

Heart Disease Prediction Using Decision Tree With Kaggle Dataset

Young-Dan Noh*, Kyu-Cheol Cho*

*Student, Dept. of Computer Science, Inha Technical College, Incheon, Korea

*Professor, Dept. of Computer Science, Inha Technical College, Incheon, Korea

[Abstract]

All health problems that occur in the circulatory system are refer to cardiovascular illness, such as heart and vascular diseases. Deaths from cardiovascular disorders are recorded one third of in total deaths in 2019 worldwide, and the number of deaths continues to rise. Therefore, if it is possible to predict diseases that has high mortality rate with patient's data and AI system, they would enable them to be detected and be treated in advance. In this study, models are produced to predict heart disease, which is one of the cardiovascular diseases, and compare the performance of models with Accuracy, Precision, and Recall, with description of the way of improving the performance of the Decision Tree(Decision Tree, KNN (K-Nearest Neighbor), SVM (Support Vector Machine), and DNN (Deep Neural Network) are used in this study.). Experiments were conducted using scikit-learn, Keras, and TensorFlow libraries using Python as Jupyter Notebook in macOS Big Sur. As a result of comparing the performance of the models, the Decision Tree demonstrates the highest performance, thus, it is recommended to use the Decision Tree in this study.

▶ **Key words:** heart disease, diseases that has high morality rate, AI, detected, Decision Tree

[요 약]

심혈관질환은 심장질환과 혈관질환 등 순환기계통에 생기는 모든 질병을 통칭한다. 심혈관질환은 2019년 사망의 1/3을 차지하는 전 세계 사망의 주요 원인이며, 사망자는 계속 증가하고 있다. 이와 같은 질병을 인공지능을 활용해 환자의 데이터로 미리 예측이 가능하다면 질병을 조기에 발견해 치료할 수 있을 것이다. 본 연구에서는 심혈관질환 중 하나인 심장질환을 예측하는 모델들을 생성하였으며 Accuracy, Precision, Recall의 측정값을 지표로 하여 모델들의 성능을 비교한다. 또한 Decision Tree의 성능을 향상시키는 방법에 대해 기술한다. 본 연구에서는 macOS Big Sur환경에서 Jupyter Notebook으로 Python을 사용해 scikit-learn, Keras, TensorFlow 라이브러리를 이용하여 실험을 진행하였다. 연구에 사용된 모델은 Decision Tree, KNN(K-Nearest Neighbor), SVM(Support Vector Machine), DNN(Deep Neural Network)으로 총 4가지 모델을 생성하였다. 모델들의 성능 비교 결과 Decision Tree 성능이 가장 높은 것으로 나타났다. 본 연구에서는 노드의 특성배치를 변경하고 트리의 최대 깊이를 3으로 지정한 Decision Tree를 사용하였을 때 가장 성능이 높은 것으로 나타났으므로 노드의 특성 배치 변경과 트리의 최대 깊이를 설정한 Decision Tree를 사용하는 것을 권장한다.

▶ **주제어:** 심장질환, 사망의 주요 원인, 인공지능, 예측, Decision Tree

-
- First Author: Young-Dan Noh, Corresponding Author: Kyu-Cheol Cho
 - *Young-Dan Noh (shdbwjd705270@gmail.com), Dept. of Computer Science, Inha Technical College
 - *Kyu-Cheol Cho (kccho@inhac.ac.kr), Dept. of Computer Science, Inha Technical College
 - Received: 2021. 12. 03, Revised: 2022. 04. 22, Accepted: 2022. 04. 25.

I. Introduction

심혈관질환은 심장질환과 혈관질환 등 순환기계통에 생기는 모든 질병을 통칭한다. 심혈관질환은 2019년 사망의 1/3을 차지하는 전 세계 사망의 주요 원인이고 사망자는 계속 증가하고 있다[1]. 또한 심장병 발병은 1990년 2억7100만 건에서 2019년 5억2300만 건으로 약 2배 증가했고 심장병 사망자 수는 1210만 명에서 1860만 명으로 증가하였다[1].

심혈관질환은 생명을 위협할 수 있고 치료시기를 놓친다면 뇌세포 손상과 같은 후유증을 발생시킬 수 있어 조기 발견이 중요하다[2]. 본 연구는 이처럼 사망률이 높고 위험성이 있는 질병을 인공지능을 활용해 환자의 데이터로 미리 예측을 하여 질병을 조기에 발견해 치료할 필요가 있어 진행되었다.

본 연구에서는 심혈관질환 중 하나인 심장질환을 예측하는 모델들을 생성하였으며 모델들의 성능을 비교하고 Decision Tree의 성능을 향상시키는 방법에 대해 기술한다. 연구에 사용된 모델은 Decision Tree, KNN(K-Nearest Neighbor), SVM(Support Vector Machine), DNN(Deep Neural Network)로 총 4가지 모델을 생성하였다.

