산업공학 종합설계 보고서 (2017-2학기)

데이터마이닝을통한치매진단예측모델링

임동희 · 손소희 · 김수정 · 안채림
아주대학교 산업공학과

Datamining modeling for dementia diagnosis

Dong-Hee Lim · So-Hee Son · Soo-Jung Kim · Chae-Rim An
Yu-Kyung Shin
Department of Industrial Engineering, Ajou University

제출일: 2017년 12월 12일

지도교수: 신현정

배경: 최근 고령화 사회로 진입함에 따라 치매 환자 수도 점점 증가하고 있다. 정확한 조기 치매 진단이 이루어진다면, 약물이 아닌 치료 방법으로 치매를 조기에 치료할 수 있다. 기존의 치매 진단 프로세스는 환자의 의료 정보 및 보호자의 설문조사를 기반으로 의사가 치매를 진단한다. 여기서 문제는 시간이 많이 소요되고 주관적이며 오류가 발생할 가능성이 있다.
연구의 목적과 내용: 본 연구에서는 이러한 문제를 해결하기 위하여 데이터 모델링을 통해 치매 진단 프로세스를 표준화할 수 있는 프로세스를 제안한다. 문제 해결 과정을 통해 치매 진단을 직관적으로 결정할 수 있도록 하는 것이 본 연구의 목적이다.
연구 수행 방법: 치매 진단에 유의한 진단 검사를 선택하기 위해 chi-square test, logistic regression, decision tree, discriminant analysis, 변수 선택법을 사용하였다. 이 과정을 통해 도출된 유의 진단 기준을 이용하여 기계학습 알고리즘을 기반으로 치매 진단 모델링을 진행한다.
연구 결과: 5개의 기계 학습 모델 중 가장 성능이 우수한 예측 모델은 평균 AUC가 가장 높은 랜덤포레스트이다. 랜덤포레스트 모델의 검증 결과 민감도는 81.69%, 특이도는 93.17%로 정상을 치매로 진단하는 제1종 오류는 약 7%, 치매를 정상으로 진단하는 제2종 오류는 약 18%가 나왔다.
연구 결과의 활용: 본 연구의 결과는 치매 진단에 있어 의사의 진단 프로세스 표준화에 도움을 줄 것이다. 또한 향상된 치매 진단 프로세스를 통해 의사뿐만 아니라 환자의 시간과 비용을 절감할 수 있을 것이다.

주제어 (Keywords) : Data mining modeling, Dementia diagnosis, Variable Selection, discriminant Analysis, Logistic Regression, Decision Tree, Random Forest

1. 서론

1.1 프로젝트배경

현재 우리나라는 공식적으로 고령사회에 진입하였다. 2017년 9월 3일 행정안전부가 발표한 8월 말 주민등록 인구에 따르면 65세 이상 노인 인구는 7,257,288명이다. 기존 통계청 예상보다 1년 빨라졌다[1]. 이처럼 점점 고령화 속도가 빨라짐에 따라, 치매 환자 또한 증가할 것으로 예상된다. 치매를 조기에 진단한다면 약물이 아닌 치료 방법으로 치매를 예방하고 경미한 단계의 치매 증상을 효과적으로 관리할 수 있다. 일반적으로 치매 진단 프로세스는 환자의 인지기능 평가, 보호자를 통한 일상생활 능력 평가 등 다양한 검사로 진행된다[2]. 환자의 신경심리검사로는 숫자 외우기 검사, RCFT(시각 및 공간 검사)를 통한 기억장애, 언어장애 및 시공간 능력에 대한 검사가 진행된다. 그 외에도 KDSQ, Stroop Test 등 다양한 검사가 진행된다. 그러나 이러한 환자의 의료 정보 및 후견인의 설문 조사를 기반으로 하는 진단 프로세스가 시간이 많이 걸리고 의사의 평가가 주관적이기 때문에 오류가 발생하기 쉽다. 따라서 이러한 문제를 해결하기 위해, 치매 진단 모델링을 통한 의사의 평가 프로세스를 표준화하는 것의 필요성이 증가하고 있다.

1.2 프로젝트목적및기대효과

본 연구에서는 치매 진단 과정을 개선하기 위해, 데이터 마이닝 모델링을 통한 치매 진단 프로세스의 표준화를 목표로 한다. 먼저, 치매 진단 프로세스 개선에 있어 중요한 것은 기존 치매 진단 검사 중 유의한 검사를 선택하는 것이다. 기존의 치매 진단 검사는 보호자를 대상으로 하는 설문조사 형태의 인지선별검사와 환자를 대상으로 하는 인지기능검사가 있다. 본 연구에서는 이러한 검사 중 유의한 검사를 선별하고, 이에 대해 기계 학습 알고리즘을 사용하여 예측 모델을 생성하고자 한다. 이 모델은 치매 진단을 위한 의사결정을 보조할 것이다. 또한 의사와 환자 모두의 시간과 비용을 효율적으로 절감할 수 있을 것으로 기대된다.

2. 방법론

2.1 모델링

2.1.1 로지스틱회귀모형

로지스틱 회귀모형은 목표변수가 범주형인 경우 분류의 목적으로 많이 쓰이며, 각 예측변수의 효과를 파악하기 좋은 모형이다. 새로운 예측변수의 값이 주어질 때 목표변수의 각 범주에 속할 확률이 얼마인지를 추정하여, 추정확률(p)을 기준치에 따라 분류하는 목적으로 사용된다. 이항 목표변수 Y에 대해 예측변수가 여러 개인 로지스틱 회귀모형의 일반적 형태는 다음과 같다.

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n \quad (1)$$

$$\text{odds} = \frac{p(y=1|x)}{1-p(y=1|x)} \quad (2)$$

(1)은 (2)의 오즈비를 활용하여 logit변환을 거쳐 나온 식이다. 오즈비는 개별 케이스가 어떤 집단에 속할 확률이 속하지 않을 확률의 몇 배인지 나타내는 비율 값이다. 추정확률 p는 개별 케이스가 어떤 집단에 속할 확률을 뜻한다. p가 기준값보다 크면 Y=1인 집단으로, 작으면 Y=0인 집단으로 분류하게 된다. 이때 분류 기준값은 사전정보, 손실함수, 정분류율 등을 동시에 고려하여 결정한다.

2.1.2 판별분석

판별분석은 둘 이상의 예측변수를 이용하여 범주형 목표변수에 관측치를 분류하는 통계적 분석방법이다. 모델링에서 사용한 판별분석은 Fisher의 선형판별분석 방법으로, 집단 간 분산은 크도록, 집단 내 분산은 작도록 차원축소 방식을 이용한다.

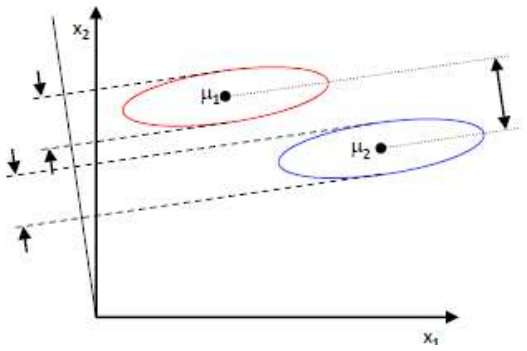

Figure1. Fisher의 차원축소

소속그룹이 이미 알려진 관측치에 대하여 예측변수 x1, x2, …, xp를 측정하고, 이 변수들의 선형 결합하여 판별에 유용한 변수 Y1, Y2, …, Ym으로 변환한다. 이 Yi들을 선형결합변수라 한다. Y의 분포가 집단 간 가장 크게 다르도록 가중치를 설정하고, 집단 간 평균차이가 가장 크게 되는 예측변수의 선형결합을 결정한다. 결정된 선형결합 Y에 대해 각 관측치 별로 집단 중심과의 거리가 더 가까운 집단에 분류한다.

