ETRI

2019
Electronics and
Telecommunications
Trends

# 디지털 헬스케어 데이터 분석을 위한 머신 러닝 기술 활용 동향

Trend of Utilization of Machine Learning Technology for Digital Healthcare Data Analysis

**우영춘** (Y.C. Woo, ycwoo@etri.re.kr) IDX 원천기술연구실 책임연구원
**이성엽** (S.Y. Lee, sylee549@etri.re.kr) IDX 원천기술연구실 UST 연구생
**최 완** (W. Choi, wchoi@etri.re.kr) IDX 원천기술연구실 책임연구원
**안창원** (C.W. Ahn, ahn@etri.re.kr) IDX 원천기술연구실 책임연구원/실장
**백옥기** (O.K. Baek, ok.baek@etri.re.kr) IDX 원천기술연구실 책임연구원/연구위원

Ⅰ. 서론
Ⅱ. 디지털 헬스케어 데이터 특징
Ⅲ. 머신 러닝 데이터 전처리 및 평가
Ⅳ. 머신 러닝 모델의 기본 유형
Ⅴ. 머신 러닝 프레임워크
Ⅵ. 설명 가능한 머신 러닝 분석 도구
Ⅶ. 결론

Machine learning has been applied to medical imaging and has shown an excellent recognition rate. Recently, there has been much interest in preventive medicine. If data are accessible, machine learning packages can be used easily in digital healthcare fields. However, it is necessary to prepare the data in advance, and model evaluation and tuning are required to construct a reliable model. On average, these processes take more than 80% of the total effort required. In this study, we describe the basic concepts of machine learning, pre-processing and visualization of datasets, feature engineering for reliable models, model evaluation and tuning, and the latest trends in popular machine learning frameworks. Finally, we survey a explainable machine learning analysis tool and will discuss the future direction of machine learning.

* DOI: 10.22648/ETRI.2019.J.340109

*본 연구는 2018년도 정부(미래창조과학부)의 재원으로 정보통신기획평가원의 지원[No.2016-0-00087, 대규모 딥러닝 고속 처리를 위한 HPC 시스템 개발]과 2018년도 정부(미래창조과학부)의 재원으로 정부출연금 주요사업의 지원을 받아 수행된 연구임[No.18ZS1230, IDX Foundation 원천기술연구].

본 저작물은 공공누리 제4유형
출처표시+상업적이용금지+변경금지 조건에 따라 이용할 수 있습니다.



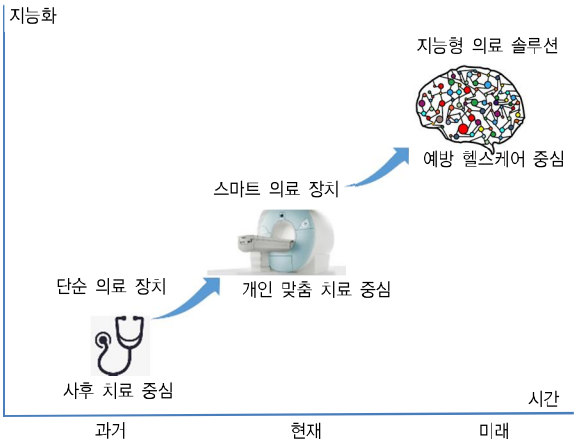
© 2019 한국전자통신연구원

## Ⅰ. 서론

여러 분야에서 이슈화되는 4차 산업혁명 시대인 현재, 전세계적으로 고령 인구의 비율이 빠르게 증가하고 있다. 급속한 고령화 사회로의 변화와 의료기술의 발전으로 병을 발견하고 치료하는 것에서 한 단계 더 나아가 병을 사전에 예방하여 삶의 질을 올리는 방향으로 사람들의 관심이 이동하고 있다. 이와 관련하여 의료 서비스는 (그림 1)에서 보는 바와 같이 아날로그 기기에 의한 사후 치료 중심에서 벗어나 ICT를 결합하여 지능형 의료 솔루션에 의한 예방 헬스케어로 급속히 변화하고 있다[1], [2]. 예방 헬스케어는 데이터 과학과 지능형 솔루션을 기반으로 헬스케어 빅데이터를 분석하여 질병이 발생하지 않도록 사전에 조치함을 의미한다.

헬스케어 빅데이터는 의료 장치, 스마트 의료 장치, 웨어러블 장치 등으로부터 폭발적으로 증가하고 있다. 이들 데이터의 80%는 텍스트, 수치, 이미지, 영상과 같은 비구조화 데이터이다. 의료 장치는 평균 초당 1,000개의 수치를 측정하여 하루에 1인당 86,400개 수치 데이터를 생성하고 있으며, 의료 장치를 사용하는 환자는 최근 평균 18% 이상 증가를 보이고 있다고 한다[3].

그동안 헬스케어 빅데이터는 데이터 간에 융합과 상호 연관성을 찾는데 어려웠지만, ICT 컴퓨팅 인프라의 성능 향상과 다양한 머신 러닝 프레임워크 등장으로 분석이 가능해졌고, 이를 바탕으로 인공지능 기반 디지털 헬스케어 분석 기술은 스마트 헬스케어 시대로 패러다임을 변화시키고 있다. 스마트 디지털 헬스케어란 빅데이터와 인공지능, 클라우드, 사물인터넷 등을 융합하여 개인이 장소에 구애받지 않고 언제든지 건강을 관리하는 개념으로 보고 있다[4].




(그림 1) 디지털 스마트 헬스케어 패러다임

헬스케어를 가능하게 한 머신 러닝[5]이란 주어진 데이터에 통계 기술을 사용하여 학습을 제공하는 컴퓨터 과학 분야로 정의하고 있다. 여기에서 학습은 컴퓨터에 데이터를 입력하여 데이터의 패턴을 인식하도록 가르치는 모델을 만들고 데이터를 이해하는 알고리즘으로 설명할 수 있다. 여기서 학습하는 과정을 훈련(Training)이라 하며, 이 과정에서 생산되는 결과물을 모델이라고 한다. 이전에 학습한 것을 토대로 이 모델을 만들면 이 모델을 이용하여 새로운 정보를 추론할 수 있다. 머신 러닝 모델은 인간 뇌로 처리할 수 없는 고성능 컴퓨터 능력을 활용하여 일련의 규칙을 결정하게 된다. 따라서 머신 러닝 모델에 많은 데이터를 입력할수록 규칙은 더 복잡해지며 더 정확한 예측 모델이 구성된다. 대규모 데이터 준비도 중요하지만, 더 정확한 예측 모델을 구성하기 위해서는 전처리(Preprocessing) 과정을 거쳐 원시(Raw) 데이터를 잘 정제하여 완전성, 정확성, 일관성 있는 데이터 셋을 준비해야 한다.

머신 러닝 프레임워크는 고성능 컴퓨팅 시스템이 보급됨에 따라 예측 정확도를 높이고 예측 성능을 높이는 방향으로 발전하고 있다. 최근에는 의사결정을 예측한 요인과 원인을 알게 하는 설명 가능한(Explainable) 인공지능을 연구하고 있다.

