데이터 마이닝 팀 프로젝트

<데이터 마이닝 기법을 이용한 청소년 문제음주 요인분석>

과목명: 데이터 마이닝(나)

교수 명: 이건호 교수님

학과 명: 산업정보시스템공학과

제출자 명: 20182480 문지원

20182523 이채연

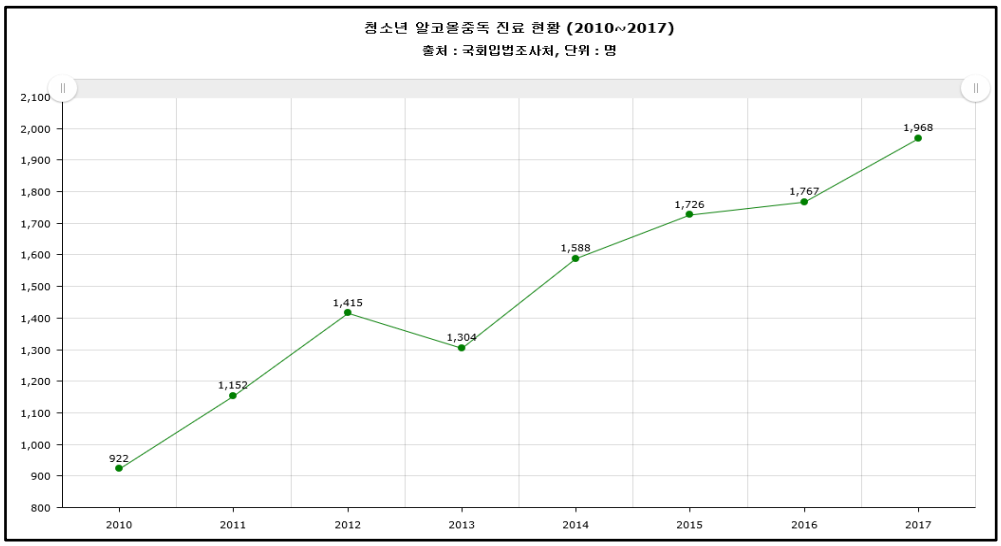
<목차>

1. 서론 및 연구배경

1) 연구의 배경

현재 우리나라의 알코올 소비량은 세계 17위로 굉장히 높고, 한, 중, 일을 비교해 봤을 때도 남, 녀를 통틀어 1위일 만큼 담배와 함께 가장 많이 남용되는 약물은 술이라고 해도 과언이 아니다. 또한, 우리나라는 술을 마시는 문화가 사회적으로 조성이 되어있고, 술에 대해 호의적인 태도를 가지고 있어 술에 대한 위험성을 쉽게 간과하고 있는 문제점이 있다. 이는 자라나는 청소년에게도 큰 문제가 되는 부분이다. 만약 청소년 음주에 영향을 미치는 요인이 어떤 것이 있는지 미리 파악한다면 이에 맞는 교육을 할 수도 있을 것이고, 일종의 경고를 할 수도 있을 것이다. 따라서 우리는 그러한 요인에는 어떤 것이 있고, 이를 관리하는 방법에는 어떤 것이 있는지 생각해 보게 되었다.

2) 연구의 필요성



<표 1. 청소년 알코올 중독 치료 현황>

2010년부터 2017년까지 청소년 알코올 중독 진료 현황을 보면 점점 증가하는 추세임을 <표 1>을 통해 확인할 수 있다. 이를 보아 현재 2020년까지도 그 수가 끊임없이 증가하고 있을 것이라고 추정된다. 청소년기의 음주는 조직을 파괴하고 학습 능력을 떨어지게 하고 기억력 저하까지 초래할 수 있다. 이러한 청소년 음주 문제는 그 원인을 다양하게 예측할 수 있는데 그 예시로는 스트레스, 학업성취, 부모의 음주여부 등 여러가지가 있다. 그 중에서도 영향을 가장 크게 미치는 요인이 무엇인지는 데이터 마이닝을 이용하여 알 수 있다. 데이터 마이닝은 과거 분류가 된 사례들을 통해 새로운 문제를 분류할 수 있는 모델을 구축하는 것이다.

우리는 이번 과제에서 데이터 마이닝 기법을 활용하여 지난 청소년 음주 사례와 그 원인을 분석하여 예측모델을 만드는 것을 목표로 하였다.

3) 연구를 통한 이점

이번 연구를 통해 나온 청소년 음주 예측 모델을 활용한다면, 원래는 모든 청소년에게 공통적으로 같은 자료를 활용해서 음주 교육을 했다면 각각의 원인에 맞는 자료를 만들어 맞춤 교육을 할 수 있을 것이다. 또한, 알코올 중독 치료가 필요한 청소년인 경우 그 학생이 가지고 있는 여러 원인들을 모델과 비교하여 보다 적합한 치료법으로 치료할 수도 있을 것이다.

2. 관련연구

음주영향요인을 데이터 마이닝 기법을 활용해 탐색하는 선행연구에 대해서 조사해보았다. 선행 연구들이 어떤 방법과 절차를 이용하여 음주영향요인을 예측하였는지를 중점적으로 알아보았다.

참고한 첫번째 연구는 <Data Mining을 이용한 음주 및 음주문제의 위험요인과 취약성요인에 관한 탐색. 김인석 외 3명>이다. 본 연구는 일상생활의 범위에서 알코올 사용과 그에 수반되는 역기능적 문제에 영향을 미치는 선행요인을 사회환경적 위험요인과 심리적 및 인구통계적 취약성 요인으로 분류하여 가장 영향력이 놓은 설명 변인을 추출하고자 하였다. 의사결정나무분석(Decision Tree)을 이용하여 음주문제에 상대적으로 높은 영향을 주는 요인을 추린 후, 신경망 분석(Neural Networks)을 통하여 앞서 도출한 요인에 대한 인공 신경망 모형을 구성하여 예측력을 확인하였다.

분석 1: 의사결정 나무분석

본 연구에서는 의사결정나무의 CART 알고리즘 방식을 사용하였다. CART 알고리즘이란 지니계수(불순도를 측정하는 지표로 데이터의 통계적 분산정도를 정량화하여 표현한 값)로 계산하는 방법이다. 개별변수의 평균값을 예측변인으로 하고 마디가 진행될수록 이에 따른 분산 감소를 지니계수를 통해 확인하는 방식을 사용하였다.

분석 2: 신경망 분석

본 연구에서는 Neural Connection 2.1(1997) 프로그램을 사용하여 신경망 분석을 실시하였다. 자료를 분석하는 방법 중 단층인식모형(Radial Basis Function; RBF)와 다층인식모형(Multiple Layer Perception; MLP)을 통합하여 사용하였다.

참고한 두번째 연구는 <청년의 문제음주에 미치는 사회생태학적 결정요인에 대한 데이터 마이닝 분석. 이숙현 외 1인> 이다. 본 연구는 사회생태학적인 관점에서 문제음주를 논의하고 문제음주에 설명력을 가지는 영향요인을 의사결정나무, 신경망, 로지스틱 회귀 모형으로 분석하였다.