2.1.3 의사결정나무(CART)

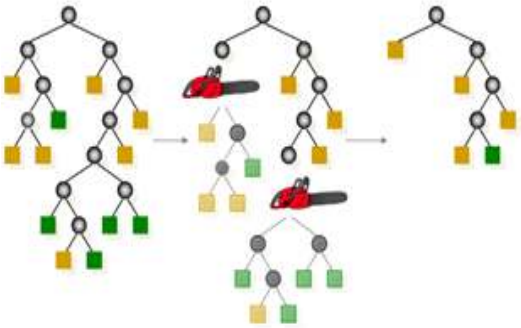

Figure2. CART 알고리즘의 의사결정나무

의사결정나무 모형은 의사결정 규칙을 나무구조로 나타내어 전체자료를 몇 개의 소집단으로 분류하거나 예측하는 분석방법이다. 상위노드로부터 하위노드로 가지 분할하는 매 단계마다 분류변수와 분류 기준 값의 선택이 중요하다. 분류변수와 분류 기준 값은 분할되는 하위노드에서 노드 내 동질성, 노드 간 이질성이 커지도록 선택된다. 즉, 노드 내의 이질성이 감소하고, 순수도가 증가하는 방향으로 나무를 형성해간다.

CART알고리즘의 의사결정나무의 경우 지니지수 값이 작아지는 방향으로 가지분할을 수행하여 항상 이진나무를 형성한다.

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2 \qquad (3)$$

CART알고리즘의 경우, 과적합 문제를 해소하기 위해 생성된 가지를 잘라내고 모형을 단순화하는 가치지기가 반드시 수행되어야 한다. 각 노드의 에러

율을 측정하여 가장 작게 나타나는 지점에서 가지치기를 수행하여 더 이상 순수도를 높일 수 없을 때 가지치기를 종료한다.

2.1.4 의사결정나무(ctree)

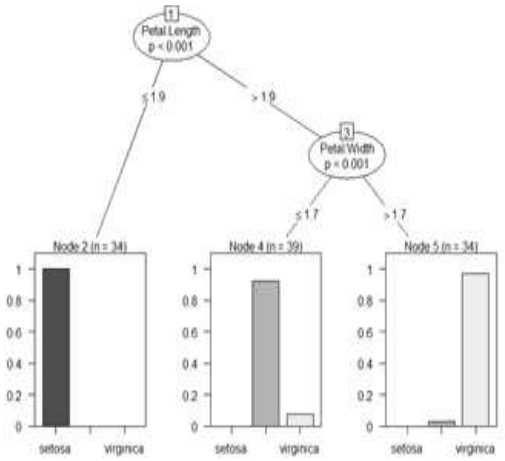

Figure3. ctree알고리즘의 의사결정나무

ctree 알고리즘의 의사결정나무는 CART 알고리즘과 같은 의사결정나무 모형의 과적합 문제, 가지치기 수행함의 불편함을 보완한 모형이다. Permutation test의 조건부 추론을 통해 통계적 유의성을 검정하는 방법으로 수행된다.

$$H_0 = \cap_{j=1}^{m} H_0^j \text{ and } H_0^j : D(\mathbf{Y}|X_j) = D(\mathbf{Y}) \quad (4)$$

$$\mathbf{T}_j(\mathcal{L}_n, \mathbf{w}) = \text{vec}\left(\sum_{i=1}^{n} w_i g_j(X_{ji}) h(\mathbf{Y}_i, (\mathbf{Y}_1, \ldots, \mathbf{Y}_n))^\top\right) \in \mathbb{R}^{pq} \quad (5)$$

(4)의 각각의 Xj와 Y의 독립을 확인하기 위한 조건부 추론을 통해 귀무가설이 기각되지 않으면 나무 형성을 멈춘다. (5)의 test 통계량을 도출하여 p값이 작은 분할변수를 선택하고, t값이 가장 큰 변수로 가지분할을 수행한다. 이 과정을 반복하여 나무를 완성시킨다. 과정 중 multiple test를 이용해 유의수준을 조정하며 수행한다.

2.1.5 랜덤포레스트

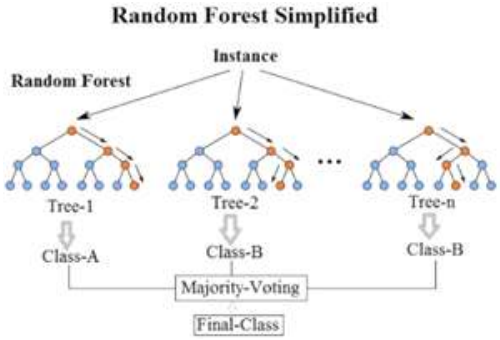

Figure4. 랜덤포레스트 모형

랜덤포레스트 모형은 여러 개의 분류모형에 의한 결과를 종합하여 분류의 정확도를 높이는 앙상블 모형으로 부트스트랩, 배깅, 앙상블 모형의 장점을 모두 취한 강력한 모형이라 할 수 있다. 부트스트랩(원 데이터에서 크기가 같은 표본을 여러 번 단순임의복원추출)을 하고, 각 샘플에 대해 나무를 형성해나간다. 이때 각 노드마다 모든 예측변수를 무작위로 추출하고, 추출된 변수 내에서 최적의 분할을 만들어 나가는 방법을 사용한다. 새로운 데이터에 대해 구성된 다중의 트리 중 다수결의 방법으로 분류한다. 이러한 랜덤포레스트 모형은 예측력이 우수하고 높은 안정성을 지니고 있다.

2.2 모델성능평가

본 연구에서 사용한 지도학습 모델의 성능을 비교하기 위해 Stratified K-fold cross validation과 ROC곡선 및 AUC를 사용하여 모델의 성능을 평가·비교하였다.

2.2.1 StratifiedK-foldcrossvalidation

K-fold cross validation(이하 K-fold

CV)은 수집된 샘플들의 검증을 위한 통계적인 분석 방법이다. K-fold CV는 분류기 성능 측정의 통계적 신뢰도를 높이기 위해서 resampling 방법을 사용한다. Data training을 통한 모델링(modeling)을 할 때 주어진 Data set에 대해서만 높은 예측 성능(prediction performance)을 보이는 과적합(Overfiting)이 발생할 가능성이 있다는 점에서 높은 오류(error)를 발생시킬 수 있기 때문에 K-fold CV를 이용하여 모델을 평가한다.

본 연구에서는 데이터 셋을 training set과 test set으로 나눈 후, training set에 대하여 K(10)개의 sub set으로 분리하였다. 하나의 sub set에 대해 90:10으로 나누어 90%은 모델 생성을 위한 training set로, 10%는 모델 평가를 위한 validation set으로 이용하였다. Cross-validation 프로세스 동안 이 과정을 10번 반복한다. 프로세스의 각 스텝마다 각 부분으로부터 나온 10개의 결과는 하나의 평가 지표로 만들기 위해 평균을 구하며, 이를 이용해 검증을 수행할 수 있다.

