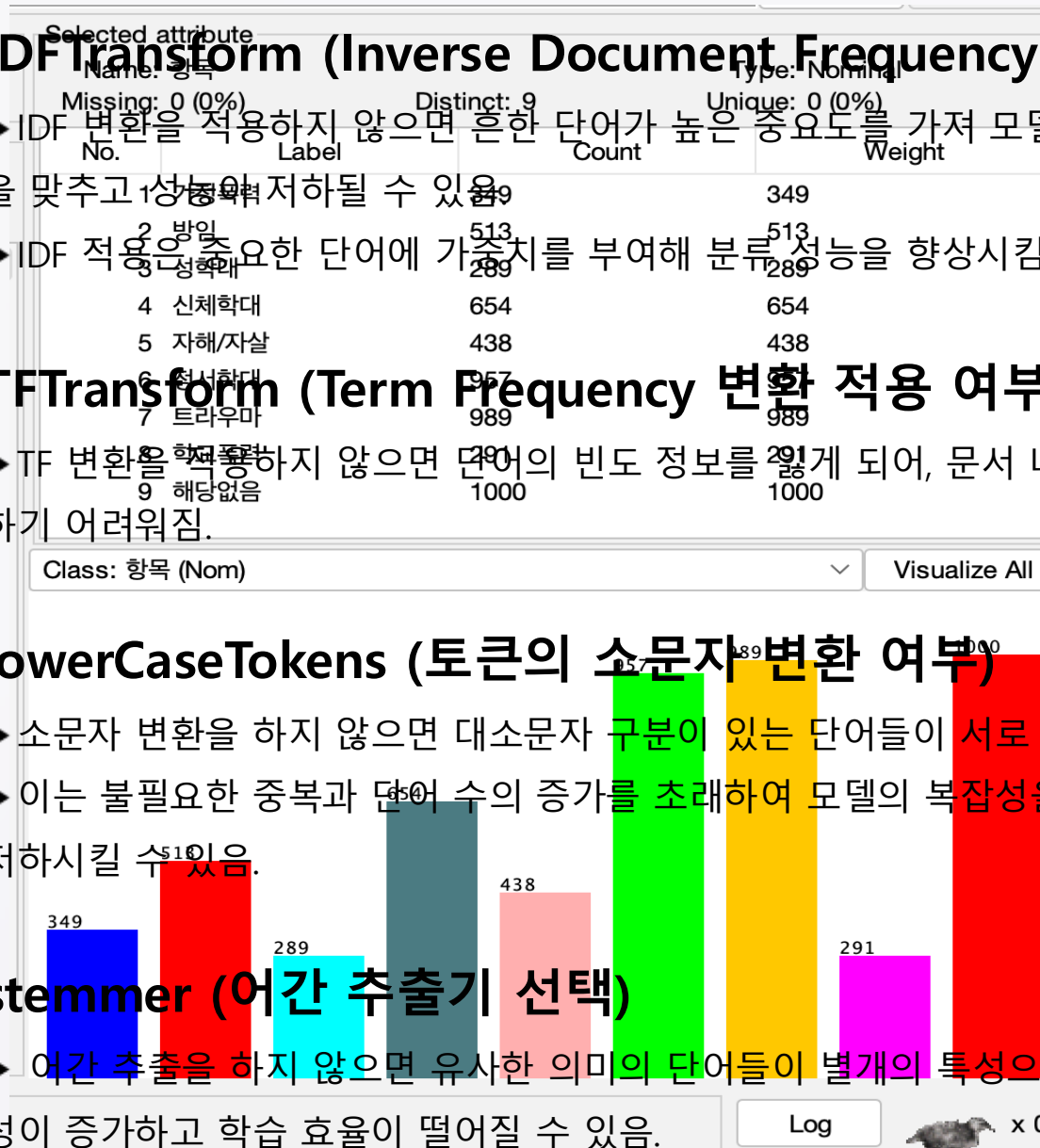
- ▶ TF 변환을 적용하지 않으면 단어의 빈도 정보를 잃게 되어, 문서 내에서 중요한 단어를 구별하기 어려워짐.