본고 에서는 머신 러닝 기술을 활용한 디지털 헬스케어 데이터 분석과 관련된 데이터셋 전처리, 머신 러닝 프레임워크, 모델 구성, 모델 평가를 소개하고, 최근 이슈화되고 있는 설명 가능한 머신 러닝 도구를 소개하고 결론을 맺는다.

## II. 디지털 헬스케어 데이터 특징

헬스케어 분야는 스마트 장치가 고급화되고 널리 보급되어 환자로부터 얻는 많은 데이터를 더 이상 사람이 인위적으로 분석할 수 없게 되었다. 헬스케어가 인공지능과 만나면서 뜨는 이유는 서론에서 설명한 바와 같이 그 중심에 빅데이터가 있다. 머신 러닝에서 데이터가 없다면 지능적 시스템을 구성할 수 없으며, 머신을 학습시키기 위해서는 데이터 예시가 필요하다. 이러한 수많은 예시 집합을 데이터셋이라 하고 이때 각 데이터에 대하여 표식을 붙여 정체를 나타내도록 하는 것이 레이블이다. 헬스케어 데이터도 이처럼 데이터와 레이블 쌍 집합을 잘 구성해야 한다. 머신 러닝을 위해 대규모 데이터셋으로부터 더 정밀한 모델을 구성함으로써 데이터 패턴이나 설명 가능한 이유를 찾아내어 전문의로 하여금 예방 헬스케어를 처방할 수 있게 한다.

앞으로 헬스케어 빅데이터와 인공지능 플랫폼이 상호 발전하기 위해서는 고품질 데이터의 제공도 선행되어야 한다고 본다. 디지털 헬스케어 원시 데이터는 복잡성, 부정확성, 데이터 누락이 많은 특성을 가지고 있다, 질병 진단은 ICD-10 표준 진단 코드에 따라 제대로 부여되어 있는지 점검이 필요하며, 데이터 특성상 개인정보보호, 제한된 데이터 등 피쳐(Feature)를 포함하는지 등에 대하여 윤리와 법적인 내용을 반드시 점검할 필요가 있다. 개인정보보호 문제를 해결하기 위한 비식별화 조치는 '개인정보 비식별 조치 가이드라인'에 따르고 다른 정보와 결합하더라도 재식별이 불가능하도록 해결해야 한다[6].

의료 및 헬스케어 데이터에는 기초생성 정보, 문진 정보(질환력), 기초검사(몸무게 등), 검체검사(혈액 등), 영상검사(MRI 등), 소견 등을 기록한 EHR(Electronic health records)[7], 건강보험공단, 건강보험심사평가원 등의 공공기관 데이터, 유전체, 단백체, 후성유전체 등 분자 수준에서 생성된 오믹스 데이터, 임상시험 데이터, 스마트 기기로부터 생성된 웰스케어 라이프로그 데이터, 기타 포털로부터 수집되는 헬스케어 데이터 등이 있다.

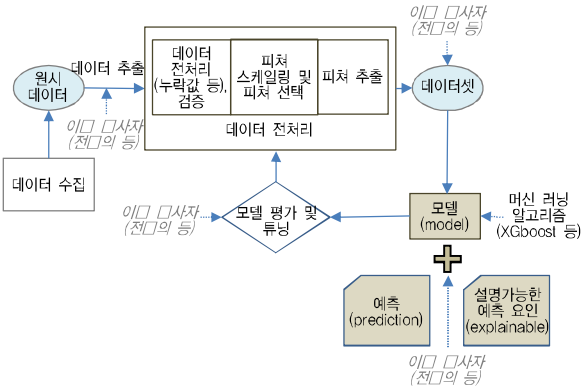
## III. 머신 러닝 데이터 전처리 및 평가

### 1. 데이터셋 전처리

실제로 원시 데이터(Raw Data)는 완전하지 않으며, 노이즈가 많고 일관성이 없고 중복된 데이터가 많다. 이런 데이터를 그대로 사용할 경우 트레이닝 과정에 상호 관련성이 미비하고 중복이 많아 예측 결과를 제대로 도출하지 못할 수 있다. 이런 데이터의 전처리 작업에 평균 80% 이상 노력이 필요하다고 알려져 있다[8]. 데이터 피쳐가 모델의 품질이나 정확성 성능에 크게 영향을 주기 때문에 원시 데이터 그대로는 머신 러닝 모델을 만드는 데 거의 사용되지 않는다.

예를 들어 헬스케어 데이터는 검사 회차 등의 이유로 데이터셋 내에 누락데이터가 있을 수 있다. 누락데이터 처리 중에서 가장 간단한 방법은 데이터셋에서 관측행을 제거하는 방법이다. 하지만 관측행이 중요한 데이터일 수 있으므로 무조건 제거하는 것은 좋지 않다. 누락값을 관측열의 평균을 취하는 방법도 한 가지 방법이다. 피쳐 스케일링은 가끔 데이터셋에서 값의 크기, 단위, 범위가 다를 수 있다. 트레이닝하면서 더 큰 피쳐값이 예측 모델을 지배할 수 있기 때문에 유클리드 거리 등의 알고리즘을 사용하여 스케일하고 표준화시켜 주어야 한다. 머신 러닝 알고리즘을 적용하기 전에 데이터셋 전처리 단계에서 최소한으로 누락데이터, 범주데이터, 피쳐 스케일을 처리할 필요가 있다. 다음은 원시데이터를 전처리하는 몇 가지 방법을 보여준다[9].

- 데이터 감소(Reduction): 중복, 필요한 데이터 속성을 선택 혹은 삭제
- 데이터 변환(Transformation): 모델의 필요에 따라 원시데이터를 지정된 형식으로 변환(Normalization,




(그림 2) 머신 러닝 분석 흐름

Aggregation(features), Generalization(standard))

- 데이터 통합(Integration): 다양한 원시 소스로부터 데이터 결함(Consistence)
- 데이터 정리(Cleaning): 누락값 채우기, 이상한 값 삭제

사람이 전문성, 통찰력을 기반으로 헬스케어 데이터를 분석하고 조직화하고 필터링함으로써 원하는 데이터를 제공하는 것을 큐레이션(Curation)이라 한다. 큐레이션에서는 데이터 레이블링, 데이터 검증분석(데이터 오류 수정, 중복제거, 불일치 데이터 제거, 충돌 데이터 조정 등), 데이터 버전 관리, 시계열 데이터와 같은 예측 데이터 분석 및 알고리즘 적용, 데이터 지속성 관리 등을 처리한다. (그림 2)는 예시를 든 스마트 헬스케어 장치 혹은 검사 장비로부터 원시 데이터를 추출하는 단계부터 관측치 예측과 설명 가능한 예측 요인 제시까지 전 단계를 보여 준다. 여기서 짚고 넘어가야 할 부분이 머신 러닝 기술자 혹은 데이터 과학자가 독단적으로 데이터를 처리할 수도 있지만, 고품질 데이터를 얻고, 고품질 모델을 만들기 위해서는 이해당사자와 협동 작업을 해야 함을 강조하고 싶다.