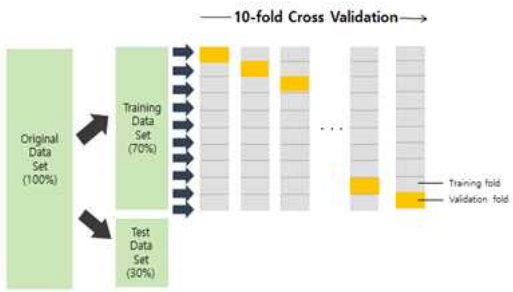

Figure5. Stratified K-fold cross validation

2.2.2 ROC곡선/AUC

ROC(Receiver Operating Characteristic) curve란 민감도(sensitivity)와 특이도(specificity)가 어떤 관계를 갖고 변하는지를 이차원 평면상에 표현한 것으로 두 분류 분석 모형을 비교 분석 결과를 가시화할 수 있다는 점에서 유용한 평가 도구이다. TPR와 FPR을 각각 x,y축으로 놓은 그래프로써 민감도(sensitivity)란 1인 케이스에 대해 1이라고 예측하는 것이고, 특이도(specificity)는 0인 케이스에 대해 0이라고 예측하는 것이다. TPR 과 FPR 은 다음과 같다.

$$\text{TPR(True Positive Rate)=민감도} \quad (6)$$

$$\text{FPR(False Positive Rate)=1-특이도} \quad (7)$$

예를 들어, TPR은 치매환자를 진찰해서 치매라고 진단한 경우이고, FPR은 치매환자가 아닌데 치매라고 진단 한 경우이다. ROC curve의 그래프가 왼쪽 꼭대기에 가깝게 그려질수록 TPR과 FPR 모두 다 맞추는 위치가 되어 분류 성능이 우수하다고 본다.

AUC(the Area Under a ROC Curve)란 분류기의 합리성을 나타내는 지표로 ROC curve의 아래 면적을 의미한다. AUC의 최대값은 1이며, 1에 가까울수록 민감도와 특이도가 모두 높은 것이니 좋은 분류기라고 할 수 있다. AUC가 0.5 아래로 떨어지면 성능이 매우 좋지 않다고 판단할 수 있다. AUC가 0.5 아래로 떨어지면 성능이 좋지 않다고 판단할 수 있다.

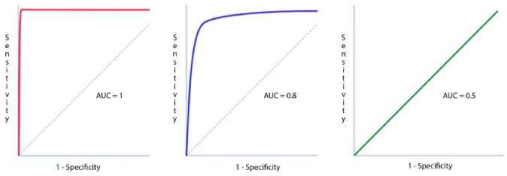

Figure6. ROC curve/AUC

3. 실험

3.1 데이터설명

Table1. 변수 목록

번호	변수명	설명	종류
1	Sex	성별	Binary
2	Phone.N	전화사용 - 현재	Nominal
3	Shopping.N	쇼핑 - 현재	Nominal
4	Shopping.P	쇼핑 - 참재	Nominal
5	Cook.N	요리하기 - 현재	Nominal
6	Cook.P	요리하기 - 참재	Nominal
7	HouseWork.N	집안일 하기 - 현재	Nominal
8	HouseWork.P	집안일 하기 - 참재	Nominal
9	Transportation.P	대중교통 이용 - 참재	Nominal
10	NearGoOut.P	근거리 외출 - 현재	Nominal
11	NearGoOut.N	근거리 외출 - 참재	Nominal
12	EatMedicine.P	약 챙겨먹기 - 현재	Nominal
13	MakeUp.P	몸단장 - 참재	Nominal
14	SayRecent.N	최근에 있었던 일 이야기하기 - 현재	Nominal
15	SayRecent.P	최근에 있었던 일 이야기하기 - 참재	Nominal
16	Hobby.N	여가활동 - 현재	Nominal
17	Hobby.P	여가활동 - 참재	Nominal
18	KDSQ-1	KDSQ-1	Nominal
19	KDSQ-13	KDSQ-13	Nominal
20	KDSQ-14	KDSQ-14	Nominal
21	KDSQ-15	KDSQ-15	Nominal
22	ClearBig	대변 가리기	Nominal
23	ClearSmall	소변 가리기	Nominal
24	Facing	세수	Binary
25	Toilet	화장실 사용	Nominal
26	Eating	식사	Nominal
27	MovingToChair	바닥에서 의자로 옮겨가기	Nominal
28	Walk	보행	Nominal
29	Clothes	옷 입기	Nominal
30	Stairs	계단 오르내리기	Nominal
31	Bath	목욕하기	Binary
32	Delusion	1망상 (잘못된 믿음)	Nominal
33	PainOfProtector	1이와 같은 행동이 감정적으로 보호자께 얼마나 고통이 됩니까?	Nominal
34	Repetition	2비정상적인 반복행동	Nominal
35	PainOfProtector2	2이와 같은 행동이 감정적으로 보호자께 얼마나 고통이 됩니까?	Nominal
36	EduBckgr	학력	Nominal
37	YearOfStudy	교육연수	Continuous
38	Time.Year	시간 지남력 - 년	Binary
39	Time.Month	시간 지남력 - 월	Binary
40	Time.Day	시간 지남력 - 일	Binary
41	Time.Weekdays	시간 지남력 - 요일	Binary
42	Time.Season	시간 지남력 - 계절	Binary
43	Place.City	장소 지남력 - 시, 도	Binary
44	Place.Current	장소 지남력 - 현재장소	Binary
45	Place.Layers	장소 지남력 - 댁 층	Binary
46	p_TG	글리코겐에 대한 단백질 표적화	Binary
47	SVLT_recall_total_score	서울언어학습검사_회상 총 점수	Continuous
48	SVLT_Delayed_recall	서울언어학습검사_지연회상	Continuous
49	SVLT_recognition_FP	서울언어학습검사_재인	Continuous
50	Digit_span_Forward_Backward	숫자 바로/거꾸로 외우기 검사	Continuous
51	Spontaneous_speech_fluency	자발적말하기 유창성	Nominal
52	Spontaneous_speech_contents	자발적말하기 내용	Nominal
53	RCFT_delayed_recall	Rey복합도형 검사	Continuous
54	RCFT_delayed_recall	Rey복합도형 검사	Continuous
55	Go_No_Go	행동신경검사	Continuous
56	COWAT_animal	통제단어 연상 검사_동물	Continuous
57	COWAT_supermarket	통제단어 연상 검사_슈퍼	Continuous
58	StroopTest_Colorreading_correct	음소적 단어 유창성, 색상 글자 스트룹 검사	Continuous
59	Babinski sign(우)	바빈스키 반사(우)	Binary
60	Diabetes	당뇨	Nominal
61	HBPressure	고혈압	Nominal
62	CDR	치매 임상 평가 척도	Nominal

목표변수인 CDR의 구성은 0, 0.5(정상)과 1, 2, 3(치매)로 Nominal type이다. 예측변수는 총 61개로 Binary type은 12개, Nominal type은 38개, Continuous type은 11개이다. Sex(성별)을 제외한 대부분의 변수가 설문조사를 기반으로 하고 있는데, Binary나 Nominal type의 변수들은 설문조사의 구체적인 문항에 해당한다. 그리고 Continuous type의 변수들은 교육연수 외에 서울언어학습검사, 행동신경검사 등의 대부분 치매 진단을 위한 검사로 구성되어있음을 알 수 있다.