모델의 생성에 사용된 환자의 데이터는 Kaggle에 공개된 데이터를 사용하였다. Kaggle은 데이터 예측모델 및 분석 대회를 주최하는 플랫폼으로 데이터 분석을 위한 데이터를 제공한다. 심장질환 데이터는 총 14개의 특성으로 구성되어 있으며 총 303개의 행으로 구성되어 있다. 그 중 target 특성은 심장질환의 여부를 표현하는 열로 0, 1의 값으로 구성되어 있으며 0은 심장질환이 없고 1은 심장질환이 있는 것을 의미한다.

본 연구의 구성은 다음과 같다. 2장에서는 본 연구와 관련된 연구에 관하여 기술하고 3장에서는 Kaggle에 공개된 환자의 데이터에 대한 설명과 데이터 전처리 기법에 대해 기술하고 Decision Tree로 심장질환의 유무를 예측해 본 결과에 대해 기술한다. 4장에서는 Decision Tree, KNN, SVM, DNN모델을 사용해 심장질환 유무를 예측한 성능에 대해 비교하며, 5장에서는 결론에 대해 기술하고 마무리한다.

II. Preliminaries

1. Related works

1.1 Decision Tree

Decision Tree는 지도 학습 모델 중 하나로 분류와 회귀가 모두 가능한 것으로 범주나 연속형 데이터 모두 예측이 가능하다.

Decision Tree는 의사결정규칙을 도표화하여 관심대상이 되는 집단을 몇 개의 소집단으로 분류하거나 예측을 수행하는 분석방법이다[3]. Decision Tree는 반복적으로 분류 규칙을 생성하여 가장 효과적인 분류 규칙 조합을 완성해 나간다[4].

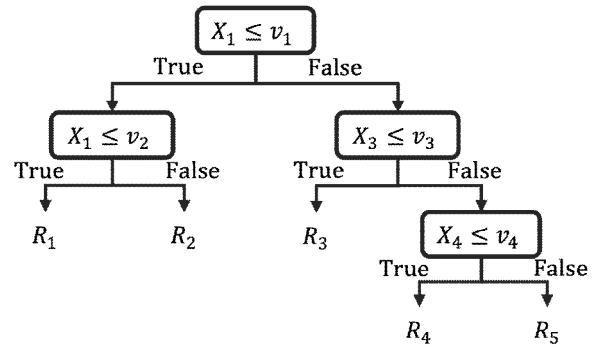


Fig. 1. Decision Tree algorithm

Fig. 1.은 Decision Tree의 구조이다. 특성 X의 값을 v 값과 비교해 점차 데이터를 분류해나가 결과 값 R을 도출해내는 트리를 생성하는 기법이다.

Decision Tree는 분석 과정을 쉽게 설명할 수 있는 장점[3]이 있지만 예측 정확도의 측면에서 비교적 최근에 개발된 모형에 비해 상대적으로 낮다는 단점이 있다[4].

1.2 KNN

KNN은 지도학습 중 하나이다. 주로 분류와 회귀에 사용되는 비모수 방식이다[5]. 데이터로부터 거리가 가까운 'k'개의 다른 데이터 레이블을 참조하여 분류하는 알고리즘으로 거리를 측정할 때에는 유클리드 거리 계산법이 대표적으로 사용된다.

KNN은 데이터가 어떤 분포를 따르는지에 대한 가정이 필요하지 않아 데이터를 분류하는 처리 과정이 빠르다는 장점이 있지만 컴퓨터 메모리를 많이 차지한다는 단점이 있다[6].

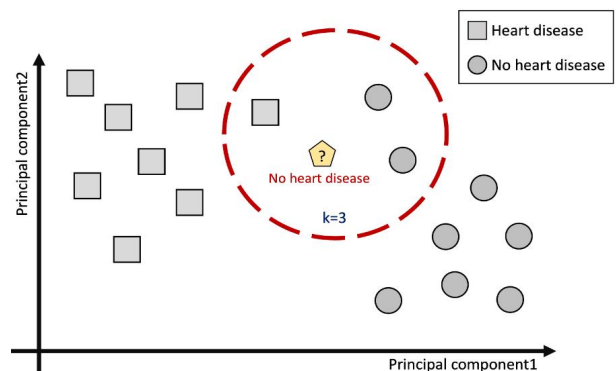


Fig. 2. KNN algorithm

Fig. 2.는 feature의 수가 2개이고 3명의 이웃을 참조해 심장질환을 예측하는 상황의 예시이다. 예측하고자하는 사람의 이웃인 세 사람 중 심장질환 환자가 1명, 심장질환이 없는 사람이 2명이므로 심장질환이 없는 사람으로 예측을 할 수 있다.

본 연구에서는 19개의 이웃을 참조하여 분류하도록 하였다.