## 2. 데이터셋 시각화 도구

데이터 시각화[10]는 데이터 과학자에게는 큰 부분을 차지하고 데이터 현황을 전체적으로 파악하는 데 도움을 준다. 대규모 데이터를 시각화로 처리하면 데이터셋을 명확하게 이해하기가 쉬워진다. 데이터 시각화에는 산포도, 선그림, 히스토그램, 바플롯, 박스플롯, 네트워크 그래프 등의 방법이 있다. 가장 널리 사용하는 파이썬 라이브러리 matplotlib를 비롯하여 몇 가지 시각화 라이브러리 및 도구를 소개한다.

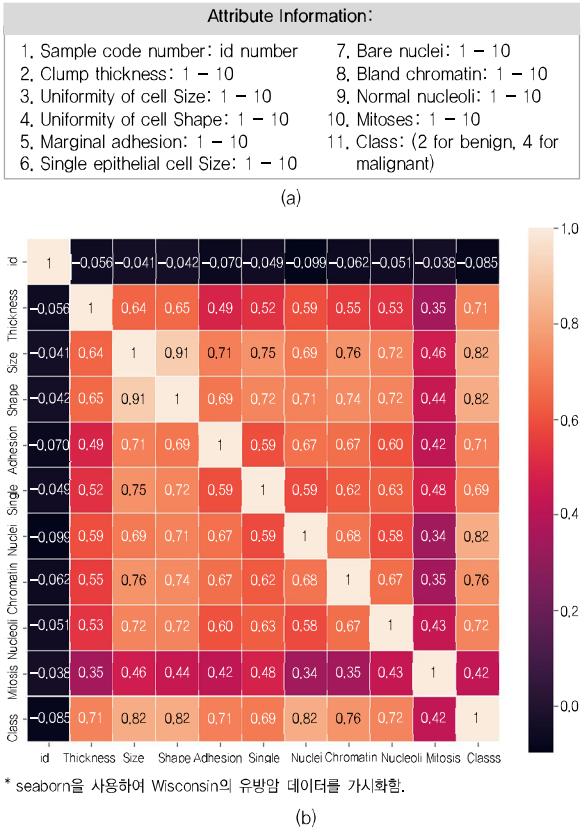
Facets[11] 데이터셋 시각화 도구는 머신 러닝은 많은 데이터셋으로부터 학습하여 패턴을 찾아 모델을 구성하는 것으로서 먼저 데이터셋을 이해하는 것도 중요하다. Facets는 머신 러닝 데이터셋을 이해하거나 분석을 위한 가시화 도구이다. 이 도구는 데이터셋으로부터 값 분포를 통계 분석하고 시각화한다. 데이터셋에서 불균형 분포, 특이한 값, 누락값(Missing Value), 특이한 트레이닝 데이터 및 테스트 데이터 패턴 등을 시각화한다. 그리고 Facets Dive는 데이터셋에서 피처 간에 관계를 볼 수 있게 하고, 데이터 포인트와 이와 연관된 데이터이거나 잘못된 레이블을 시각적으로 보여 주는 대시보드 기능을 제공한다.

이외에도 파이썬 라이브러리와 같이 데이터셋 시각화하기 위하여 응용할 수 있는 도구들이 많이 있다. 데이터셋 시각화 도구에는 Matplotlib, Seaborn, Bokeh, Pygal, Plotly, Missingno, Folium, Spyder 등이 있고, 네트워크 그래프 기반 데이터 조작 및 가시화 도구에는 pyplot, PyGraphviz, NetworkX, Pydot 등이 있다[〈표 1〉 참조].

Wisconsin 유방암 데이터(Breast Cancer Wisconsin (Original) Data Set)는 699명 관측치가 있고 피처는 9개로 구성되어 있으며 각 피처의 값으로 1~10 값을 가진 데이터셋이다[12]. 본고에서 이 사례 데이터를 기반으로 파이썬 프로그램으로 가시화해 보고, 모델을 만들고, 이에 대한 지표 도출해 본다. 마지막으로 최근 이슈화되고 있는 예측 결정에 도달하게 한 설명 가능한 예측 요인 분석 도구를 적용해 본다.

〈표 1〉 데이터 가시화 도구

| 패키지 | 주요 기능 |
|---|---|
| matplotlib | 최초 파이썬 시각화 라이브러리로써 데이터셋을 2차원의 그래프로 표현 |
| seaborn | matplotlib를 기반의 시각화 라이브러리로써 더 세련된 디자인을 지원하며 사용법이 유사[13] |
| Bokeh | 웹브라우즈에서 데이터셋 시각화를 지원하는 라이브러리 |
| pygal | Bokeh 와 같이 웹 브라우드에 삽입할 수 있는 대화형 시각화 도구 |
| plotly | 데이터셋의 영역 차트, 통계 차트, 하트 차트 등 차트 생성 그래픽 라이브러리 |
| missingno | 누락된 데이터 상태를 포함하여 데이터셋을 전체적으로 측정해 볼 수 있는 도구[14] |
| folium | 지리데이터를 시각화에 유용한 라이브러리 |
| spyder | 데이터 분석가가 데이터 탐색, 대화식 실행, 데이터셋의 시각화를 지원 |
| blaze | Numpy 기능을 확장한 데이터셋 가시화 및 대시보드 지원 라이브러리[15] |
| pyplot | MATLAB과 유사한 데이터셋 플로팅 라이브러리(matplotlib 모듈) |
| Geoplot | 파이썬에서 지리 데이터 가시화 라이브러리 |
| PyGraphviz | 그래프를 텍스트로 설명하고, 편집하고, 작성할 수 있는 시각화 도구 |
| NetworkX | 네트워크 구조 표현 및 분석, 생성, 변경을 제공하는 파이썬 패키지 |
| igraph | 그래프를 생성하고 조작하는 라이브러리(웨이브 그래프 등 지원)[16] |
| Pydot | GraphViz에서 사용하는 DOT 언어로 구분 분석하는 파이썬 인터페이스 제공 |




(그림 3) (a) Breast cancer 데이터 속성과 (b) 가시화

(그림 3)은 유방암 데이터에 대하여 seaborn 파이썬 라이브러리를 이용하여 히트맵으로 가시화한 것이다. 히트맵은 데이터의 그래픽 표현으로 많은 데이터 분석 상황을 시각적으로 알 수 있게 한다. 이 외에도 앞에서 설명한 각 라이브러리에서는 시각화 메소드를 많이 제공하므로 데이터 분석자에게 맞게 사용하면 된다.