3.2 데이터전처리

3.2.1 결측치처리

목표 변수(CDR)는 우리가 분류를 해야 하는 대상으로써, 해당 결측치를 대체하는 것은 결과를 왜곡시키고 무의미하다고 판단하여 해당 관측치들을 제거하고, 예측변수들에 대해서 결측치 대체를 하였다.

예측변수들에 대해서 결측치 대체를 위해 사용한 방법은 3가지이다. 다중대체법(mice)과 피어슨 잔차, 그리고 SAS E-miner의 결측치 처리 방식을 사용하였다.

다중대체법(mice)은 데이터 셋의 결측 패턴이 결측과 무응답을 가진 자료로써 일반적 결측 메커니즘에 따른다는 가정 하에 결측치를 대체하게 된다. 1회마다 한 개의 예측변수에 대해 나머지 예측변수들을 모형화하여 연쇄적으로 결측치를 대체하게 된다. 해당 방식은 R의 mice 패키지를 사용하였으며 그 결과 원 데이터 대한 결측률 19.28%에서 1.66%로 줄어들었다.

피어슨 잔차는 두 개의 예측변수들의 수준 간에 유의미한 차이가 있는지 확인할 때 사용을 하게 된다. 한 예측 변수의 수준이 다른 예측변수의 수준과 비교하여 관측 도수와 기대 도수가 차이가 있을 때 서로 연관성이 있다고 판단하여, 한 예측변수의 결측치들을 다른 예측 변수의 수준과 비교하여 대체하였다. 그 결과 60개의 예측 변수 중 12개 변수들을 대체할 수 있었지만 결측 연관성이 높은 변수들이 많아 모두 대체하기에는 한계가 있었다.

마지막으로 SAS E-miner의 결측치 처리 방식을 이용하였다. 이산형 변수에

대해서는 데이터의 최빈값을 기준으로 대체하는 Count 방식을 사용하였고, 연속형 변수는 의사결정 알고리즘으로 분류하여 대체를 하는 Tree 방식을 사용하였다. 그 결과 이산형 변수와 연속형 변수 약 10% 와 53%의 결측률을 완전히 처리 할 수 있었다.

최종적으로 우리는 모든 결측값을 처리할 수 있는 SAS E-miner 방식을 선택하게 되었다.

3.2.2 이상치처리

이상치 처리의 방식은 다음과 같다. 먼저 CrossTable로 판단하여 관련 있는 목표변수끼리 이상치를 대체하였다.

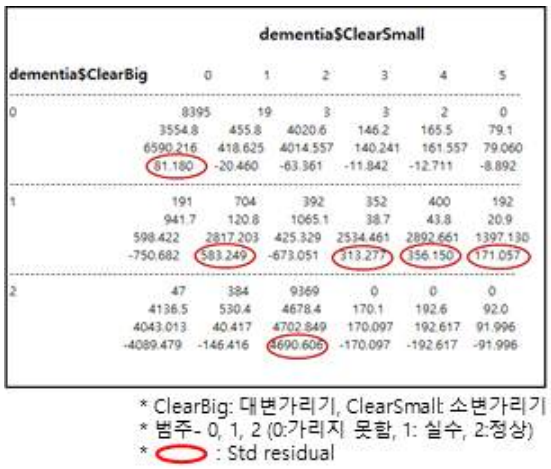

Figure7. CrossTable을 이용한 이상치 처리 예시

'ClearBig'과 'ClearSmall'은 원래 0(가리지 못함), 1(실수), 2(정상)의 수준으로만 구성되어야 하는데, Figure7을 보면 'ClearSmall'의 수준 3, 4, 5가 추가로 존재하고 있다. 'ClearSmall'의 수준 3, 4, 5를 CrossTable을 통해 본 결과 'ClearBig'의 수준의 1과 깊은 관련성을 보였고, 3, 4, 5의 수준을 1로 대체하게 되었다.

다음으로 범주가 주어지지 않은 예측변수에 대해서는 히스토그램으로 이상 기준을 판단하여 이상치를 처리하였다.

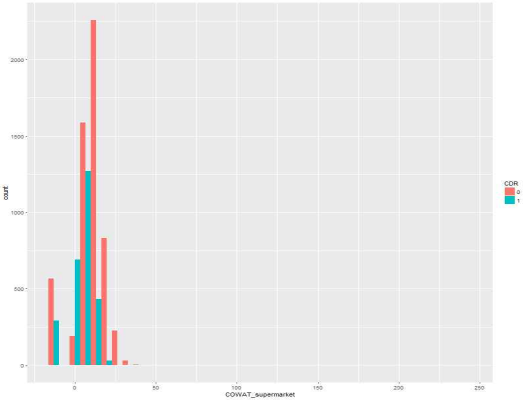

Figure8. 히스토그램을 이용한 이상치 판단 예시

Figure 8을 보면 'COWAT supermarket'은 대부분의 범위가 0~44 사이에 있으며 하나의 관측치만 245의 값을 가졌다. 마찬가지로 'RCFT_delayed_recall'에서 36을 초과하는 5개의 관측치, 'StroopTest_Colorreading_correct'에서 789의 값을 가지는 1개의 관측치 등 대부분의 범위에서 크게 벗어난다고 판단이 되는 총 11개의 관측치들을 이상치라 보고 제거하였다.

마지막으로 목표변수(CDR)에 대해 무의미한 변수나 범주를 히스토그램으로 판단한 후 제거 하였다.

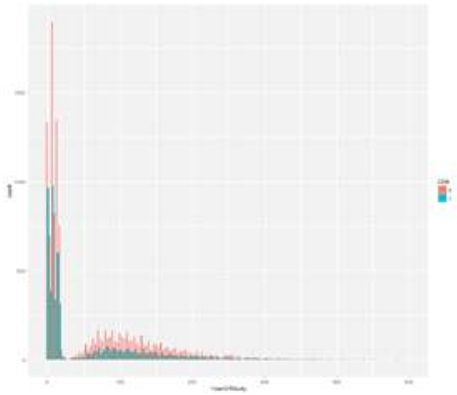

Figure9. 히스토그램을 이용한 의미 없는 변수 판단 예시

Figure9의 연속형 변수 'p-TG'의 경우 그 값이 변함에 따라 목표변수(CDR)의 비율(정상:치매 = 2:1)이 변화가 없으므로 무의미한 변수라고 판단하였다. 또한 명목형 변수에 대해서 '-1000'의 수준에서 2:1의 목표변수(CDR)의 비율을 가지는 경우 역시 무의미하다고 판단하여 관측치를 제거하였다. 다음과 같은 방식으로 총 2483개의 관측치를 제거할 수 있었다.

3.2.3 전처리 결과

데이터 전처리를 한 결과 총 62개의 변수 중 59개의 변수를 사용하게 되었으며 관측치의 개수는 21094개에서 18600개로 줄어들었다.