## lowerCaseTokens (토큰의 소문자 변환 여부)

- ▶ 소문자 변환을 하지 않으면 대소문자 구분이 있는 단어들이 서로 다른 단어로 인식됨.
- ▶ 이는 불필요한 중복과 단어 수의 증가를 초래하여 모델의 복잡성을 높이고 일반화 성능을 저하시킬 수 있음.

## stemmer (어간 추출기 선택)

- ▶ 어간 추출을 하지 않으면 유사한 의미의 단어들이 별개의 특성으로 취급되어 모델의 복잡성이 증가하고 학습 효율이 떨어질 수 있음.



Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Classifier

Choose

SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C 250007" -calibrator "weka.classifiers.functions.Logistic"

Test options

Use training set

Supplied test set

Cross-validation

Percentage split

Set...

Folds

%

10

66

More options...

(Nom) 항목

Start

Stop

Result list (right-click for options)

15:24:17 - bayes.NaiveBayesMultinomialText

15:25:43 - bayes.NaiveBayesMultinomialText

15:25:46 - bayes.NaiveBayesMultinomialText

15:25:48 - bayes.NaiveBayesMultinomialText

15:25:50 - bayes.NaiveBayesMultinomialText

15:29:22 - functions.SMO

15:29:42 - functions.SMO

15:29:46 - functions.SMO

15:29:50 - functions.SMO

15:29:54 - functions.SMO

15:29:58 - functions.SMO

Classifier output

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances

Incorrectly Classified Instances

Kappa statistic

Mean absolute error

Root mean squared error

Relative absolute error

Root relative squared error

Total Number of Instances

=== Detailed Accuracy By Class ===

TP Rate

FP Rate

Precision

Recall

F-Measure

MCC

ROC Area

PRC Area

Class

=== Confusion Matrix ===

a

b

c

d

e

f

g

h

i

<-- classified as

Status

OK

Log

x 0

## 우수한 모델의 성능

정확도 : 95.5292%

RMSE : 0.2816

Kappa Statistic: 0.9482

우연에 의한 일치를 배제한 상태에서 실제로 평가자 간의 일치가 얼마나 높은지를 -1과 1 사이의 값으로 표현함.