## 3. 피쳐 공학

머신 러닝은 중복되고 명확하지 않은 피쳐를 가지고는 정확한 예측 정답을 내지 못한다. 피쳐 공학(Feature Engineering)은 머신 러닝 모델의 성능을 위해 기존 데이터로부터 새로운 피쳐를 생성하는 것이다. 위키피디어에 따르면 피쳐 공학[17]을 도메인 지식을 활용하여 피쳐를 만들어내는 과정으로 설명하고 있다. 피쳐 추출(Feature Extraction)은 모든 변수를 조합하여 원본 데이터르부터 잘 표현할 수 있는 중요 성분을 가진 새로운 변수를 추출하는 방법이다(예를 들어, 피쳐 A, B, C D를 사용하여 가, 나, 다를 만드는 것으로서 생년월일과

현재 날짜로부터 나이를 유도하거나, 원본 이미지로부터 픽셀 정보 추출하는 경우). 원시 데이터 값을 그대로 입력 피쳐로 사용하면 머신 러닝 모델이 실제로 큰 피쳐 값으로 편향되는 되므로 이때는 피쳐 스케일링이 필요하다. 주로 사용하는 방법에는 표준 스케일링, min-max 스케일링, Robust가 있다. 데이터셋에서 더 많은 피쳐를 가지는 것은 모델을 복잡하게 하고 해석을 더 어렵게 하고 과적합이 발생하기도 한다. 변수 간에 중첩이 있는지, 어떤 변수가 중요한 변수인지, 어떤 변수가 타겟에 영향을 크게 주는 변수인지를 분석할 필요가 있다. 피쳐 선택(Feature Selection)은 피쳐의 최적의 수를 선택함으로써 과적합을 방지한다. 중첩되는 변수를 찾을 때 주로 사용하는 방법은 상관분석(Correlation)을 사용하고, 상관분석 계수가 높거나 분산팽창지수(VIF: Variance Inflation Factor)가 높은 중첩되는 변수들 중 하나만을 선택하는 방법이 있다. 타켓 레이블(종속 변수)에 영향을 크게 미치는 변수는 랜덤포레스트(Random Forest) 또는 XGBoost 등 머신 러닝 기법을 이용해 변수 중요도(Variable Importance)를 찾아낸 후 몇 가지 변수만 선택하는 방법도 있다.

## 4. 모델 평가

원시 데이터를 수집하고 현장 실무자 등 이해당사자와 협의를 통하여 전처리한 데이터셋을 만들고 모델을 만든 후, 다음 단계로 과연 이 모델이 예측 성능이 어떻게 되는지 찾아보는 단계가 모델 평가이다. 예측 정확도 성능이 우수하고 유명한 알고리즘을 사용하였다 하여 좋은 모델을 만들었다고는 말할 수 없다.

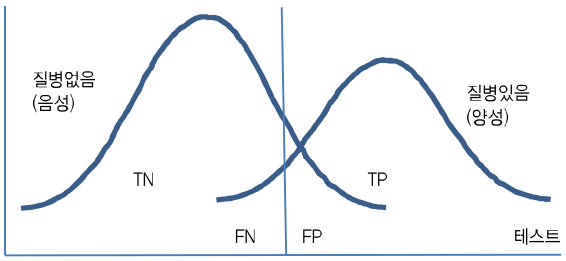
(그림 4a)는 분류행렬(Confusion Matrix)로 머신 러닝 분류 성능을 평가하는 데 사용된다. 이는 전체 데이터셋에 대하여 데이터 포인터와 예상 클래스 레이블에 실제 레이블과 비교를 반복하여 행렬 형식으로 표현한 것이다. 실제 데이터 레이블을 알고 예측(Predicted) 데이터

| | | 예측 결과(Predicted 레이블) | |
|---|---|---|---|
| | | n'<br>(Predicted) (0)(양성) | p'<br>(Predicted)(1)(음성) |
| 실제 결과(True레이블) | n<br>(True)<br>(0)<br>(양성) | 진짜 음성(True negative)<br>(true 클래스 레이블이 predicated 클래스 레이블과 동일한 negative 클래스의 총 인스턴스 수)<br>(Specificity) | 거짓 양성(False positive)<br>(모델이 positive로 예측하여 잘못 분류한(misclassified) negative 인 클래스의 총 인스턴스 수<br>(Type I error) |
| | P<br>(True)<br>(1)<br>(음성) | 거짓 음성(False negative)<br>(negative인 것으로 예측하여 잘못 분류한(misclassified) positive인 클래스의 총 인스턴스 수<br>(Type II Error) | 진짜 양성(True positive)<br>(true 클래스 레이블이 predicated 클래스 레이블과 동일한 긍정적인(positive) 클래스 총(instances) 수)<br>(Sensitivity) |

(a)

```
print (metrics.confusion_matrix(y_test, y_pred_class))

[[ 104      3]
 [     3   61]]
```




(b)

(그림 4) (a)분류 행렬과 (b)유방암 데이터의 분석(XGBoost적용) 사례

레이블과 비교할 수 있는 분류 모델의 성능을 평가하는 데 유용하다. 각 열(Column)은 모델의 예측을 기반으로 분류된 인스턴스(Classified Instance) 수를 나타내고 행렬의 각 행은 실제(Actual, True) 클래스 레이블(Class Label)을 기반으로 인스턴스 수를 나타낸다. 예를 들어, 민감도(Sensitivity)는 질병에 걸렸을 때 걸렸다는 검사 결과에 대한 비율이고 특이도(Specificity)는 질병에 안 걸렸을 때 안 걸렸다는 검사 결과에 대한 비율을 나타낸다.

헬스케어 데이터 등을 대상으로 머신 러닝 학습 결과를 평가하는 지표(Metric)에는 정확도(Accuracy), 정밀도(Precision), 재현율(Recall), F-점수(F Score) 등이 있다[18].

정확도는 분류 성능의 보편적인 지표이고 모델의 전체적인 예측 비율을 나타낸다(전체 데이터 수 대비하여

정답과 일치한 수의 비율, (TP + FN) / (TP + FP + TN + FN). 100명을 대상으로 양성 60명을 예측했다면 정확도는 60%가 된다. 결국은 100명 모두를 양성으로 예측하였더라도 경우 과반이 넘는 60% 정확도를 보이므로 신빙성이 떨어진다. 또한, 데이터 분포가 쏠릴 경우 정확도에도 신빙성이 없다. 정밀도는 데이터 분포를 고려하여 출력 결과가 정답을 맞춘 지표를 말한다(TP / (TP + FP). 이는 양성(맞음)으로 예측한 결과에서 실제로 맞는 것의 비율을 의미한다. 이는 전체 정확도는 감소하더라도 최대 양성 수를 찾는 것이 중요할 경우 이 지표를 사용한다.

재현율은 민감도(Sensitivity), 적중률(Hit Rate), 커버리지(Coverge)라고도 하는데, 실제로 양성(맞음)인 것 중에서 양성이라고 예측한 수를 비율로 나타낸다(TP / (TP + FN). 예를 들어 실제로 암인 사람 중에서 암으로 예측한 사람의 비율이다.

F-점수는 정밀도와 재현율을 조합으로 만든 점수를 나타낸다. 정확한 예측이 필요할 경우 정밀도를 중요시하고, 혹시 잘못 예측하더라도 빠뜨리는 것이 없도록 하려면 재현율을 중요시한다.

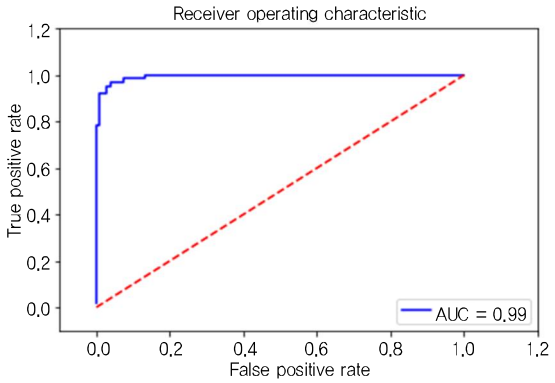
특이도(Specificity)는 민감도와 반대로 실제로 음성 것 중에서 음성이라고 예측한 수를 비율로 나타낸다(TN / (FP + TN). 머신 러닝 분류 모델 성능을 보여주는 그래프로 ROC(Receiver Operating Characteristic), AUC (Area Under Curve)가 있다[19].