3.3 유의변수도출

치매 진단 모델링을 하기 전, 치매 진단을 위한 다양한 검사 중 중요한 진단 검사를 선정하고자 한다. 앞서 언급한 바와 같이 치매 진단을 위한 검사들이 예측변수이기 때문에, 즉 이 과정은 치매 판단에 유의한 영향을 미치는 변수를 찾아내는 것이다.

본 연구의 목적은 치매 진단이 목적이므로 전처리가 완료된 데이터에서 CDR(치매 임상 평가 척도)을 nominal type(0, 0.5, 1, 2, 3)에서 binary type(0:정상, 1:치매)로 변환 후 진행하였다. 유의 변수 도출 과정에서 사용된 방법은 Table.2에 기재된 7가지 방법을 사용하였다.

R을 이용하여 Chi-square test, 로지스틱 회귀(단계적 방법), 의사결정나무(CART, ctree), 판별분석을 진행하였다. SAS를 이용하여 의사결정나무(최대가지:2, 최대깊이:8), 변수 선택 방법을 사용하여 진행하였다.

Table2. 유의 변수 선택 방법

유의변수 선택 방법	유의 판단 기준	ACC (정확도)	유의 변수 개수
Chi-square test	Chisq p-value < 0.05	-	45
Logistic Regression (stepwise)	p-value < 0.05	0.900	39
Decision Tree (CART)	Gini Index ↓	0.8677	21
Decision Tree (ctree)	Permutation test p-value	0.8597	34

Discriminant Analysis	Accuracy ↑	0.8807	43
Decision Tree (SAS)	Chisq p-value ↓	0.8900	31
Variable Selection (SAS)	R-square ↑	0.8803	28

Table3. 유의 변수 선택 방법 예시

	KDSQ .13	Walk	Bath	Eating	Toilet	Delusion
chi-square	✓	✓	✓	✓	✓	✓
로지스틱 회귀 (stepwise)	✓	✓	✓	✓	✓	
의사결정나무 (CART)	✓					
의사결정나무 (ctree)	✓	✓	✓			✓
판별분석	✓	✓	✓	✓	✓	
의사결정나무 (SAS)	✓	✓	✓			
변수 선택 방법 (SAS)	✓	✓		✓		

각 방법에 따라 유의 변수를 도출한 결과, 변수 선택 결과가 다양하게 선택되어 있었기 때문에 변수 선택에 있어 한 방법만을 선택할 수 없었다. 따라서 7개의 방법에서 3개 이상의 방법이 유의하다고 선택한 변수들을 최종적으로 유의한 변수라고 판단하였다. 예를 들어, 아래의 Table3.에서 KDSQ.13은 7개의 방법에서 모두 유의한 변수로 선택되었기 때문에 유의변수로 선택하였고, Delusion의 경우 2개의 방법인 Chi-square test와 의사결정나무(ctree)에서만 선택이 되었기 때문에 유의변수로 선택하지 않았다. 이러한 방법으로 유의변수를 도출한 결과, 총 45개의 유의변수를 도출할 수 있었다.

3.4 모델링

```
##data 에 분석할 데이터 할당.
data <- read.csv("dementia_SAS_RAW_유의변수탐색(0,1)_모델7개.csv")

##data 변수 type 정의 예시(Binaray/Norminal -> Factor, Continuous -> Numeric)
data$CDR<-as.factor(data$CDR)
data$Phone.N<-as.factor(data$Phone.N)
data$Shopping.N<-as.factor(data$Shopping.N)
data$Shopping.P<-as.factor(data$Shopping.P)

data$SVLT_recall_total_score<-as.numeric(data$SVLT_recall_total_score)
data$SVLT_Delayed_recall<-as.numeric(data$SVLT_Delayed_recall)
data$SVLT_recognition_FP<-as.numeric(data$SVLT_recognition_FP)
data$Digit_span_Forward_Backward<-as.numeric(data$Digit_span_Forward_Backward)
data$RCFT_delayed_recall<-as.numeric(data$RCFT_delayed_recall)
data$Go_No_Go<-as.numeric(data$Go_No_Go)
```

Figure10. 변수 타입 지정 R 코드

Figure10.은 data의 변수별로 변수 타입을 지정해주는 코드이다. Binary type과 Nominal type은 Factor형태, Continuous type은 Numeric 형태로 지정해주었다.

```r
total <- data

set.seed(1234)
idx =sample(1:nrow(total), nrow(total)*0.7)

# train set과 test set으로 나누기
train <- total[idx,]
test <- total[-idx,]

# k=10인 k-fold를 진행
k1=10
set.seed(1234)
list = 1:k1

# k-fold를 위해 sub-set id 부여하기
library(dismo)
# k-fold는 CDR에 대해서 계층적(Stratifeid)하다.
train$id <- kfold(train, k=k1,by=train$CDR)

# 예측에 대한 결과를 담는 변수 생성
result.validation = data.frame()
Acc_1 = NULL
Model_list = list()
Auc_1 = NULL
```

Figure11. k-fold를 위한 데이터 분할 R 코드

변수 타입이 지정된 데이터는 train과 test set으로 분할하고 train set에 대해 k-fold를 하기 위해 fold 번호를 부여해 주었다.

```r
for(i in 1:k1){

  # train-set과 validation-set으로 나누기
  semi_train <- subset(train, id %in% list[-i])
  validation <- subset(train, id %in% c(i))


  # 모델링 하기
  rpartmod <- rpart(CDR ~.-id , data= semi_train, method = "class")

  ptree<-prune(rpartmod, cp = rpartmod$cptable[which.min(rpartmod$cptable[,"xerror"]),"CP"])

  # 모델 저장 하기
  Model_list <- append(Model_list,list(ptree))


  # 예측하기
  predict.y <- try(predict(ptree,validation),silent = T)

  if(class(predict.y)!="try-error"){

    Real = as.character(validation$CDR)
    Predict = as.character(round(predict.y[,2]))
    validation$Predict <- as.character(round(predict.y[,2]))


    Acc_1 <- append(Acc_1, sum(Real==Predict)/length(Real) )
    result.validation <- rbind.data.frame(result.validation,
                                          validation)


    # ROC 곡선 및 AUC 값 구하기
    pr.reduced <- prediction(as.numeric(predict.y[,2]),as.numeric(Real))
    prf.reduced <- performance(pr.reduced, measure = "tpr", x.measure = "fpr")

    png(filename = paste("p_1_",i,".png"),width = 2000,height = 2000)
    plot(prf.reduced)
    dev.off()

    auc.reduced <- performance(pr.reduced, measure = "auc")
    auc.reduced <- auc.reduced@y.values[[1]]
    Auc_1 <- append(Auc_1, auc.reduced)


  }else{
    # 예측을 하는 데 오류가 난 경우
    print(paste(i,"오류"))
  }

}
```

Figure12. k-fold를 이용한 모델 생성 및 평가 R 코드

Figure12.는 k-fold를 이용해 총 10번의 data set을 생성하고, 각각 validation set에 대해 평가하는 코드이다. validation set에 대한 AUC와 Accuracy는 각각 저장하여 모델의 평균적인 성능을 평가하고 최종 모델을 결정하는 데 사용한다.