분석 1 : 의사결정나무모형

본 연구는 CHAID 알고리즘을 사용하여 분석하였다. 이것은 두 변수간의 통계적 관계를 찾을 수 있다. 전체 자료를 설명변수의 범주의 쌍에 대한 반응변수의 유의적이지 않은 쌍들이 없을 때 까지 둘 이상의 하위 노드(child node)로 반복적으로 분할한다. 의사결정 트리의 최대가지는 2, 최대 깊이는 6, 최소 범주형 크기는 20으로 설정하여 분석하였다.

분석 2 : 신경망모형

본 연구에는 MLP(Multilayer Perception) 구조의 신경망 모형을 분석하였다. 은닉층 한 개에 3개의 은닉마디를 설정하였다. 신경망의 최적화 과정은 50회 반복을 통해 추정되었다.

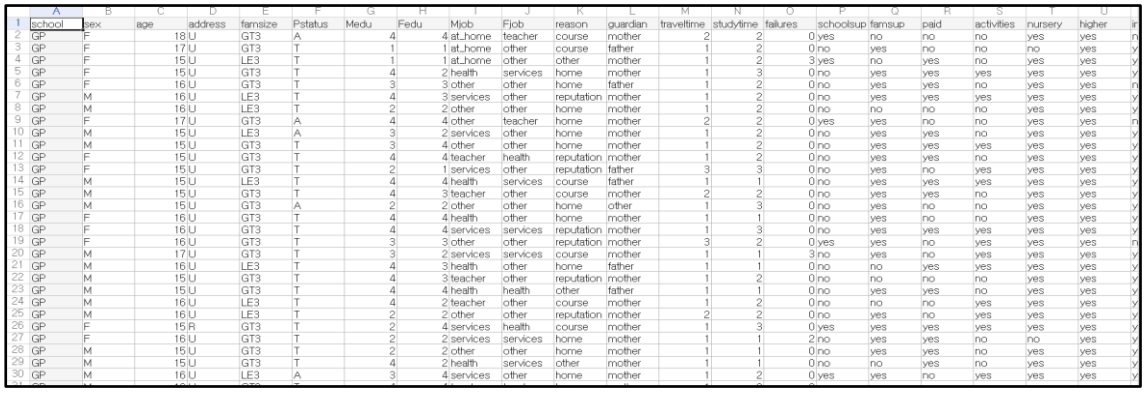
분석 3: 로지스틱 회귀모형

로지스틱 회귀모형을 사용하여 성별, 연령, 흡연여부 등의 여러 요인들이 청년의 문제음주에 유의미한 영향을 미치는지 확인하였다.

3. 본론

1) 데이터 소개

데이터는 중등 학교의 학생들의 알코올 섭취량과 학생들의 사회적, 성별 및 학습 정보가 포함되어 있고, 30개의 attribute와 382개의 row들로 구성되어 있다. 이 데이터는 데이터 사이언스 경진 대회 플랫폼 Kaggle에서 제공하는 Data set이다.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	reason	guardian	traveltime	studytime	failures	schoolsup	famsup	paid	activities	nursery	higher
2	GP	F	18	U	GT3	A	4	4	at_home	teacher	course	mother	2	2	0	yes	no	no	no	yes	yes
3	GP	F	17	U	GT3	T	1	1	at_home	other	course	father	1	2	0	no	yes	no	no	no	yes
4	GP	F	15	U	LE3	T	1	1	at_home	other	other	mother	1	2	3	yes	no	yes	no	yes	yes
5	GP	F	15	U	GT3	T	4	2	health	services	home	mother	1	3	0	no	yes	yes	yes	yes	yes
6	GP	F	16	U	GT3	T	3	3	other	other	home	father	1	2	0	no	yes	yes	no	yes	yes
7	GP	M	16	U	LE3	T	4	3	services	other	reputation	mother	1	2	0	no	yes	yes	yes	yes	yes
8	GP	M	16	U	LE3	T	2	2	other	other	home	mother	1	2	0	no	no	no	no	yes	yes
9	GP	F	17	U	GT3	A	4	4	other	teacher	home	mother	2	2	0	yes	yes	no	no	yes	yes
10	GP	M	15	U	LE3	A	3	2	services	other	home	mother	1	2	0	no	yes	yes	no	yes	yes
11	GP	M	15	U	GT3	T	3	4	other	other	home	mother	1	2	0	no	yes	yes	yes	yes	yes
12	GP	F	15	U	GT3	T	4	4	teacher	health	reputation	mother	1	2	0	no	yes	yes	no	yes	yes
13	GP	F	15	U	GT3	T	2	1	services	other	reputation	father	3	3	0	no	yes	no	yes	yes	yes
14	GP	M	15	U	LE3	T	4	4	health	services	course	father	1	1	0	no	yes	yes	yes	yes	yes
15	GP	M	15	U	GT3	T	4	3	teacher	other	course	mother	2	2	0	no	yes	yes	no	yes	yes
16	GP	M	15	U	GT3	A	2	2	other	other	home	other	1	3	0	no	yes	no	no	yes	yes
17	GP	F	16	U	GT3	T	4	4	health	other	home	mother	1	1	0	no	yes	no	no	yes	yes
18	GP	F	16	U	GT3	T	4	4	services	services	reputation	mother	1	3	0	no	yes	yes	yes	yes	yes
19	GP	F	16	U	GT3	T	3	3	other	other	reputation	mother	3	2	0	yes	yes	no	yes	yes	yes
20	GP	M	17	U	GT3	T	3	2	services	services	course	mother	1	1	3	no	yes	no	yes	yes	yes
21	GP	M	16	U	LE3	T	4	3	health	other	home	father	1	1	0	no	no	yes	yes	yes	yes
22	GP	M	15	U	GT3	T	4	3	teacher	other	reputation	mother	1	2	0	no	no	no	no	yes	yes
23	GP	M	15	U	GT3	T	4	4	health	health	other	father	1	1	0	no	yes	yes	no	yes	yes
24	GP	M	16	U	LE3	T	4	2	teacher	other	course	mother	1	2	0	no	no	no	yes	yes	yes
25	GP	M	16	U	LE3	T	2	2	other	other	reputation	mother	2	2	0	no	yes	no	yes	yes	yes
26	GP	F	15	R	GT3	T	2	4	services	health	course	mother	1	3	0	yes	yes	yes	yes	yes	yes
27	GP	F	16	U	GT3	T	2	2	services	services	home	mother	1	1	2	no	yes	yes	no	no	yes
28	GP	M	15	U	GT3	T	2	2	other	other	home	mother	1	1	0	no	yes	yes	no	yes	yes
29	GP	M	15	U	GT3	T	4	2	health	services	other	mother	1	1	0	no	no	yes	no	yes	yes
30	GP	M	16	U	LE3	A	3	4	services	other	home	mother	1	2	0	yes	yes	no	yes	yes	yes

<그림 1. Data set>

2) 데이터 전처리

데이터 분석에서 전처리 과정은 데이터 분석가가 80%의 시간을 데이터 수집 및 전처리에 사용한다고 할 정도로 비중있고 중요한 과정이다. 이번 과제에서는 Jupyter를 사용하여 데이터 전처리 및 분석을 실시하였다. 데이터를 전처리 하기 전, CSV 파일을 열어 데이터의 속성을 자세히 살펴보았고, 이 중, 어머니의 직업과 아버지의 직업에 교사, 건강관리 관련, 공무원 서비스, 집에서, 기타 와 같은 항목이 동일하게 있는 것을 확인하였고 이는 후에 요인 별 결과에 미치는 중요도를 분석하는 데에 있어 영향을 미칠 것으로 생각되었다. 따라서 각각의 항목을 어머니의 직업이면 mhealth, 아버지의 직업이면 fhealth 등으로 바꾸는 작업을 해주었다. 수정한 데이터를 통해 중요도를 조사하여 데이터로부터 최대한 insight를 추출해 내려고 노력하였다.

```
In [1]: import numpy as np
        import pandas as pd

In [2]: df=pd.read_csv('student-mat1.csv')
```

<그림 2. 데이터 불러오기>

먼저, Jupyter에서 본인의 드라이브에 있는 데이터 CSV 파일을 불러오고, 마이닝을 하기 위해 Pandas와 Numpy를 쓰기 위해 이를 불러왔다.