1.3 SVM

SVM은 두 클래스 사이의 최대 여백으로 분리되는 하이퍼 평면을 식별하는 이진 분류기이다[7]. 주로 분류와 회귀 분석에서 사용되며 주어진 데이터가 어느 카테고리에 속할지 판단한다. 또한 SVM은 비선형 함수를 이용하여 데이터간의 거리가 최대가 되도록 분류하고 최적화된 결정 경계를 찾는 특징도 가지고 있다[8].

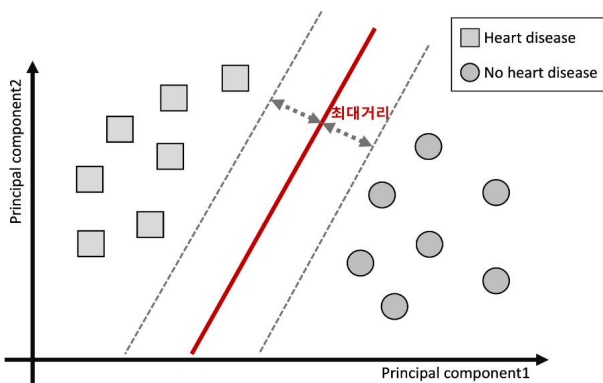


Fig. 3. SVM algorithm

Fig. 3.은 feature가 2개 일 때 심장질환을 분류하는 상황의 예시이다. 데이터가 2차원이라면 SVM의 결정경계는 선이고 3차원이라면 결정경계는 평면으로 표현된다[7].

SVM은 이진분류 모델이기 때문에 다중분류에 적용하게 될 경우 성능이 저하되고 학습 시간이 길어진다는 단점이 있다[9].

1.4 DNN

DNN은 기계학습법 중 하나로 기존의 인공신경망에 비해 더 깊고 넓은 구조를 가지고 있으며, 기존 인공신경망의 역전파 알고리즘의 한계를 극복하고 복잡한 구조를 풀 수 있게 하였다[10]. 또한 DNN은 간단한 구조의 신경망으로도 매우 다양한 분야의 예측 모델을 만들 수 있다는 장점이 있다[11].

DNN은 입력층과 출력층 사이에 2개 이상의 은닉층으로 구성되어있으며 본 연구에서의 DNN모델의 layer구성방식에 대해서는 4장에서 기술한다.

III. Predict Heart Disease With Decision Tree

본 연구에서는 Kaggle에 공개된 환자의 데이터[12]를 가지고 연구를 진행하였다. 본 3장에서는 데이터의 구성과 데이터의 전처리 기법에 대해 기술하고 본 연구에서 제안하는 Decision Tree로 심장질환의 유무를 예측해 본 결과에 대해 기술한다.

1. Data explore

CSV데이터는 총 14가지의 특성(열)으로 구성되어있다. 나이, 성별, 가슴 통증 유형 등의 특성과 심장질환의 유무를 나타내는 target 속성으로 구성되어 있다. target 특성의 값은 0이면 심장질환이 없는 것을 의미하고 1이면 심장질환이 있는 것을 의미한다.

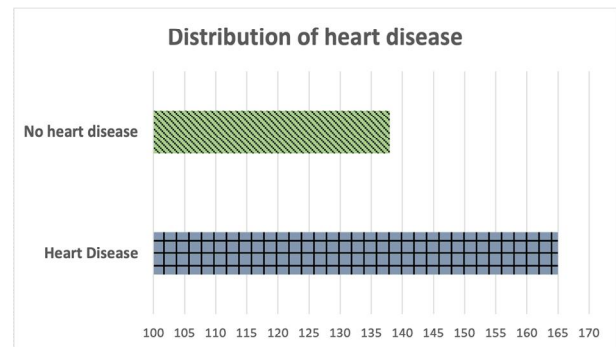


Fig. 4. Distribution of heart disease

Fig. 4.은 전체 데이터 중 심장질환 환자의 수와 심장질환을 가지지 않은 사람의 수를 나타낸다. 데이터는 총 303개가 있으며 그 중 심장질환 환자는 165명이며, 심장질환을 가지지 않은 사람의 수는 138명이다.

2. Data analysis

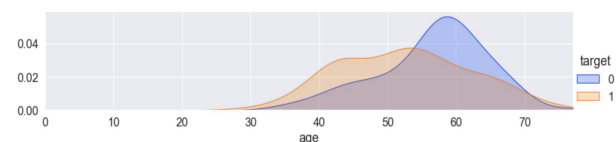


Fig. 5. Presence of heart disease by age

Fig. 5.는 나이에 따른 심장질환 유무의 분포도를 보여준다. 20대에서 70대까지 매우 다양한 연령이 존재하는 것을 볼 수 있으며 대체적으로 40~60세 사이에 target 값이 1인 심장질환 환자가 많은 것을 볼 수 있다. 또한

describe()메서드를 사용해 나이의 최솟값과 최댓값을 보았을 때 최소 연령은 29세, 최대 연령은 77세인 것을 알 수 있었다.