3.4.1 로지스틱회귀분석

```r
# formula 식 만들기
formula <- CDR ~Bath + ClearBig + ClearSmall + Clothes + Cook.P + Diabetes + EatMedicine.P +
  Eating + EduBckgr + Hobby.N + Hobby.P + HouseWork.N + HouseWork.P + KDSQ.1 + KDSQ.13 + KDSQ.14 +
  KDSQ.15 + MakeUp.P + MovingToChair + NearGoOut.N + NearGoOut.P + PainOfProtector + PainOfProtector2 +
  Phone.N + Place.City + Place.Current + Place.Layers + SayRecent.N + Shopping.N + Shopping.P +
  Stairs + Time.Day + Time.Month + Time.Season + Time.Weekdays + Time.Year + Toilet + Transportation.P +
  Walk + COWAT_animal + COWAT_supermarket + Go_No_Go + RCFT_delayed_recall + SVLT_Delayed_recall +
  StroopTest_Colorreading_correct

# 로지스틱 회귀 모델링
model.full <- glm(formula, data = train, family = "binomial")

# 단계적 방법
reduced.model=step(model.full, direction="both")
```

Figure13. 로지스틱 회귀 모델링 및 단계적 방법 R 코드

```
> summary(model2)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.3209  -0.3575  -0.1896   0.1808   3.3933

Coefficients:
                                Estimate Std. Error z value Pr(>|z|)
(Intercept)                     3.300359   0.495832   6.656 2.81e-11 ***
Bath1                          -0.381950   0.100778  -3.790 0.000151 ***
ClearBig1                       0.439932   0.107557   4.090 4.31e-05 ***
ClearBig2                       0.071354   0.212786   0.335 0.737373
Clothes1                       -0.625890   0.097878  -6.395 1.61e-10 ***
Clothes2                       -0.404873   0.231533  -1.749 0.080349 .
Cook.P1                         0.218548   0.091274   2.394 0.016647 *
Cook.P2                         0.268007   0.097833   2.739 0.006155 **
Cook.P3                         0.303054   0.146824   2.064 0.039012 *
Diabetes1                      -0.040117   0.117398  -0.342 0.732560
Diabetes2                       0.921936   0.284115   3.245 0.001175 **
Eating1                         0.524079   0.236941   2.212 0.026977 *
(가운데 71개 변수 중략)
COWAT_animal                   -0.083461   0.019552  -4.269 1.97e-05 ***
COWAT_supermarket              -0.103863   0.015952  -6.511 7.46e-11 ***
RCFT_delayed_recall            -0.067493   0.016080  -4.197 2.70e-05 ***
SVLT_Delayed_recall            -0.121830   0.036283  -3.358 0.000786 ***
StroopTest_Colorreading_correct -0.005992   0.002406  -2.491 0.012749 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 16359  on 13019  degrees of freedom
Residual deviance:  7011  on 12932  degrees of freedom
AIC: 7187

Number of Fisher Scoring iterations: 7
```

Figure14. 로지스틱 회귀 모델 결과 예시

로지스틱 회귀 결과 'Bath'의 0 수준이 유의하다고 나왔으며 'ClearBig'의 0 수준 역시 유의하다고 나왔음을 알 수 있다. 예측변수 모두 유의함을 보였다.

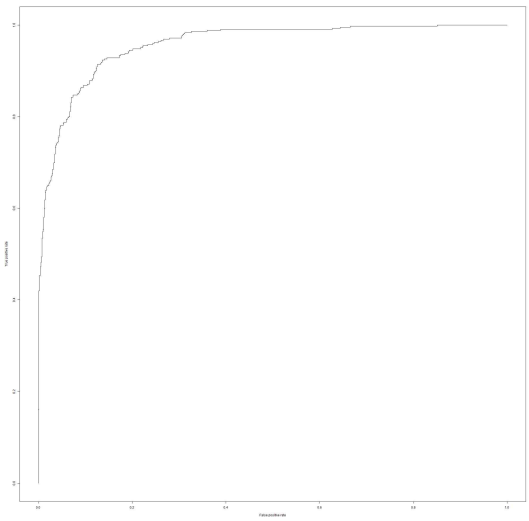

Figure15. 로지스틱 회귀 모델 ROC곡선 예시

로지스틱 회귀 분석을 k-fold를 통해 성능 평가를 한 결과 k=10 중 최대 AUC는 0.9569784였으며, 평균 AUC는 0.94913이 나왔다.

3.4.2 판별분석

```
61  # 판별 분석 모델링
62  library(MASS)
63  model <- lda(CDR ~.-id , data= semi_train)
```

Figure16. 판별분석 모델링 R 코드

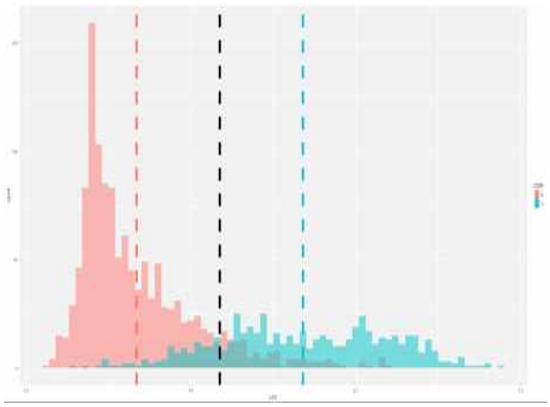

Figure17. 선형결합 변수 히스토그램 결과

판별분석 통해 만들어진 선형결합 변수에 'CDR'에 따라 히스토그램을 그린 결과 Figure17.과 같이 나왔다. 빨간 점선은 'CDR'=0인 그룹의 평균이고, 파란 점선은 'CDR'=1인 그룹의 평균으로써, 두 점선의 평균인 가운데 점선을 기준으로 왼쪽의 관측치는 'CDR'을 0으로 오른쪽의 관측치는 1로 예측을 하게 된다.

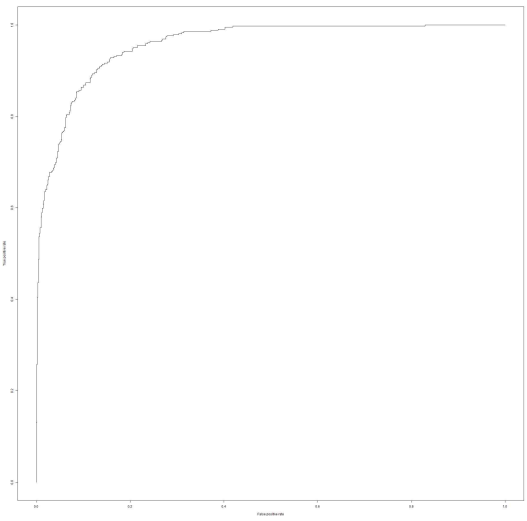

Figure18. 판별분석 모델 ROC곡선 예시

판별 분석을 k-fold를 통해 성능 평가를 한 결과 k=10 중 최대 AUC는 0.9570이였으며, 평균 AUC는 0.94365이 나왔다.