민감도와 특이도 관계를 그래프로 나타낸 것이 ROC 곡선이다. 여기서 ROC 곡선의 아래 면적 값을 AUC라 한다. 모델로 예측한 값이 100% 정확하다면 1.0이 된다. 모델 평가에서 성능을 도출하는 것은 데이터 학습 예측을 헬스케어에 적용 시 품질을 확보하기 위한 단계이다. 정확성(Accuracy) 성능이 높다고 하여 헬스케어 분야에서 품질이 좋다는 것이 아니므로 평가 지표를 적절히 활용할 필요가 있다. (그림 5)는 '0'으로 예측한 데이터의 97%(Precision)만 실제로 '0' 이며, 예측한 데이터의 95는 '1' 임을 나타낸다. 그리고 실제 '0'인 데이터의 97%(Recall)는 '1'로 예측하였고 이 중에서 '1'인 데이터 중의 95%가 '1'로 양성(맞음)을 나타낸다. (그림 6)은 클래스 예측 기준값에 대한 ROC 곡선으로 AUC는 '1'에 가까운 0.99 에 가까워서 민감한 모델이라 한다.

```
from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred_class))

              precision    recall  f1-score   support
           0       0.97      0.97      0.97       107
           1       0.95      0.95      0.95        64
   avg/total0.96       0.96      0.96       171
```

(그림 5) 유방암 데이터의 모델 평가 지표




(그림 6) 유방암 데이터의 모델 평가 결과의 ROC 곡선

## Ⅳ. 머신 러닝 모델의 기본 유형

머신 러닝 모델의 기본 유형 중에서 분류(Classification)는 감독 학습(Supervised Learning)의 일종으로 데이터 속성과 레이블 혹은 카테고리 간의 연관성을 학습하여 새로운 관측자 데이터의 카테고리를 결정하는 것이다. 분류는 레이블 된 데이터 집합을 사용하여 모델을 트레이닝한다. 예제로 많이 사용하는 이미지 분류도 이 분류 문제 중의 하나이다. 예를 들어 유방암의 유무를 예측하는 것이 '맞다', '아니다'로 구분하는 문제라 하면 이진(Binary) 분류라 부르고, 건강 상태 등급을 'A, B, C, D'로 나누는 문제라면 멀티클래스(Multi-class)분

류라 부른다. 널리 사용되는 분류기(Classifier)에는 트리 구조 기반의 결정트리(Decision Tree), 베이시안 확률 기반의 나이브 베이즈(Naïve Bayes), 다수결 투표 기반의 KNN(K-Nearest Neighbors Algorithm) 등이 있다[20].

회귀 분석(Regression)은 연속된 값과 함수관계를 관찰하고 예측하는 문제이다. 데이터셋을 가장 잘 설명하는 직선이나 함수 곡선을 찾고자 할 때 회귀를 사용하는 것과 같다[21]. 이 곡선에 대하여 임의 입력에 대하여 새로운 출력을 예측하는 하는 문제로 볼 수 있다. 예를 들어 어떤 만성 환자의 재입원 시기를 예측하는 경우로 환자들의 나이, 성별, 만성 환자들의 재입원한 시기 등의 데이터셋의 데이터 패턴을 트레이닝하여 회귀 분석 모델을 얻는다. 예측 변수가 바이너리이고 모든 예측이 서로 독립적이고 데이터 누락 없을 때 동작하는 단점이 있다. 널리 사용되는 회귀 분석에는 입력과 출력 사이에 선형 관계를 가정하는 선형 회귀(Linear Regression), 비선형 로그 변환 적용 기반의 로지스틱 회귀, 비확률적 선형 분류 모델 기반의 SVM(Support Vector Machine) 회귀 등이 있다.

클러스터링(Clustering)은 피쳐(Feature)가 유사한 데이터들을 그룹화하는 머신 러닝 모델 유형이다. 이는 레이블 되지 않은 데이터를 그룹화하는 경우이므로 비지도학습(Unsupervised Learning) 학습 방법이라고 한다[22]. 머신 러닝 분야에서는 그룹화 구분 능력에 더 중점을 둔다. 예를 들어 특정 지역(산업 공장 오·폐수 문제, 등)에 사는 사람들이 암 발병률이 높다는 것을 보여주는 모델로 이용할 수 있다. 클러스터링 알고리즘으로 환자 그룹을 나누었다면 그 다음에는 인위적으로 레이블을 지정할 수 있다. 널리 사용되는 클러스터링 알고리즘에는 K개 그룹을 위한 데이터 분할법 기반의 K-평균(K-Means) 알고리즘, 데이터 포인터의 가우스 분포를 가정한 GMM(Gaussian Mixture Model), 데이터셋을 계층적 혹은 파티션 기반으로 그룹화하는 계층 클러스터링(Hierarchical Clustering) 알고리즘 등이 있다.

딥러닝(Deep Learning)은 뉴럴 네트워크가 뇌의 동작을 모방하여 이를 알고리즘화하는 것으로 네트워크를 구성하기 위해 레이어로 연결되는 많은 뉴런이라는 노드를 포함한다[23]. 최소 레이어로 입력층, 출력층은 가져야 하며, 이 레이어 사이에 더 많은 정보를 얻기 위하여 여러 은익층을 두어 심층 레이어로 구성할 수 있다. 하지만 신경망은 헬스케어 이미지나 뇌 이미지와 같이 고도로 구조화된 문제를 해결 하는 데는 유용하지만 '설명 가능하지 않은' 블랙박스인 것을 단점으로 보고 있다.

## V. 머신 러닝 프레임워크

### 1. 랜덤포레스트(Random Forest) 프레임워크

머신 러닝에서 랜덤 포레스트(Random Forest)는 여러 개의 결정 트리를 생성한 다음 각 트리 예측값 중에서 가장 많은 선택을 받은 레이블로 예측하는 앙상블 학습 방법의 한가지로, 학습 단계와 테스트 단계로 구성된다[24]. 이 방법은 트레이닝 데이터에 과대적합(Over-fitting)하는 단일 결정 트리 학습의 단점을 해소한 방법으로 분류, 회귀 등에 널리 사용된다. 이 방법의 피쳐는 임의성(Randomness)으로 데이터 포인트를 무작위로 선택하는 것이고, 테스트에서 노드 분기 특성을 무작위로 선택하는 방법이다. 그리고 노이즈가 있는 데이터에 대해 배깅(Bagging, Bootstrap Aggregating)과 임의 노드 최적화의 앙상블 방법으로 처리된다. 이 방법은 상대적으로 측정 가능한 피쳐를 도출 가능한 것이 장점이나 데이터셋이 적은 경우에 예측 성능은 높지 않다.