3. Data preprocessing

데이터의 전처리란 분석에 적합하게 데이터를 가공하는 작업을 의미한다.

Fig. 5.에서처럼 나이의 분포가 폭넓게 나타나 나이 데이터에 대해서는 Data binning 기술을 사용하였다. Data binning이란 데이터의 구간을 나누어 각각 하나의 카테고리별로 나이를 담는 것으로 10대, 20대와 같이 나누어 데이터 전처리를 하는 것이다. 나이를 그대로 훈련에 사용하면 29~77세까지 너무 많은 정보가 제공되어 훈련에 방해가 될 수 있기 때문에 Data binning 기술을 사용하였다.

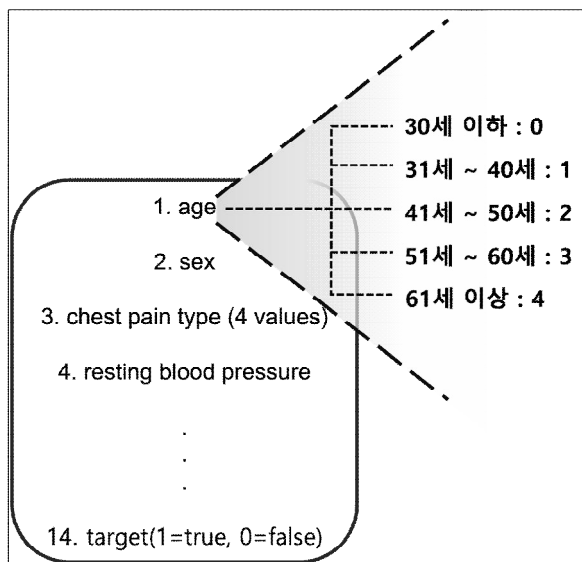


Fig. 6. Data preprocessing

Fig. 6.은 데이터 전처리 과정을 나타낸다. 본 연구에서는 30세 이하와 61세 이상의 나이로 카테고리를 나누고 중간 값들은 10살 간격으로 카테고리를 나누어 총 5개의 카테고리로 나누었다. 각각의 카테고리는 0~4까지의 숫자로 표현한다. 실제로 트리의 최대 깊이를 설정하지 않았을 때, Data binning과정을 거쳤을 때의 테스트 데이터의 정확도가 Data binning 과정을 거치지 않았을 때의 테스트 데이터의 정확도보다 3.3%높은 모습을 보여주었다.

target 속성은 예측을 하기 위한 속성이므로 데이터에서 해당 열을 제거한 후 Data binning 과정을 거친 데이터에서 훈련 데이터와 테스트 데이터를 80%, 20%의 비율로 나누었다. 본 연구에서는 이와 같은 데이터 전처리 과정을 거쳐 모델에 적용하였다.

4. Training conditions

본 논문에서는 훈련을 할 때 특별한 조건을 주어 훈련을 하였다. 반복문을 돌려 특정 정확도 값을 넘을 때까지 다시 훈련과 테스트 데이터를 분리하고 훈련을 하되 1분이 넘도록 반복문이 수행된다면 정확도 값을 낮춰서 설정해 다시 훈련을 하도록 하였다. 이를 통해 최대의 정확도를 도출하고자 하였으며 여기서 1분은 임의로 설정한 시간이다.

5. Prediction with Decision Tree

5.1 Decision Tree model without max depth

의사결정 트리 분류 모듈로는 sklearn.tree 패키지의 DecisionTreeClassifier를 사용하여 훈련하였다.

트리의 최대 깊이를 지정하지 않고 훈련을 하였을 때, 훈련 데이터의 예측 정확도는 100%, 테스트 데이터의 예측 정확도는 93.4%로 나왔다.

이처럼 트리의 깊이를 지정하지 않으면 훈련 데이터를 과도하게 학습하는 과적합이 발생하게 되며 훈련 데이터에 대해선 정확도가 높지만 테스트 데이터에 대해선 정확도가 떨어지는 모습을 보여준다. 따라서 이 과적합을 해결하는 것이 중요한데 그 방법 중 하나로 트리의 최대 깊이를 지정하였다.

5.2 Decision Tree model with a max depth of 3

과적합을 해결하기 위해 최대 깊이를 3으로 지정하고 훈련한 결과 훈련 데이터의 예측 정확도는 83.1%이며, 테스트 데이터의 예측 정확도는 95.1%로 나왔다. 트리의 최대 깊이를 지정하지 않았을 때보다 훈련 데이터의 예측 정확도는 내려갔지만 테스트 데이터의 예측 정확도는 1.7% 올라간 것을 보아 과적합이 해결된 모습을 볼 수 있다.