## 개선이 필요한 오분류율

해당없음 오분류율: 0.883

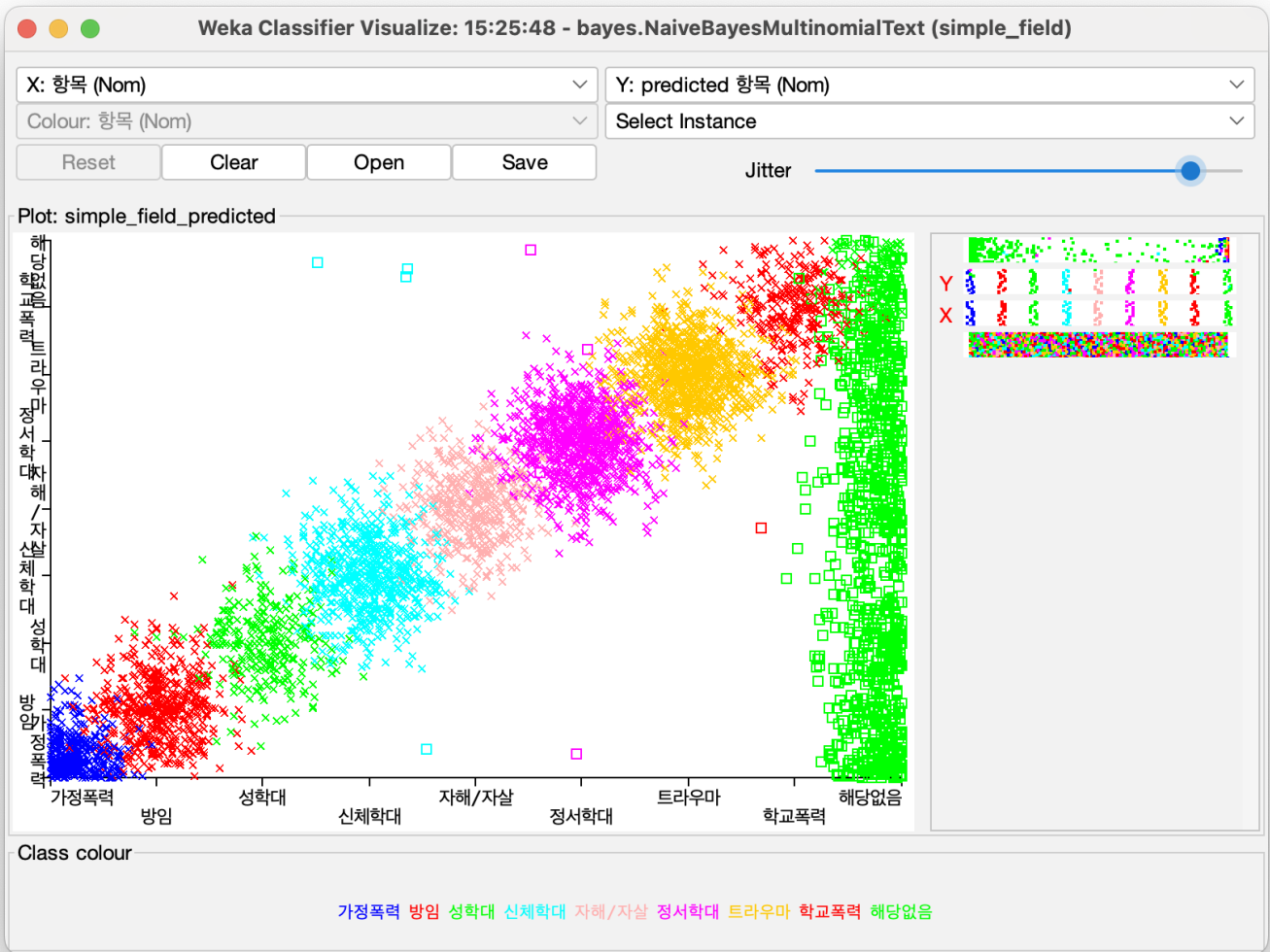
나이브 베이즈 텍스트에 비하여 많이 개선된 수치이지만 95%의 정확도를 고려할 때 여전히 개선이 필요하다.

데이터 불균형의 영향을 많이 받음.