그 후, 데이터들을 하나씩 확인해보며 필요한 데이터인지, 의미가 있는 데이터인지를 생각하였다. 이 Data set에서는 알코올 소비를 주중과 주말로 나눠서 1부터 5의 수준으로 나타내었는데, 일주일치 소비량으로 판단하고자 이 두 attribute를 하나의 attribute로 합쳐주었다.

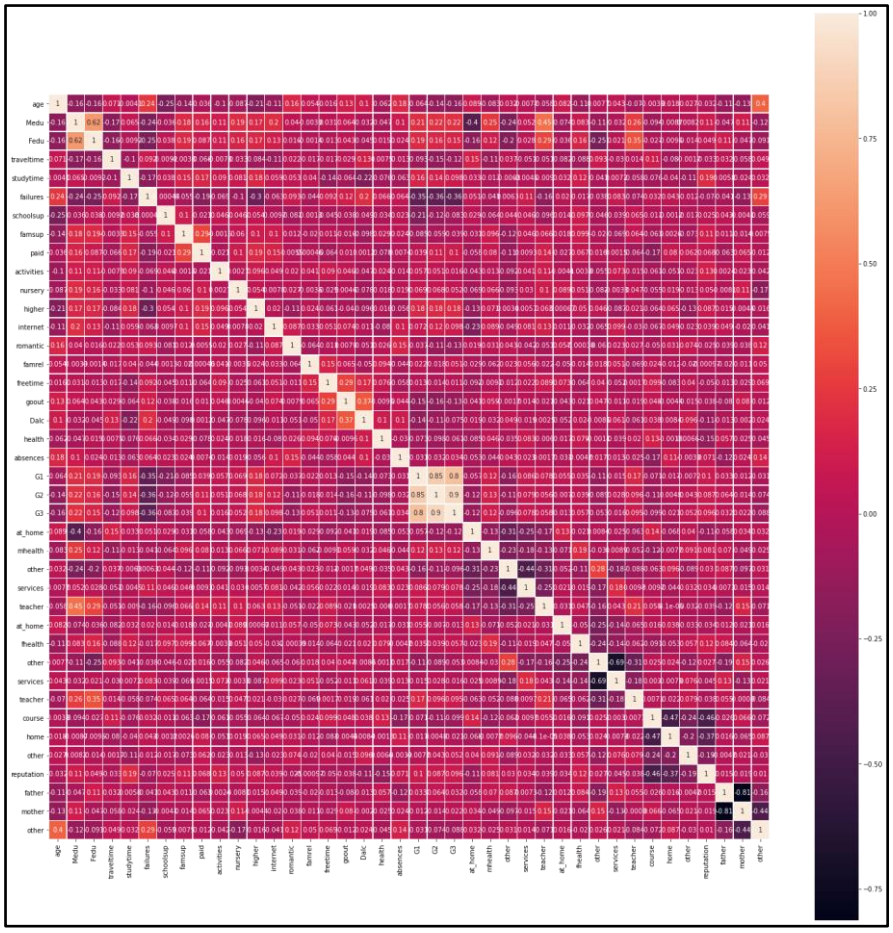
또한, 데이터를 확인해 본 결과 Missing value는 확인되지 않았고, 추가 교육 지원(schoolup), 가족 교육 지원(famsup), 과외활동(activities), 고등교육(higher), 인터넷

(internet), 연애관계(romantic) 는 이진변수로 binary 값을 넣었다. yes에는 1을 대입하고, no에는 0을 대입하였다. data 값이 이진으로 되어있는 school(GP, MS), sex(Female, Male), address(도시U, 시골 R), famsize(3보다 작거나 같음, 3보다 큼), 부모의 동거 상태(Pstatus)(동거 T, 떨어져 있음 A)도 binary 값을 넣어 각각 1 또는 0으로 데이터를 처리하였다.

일주일치 소비량으로 판단하고자 이전 작업에서 주중 알코올 소비와 주말 알코올 소비를 Dalc로 합쳐 주었는데 이는 1부터 10까지의 수준으로 나타나 있고, 이를 5보다 큰 수준이면 중독, 5보다 작거나 같으면 보통 소비량으로 하여 중독이면 1, 중독이 아니면 0으로 binary 값을 넣어주었다.

다음으로는 범주 특성의 One-hot-encoding 작업을 해주었다. Mjob, Fjob, reason, guardian 데이터는 범주형 데이터이다. 범주형 데이터가 숫자이면 연속형 데이터로 인식되어 범주형 특성이 무시되므로 실제와 오차가 발생한다. 이에 대해 각 변수의 하위 범주로 0,1의 bool 형태로 구분하여 하위변수를 생성하여 처리하는데 이를 dummy variable이라 한다. 각각의 attribute를 dummy variable로 만든 후에 dummy와 column들을 합병 후 범주형 데이터들을 column에서 삭제함으로써 One-hot-encoding 작업을 실시하였다.

원인 별로 독립성을 가지는지 혹은 서로 상관관계가 얼마나 미치고 있는지 확인하기 위해 상관관계 분석을 해주었다.



<그림 3. 원인 별 상관관계>

그 결과 G1, G2, G3간의 상관 정도가 높아 독립이 아님을 알 수 있다. 이를 '다중공선성'이라 하는데 이는 데이터 분석 시 부정적인 영향을 미치게 된다. 따라서 이 3개의 attribute 값들을 1개로 만들기 위해 평균치를 만들어 사용하였다.

이후 scaling을 통해 값들을 scaling 해주었다. 스케일링의 종류에는 min-max, standardization, maxabs, robust 등이 있는데 이 중에서 중앙값(Median)과 IQR(Interquartile Range)을 사용하여 outlier의 영향을 최소화 할 수 있다는 장점을 가진 robust scaler를 사용하였다.

3) 활용 알고리즘

앞서 조사한 선행연구들은 주로 의사결정나무분석과 신경망 분석을 이용하여 데이터 분석을 하였다. 본 연구 역시 청소년의 음주에 영향력이 높은 변인을 선별하려 하기에 선행연구와 마찬가지로 분류(classifier) 종류의 모델을 사용하고자 하였다. 그 중 인공신경망과 의사

결정 나무는 최근 연구와 응용 양쪽에서 가장 활발한 모형이다. 따라서 나무 모양으로 지도 학습을 하는 모델 (Decision Tree, Random Forest, Extra Tree Classifier) 과 인공 신경망 모델을 사용하여 데이터를 분석할 것이다.