5.3 Decision Tree model with a max depth of 3 and changing the node's feature placement

의사결정 트리에서는 정보 획득량이 큰 순서대로 질문을 배치하는 것이 좋다. 즉, 특성(feature)을 어떻게 배치하느냐에 따라 다양한 Decision Tree를 만들 수 있고 정확도도 다양해질 것이다.

Fig. 7.은 노드의 특성배치를 변경하고 트리의 최대 깊이를 3으로 지정한 트리의 구조이다.

훈련 데이터와 테스트 데이터로 나누기 전 데이터의 열 순서를 변경하였다. 트리의 깊이를 3으로 지정하고 훈련을 했을 때와 비교하면 훈련 데이터의 예측 정확도는 82.2%로 동일했고 테스트 데이터의 예측 정확도는 약 1.6% 가량 증가해 96.7%의 정확도를 보여주었다.

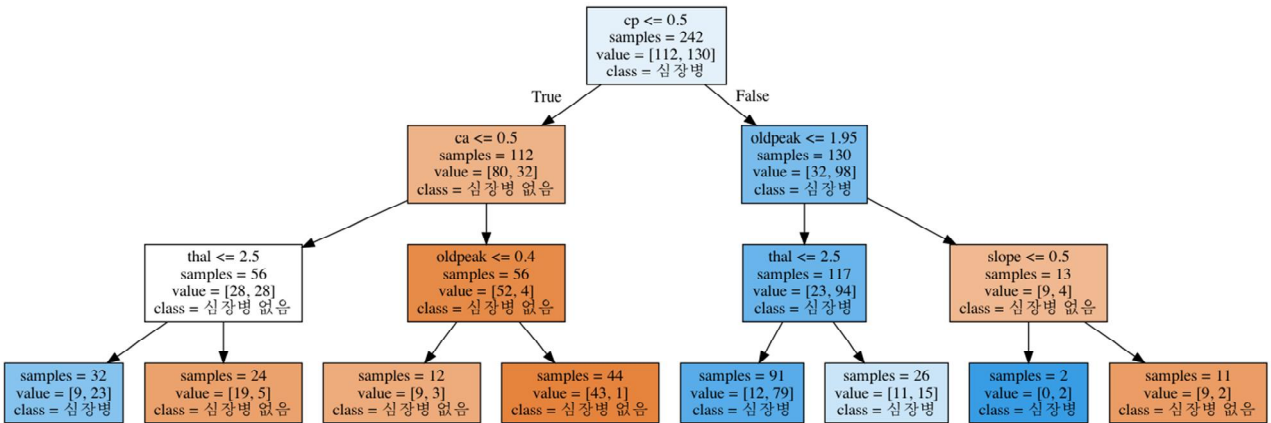


Fig. 7. Changing a node's feature placement

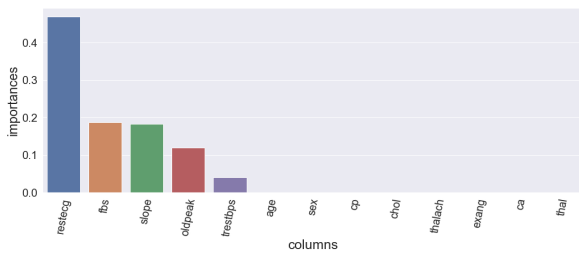


Fig. 8. Feature importance

Fig. 8.은 특성의 중요도를 그래프로 표현한 그림이다. 특성의 중요도는 각 특성들이 모델의 예측 결과에 얼마나 큰 영향을 미쳤는지 해당 특성의 중요도를 평가하는 지표이다. 그래프를 살펴보면 총 12가지의 특성들 중에서 restecg 특성이 중요도가 가장 높은 것을 보아 모델의 예측 결과에 가장 큰 영향을 끼친 것을 알 수 있다.

Table 1. Decision Tree prediction accuracy

	Train data accuracy(%)	Test data accuracy(%)
No max depth	100%	93.4%
Max depth-3	83.1%	95.1%
Max depth-3 & Change node's feature placement	82.2%	96.7%

Table 1.은 트리의 최대 깊이를 지정하지 않은 모델, 트리의 최대 깊이를 3으로 지정한 모델, 트리의 최대 깊이를 3으로 지정하고 노트의 특성 배치를 변경한 모델의 훈련 데이터와 테스트 데이터의 예측 정확도에 대한 표이다.

본 연구에서는 최대깊이를 3으로 지정하고 노트의 특성 배치를 변경한 Decision Tree 모델로 KNN, SVM, DNN 모델과 비교한다.

IV. Experiment

앞서 설명했던 것과 같이 Kaggle에 공개된 환자의 데이터를 사용해 심장질환의 유무를 예측해 보았으며. 실험을 통해 Decision Tree, KNN, SVM, DNN 모델을 사용해 심장질환 유무를 예측한 성능을 비교해 보고자한다.