3.4.3 의사결정나무(CART)

```
106 # CART 의사결정나무 모델
107 library(rpart)
108 rpartmod <- rpart(CDR ~.-id , data= semi_train, method = "class")
109 # 가지치기(Pruning)
110 ptree<-prune(rpartmod, cp = rpartmod$cptable[which.min(rpartmod$cptable[,"xerror"]),"CP"])
111 summary(ptree)
112
113 png(filename ="ptree.png",
114     width=7000,
115     height=6000
116 )
117 plot(ptree)
118 dev.off()
```

Figure19. CART 의사결정나무 모델링 R 코드

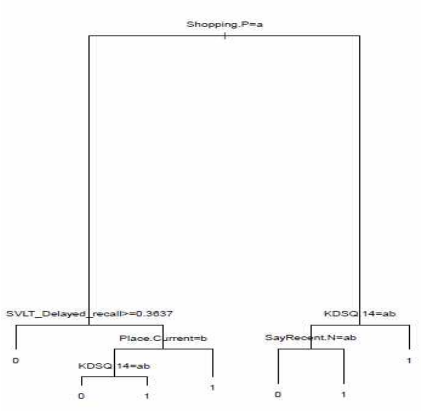

Figure20.CART 의사결정나무 결과 Plot

지니지수가 작아지도록 변수가 선택되며, 이에 따라 가장 유의한 변수는 Shopping.P, 그 다음으로 SVLT_Delayed_recall과 KDSQ.14가 선택된 것이다. 이러한 과정으로 이진트리를 형성해 나가며, 다시 각 노드의 에러율을 측정하여 가지치기를 실행하여, 나무를 잘라 최종 모델을 만든다.

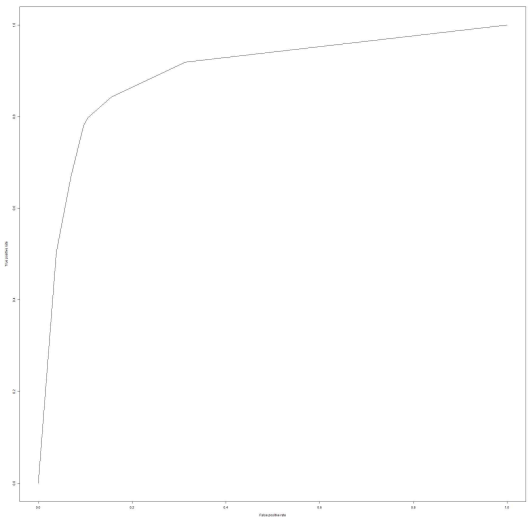

Figure21. CART 의사결정나무 모델 ROC곡선 예시

CART 의사결정나무 모델을 k-fold를 통해 성능 평가를 한 결과 k=10 중 최대 AUC는 0.8933785였으며, 평균 AUC는 0.87772이 나왔다.

3.4.4 의사결정나무(ctree)

```
# ctree 의사결정나무 모델
library(party)
ctree<-ctree(CDR~.-id,data=semi_train,controls=ctree_control(maxdepth = 7))
summary(ctree)

png(filename ="ctree.png",
    width=7000,
    height=6000
  )
plot(ctree)
dev.off()
```

Figure22. ctree 의사결정나무 모델링 R 코드

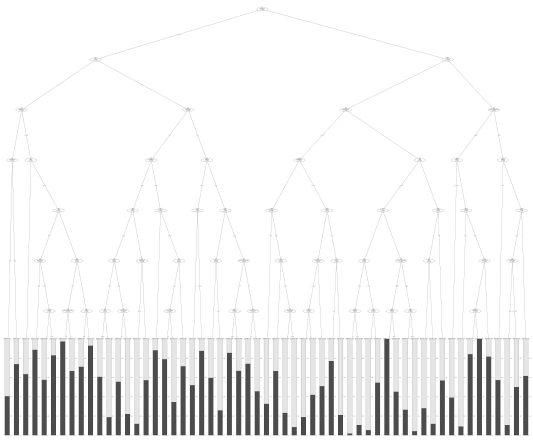

Figure23. ctree 의사결정나무 결과 Plot

Figure 23.은 ctree 의사결정나무의 결과 plot의 사진이다. 잘 보이지 않아 부분적으로 잘라 설명하겠다.

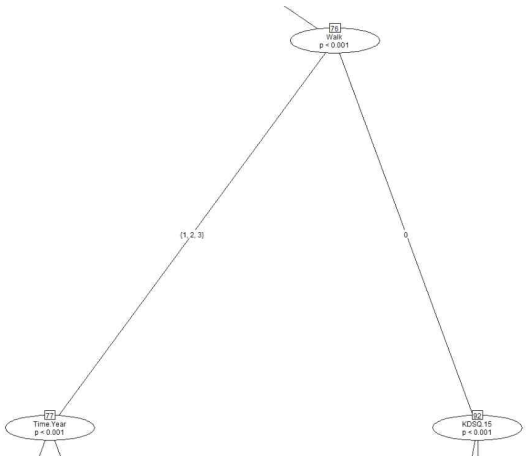

Figure24. ctree 의사결정나무 결과 Plot 부분 1

앞의 노드와 강한 연관성을 지닌 변수로 walk변수가 선택되었고, 각각 범주 수준이 1, 2, 3 그리고 0으로 각각에 대해서 permutation test 통계량이 가장 큰 변수인 Time.Year과 KDSQ.15의 변수로 가지가 분할되었다.

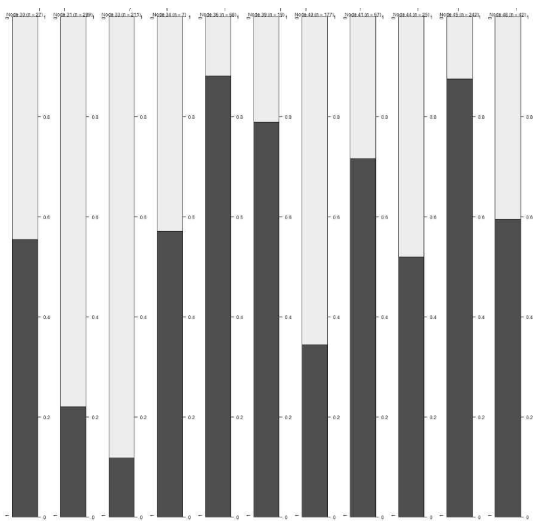

Figure25. ctree 의사결정나무 결과 Plot 부분 2

Figure 25.은 검은색으로 칠해진 부분만큼이 목표변수 CDR이 0일 때 차지하는 비율을 뜻한다. 즉, 검은색이 차지하는 비율이 0.8이면 CDR의 범주 0:1=80:20(%)가 되는 것이다.

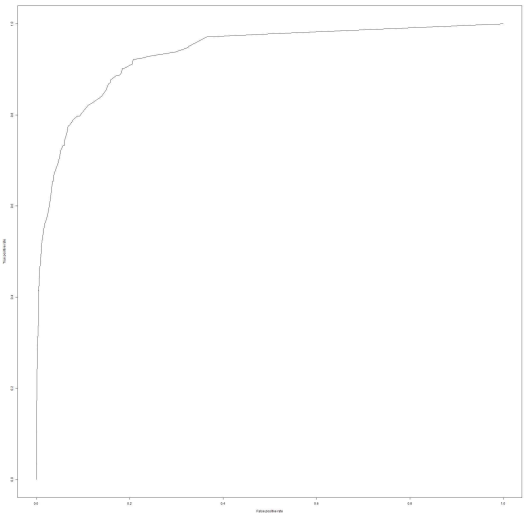

Figure26.ctree 의사결정나무 모델 ROC곡선 예시

ctree 의사결정나무 모델을 k-fold를 통해 성능 평가를 한 결과 k=10 중 최대 AUC는 0.9335486이었으며, 평균 AUC는 0.91807이 나왔다.