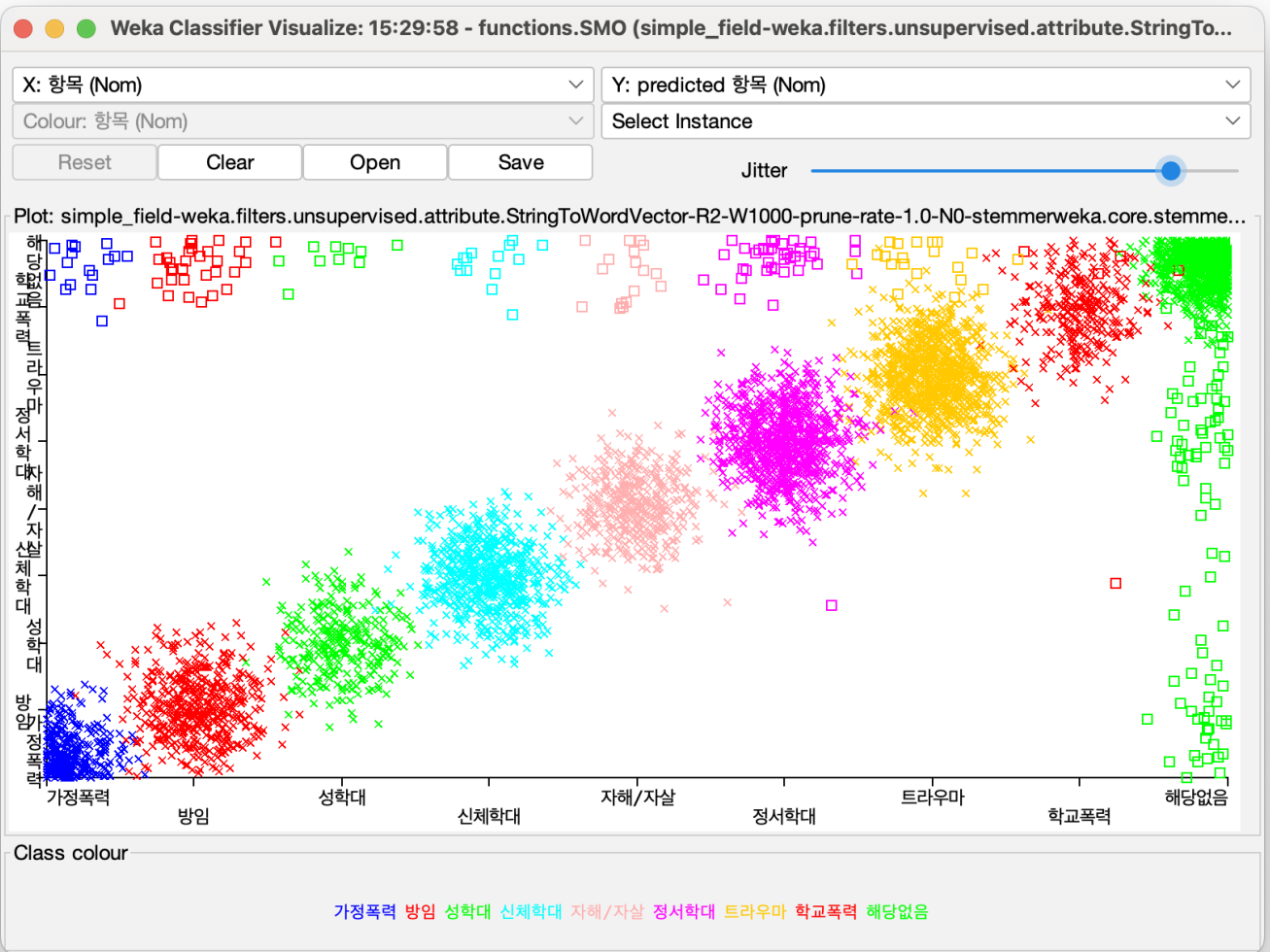


# Visualization Comparison

나이브 베이즈 텍스트는 " 해당없음 " 을 SMO에 비하여 올바르게 분류하지 못함.



나이브 베이즈 텍스트 혼돈 분포 시각화



SMO 혼돈 분포 시각화

# Python을 통한 분석

## 단원 목표

- 나이브 베이지 텍스트와 SMO를 적용한 뒤  
Weka와 분석 과정 비교를 수행한다.
- Python과 Weka의 장단점을 분석한다.
- Python만 가능한 분석 방법을 소개한다.



# Python을 통한 구현

## 나이브 베이즈 텍스트

- Sklearn의 **MultinomialNB**를 통해 구현
- Konlpy의 **Open Korean Text(Okt)**로 토큰화
- **TF-IDF**로 텍스트 벡터화
- 정확도: **88.77%**

## SMO

- Sklearn의 **SVC**를 통해 구현
- Konlpy의 **Open Korean Text(Okt)**로 토큰화
- **TF-IDF**로 텍스트 벡터화
- 정확도: **95.43%**