Decision Tree 는 대용량의 데이터로부터 입력변수를 탐색하고, 입력변수의 상호작용 즉, 입력변수들의 결함이 목표변수에 미치는 영향을 도출하는 데이터 마이닝 방법론이다. (이숙현 외 1인.2018) 이는 나무구조로 표현되어 있어 사용자가 모형을 쉽게 이해할 수 있으며 두 개 이상의 변수가 결합하여 목표변수에 어떻게 영향을 주는지 쉽게 알 수 있다는 장점이 있다. 이번 프로젝트에서 사용되는 데이터 역시 청소년의 음주 정도에 따른 다양한 요인들이 있는데, 이 요인들이 목표변수인 청소년의 음주정도에 어떠한 영향을 주는지 쉽게 파악할 수 있다는 점에서 해당 연구에 매우 적합하다. 그러나 단점으로는 학습 데이터에 오버피팅 하는 경향이 있다. 가지치기 와 같이 오버피팅을 최소화하는 전략이 있지만 부족하다.

Random Forest 는 훈련을 통해 구성해 놓은 다수의 나무들로부터 분류 결과를 취합하여 결론을 얻는 인기투표와 같은 형식이다. 다수의 의사결정 트리를 만든 후 나무들의 분류를 집계하여 최종적으로 분류한다. 이는 Decision Tree에서 발생한 오버피팅 문제를 해결할 수 있다. 몇몇의 나무들이 오버피팅을 보일 수 있지만 다수의 나무를 기반으로 예측하기 때문에 오버피팅을 보이는 나무들에 대한 영향력이 줄어들어 좋은 성능을 보인다. 랜덤 포레스트 모델에서 나무들을 생성하기 위해 배깅(bagging)이라는 프로세스를 활용한다. 배깅은 전체 학습 데이터 셋 중에 중복을 포함하여 임의로 일부를 뽑은 후 의사결정 트리를 만드는 것이다. 랜덤 포레스트 모형에 사용되는 매개변수 중 n_estimatior 는 만들 트리 개수로 클수록 좋다. 본 프로젝트에서는 500을 사용하였다. n_job이란 사용할 코어수로 -1로 지정하면 모든 코어가 사용 가능하기 때문에 -1로 지정해주었다. 이 모델은 성능이 매우 뛰어나고 데이터의 스케일을 맞추지 않아도 된다는 장점이 있는 반면 차원이 높은 데이터에서는 잘 작동하지 않는다는 단점이 있다.

Extra Tree Classifier는 랜덤 포레스트 모델의 변종으로 포레스트 트리의 각 후보 특성을 무작위로 분할하는 식으로 랜덤 포레스트보다 훨씬 더 무작위성을 증가시킨다. 트리를 더욱 무작위하게 만들기 위해 보통의 결정 트리처럼 최적의 임계값을 찾는 것이 아닌 후보 특성을 사용해 무작위로 분할한 다음 그 중에서 최상의 분할을 선택한다. 때문에 일반적인 랜덤 포레스트보다 속도가 훨씬 빨라질 수 있다. 또한 랜덤 포레스트 보다 더 폭넓은 시각으로 특성들을 평가한다는 장점이 있다.

Neural Network는 컴퓨터를 학습시키기 위해 인간의 뉴런의 동작 원리에 기초해 인공적으

로 구축한 신경망을 의미한다. 신경망 분석은 복잡하고 비선형이며 관계성을 가지는 데이터를 통해 최적결과를 예측하는데 적합한 기법이다. 신경망 모형은 입력과 출력 마디에 연속형과 인산형 변수 모두를 이용할 수 있다는 점에서 적용범위가 의사결정 나무 모형보다 넓다는 장점을 가진다. 또한 비선형이고 상호관련성을 가진 모델에서 예측력 높은 출력변수를 도출하는데도 우수한 기법이다. 단층 퍼셉트론에서는 활성 함수가 1개 밖에 없는 구조로 비선형적으로 분리되는 데이터에 대해서는 제대로 된 학습이 어렵다는 단점이 있다. 반면 다층 퍼셉트론은 입력층과 출력층 사이에 하나 이상의 중간층을 두어 비선형적으로 분리되는 데이터에 대해서도 학습이 가능하다. 따라서 이번 프로젝트에서는 다층 퍼셉트론 (MLP) 기법을 사용하여 분석할 것이다.

4. 분석 및 평가

1) 모델 평가

모델을 평가하는 척도로서 Confusion Matrix를 사용하였다. Confusion Matrix란 테스트 데이터 셋에 대한 분류기와 분류의 성능을 평가하는 행렬이다.

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes (positive)	a: TP (true positive)	b: FN (false negative)
	Class=No (negative)	c: FP (false positive)	d: TN (true negative)

<그림 4. Confusion Matrix>

- Accuracy = $\frac{TP + TN}{TP + TN + FP + FN}$
- Precision = $\frac{TP}{TP + FP}$
- Recall = $\frac{TP}{TP + FN}$

Precision이란, "정밀도" 라고 불리며 예측 모델이 True라고 평가한 것들 중에 실제로 True인 것들의 비율을 뜻한다. Recall은 "재현율" 이라고 불리며 실제 정답 중에서 모델이 맞춘 정답의 비율을 뜻한다. 이 때, 정밀도와 재현율은 trade-off 관계를 갖게 되는데 이 말은 정밀도가 늘어나면 재현율이 줄어들고, 재현율이 늘어나면 정밀도가 줄어든다는 것을 말한다. 때문에 이 두가지를 모두 고려한 분류 평가 수치가 필요한데 이것이 바로 F1 score이다. F1 score는

precision, recall과 같은 측정치들로 값을 따졌을 때 모델 평가에서 문제가 생겨 한 쪽으로 편향된 결과가 나을 수 있는데 이를 보완하기 위해 precision과 recall를 조화 평균한 값으로 표현한 값이다.

```
from sklearn.metrics import classification_report,confusion_matrix
from sklearn.preprocessing import RobustScaler
from sklearn.metrics import accuracy_score
from sklearn.metrics import f1_score

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1)
clf = DecisionTreeClassifier()
clf = clf.fit(X_train,y_train)

scaler= RobustScaler()
scaler.fit(X)

X_train=scaler.transform(X_train)
X_test=scaler.transform(X_test)

y_pred = clf.predict(X_test)

print(confusion_matrix(y_test,y_pred))

print("Accuracy:", metrics.accuracy_score(y_test, y_pred))
print("Precision:", metrics.precision_score(y_test, y_pred))
print("Recall:", metrics.recall_score(y_test, y_pred))
f1_score(y_test,y_pred)
```

```
[[59 38]
 [ 3 19]]
Accuracy: 0.6554621848739496
Precision: 0.3333333333333333
Recall: 0.8636363636363636

0.481012658227848
```

<그림 5. 의사결정나무를 활용한 예측모델>

먼저, 스케일링 후 의사결정나무 기법을 활용하여 예측 모델을 만들어 낸 결과이다. Test 데이터는 30%의 데이터로 설정하였고, 이 때 나온 정확도는 약 65%이다. 90% 이상의 값은 아니지만 그리 낮은 값이 아닌 정확도라고 생각하였다.

```
from  sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import f1_score

model = RandomForestClassifier(n_jobs=-1 ,n_estimators=500, random_state=40)
model =model.fit(X_train, y_train)

y_pred = model.predict(X_val)

print("Accuracy:", metrics.accuracy_score(y_val, y_pred))
print("Precision:", metrics.precision_score(y_val, y_pred))
print("Recall:", metrics.recall_score(y_val, y_pred))
```

```
Accuracy: 0.8192771084337349
Precision: 1.0
Recall: 0.0625
```