1. Experiment environment

실험에 앞서 본 연구를 위한 실험 환경은 Table 2.와 같이 구성하였다. 모델의 훈련과 테스트 및 성능 비교를 위한 Accuracy, Precision, Recall 값은 Python을 사용해 도출하였다.

Table 2. Experiment environment

OS	macOS Big Sur
IDE	Jupyter Notebook
Programming Language	Python
Library	scikit-learn, Keras, TensorFlow

2. Comparison of test accuracy of DNN model

4가지의 모델들의 성능을 비교하기에 앞서 동일하게 구성된 DNN 모델의 layer 구성에 대해 기술하고자한다.

DNN 모델을 구성할 때 입력층(Input layer)과 출력층(Output layer)은 각각 1개씩 만들고 중간층 은닉층(Hidden layer)은 4개로 구성하였다.

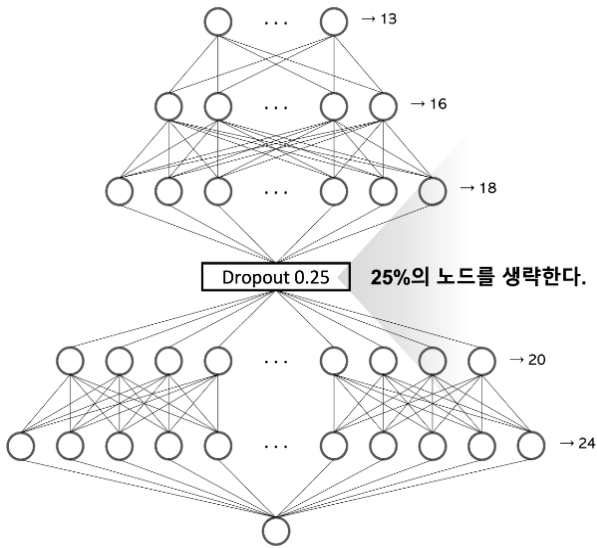


Fig. 9. The structure of the DNN model used for training

Fig. 9.는 훈련을 하기 위해 구성한 DNN모델의 구조이다. 훈련에 사용된 데이터의 특성은 target 특성을 제외하고 13개 이므로 입력층에서는 총 13개의 입력이 들어가 16개의 출력이 나오게 구성하였고, 출력층은 심장질환이 있는지 아닌지 판단하기 위한 0 또는 1의 값이 출력되어야 하므로 24개의 입력이 들어가면 1개의 출력이 나오게 구성하였다. 또한 중간의 은닉층 4개를 구성하였는데 그 중 하나는 Dropout 층으로 과적합을 방지하기 위한 층이다. Dropout은 학습할 때 각 레이어의 모든 노드에 대해 학습하는 것이 아니라 은닉층의 노드를 무작위로 선택하여 학습을 할 수 있게 해주고 생략된 노드는 학습에 영향을 끼치지 않게 해준다[13].

이와 같이 DNN 모델을 구성해 훈련을 한 모델에 테스트 데이터를 적용해 보았을 때 최대 93.4%의 정확도를 보여주었다.

3. Accuracy comparison according to the number of categories of age data

3장에서 설명한 것과 같이 나이 데이터는 데이터의 구간을 나누어 각각 하나의 카테고리별로 나이를 담은 Data binning을 하였는데 그 카테고리의 개수를 5개와 9개로 했을 때의 정확도를 비교해보고자 한다.

KNN은 19개의 이웃을 참조해 예측하도록 하였고, SVM은 kernel을 linear로 주었다. KNN은 참조하는 이웃의 개수를 조절해보며 여러 번 훈련을 해본 결과, 19개의 이웃을 참조하는 것이 가장 적절한 것으로 보여 19개의 이웃을 참조하도록 구성하였다.

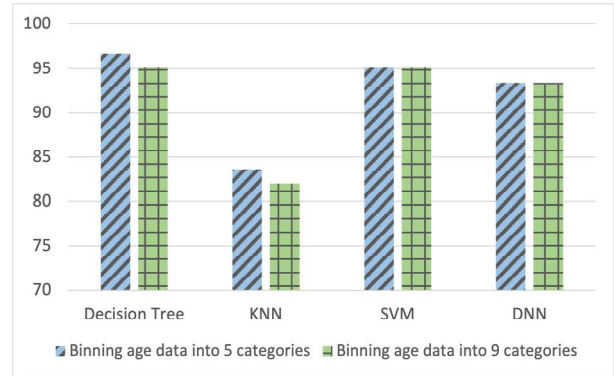


Fig. 10. Comparison of test data accuracy with 9 and 5 categories of age

Fig. 10.은 3장의 데이터 전처리 과정에서 설명한 것과 같이 5개의 카테고리로 나누었을 때와, 30세 이하와 66세 이상으로 나누고 중간 값들은 5살 간격으로 나누어 9개의 카테고리를 분류했을 때의 모델들의 정확도이다. 파란색의 빗금 바는 나이를 5개의 카테고리로 나누었을 때의 정확도이고 연두색의 격자무늬 바는 나이를 9개의 카테고리로 나누었을 때의 정확도를 나타낸다.