## Weka와 분석과정 비교

- 동일한 하이퍼 파라미터에서
- ❖ 정확도 및 F1 score가 더 높음.
    - 나이브 베이즈: **82% -> 88%**
    - SMO: 95% -> 95%
- => 한국어에 특화된 **OKT**를 사용한 덕분.
- ❖ 더 다양한 토큰나이저 사용 가능
  - ❖ 코드를 작성하는데 시간이 소모됨



# Python만의 방법 == KoBERT

## SKTBrain/KoBERT

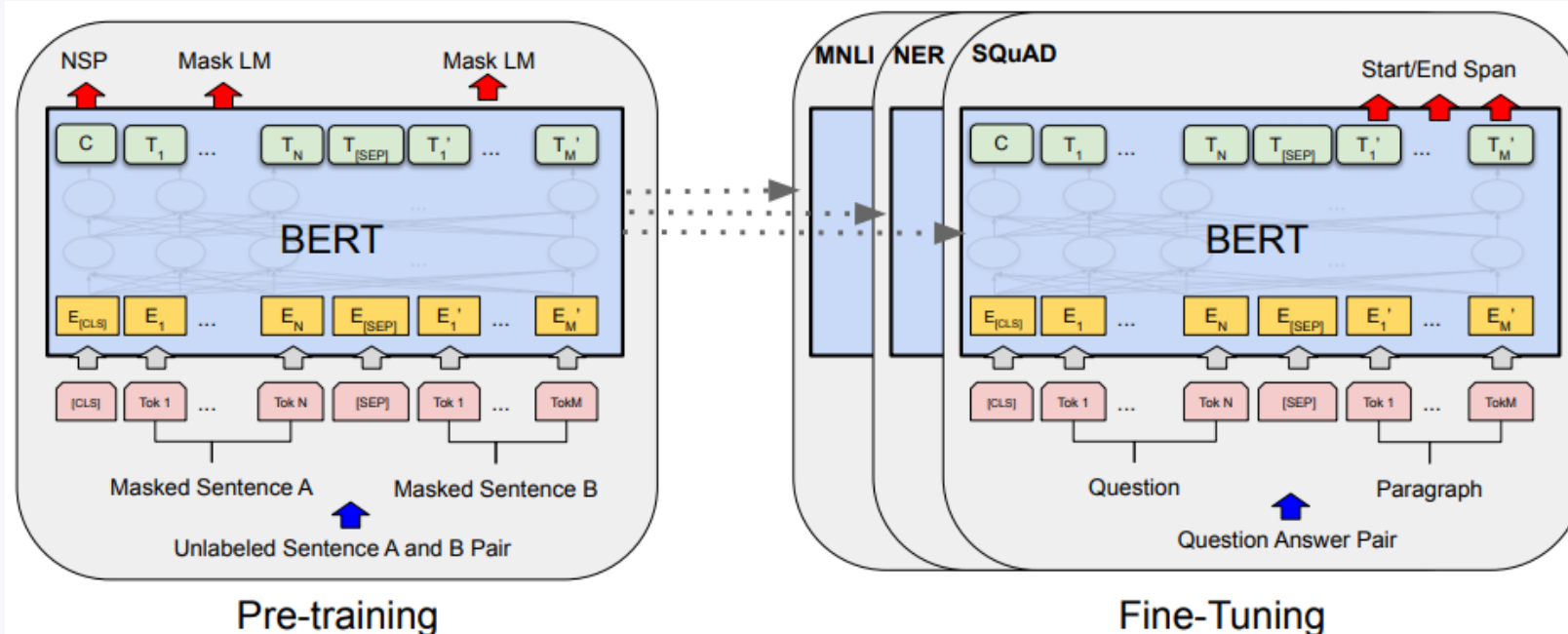
Korean BERT pre-trained cased (KoBERT)



Contributors 5 Issues 3 Stars 627 Forks 159

**KOBERT**는 **Korean Bidirectional Encoder Representations from Transformers**의 약자로, **SKT Brain**에서 공개한 일종의 기계번역 모델이다.

**KOBERT**는 2018년 **Google**에서 발표한 “**BERT**”의 한국어 버전 모델로서 텍스트 분류, 개체명 인식, 감정 분석, 기계 번역 등의 자연어 처리(**Natural Language Processing**) 작업에서 뛰어난 성능을 발휘할 수 있다.