<그림 6. random forest 기법을 활용한 예측모델>

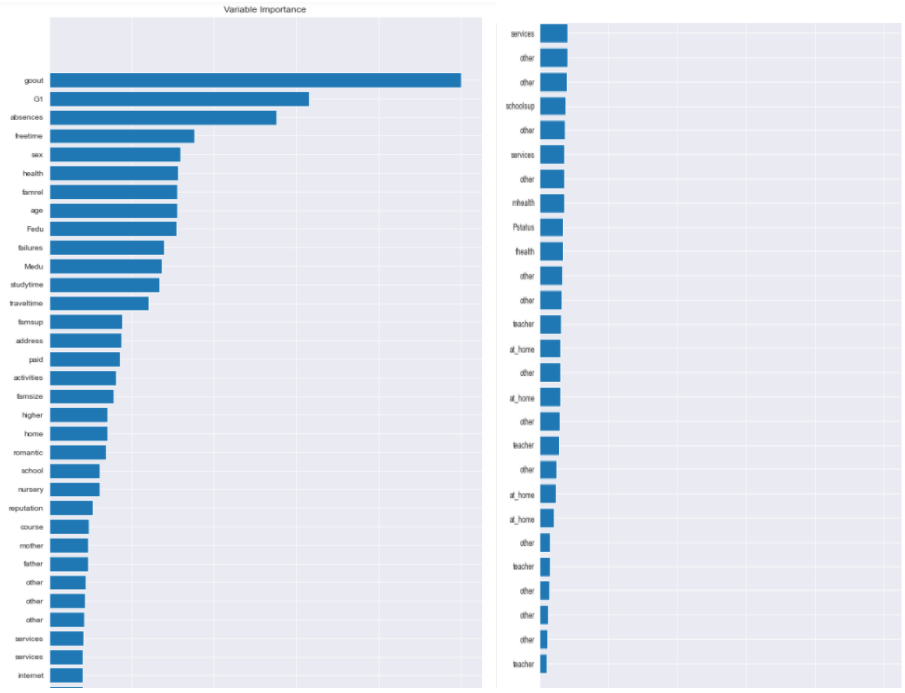
Random Forest 분류기법을 활용하여 결정 트리를 500개로 설정하였고, Test 데이터 수는 전체 데이터의 30%로 설정하였다. 또한 validation을 Train 데이터 수의 30% 비중으로 배정하여 교차검증을 진행하였다. 그 결과 정확도가 약 81%로 의사결정나무를 활용했을 때의 정확도보다 높아졌다는 것을 확인하였다. 또한, 정밀도 또한 33%에서 100%로 증가하였다.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1)
X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, train_size=0.7, random_state=10)
model = ExtraTreesClassifier(n_jobs=-1 ,n_estimators=500, random_state=40)
model =model.fit(X_train, y_train)



y_pred = model.predict(X_val)

print("Accuracy:", metrics.accuracy_score(y_val, y_pred))
print("Precision:", metrics.precision_score(y_val, y_pred))
print("Recall:", metrics.recall_score(y_val, y_pred))
```

```
Accuracy: 0.8313253012048193
Precision: 1.0
Recall: 0.125
```

<그림 7. Extra Tree Classifier를 활용한 예측모델>

다음으로는 Random Forest에서 조금 변형된 Extra Tree Classifier를 사용하여 모델을 예측해 보았다. 모델 생성 시 Test set을 전체 데이터의 30%로 정하고 결정 트리를 500개, validation은 Train 데이터의 30% 비중으로 설정하였다. 결과를 보면 정확도가 약 83%로 random forest를 사용했을 때보다 증가한 것을 알 수 있고 정밀도는 100%로 똑같지만 재현율은 6% 에서 12.5%로 증가하였다. 이후 결과에 대한 속성들의 상관성의 중요도 수치를 그래프로 표현해 보았다.



<그림 8. 속성들의 상대적 중요성>

그래프에서 나온 속성들의 상대적 중요성을 바탕으로 중요성이 약 10%인 값들은 모두 지웠다. 이런 방식으로 속성들을 지우고 다시 나머지 상대적 중요성을 그래프로 나타내 가며 결론적으로는 상대적 중요성이 약 20% 이하인 값들을 모두 지웠다. 이렇게 최소화한 변수들

을 활용하여 모델 평가를 진행하였다.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1)
X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, train_size=0.7, random_state=1)

scaler= RobustScaler()
scaler.fit(X)

X_train=scaler.transform(X_train)
X_val=scaler.transform(X_val)
X_test=scaler.transform(X_test)

clf = DecisionTreeClassifier()
clf = clf.fit(X_train,y_train)
y_pred = clf.predict(X_val)

print(confusion_matrix(y_val,y_pred))

print("Val_Accuracy:", metrics.accuracy_score(y_val, y_pred))
print("Val_Precision:", metrics.precision_score(y_val, y_pred))
print("Val_Recall:", metrics.recall_score(y_val, y_pred))
print("Val_f1_score:",metrics.f1_score(y_val,y_pred))

real_y = clf.predict(X_test)
print(confusion_matrix(y_test,real_y))

print("Real_Accuracy:", metrics.accuracy_score(y_test, real_y))
print("Real_Precision:", metrics.precision_score(y_test, real_y))
print("Real_Recall:", metrics.recall_score(y_test, real_y))
print("Real_f1_score:",metrics.f1_score(y_test,real_y))
```

```
[[55 14]
 [ 9  5]]
Val_Accuracy: 0.7228915662650602
Val_Precision: 0.2631578947368421
Val_Recall: 0.35714285714285715
Val_f1_score: 0.30303030303030304
[[84 13]
 [10 12]]
Real_Accuracy: 0.8067226890756303
Real_Precision: 0.48
Real_Recall: 0.5454545454545454
Real_f1_score: 0.5106382978723404
```