### 2. 애다부스트(Adaboost)와 그래디언트 부스팅 (Gradient Boosting) 프레임워크

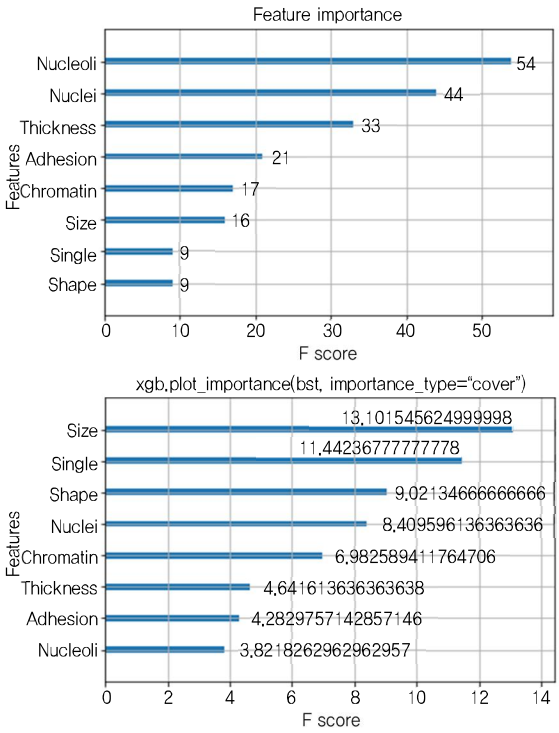
부스팅 학습방법은 여러 개의 간단한 모델을 사용하거나 기본 학습기 혹은 성능이 약한 학습기를 여러 개 연결하여 성능이 높은 학습기를 만드는 앙상블 학습 방

법으로 더 나은 학습 모델로 진행시켜 학습하는 방법이다[25]. 이 방법에는 애다부스트(Adaboost)와 그래디언트 부스팅(Gradient Boost) 방법이 있다.

애다부스트[26]는 전체 학습 데이터셋을 이용하여 첫 모델을 만든 후 상대적으로 가중치를 높인 후 두 번째 모델에 대하여 이 가중치로 업데이트하여 모델을 학습해 가면서 학습이 어려운 데이터에 대하여 적합(Fit)되도록 반복 처리하는 방법이다. 그래디언트 부스팅은 데이터셋에 대하여 모델을 만든 후 모델의 오차(잔여 오차, Residual Error)를 수정 즉, 손실 함수(Cost Function)를 정의하고 경사 하강법을 사용하여 다음에 추가될 트리가 예측해야 할 값을 보완하는 방법으로 모델을 추가해 주는 방법이다[27]. 이 방법에는 특성 중요도가 존재하며 트리의 개수가 많아지는 경우 과적합 될 수 있으므로 변수 튜닝이 필요하다. 고려할 매개 변수로서 학습률(Learning Rate)이 있다. 이것은 트리 오차를 얼마나 보완할 것인지를 나타내는 변수로써 값이 크면 복잡한 모델을 만든다. 그래디언트 부스팅은 매개 변수 튜닝이 필요하며 트레이닝 시간이 길며, 트리 기반 모델의 특성으로 고차원 데이터셋에는 잘 동작하지 않는 단점이 있다. 최근 연구로는 그래디언트 부스팅 예측 문제 성능을 높인 가속 그래디언트 부스팅(Accelerated Gradient Boosting) 기술이 있다[28].

### 3. XGBoost 프레임워크

XGBoost 는 eXtreme Gradient Boosting으로 구조화된 데이터 혹은 표 형식의 데이터에 적용하여 예측 모델 구축을 지원하는 강력한 그래디언트 부스트 의사 결정 트리를 구현한 머신 러닝 프레임워크로 속도와 성능을 크게 향상시킨 것이다[28]. 가장 일반적으로 사용하는 기본 학습자는 트리를 구성할 때 CART(Classification & Regression Trees)와 같은 회귀 트리 알고리즘을 사용하고 에러를 낮추는 부스팅 기법을 사용한다. 이는 임의의 데이터를 생성 후, 이 데이터들을 데이터 속성값을 기준으로 분류하는 분류 함수를 추가하고, 이 함수들을 노드로 표현하는 방식이다. 트리는 많은 분류 함수로 구성되며 노드가 많을수록 분류는 잘 동작하게 된다. 이 프레임워크는 머신 러닝 예측 모델로 사용하고 있지만 전체 트리에서 변수의 결정 기여도를 알 수 있음으로 피쳐 중요도(Feature Importance) 패키지를 내장하고 있으므로 학습 해석 모델로 사용될 수 있다. 그래디언트 부스팅에 대하여 알고 있는 통계 학자, 데이터 과학자들은 분류, 회귀 관련 문제에 관심이 있고 그래디언트 부스팅을 사용하고자 한다면 XGBoost를 많이 사용하고 있다. 이는 파이썬을 비롯한 여러 가지 언어를 지원하며 특히 고성능 처리를 위해 GPU 환경, 클러스터 환경을 지원한다. 다음과 같이 성능 면에서 우수한 특징은 가진다. 딥러닝 문제가 아닌 머신 러닝 분류 문제의 경우 XGBoost가 널리 사용되고 있다.




(그림 7) Breast Cancer 데이터[12]의 XGBoost 적용 분석 및 피쳐 중요도 ([29]로 구성)

- CPU 코어를 사용하여 트리 구조를 병렬화 기능
- 큰 모델을 위한 분산 컴퓨팅 인프라 지원 기능
- 큰 데이터셋(메모리 크기를 넘어선)을 위한 Out-of-Core Computing 기능

(그림 7)은 앞에서 설명한 유방암 데이터를 XGBoost로 분석한 결과이며 예측 결과에 영향을 미친 중요한 피처를 설명한다. 추가로 20017년 1월에 발표된 Light GBM 프레임워크가 있다[30]. 이는 트리 노드가 많이 생겨 과적합하는 XGBoost 보다 Leaf-Wise 수직 방향으로 노드를 적게 구성함으로써 과적합 발생을 낮추고 손실(Loss)을 줄인다. 또한, 많은 데이터를 분할 별로 스케일아웃(Scale-Out) 병렬로 학습이 가능하며 정확도(Accuracy)가 높인 기술이다.

### 4. 캣부스트(CatBoost) 라이브러리

캣부스트(CatBoost)는 Category Boosting의 약자로 얀덱스사가 개발한 의사결정 트리 라이브러리를 강화한 그래디언트 부스팅(Gradient Boosting) 메모드이다. 하이퍼 파리미터 튜닝 부담을 줄여 주며, CPU 및 GPU 성능 향상을 지원한다[31]. 다른 알고리즘과 크게 다른 2가지 기능을 지원한다. 하나는 결정트리 구성시 트레인과 테스트 간 관측치 오차 처리를 구하고 비편향 전차(Unbiased Residual) 구하고 가중치를 업데이트하는 방법으로 과적합(Overfitting)을 줄여주는 순차 부스팅(Ordered Boosting)을 구현한 것이고 둘째는 범주(Category, 예: Korea, Japan, China)를 다른 수준 예를 들어 숫자로 치환하지 않아도 되는 비수치 범주화 기능을 지원하는 것이다.