나이를 5개와 9개의 카테고리로 나누었을 때 Decision Tree의 정확도는 각각 96.7%, 95.1%로 가장 높은 정확도를 보여주며 KNN은 각각 83.6%와 82%로 가장 낮은 정확도를 보여준다.

SVM과 DNN은 나이의 카테고리를 5개와, 9개로 나누었을 때의 정확도가 동일했지만 Decision Tree, KNN의 정확도는 카테고리를 5개로 나누었을 때 더 높은 것을 볼 수 있었다. 따라서 전반적인 모델들의 정확도를 비교할 때 나이의 카테고리를 5개로 분류하는 것이 더 정확도를 높이는 방법이라고 생각된다.

4. Comparison of Decision Tree, KNN, SVM, and DNN

본 연구에서는 4가지 모델의 성능을 비교하기 위해 Accuracy, Precision, Recall을 측정하였다[14]. Accuracy, Precision, Recall 모두 값이 클수록 좋은 모델로 평가된다.

나이의 카테고리를 5개로 분류했을 때의 모델을 비교하며 Decision Tree는 최대 깊이를 3으로 지정하고 노드의 특성배치를 변경한 모델을 기준으로 비교한다.

Table 3. Comparison of accuracy, precision and recall of 4 model

	Accuracy(%)	Precision	Recall
Decision Tree	96.7	0.97	0.97
KNN	83.6	0.86	0.86
SVM	95.1	0.97	0.94
DNN	93.4	0.88	1.0

Table 3.은 4가지 모델의 Accuracy, Precision, Recall에 대한 표이다.

Decision Tree가 가장 높은 Accuracy를 보여주며 KNN이 83.6%로 가장 낮은 Accuracy를 보여준다. Precision에서는 0.97인 Decision Tree와 SVM이 가장 높은 모습을 볼 수 있으며, 마지막으로 Recall 값은 DNN이 1.0으로 가장 높은 것을 볼 수 있다.

세 가지 Accuracy, Precision, Recall를 비교했을 때 Recall값은 DNN이 가장 높지만 Accuracy, Precision은 Decision Tree가 가장 높은 것을 볼 수 있다. 따라서 본 연구에서는 Decision Tree를 사용한 심장질환 예측을 권장한다.

V. Conclusions

본 연구는 전 세계 사망의 선두 원인인 심혈관질환 중 하나인 심장질환을 인공지능을 활용해 환자의 데이터로 미리 예측하고자 시작되었다. 이와 같은 사망률이 높고 치료시기를 놓치면 좋지 않은 결과를 초래할 수 있는 질병을 인공지능을 통해서 조기에 발견해 치료할 수 있다면 수많은 인명을 구할 수 있을 것으로 기대된다.

심장질환 예측에 적합한 모델을 찾기 위해 Decision Tree, KNN, SVM, DNN의 총 4가지 모델을 구성해 비교하였으며 성능 비교의 지표로는 Accuracy, Precision, Recall을 사용하였다. 비교 결과 본 연구에서는 가장 성능이 좋은 것으로 보인 Decision Tree를 사용한 예측을 권장한다. 또한 Decision Tree는 트리의 최대깊이와 같은 것을 설정하지 않았을 경우보다 노드의 특성배치를 변경하고 트리의 최대 깊이를 3으로 지정하고 사용하였을 때 가장 성능이 높은 것으로 나타났다. 그러므로 Decision Tree를 사용할 경우에는 노드의 특성 배치 변경과 트리의 최대 깊이를 설정하고 사용하는 것을 권장한다. 현재 Accuracy, Precision, Recall 3가지 정보를 가지고 성능을 비교하였지만, 차후 연구에서는 환자가 아닌 사람을 환자로 분류한 비율인 오탐율에 대한 정보도 추가한다면 좋을 것이라고 생각된다.

본 연구에서는 비교적 새로운 방식인 최근의 방식인 딥러닝보다 고전적인 방식인 Decision Tree의 정밀도가 더 높게 나왔다. 이것은 모델의 정의, 학습방식과 파라미터 튜닝에 문제가 있을 수도 있다는 것을 의미한다. 차후 연구에서는 이 문제를 수정한다면 딥러닝에서의 정밀도를 높일 수 있을 것이라고 생각된다.

본 연구에서는 Kaggle에서 제공하는 총 303명의 환자의 데이터를 가지고 모델을 생성해 훈련을 하였는데 더 많은 환자의 데이터를 모집해 훈련을 한다면 더욱 성능이 좋은 모델을 만들 수 있을 것이라고 예상된다.