<그림 9. 최소화된 데이터로 의사결정나무를 이용한 예측 모델>

Test set을 전체 데이터의 30% 비중으로 설정하였고 교차검증은 Training set의 30%로 설정하였다. 이 때, 스케일링을 한 번 더 사용하여 정확성을 높였다. 정확도는 약 72% 가 나왔다.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1)
X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, train_size=0.7, random_state=1)

model = RandomForestClassifier(n_jobs=-1 ,n_estimators=300, random_state=1)

model =model.fit(X_train, y_train)
y_pred = model.predict(X_val)


print(confusion_matrix(y_val,y_pred))

print("Val_Accuracy:", metrics.accuracy_score(y_val, y_pred))
print("Val_Precision:", metrics.precision_score(y_val, y_pred))
print("Val_Recall:", metrics.recall_score(y_val, y_pred))
print("Val_f1_score:",metrics.f1_score(y_val,y_pred))

real_y = model.predict(X_test)
print(confusion_matrix(y_test,real_y))

print("Real_Accuracy:", metrics.accuracy_score(y_test, real_y))
print("Real_Precision:", metrics.precision_score(y_test, real_y))
print("Real_Recall:", metrics.recall_score(y_test, real_y))
print("Real_f1_score:",metrics.f1_score(y_test,real_y))

plot_feature_importance(model, X_train, (10, 9))
```

```
[[67  2]
 [12  2]]
Val_Accuracy: 0.8313253012048193
Val_Precision: 0.5
Val_Recall: 0.14285714285714285
Val_f1_score: 0.22222222222222224
[[93  4]
 [16  6]]
Real_Accuracy: 0.8319327731092437
Real_Precision: 0.6
Real_Recall: 0.2727272727272727
Real_f1_score: 0.37499999999999994
```

<그림 10. 최소한의 데이터로 Random Forest를 활용한 모델 예측>

이는 Random Forest를 사용한 예측 모델이다. 처음 데이터에서 Test set의 비중을 전체 데이터의 30%로 두었고, 트리의 개수를 300개로 설정하였다. 정확도는 의사결정나무 기법보다 약 11%가 높은 83%가 나왔다. 또한, 검증과정을 통해 과 적합 여부도 판단을 하였는데, validation으로 평가한 것과 test로 평가한 결과가 유사하여 과 적합한 모델이 아님을 확인할 수 있었다. 이전 모델보다 정확도가 올랐지만 F1 score를 비교해 보았을 때는 의사결정 나무

를 이용한 모델보다 조금 떨어지는 것을 볼 수 있었다.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1)
X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, train_size=0.7, random_state=1)

model = ExtraTreesClassifier(n_jobs=-1 ,n_estimators=500, random_state=40)

model =model.fit(X_train, y_train)

y_pred = model.predict(X_val)
print(confusion_matrix(y_val,y_pred))
print("Val_Accuracy:", metrics.accuracy_score(y_val, y_pred))
print("Val_Precision:", metrics.precision_score(y_val, y_pred))
print("Val_Recall:", metrics.recall_score(y_val, y_pred))
print("Val_f1_score:",metrics.f1_score(y_val,y_pred))

real_y = model.predict(X_test)
print(confusion_matrix(y_test,real_y))

print("Real_Accuracy:", metrics.accuracy_score(y_test, real_y))
print("Real_Precision:", metrics.precision_score(y_test, real_y))
print("Real_Recall:", metrics.recall_score(y_test, real_y))
print("Real_f1_score:",metrics.f1_score(y_test,real_y))

plot_feature_importance(model, X_train, (10, 9))
```

```
[[67  2]
 [10  4]]
Val_Accuracy: 0.8554216867469879
Val_Precision: 0.6666666666666666
Val_Recall: 0.2857142857142857
Val_f1_score: 0.4
[[92  5]
 [15  7]]
Real_Accuracy: 0.831932773109244
Real_Precision: 0.5833333333333334
Real_Recall: 0.3181818181818182
Real_f1_score: 0.4117647058823529
```

<그림 11. 최소한의 데이터로 Extremely Randomized Tree Classifier를 활용한 예측모델>

Extremely Randomized Tree Classifier를 사용한 예측 모델이다. 이 모델은 Random Forest 보다 무작위성을 증가시키고, Overfitting 현상을 방지할 수 있는 모델이다. 모델 생성 시 Test set을 전체 데이터의 30%로 설정하였고, 결정 트리를 500개로 설정하였다. 또한, Train Set 의 30% 비중을 교차검증 하였다. 그 결과는 Random Forest 모델을 사용했을 때보다 정확도가 약 2% 증가한 85%가 나왔고 F1 score을 비교했을 때도 약 18% 증가한 40%가 나왔다.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1)
X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, train_size=0.7, random_state=1)
scaler= RobustScaler()
scaler.fit(X_train)

X_train=scaler.transform(X_train)
X_val=scaler.transform(X_val)
X_test=scaler.transform(X_test)

from sklearn.metrics import classification_report,confusion_matrix
from sklearn.neural_network import MLPClassifier

mlp= MLPClassifier(hidden_layer_sizes=(11,2),random_state=3,
                   activation='relu', solver='adam')

mlp.fit(X_train,y_train)

predictions=mlp.predict(X_val)
print(confusion_matrix(y_val,predictions))
print("Val_Accuracy:", metrics.accuracy_score(y_val, predictions))
print("Val_Precision:", metrics.precision_score(y_val, predictions))
print("Val_Recall:", metrics.recall_score(y_val, predictions))
print("Val_f1_score:",metrics.f1_score(y_val,predictions))



real_y = mlp.predict(X_test)
print(confusion_matrix(y_test,real_y))