## Ⅵ. 설명 가능한 머신 러닝 분석 도구

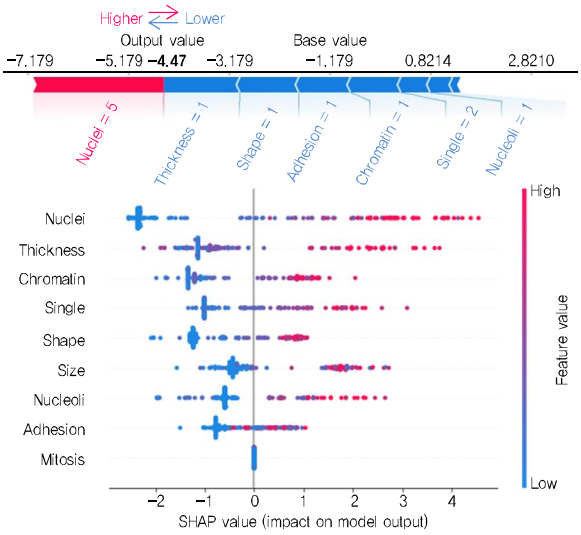
머신 러닝 사용자는 모델을 신뢰할 수 있어야 한다. 만약 수술을 앞둔 환자에게 전문의사가 “모델이 그렇게 결정했습니다”라고 하면 환자는 수술을 받지 않을 것이다. 머신 러닝에도 신뢰가 필요하며 최근까지 머신 러닝 모델은 블랙박스라고 말한다. 그래서 모델 신뢰를 위해 예측 결과에 대하여 이론적 근거를 알면 머신 러닝 사용자가 그 모델에 의한 예측을 신뢰 유무를 결정하는 데 도움이 될 것이다. 설명 가능한(Explainable) 머신 러닝은 사람이 복잡한 머신 러닝 모델의 동작을 이해할 수 하며 예측 결정에 도달한 방법이나 요인을 이해할 수 있게 한다[32].

### 1. LIME 패키지

LIME(Local Interpretable Model-Agnostic Explanations)은 모델을 예측한 결과에 대한 이해를 설명하는 도구이다. 선형 회귀와 같은 모델을 사용하여 여기에서 입력과 출력의 관계 즉 데이터에 대하여 모델 근사치를 구성하여 단일 예측을 설명하는 메소드 이다[33]. 다시 말해서 데이터 피처값을 바꾸어서 출력 예측이 어떻게 변하는지를 설명하는 방법이다. 이는 사람이 실험을 통하여 출력을 관찰하는 것과 유사하다. 이 방법은 복잡한 비선형 데이터셋은 해석 불가능할 수 있어 선명 모델에만 적용되는 단점이 있다.

### 2. SHAP 패키지

SHAP(Shapley Additive exPlanations)는 해석 가능한 머신 러닝은 사람이 블랙박스로 된 이해할 수 없는 분류 모델에 대하여 예측 결과를 이해하도록 한다[34]. 이는 머신 러닝 모델에 독립적 방식으로 샤플리 밸류(Shapley Value, 협업 게임이론에서 소개됨)로서 피쳐(Feature) 중요도를 피쳐에 할당시키는 방식으로 머신 러닝 모델에 의한 데이터 분석을 이해 가능한 것으로 설명한다. 샤플리 밸류는 모델에서 예측 변동 위험에 긍정적 혹은 부정적으로 기여하는 정보를 측정한다. 이것은 로지스틱 회귀 분석에서 중요성을 표현하는 것과 유사




(그림 8) 유방암 데이터의 XGBoost분석에 대하여SHAP에 의한 예측요인 설명([33]의 재구성)

하다. 여기서 중요도 계수(coefficient) 크기로 각 피처의 영향력을 결정할 수 있다. SHAP 값은 트리 기반 모델에 대해 계산되므로 비선형 및 정확한 모델을 구성하도록 하며 영향력을 미치는 요인을 찾도록 한다. 이상과 같이 샤플리 밸류와 SHAP 라이브러리는 머신 러닝 알고리즘이 분석한 결과 패턴을 이해 가능한 것으로 밝히는 도구이다. 모든 데이터셋 수준에서보다는 각기 데이터에 대하여 피처의 영향력을 고려한 후에 피처 조합의 상호 작용을 분석함으로써 인사이트(Insight)를 가지게 한다. 이 패키지의 단점으로는 SHAP 값이 피처의 강한 연관성(Correlation)에 민감하다는 것과 SHAP 값이 예측 모델에 대하여 위험도에 부정적 영향을 줄 수도 있다는 것이다. 이점에서 예측을 정확하게 설명하기 위해서는 무작위 대조 시험(RCT: Randomized Controlled Trial)이 필요하다. (그림 8)은 예측한 결과에 대하여 피처가 얼마만큼 영향을 주었는가에 대한 예측 요인을 설명한다.

### 3. Treeinterpreter 패키지

Scikit-learn의 결정트리와 랜덤포레스 예측을 해석하기 위해 구현된 패키지이다. 데이터셋을 랜덤포레스트로 트레이닝시키고 테스트셋으로 예측한 후에 설명 가능성을 찾아보는 방법이다[35]. 예를 들어 2개의 임의 데이터에 대하여 어떤 피처가 얼마나 기여했는지 따져보는 것이다.

### 4. XGBoostExplainer 패키지

이 패키지는 XGBoost 모델을 투명하고 결정 트리를 해석 가능한 것으로 만드는 일종의 화이트 박스 패키지이다[36]. XGBoost 의사 결정 트리를 보면 각 피처가 예측하는데 어떻게 영향력을 미치고 있는가를 알 수 있다. XGBoost 프레임워크는 내장된 메소드 중에서 각 변수에 대한 중요도(Importance) 그래프를 보여 주는 기능을 제공한다. 이는 모든 변수 측면에서 가장 중요한 변수라는 것을 알려 주지만 특정 한 개 데이터에 대해서는 중요도가 높다는 것이 아니다. 이 패키지는 블랙박스에서 예측을 분해하여 각각 변수들로 최종 예측에 기여했는지를 확인하여 모든 각각 변수의 기여도를 보여준다.

## Ⅶ. 결론

머신 러닝은 정교한 알고리즘을 코딩하여 정확성을 높이는 것뿐만 아니라, 충분한 데이터만으로도 성능이 높은 모델을 구성할 수 있다. 머신 러닝의 발전은 실제로 알고리즘이 아니라 충분한 데이터셋에서 일어난다고 볼 수도 있다. 스마트 헬스케어 장치가 보급됨에 따라 헬스케어 빅데이터는 오히려 머신 러닝 기술을 선도하게 되었다. 한편으로 머신 러닝은 예방 헬스케어 시대 가능성을 열어 주고 있고 특정 환경에 적합한 데이터를 사용하여 더욱 정확성을 높일 수 있으며, 예측 모델은 정교해 질 수 있는 것이다. 머신 러닝의 사용이 지속적으로 증가함에 따라 머신 러닝 모델이 하는 일을 이해하고 설명하고 정의해야 할 필요성이 커지고 있다. 본 고에서는 헬스케어 데이터를 정제하는 전처리, 피처 처리

그리고 데이터 이해를 위한 시각화 기술을 조사하였다. 그리고 머신 러닝 기본 모델 유형과 모델 평가를 살펴보고, 최근 인기 높은 머신 러닝 프레임워크인 XGBoost를 유방암 사례에 적용해 보았다. 여기서 한가지 짚고 넘어갈 것은 선호하는 머신 러닝 패키지를 한가지만을 사용해서는 안 된다. 데이터 피쳐, 데이터셋에 따라 예측 모델이 잘못 구성될 수 있으므로 반드시 랜덤포레스트, 서포트벡터머신(SVM), 결정 트리 모델, 딥러닝 모델 등 다양한 머신 러닝 프레임워크를 사용하여 구성하여 그 모델을 평가해 본 후에 머신 러닝 모델을 선정할 필요가 있다. 마지막으로 머신 러닝 모델의 예측 정확성과 함께 떠오르는 것으로 그 모델이 예측 결정에 도달하게 한 방법은 무엇인지, 예측한 요인이 무엇인지를 밝히는 설명 가능한 머신 러닝 도구 패키지 사례를 살펴보았다. 앞으로 머신 러닝은 거대 빅데이터 분석과 지식 기반으로 더 정교하고 설명 가능한 머신 러닝으로 발전할 것으로 본다.