## KoBERT를 사용한 이유

- ❖ **한국어에 특화된** 토큰나이저  
=> 한국어의 문맥을 잘 이해함.
- ❖ Attention 기반의 꼼꼼한 문맥 분석  
=> **1:1로 토큰의 영향력을 비교**

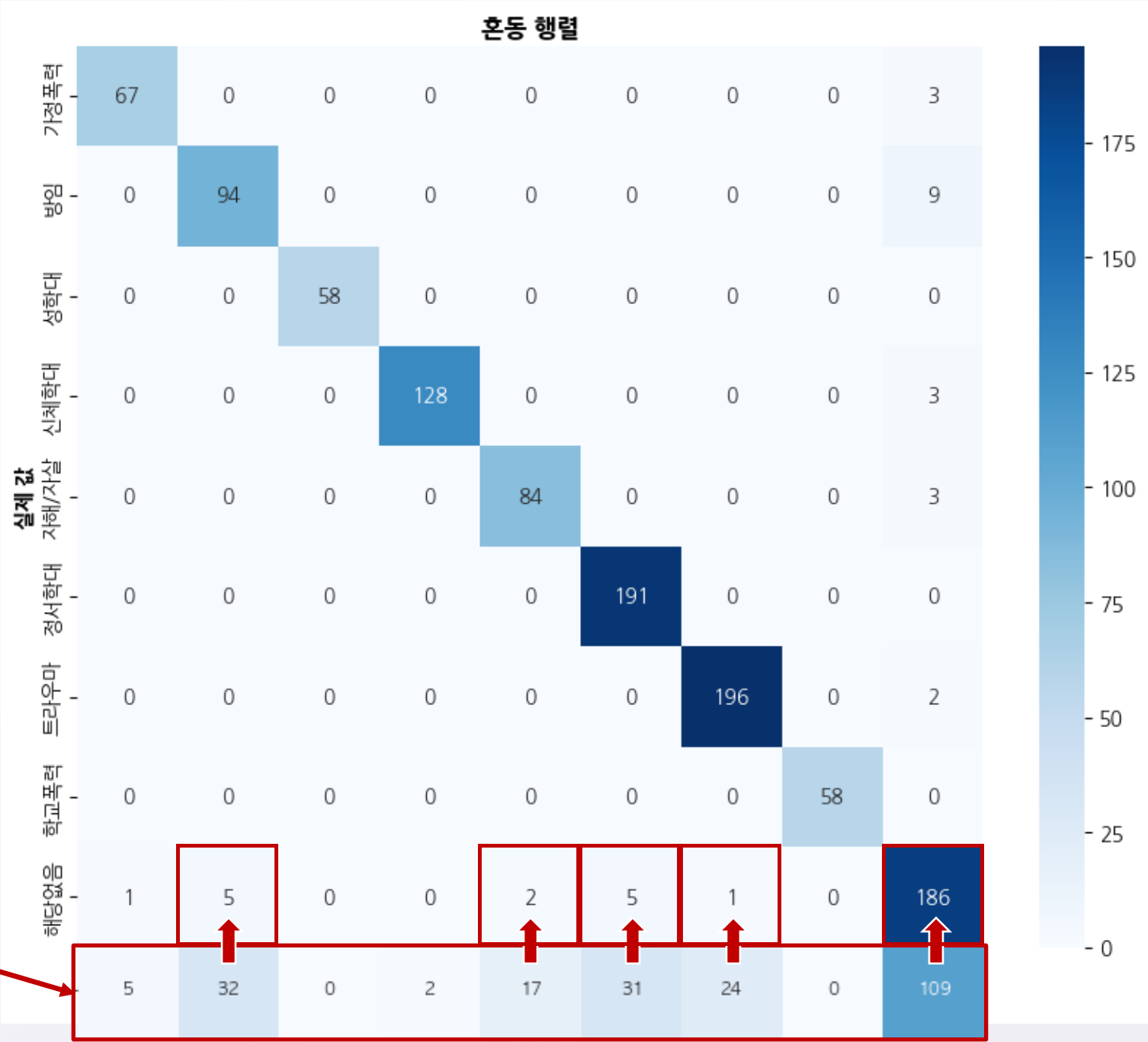
# 결과 평가 및 비교 분석

## 모델 학습 결과

지표	KoBert	SMO	Naive bayes
정확도	96.9%	95.4%	89.7%
정밀도	90.3%	93.0%	90%
재현율	93%	80.0%	89%

## 기타 요소 비교

지표	KoBert	SMO	Naive bayes
학습 시간	7분 32초	12초	5초
모델 크기	368MB	1MB 이내	1MB 이내



정확도가 모든 것을 대표 할 수 없다

# 모델 활용 방안

## 아동학대 조기 발견 및 예방

- ❖ **비침해적 학대 징후 파악:** 아동과의 대화를 통해 자연스럽게 학대 징후를 파악할 수 있다.
- ❖ **신속한 대응 가능:** 인공지능 모델을 활용하여 학대 여부를 빠르게 판단하고, 필요한 조치를 신속하게 취할 수 있다.
- ❖ **건강한 성장 지원:** 안전한 환경에서 아동이 건강하게 성장할 수 있도록 지원한다.

## 다양한 사회적 분야에서의 활용

- ❖ **교육 현장:** 교사들이 학생들과의 상담 시 학대 징후를 파악하는 데 도움을 줄 수 있다.
- ❖ **상담 센터 및 복지 기관:** 전문 상담사들의 업무 효율을 높이고, 더 많은 아동들을 지원할 수 있다.