print("Real_Accuracy:", metrics.accuracy_score(y_test, real_y))
print("Real_Precision:", metrics.precision_score(y_test, real_y))
print("Real_Recall:", metrics.recall_score(y_test, real_y))
print("Real_f1_score:",metrics.f1_score(y_test,real_y))
```

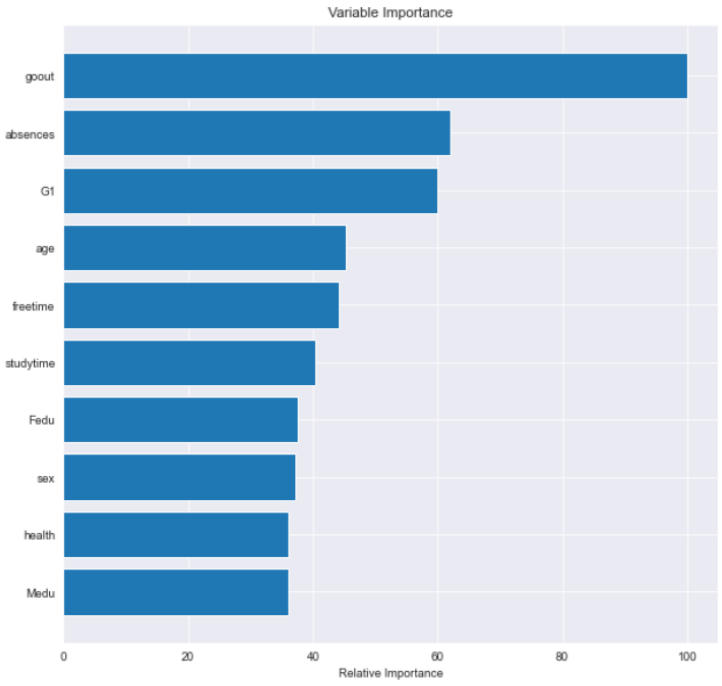
```
[[69  0]
 [12  2]]
Val_Accuracy: 0.8554216867469879
Val_Precision: 1.0
Val_Recall: 0.14285714285714285
Val_f1_score: 0.25
[[94  3]
 [19  3]]
Real_Accuracy: 0.8151260504201681
Real_Precision: 0.5
Real_Recall: 0.13636363636363635
Real_f1_score: 0.21428571428571427
```

<그림 12. 최소한의 데이터로 인공신경망 기법을 활용한 예측모델>

마지막으로, 인공신경망 기법을 활용한 예측 모델의 결과이다. robust scaling을 통해 데이터 값들을 스케일링 해주었고, 이를 validation과 test 과정을 통해 검증하였다. 정확도는 약 85%로 이전의 Extreme Randomized Tree Classifier 모델과 유사한 값이었고, 오히려 F1 score가 25%로 떨어진 수치를 보였다. 이를 통해 이번 예측 모델 생성에서는 Extreme Randomized Tree Classifier가 가장 적합한 모델이라는 결론을 얻을 수 있었다.

2) 결과 분석

ExtraTreeClassifier는 보다 폭넓은 시각으로 모델의 특성 중요도를 평가할 수 있기 때문에 해당 모델을 사용하여 모델의 특성 중요도를 시각화 해보았다. 상대적으로 중요도가 낮은 특성은 제거하고 상위 10개의 특성을 나타냈다.



<그림 13. 모델의 특성 중요도>

Extra Tree Classifier는 보다 폭넓은 시각으로 모델의 특성 중요도를 평가할 수 있기 때문에 해당 모델을 사용하여 모델의 특성 중요도를 시각화 해보았다. 상대적으로 중요도가 낮은 특성은 제거하고 상위 10개의 특성을 나타냈다.

1) goout(외출) : goout 특성은 친구와 함께 외출하는 정도를 숫자 1(매우 낮은) 에서 숫자

5(매우 높은) 으로 나타내었다. 특성 중요도 평가 결과 친구와 함께 외출하는 것이 청소년 알코올 소비에 가장 큰 영향을 준다는 것을 알 수 있었다. 선행연구에서도 성소년의 음주영향요인 중 또래 영향은 직접적인 영향을 미친다고 하였다. 따라서 우리 과제에서도 친구관계에 민감한 청소년기일수록 또래 동조성향이 강해 친구들과 함께 몰려다니며 잦은 외출을 할수록 알코올에 노출될 가능성이 높아질 것이라 추론할 수 있다.

2) absences(부재) : 부재 특성은 학교 결석 수를 0에서 93까지의 숫자로 나타내었다. 특성 중요도 평가 결과 학교 결석수가 음주 정도에 많은 영향을 준다는 것을 알 수 있었다. 알코올 섭취량이 많은 청소년일수록 불량할 가능성이 높고 그에 따라 학교 결석수가 많아진다는 추론이 가능하다.

3) G1(성적) : 성적 특성은 등급을 숫자 0에서 20까지 나타내었다. 원래 데이터에서는 G1이 1기 학년의 등급만을 의미하는 특성이지만 G1과 G2, G3의 상관관계를 고려하여 셋의 평균을 G1으로 설정하였다. 특성 중요도 평가 결과 성적이 청소년의 알코올 섭취 정도에 영향을 준다는 것을 알 수 있었다. 학업성적은 비행에 대하여 부정적인 인과적 영향을 미친다는 연구가 있다. 해당 연구에서 청소년들은 비행 때문에 학업성적이 낮아지는 것이 아닌, 낮은 학업성적으로 인하여 비행을 저지르게 된다고 주장하였다. 따라서 낮은 학업성적이 청소년의 알코올 소비를 늘린다는 추론이 가능하다.

4) age(연령) : 연령특성은 학생의 연령을 숫자 15에서 22까지로 나타내었다. 학년에 있어서 고 학년일수록 문제음주의 가능성이 상승한다는 선행 연구가 있다. 이는 고학년이 됨에 따라 가중되는 입시를 비롯한 사회적 요구에서 오는 부담이 그 실제적 원인이 된 것으로 예측된다 (박소연 외 1인. 2013). 특성 중요도를 분석해 본 결과에서도 연령이 청소년 음주에 비교적 높은 영향을 준다는 것을 알 수 있다.

5) freetime : 프리타임은 방과 후 자유 시간의 정도를 숫자 1(매우 낮은) 에서 숫자 5 (매우 높음) 으로 나타내었다. 여가 권태가 청소년의 상습적인 물질남용에 영향을 준다는 선행연구가 있다 (원형중.1999) . 여가 권태는 적정수준보다 많은 여가시간을 갖고 있거나 너무 바빠서 여가시간을 낼 수 없을 때에도 발생한다고 한다. 따라서 여가시간을 얼마나 갖는지가 청소년 음주에 유의미한 영향을 줄 수 있다고 추론할 수 있다.

6) studytime(학습시간) : 학습시간은 주간 학습시간을 숫자 1에서 10으로 나타내었다. 자기통제력이 높은 학생일 수록 학교수업 참여도가 높고 개인학습시간이 많으며 학업성적이 높다는 선행연구 결과가 있다 (하창순 외 3인. 2006) . 따라서 학습시간이 높은 학생은 자기통제력이 강해 과도한 음주를 할 가능성이 적어진다는 추론이 가능하다.

7) Fedu(아버지의 교육) : Fedu 특성은 아버지의 학력 수준을 숫자 : 0 - 없음, 1 - 초등 교육 (4 학년), 2 - 5 ~ 9 학년, 3 – 중등 교육 또는 4 – 고등 교육으로 나타내었다. 선행 연구에 따르면 부모의 사회, 경제적 지위가 청소년의 우울, 스트레스에 영향을 미친다. 그 중에서도 아버지의 높은 교육수준은 자녀로 하여금 학업에 큰 기대치를 부여할 수 있고, 서로 간의 불일치하는 목표로 인한 갈등이 청소년의 정신건강을 위협할 가능성이 있다고 한다. 본 프로젝트에서 이는 뒤에서 언급할 건강에 대한 요인과 일맥상통하며 청소년의 음주에 영향을 미친다는 것으로 추론할 수 있다.

8) sex(성별) : 섹스 특성은 학생의 특성을 여성 남성으로 나타내었다. 선행연구에 따르면 여학생일수록 문제음주의 개연성이 상승한다는 것을 알 수 있다. 따라서 이번 프로젝트 역시 성별의 특성 중요도가 높게 나왔다는 점에서 성별에 따라 알코올 소비에 차이가 있고 여학생의 경우가 남학생보다 알코올 중독 위험에 더 많이 노출된다는 것을 추론할 수 있다.

9) health(건강) : 건강 특성은 현재 건강 상태를 숫자 1에서 5까지로 나타내었다. 선행 연구에 따르면 청소년의 음주는 스트레스, 우울, 자살 등 정신건강과 관계가 있다고 하였다. 이번 프로젝트에서도 건강의 중요도가 높게 나왔다는 점에서 정신적 건강을 비롯한 건강 수준에 따라 청소년 알코올 소비에 영향을 준다는 것을 알 수 있다.