## 약어 정리

| | |
|---|---|
| AUC | Area Under Curve |
| CART | Classification & Regression Trees |
| GMM | Gaussian Mixture Model |
| HER | Electronic Health Records |
| ICD-9 | International Classification of Diseases |
| KNN | K-Nearest Neighbors Algorithm |
| LIME | Local Interpretable Model-Agnostic Explanations |
| RCT | Randomized Controlled Trial |
| ROC | Receiver Operating Characteristic |
| SHAP | SHapley Additive exPlanations |
| SVM | Support Vector Machine |
| VIF | Variance Inflation Factor |

## 참고문헌


[1] 서경원 외, “스마트 헬스케어 의료기기 기술,” 표준전략보고서, 식품의약품안전평가원, 2018. 8.

[2] 송영준, “4차 산업혁명과 디지털 헬스케어 정책,” 주간기술동향, 2018. 2.

[3] 정성원, “Healthcare에서 빅데이터의 활용,” 제 5회 임상연구방법론 워크숍, 가톨릭의대의생명산업연구원, 서울, 2016. 11. 5, pp. 18-29.

[4] IBM, “Bigdata in Healthcare: Tapping New Insight to Save Lives,” IBM Big Data & Analytics Hub, 2014. https://www.ibmbigdatahub.com/infographic/big-data-healthcare-tapping-new-insight-save-lives

[5] Wikipedia, “Machine Learning,” https://en.wikipedia.org/wiki/Machine_learning

[6] 정일영, 구원모, “헬스케어 생태계 구축을 위한 데이터 통합 방안,” 동향과 이슈, 제46호, 2018. 1, pp. 1-38.

[7] MIT Critical Data, *Secondary Analysis of Electronic Health Records*, Springer International Publishing: NY, USA, 2016.

[8] G. Press, “Cleaning Big Data: Most Time-Consuming, Least Enjoyable Data Science Task, Survey Says,” Forbe, Mar. 23, 2016.

[9] S. Christa, V. Suma, and L. Maduri, “An Effective Data Preprocessing Technique for Improved Data Management in a Distributed Environment,” *ACCTHPCA*, vol. 3, July 2012, pp. 25-29.

[10] SAS, “Data Visualization Techniques: From Basics to Big Data with SAS® Visual Analytics,” SAS White Paper, 2018

[11] P. van der Laken, “Facet,” Google, June 2017. https://github.com/PAIR-code/facets

[12] WIlliam H. Wolberg (physician), University of Wisconsin Hospitals. Madison, Wisconsin, USA, Breast Cancer Wisconsin (Original) Data Set, https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(original)

[13] Tutorials Point, “Seaborn,” TutorialsPoint, 2017. https://www.tutorialspoint.com/seaborn/seaborn_tutorial.pdf

[14] A. Bilogur, “Missingno: A Missing Data Visualization Suite,” *J. Open Source Softw.*, Feb. 27, 2018, doi: 10.21105/joss.00547

[15] Continuum Analytics, “Blaze Documentation,” 2018. https://blaze.readthedocs.io/en/latest/index.html

[16] G. Csardi and T. Nepusz, *igraph Reference Manual*, Harvard University: Cambridge, MA, USA, 2013.

[17] Wikipedia, “Feature Engineering,” https://en.wikipedia.org/wiki/Feature_engineering

[18] A. Zheng, *Evaluating Machine Learning Models*, O’reilly: Sebastopol, CA, USA, 2015.

[19] Medcalc, “ROC Curve Analysis,” https://www.medcalc.org/manual/roc-curves.php

[20] F.Y. Osisanwo et al., “Supervised Machine Learning

Algorithms: Classification and Comparison," *Int. J. Comput. Trends Technol.*, vol. 48, no. 3, June 2017, pp. 128-138.

[21] P. Harrington, *Machine Learning in Action*, Manning Publications Co.: Shelter Island, NY, USA, 2012, pp. 83-100.

[22] M. Namratha and T.R. Prajwala, "A Comprehensive Overview of Clustering Algorithms in Pattern Recognition," *IOSR J. Comput. Eng.*, vol. 4, no. 6, 2012, pp. 23-30.

[23] L. Arnold et al., "An Introduction to Deep Learning," in *Proc. Eur. Symp. Artif. Neural Netw.*, Bruges, Belgium, Apr. 27-29, 2011, pp. 477-488.

[24] Wikipedia, "Random Forest," https://en.wikipedia.org/wiki/Random_forest

[25] Wikipedia, "Boosting," https://en.wikipedia.org/wiki/Boosting_(machine_learning)

[26] R.E. Schapire, "The Boosting Approach to Machine Learning, An Overview," in *MSRI Workshop on Nonlinear Estimation and Classification*, Springer: Heidelberg, Germany, 2002, pp. 3-4.

[27] A. Natekin and A. Knoll, "Gradient Boosting Machines, a Tutorial," *Front. Neurorobot.*, July 21, 2013, doi: 10.3389/fnbot.2013.00021.

[28] G. Biau, B. Cadre, and L. Rouviere, "Accelerated Gradient Boosting," arXiv:1803.02042, May 2018.

[29] J. Brownlee, "XGBoost with Python, Gradien Boosted Trees with XGBoost and Scikit-learn," Machine Learning Mastery, Sept. 19, 2016.

[30] G. Ke et al., "LGBM LightGBM: A Highly Efficient Gradient Boosting Decision Tree," *Conf. Neural Inform. Process. Syst.*, Long Beach, CA, USA, 2017, pp. 1-9.

[31] A. Veronika, D.V. Ershov, and A. Guli, "CatBoost: Gradient Boosting with Categorical Features Support," Yandex, 2017. https://catboost.ai/

[32] M. Du, N. Liu, and X. Hu, "Techniques for Interpretable Machine Learning," arXiv:1808.00033, July 2018.

[33] M.T. Ribeiro, S. Singh, and C. Guestrin, "Why Should I Trust You?" *Proc. ACM SIGKDD Int. Conf. Knowled. Discovery Data Mining*, San Francisco, CA, USA, Aug. 13-17, 2016, pp. 1135-1144.

[34] S.M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," *Conf. Neural Inform. Process. Syst.*, Long Beach, CA, USA, 2017, pp. 1-10.

[35] A. Saabas, "treeinterpreter, 2015. https://github.com/andosa/treeinterpreter

[36] D. Foster, "xgboostExplainer," 2017. https://github.com/AppliedDataSciencePartners/xgboostExplainer