REFERENCES

- [1] Gregory A. Roth et al., "Global Burden of Cardiovascular Diseases and Risk Factors, 1990-2019: Update From the GBD 2019 Study", *JOURNAL OF THE AMERICAN COLLEGE OF CARDIOLOGY*, Vol. 76, No. 25, pp.2982-3021, Dec 2020. DOI : 10.1016/j.jacc.2020.11.010
- [2] The importance of cardiovascular disease checkup, http://iheartwell.com/html/community_info_read.php?id=1746&mode=&cate=&page=7&key=&keyword=
- [3] Yongjung Son, Hyunduk Kim, "Forecasting Export & Import Container Cargoes using a Decision Tree Analysis", *Journal of Korea Port Economic Association*, Vol. 28, No. 4, pp.193-207, Dec 2012.
- [4] Youtaek Jeon, HyungJun Cho, "Model based hybrid decision tree", *Journal of the Korean Data And Information Science Society*, Vol. 30, No.3, pp.515-524, May 2019. DOI: 10.7465/jkdi.2019.30.3.515
- [5] Altman, N. S., "An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression", *The American Statistician*, Vol. 46, No. 3, pp.175-185, 1992, DOI: 10.1080/00031305.1992.10475879
- [6] Taewoo Lee, Jungwoo Kim, Deokyo Kim, Jaeho Lee, "Performance Evaluations and Feature Vector by Implementing Hybrid Filter in BLE-Based Fingerprinting", *The Journal of Korean Institute of Communications and Information Sciences*, Vol. 44, No. 8, pp.1556-1565, Aug 2019. DOI: 10.7840/kics.2019.44.8.1556
- [7] Kyong-Rok Lee, "A Study on SVM-Based Speaker Classification Using GMM-supervector", *Journal of IKEEE*, Vol. 24, No.4, pp.102-107, Dec 2002. DOI: 10.7471/ikeee.2020.24.4.1022
- [8] Jeong-II Go, Eui-Young Lee, Min-Jae Lee, Seong-Dae Choi, Jang-Wook Hur, "Corrosion Failure Diagnosis of Rolling Bearing with SVM", *Journal of the Korean Society of Manufacturing Process Engineers*, Vol. 20, No. 9, pp.35-41, Sep 2021. DOI: 10.14775/ksmp.2021.20.09.035
- [9] Yang Seok Kim, Do Hwan Lee, Seong Kook Kim, "Fault

Classification for Rotating Machinery Using Support Vector Machines with Optimal Features Corresponding to Each Fault Type”, The Korean Society of Mechanical Engineers, Vol.34, No.11, pp.1681-1689, Nov 2010, DOI:10.3795/KSME-A.2010.34.11.1681

- [10] Won-Bin Oh, Ill-Soo Kim, Tae-Jong Yun, Bo-Ram Lee, Chung-Woo Lee, Ki-Young Park, Byeong-Ju Jin and Yu-Cheol Lee, “A Study on the Prediction of Real-Time Bead Width Using a DNN Algorithm in GTA Welding”, Journal of Welding and Joining, Vol.38, No.6, pp.593-601, Dec 2020. DOI: 10.5781/JWJ.2020.38.6.10
- [11] Gyeonggon Kim, Chansoo Park, Wooyeong Kim, Jeeyeon Jeon, Miyeon Jeon, Choongsik Bae, “The Effect of Natural Gas Substitution Ratio and Diesel Injection Timing on Accuracy of In-cylinder Pressure Prediction DNN Model from Vibration Signal in a CNG-Diesel Dual-Fuel Engine”, Transaction of the Korean Society of Automotive Engineers, Vol. 29, No. 10, pp.909-919, Oct 2021. DOI: 10.7467/KSAE.2021.29.10.909
- [12] Patient data from Kaggle, <https://www.kaggle.com/ronitf/heart-disease-uci>
- [13] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting”, The Journal of Machine Learning Research, Vol. 15, No. 1, pp. 1929-1958, Jan 2014.
- [14] Yunseok Rhee, “Malicious Code Detection Method Using LSTM Learning on the File Access Behavior”, The Journal of Korean Institute of Information Technology, pp.25-32, Vol. 18, No. 2, Feb 2020. DOI : 10.14801/jkiit.2020.18.2.25

Authors



Young-Dan Noh received the A.S. degree in Computer Information Engineering from Inha Technical College, Korea, 2021 respectively. Ms. Noh entered the Inha Technical College in 2018 and graduated in 2021.

She has experience in IT practice and interested in IT and the fourth industrial revolution.



Kyu-Cheol Cho received the B.S., M.S. and Ph.D. degrees in Computer Science and Information Engineering from Inha University, Korea, in 2005, 2007 and 2013, respectively. Dr. Cho joined the faculty of the Department

of Computer Science at Inha Technical College, Incheon, Korea, in 2016. He is currently a assistant professor in the Department of Computer Science, Inha Technical College. He is interested in cloud computing, green IT and web programming.