10) Medu(어머니의 교육): Medu 특성은 어머니의 학력수준을 숫자 : 0 - 없음, 1 - 초등 교육 (4 학년), 2 - 5 ~ 9 학년, 3 – 중등 교육 또는 4 – 고등 교육으로 나타내었다. 아버지의 교육 수준과 달리 어머니의 교육수준이 높은 경우 청소년기에 대한 이해도가 높을 가능성이 있어 청소년의 정서적 건강에 좋은 영향을 미친다는 선행연구가 있다. 이를 통해 아버지의 교육처럼 어머니의 교육이 청소년의 음주에 영향을 미친다는 것을 이번 프로젝트의 결과로써 확인할 수 있었다.

5. 결론

1) 결과 활용 방안

위와 같이 데이터 마이닝 기법을 이용해 얻은 결과를 청소년 음주와 관련해 실생활에 적용할 방법에 대해 생각해보았다.

<청소년 알코올 중독 사전예방 프로그램>

학교와 같은 교육기관에서는 학생들의 기본적인 인적사항과 교우관계 가족구성 등에 대한 데이터를 갖고있다. 따라서 이를 활용하여 알코올 중독 위험이 높은 학생들을 대상으로 집중적인 교육 프로그램을 계획하고 진행할 수 있을 것이다.

<알코올 중독 고위험 청소년 분류>

데이터 마이닝을 통하여 파악된 요인들을 데이터에 포함된 학생들의 분석에만 활용하는 것이 아닌 더 많은 숫자의 학생들에게 활용할 수 있다. CBR기법을 사용하여 비슷한 속성을 가진 학생들을 분류할 수 있다. 이렇게 속성별로 학생들을 분류할 경우 비슷한 그룹별로 집중관리가 가능하다는 장점이 있다.

<맞춤형 교육 프로그램>

예를 들어, 청소년 음주에 영향을 가장 많이 미치는 요인 중 하나가 높은 외출인데, 선행 연구에 따르면 친구가 음주를 하는 경우 본인도 음주를 할 확률이 높다고 하였다. 따라서 주변에 음주를 하는 친구가 있는 청소년이라면 알코올에 대한 위험성과 정보를 제공하고 어떻게 하면 효과적으로 유혹에서 벗어날 수 있는지에 대한 교육이 가능할 것이다. 또한, 음주로 치료가 필요한 청소년인 경우 현재 처해있는 원인에 대한 치료를 받을 수도 있지만 더 나아가 발생할 수 있을 법한 원인에 대해서도 맞춤 교육 프로그램을 통해 예방할 수도 있을 것이다.

2) 문제점 및 한계

첫째, 데이터 자체의 문제이다. 이번 프로젝트에서 사용한 데이터는 외국 학교의 학생들을 대상으로 한 데이터이다. 때문에 한국 학생들의 정서나 이유와 맞지 않는 부분이 있을 수도 있다고 생각하였다. 또한, 데이터는 382명의 학생들의 결과를 나타내 주고 있다. 더 많은 학생들을 분석했더라면 지금과는 다른 결과가 나올 수도 있고 보다 정확한 결론을 내릴 수 있었을 것 같아 아쉬움이 남는다.

둘째, 우리가 결론으로 내린 그래프는 각 attribute의 상대적 중요성을 나타낸 그래프이다. 때문에 정확히 어떤 성별이 영향을 더 미치는지, 어떤 연령대에서 청소년 음주가 많이 발생하는지 등은 그래프 상으로 파악할 수 없어 선행연구를 통해 추측할 수 밖에 없었다.

셋째, 알고리즘 종류의 한계이다. 이번 프로젝트에서는 의사결정나무 모델과 여기서 파생된 2개의 모델, 그리고 인공신경망을 활용한 분석 결과를 활용하였다. 이는 분류 모델로써 분류 기법 외에 CBR, Clustering, Regression 과 같은 알고리즘을 활용했다면 우리가 추출한 요인들 중에서도 동질적인 특성을 가진 요인들을 따로 뽑아낼 수 있었을 것이다. 이를 통해 예측

의 정확도를 보다 높일 수 있을 것이라고 생각한다.

3) 향후 발전 방안

이번 프로젝트를 통해 향후 발전시킬 수 있는 방안을 몇 가지 생각해 보았다.

첫째, 속성에 대한 방안이다. 이번 프로젝트에서는 다양한 데이터들을 각 속성으로 나누었다. 하지만 필요 없다고 생각하는 속성들을 직접 판단하여 지워가며 간단히 만들고 이를 예측 모델을 활용하여 예측하였다. 청소년 음주에 영향을 미치는 요인은 개인마다 그 정도가 확연히 다를 것이라고 생각한다. 때문에 이를 속성들의 상대적 중요도만 보고 지워 나가지 않고 개인의 성향과 정도를 반영했다면 더 유의미한 결과를 낳지 않았을까 생각한다.

둘째, 데이터에 대한 방안이다. 이번 프로젝트에는 해외 중고등학생을 대상으로 한 데이터를 사용하였다. 앞서 언급했듯이, 해외 학생들은 우리나라 학생들에 비해 개방적인 경향이 있으며 교육에 있어도 차이가 존재한다. 따라서 해외의 데이터를 우리나라 학생들에게 대입하여 일반화하기에는 다소 무리가 있다. 따라서 향후 연구에서는 우리나라 중고등 학생들의 데이터를 수집하여 보다 정확한 연구를 진행할 수 있을 것이다.

셋째, 알고리즘 종류에 대한 방안이다. 이번 프로젝트에서는 의사결정 나무와 여기에서 파생된 알고리즘, 인공신경망을 사용하여 모델을 예측하였다. 이런 트리 기반의 분류모델 이외에도 SVM 기법이나 Clustering 기법을 적용해 볼 수도 있다. 또한, 의사결정나무 알고리즘에서 결정 트리의 개수의 값을 개선하는 방법도 있다. 인공신경망 분석에서 우리는 Hidden layer층과 뉴런 값의 설정을 임의로 설정했지만 이를 반복문을 사용한다던지 하는 방식으로 최적의 값으로 설정한다면 더 높은 예측율을 기대할 수 있을 것이라 생각한다.

6. 참고문헌

김순규, 김숙경,「청소년 음주에 영향을 미치는 결정요인 분석」,「한국청소년학회」,2009년,20쪽

김인석 외 2명,「Data Mining을 이용한 음주 및 음주문제의 위험요인과 취약성요인에 관한 탐색」,「한국심리학회」,2001년,21쪽

정명숙, 이동수,「청소년의 음주영향요인과 음주행위 관계에 있어 성별에 따른 조절효과 분석」,「한국청소년학회」,2008년,30쪽

이숙현, 문상호「청년의 문제음주에 미치는 사회생태학적 결정요인에 관한 데이터 마이닝

분석」,「사회복지연구」,2018년,36쪽

이종원,「청소년의 학업성적과 비행, 비행친구의 관계에 대한 종단연구」,「한국청소년정책연구원」,2013년,82쪽

박소연, 이홍직,「청소년의 문제음주에 영향을 미치는 요인에 관한 연구 인구사회학적 특성 요인, 건강 요인, 일탈행동 요인을 중심으로」,「한국웰니스학회」,2013년,12쪽

김현옥, 전미숙,「청소년의 흡연, 음주와 정신건강과의 관계」,「한국보건간호학회지」,2007년,13쪽

윤명숙 외 2명,「청소년의 학업스트레스 및 또래애착이 우울에 미치는 영향. 인터넷 사용과 음주행위의 매개효과를 중심으로」,「경성대학교 사회과학연구소」,2009년,26쪽

박다혜, 장숙량,「부모의 사회 경제적 지위가 청소년의 스트레스, 우울, 자살생각에 미치는 영향」,2013년,10쪽

하창순 외 3명,「청소년의 자기 통제력, 학교수업 참여도, 개인 학습시간과 학업성적 간의 관계」,「한국청소년정책연구원」,2006년,20쪽

원형중,「청소년의 중독성 물질남용, 여가참여, 여가권태, 여가만족의 인과모형 설정을 위한 연구. 서울 시내 남자고등학생을 중심으로」,「한국여가크리에이션학회」,1999년,13쪽