- ❖ **정책 수립 지원:** 데이터를 기반으로 한 분석을 통해 정부 및 관련 기관의 정책 수립에 기여할 수 있다.



# Weka와 Python의 비교

## Weka의 장단점 분석

특성	Weka	Python
최종 모델 성능	95%, 재현율 한계 있음.	96%, 종합적 분류 수행
사용 편의성	GUI 제공, 초보자 친화적	코딩 필요, 유연성 높음
알고리즘에 대한 이해도	요구됨 + 지원되는 데이터 유형을 알려줌	더 크게 요구됨
정보량	고정된 많은 정보량	목적한 만큼의 정보량(아는 만큼 획득)
시각화	간단하며 고정된 표현 가능.	자유 표현 가능, 코딩 구현 요구됨
최적 하이퍼 파라미터 탐색	간단한 적용, 수동적 탐색	코딩 필요, GridSearchCV를 통한 자동화

# 과제 수행의 느낀점과 의의

## 과제 수행의 느낀점, 다시 할 경우...

- ❖ Weka의 필요성을 확인 할 수 있었다.
- ❖ 불필요한 공부는 없다는 것을 재확인 할 수 있었다.
- ❖ 과제를 다시 수행할 경우 Weka를 통해 다양한 데이터를 적용한 뒤 다양한 데이터에 대한 다양한 접근 시도.
- ❖ 시간을 더 투자하여 주제에 대한 더 깊이 있는 분석

## 과제 수행의 의의

- ❖ Weka를 통해 간단히 데이터 모델링 및 분석.
  - ❖ 데이터 모델링 및 분석의 유효성 검증
  - ❖ 시각화를 통해 다른 사람에게 주제 소개 및 데이터 분석 예상 결과를 제시 및 설득.
- ❖ 위 내용을 기반으로 Python을 통해 정밀한 모델링 및 시각화 수행.