3.4.5 랜덤포레스트

```
107 # 랜덤포레스트 모델
108 library(randomForest)
109 rf.dementia <-randomForest(CDR~.-id , data=semi_train, ntree =100, mytr=5,importance=T)
```

Figure27. 랜덤포레스트 모델링 R 코드

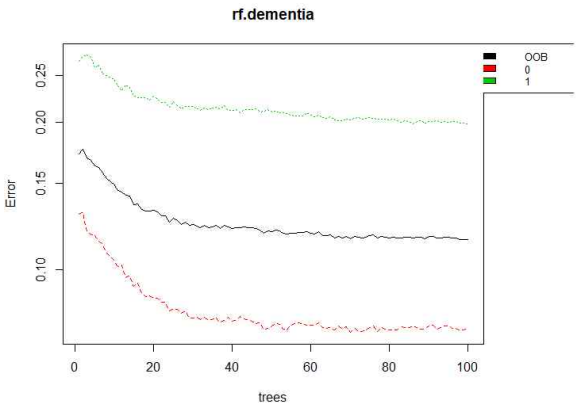

Figure28. 랜덤포레스트 결과 plot

Figure 28.은 랜덤포레스트의 모델의 결과물이다. 랜덤포레스트는 여러 개의 의사결정나무로 구현된다. 랜덤포레스트 모델을 출력하면 모델 훈련에 사용되지 않은 데이터를 사용한 에러 추정치가 'OOB(Out Of Bag) estimate of error rate' 항목으로 출력된다. dementia에 대한 모델에서는 OOB 에러가 11.83%이다.

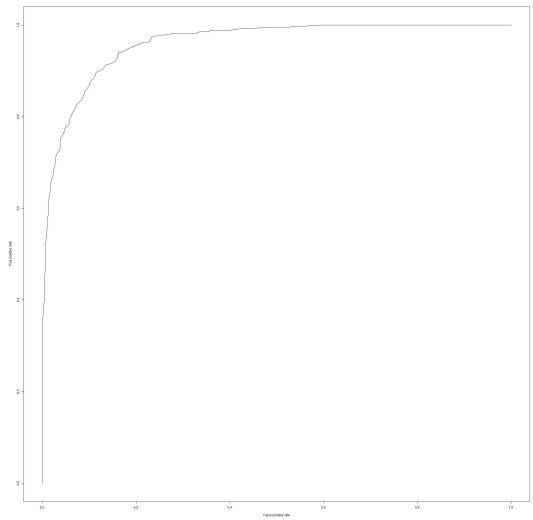

Figure29. 랜덤포레스트 모델 ROC곡선 예시

랜덤포레스트 모델을 k-fold를 통해 성능 평가를 한 결과 k=10 중 최대 AUC는 0.9604043이였으며, 평균 AUC는 0.95493이 나왔다.

3.5 모델링

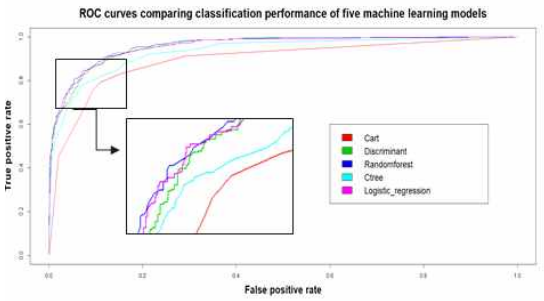

Figure30. 모델 ROC곡선

5개의 치매 진단 모델들의 성능 평가를 통해, 가장 성능이 높은 모델을 최종 모델로 선택하고자 한다. 성능 평가 지표는 AUC를 사용하였다. CART 알고리즘 기반의 의사결정나무 모델은 0.87772로 가장 낮았고, 랜덤 포레스트는 0.95011로 가장 높은 AUC를 보였다. 따라서 랜덤 포레스트 모델을 최종 치매 진단 모델로 선정하였다.

4. 결과

AUC가 가장 높은 랜덤포레스트 모델을 최종 치매 진단 모델로 선정한 후, Test set에 대해 검증을 실시한 결과 정분류율(Accuracy)은 약 90%를 보였으며, 치매로 진단하는 검정력(sensitivity)은 약 82%이다.

5. 결론및토의

인구수가 점점 증가함에 따라 치매 환자 또한 증가하고 있다. 치매를 진단

하기 위하여 다양한 검사가 실시되고 있는데, 현재 진행되는 검사를 통한 의사의 진단 과정에서 시간이 많이 소요되고 주관적이며 오류가 발생하기 쉽다. 이 연구는 치매 진단 예측 모델을 만듦으로써 이러한 문제를 해결하는 데에 의의가 있다. 목표변수 CDR을 제외한 기존의 변수 61개에서 45개로, 유의한 치매 진단 검사를 약 25% 감소할 수 있었다. 또한 모델링에 있어 과적합 되지 않은 좋은 성능의 모델을 선택하기 위해 K-fold cross validation을 이용하였다. 그 결과 0.9604043으로 가장 높은 성능을 가진 랜덤포레스트 모델을 치매 진단 예측 모델로 구성하였다.

랜덤포레스트가 가장 성능이 좋게 나온 이유를 생각해본 결과, 랜덤포레스트의 원리에 대해 다시 한 번 생각해 보게 되었다. 랜덤포레스트는 표본에 대해 여러 번 단순임의 복원 추출 후 무작위 변수 선택을 통한 최적의 분할로 트리가 형성이 된다. 다중의 여러 트리를 만들게 됨으로써 다양성이 극대화되고 예측력이 우수해져 가장 높은 성능을 보인 것이라고 생각된다.

최종 랜덤포레스트 모델의 검증 결과 민감도는 81.69%, 특이도는 93.17%로 정상을 치매로 진단하는 제1종 오류는 약 7%, 치매를 정상으로 진단하는 제2종 오류는 약 18%가 나왔다. 의료 분야에서는 제1종 오류 보다는 제2종 오류가 더 치명적인 것으로 보고 있으므로 제2종 오류를 낮춰 치매에 대한 검정력을 높이는 것이 바람직하다. 하지만 제1종 오류와 제2종 오류가 trade off 관계에 있어 높은 검정력으로 인해 제1종 오류가 증가할 수 있다. 그런데 제1종 오류의 증가는 의료 산업에 이익을 도모하게 되지만, 사회적으로는 허위 진단으로 인한 손실이 증가하게 된다. 즉, 이에 따라 치매 진단을 위한 예측 모델링은 의료산업의 이해관계에 따라 검정력을 고의로 조절하는 것이 아닌, 사회적 이익을 고려하여 이뤄져야 한다.

또한 현 연구에는 설문조사에 대한 검사가 대부분이지만, 이후 뇌 영상, 개인 유전체 등 환자의 정밀한 정보를 추가하여 보다 더 높은 정확도의 치매 진단이 이루어질 수 있을 것이다.

6. 역할

이름 (학번)	역할분담
임동희 (201220236)	데이터 전처리(mice, Histogram) Discriminant Analysis(R) Chi-square Test(R) 최종 보고서 작성
손소희 (201320217)	데이터 전처리(pearson Residual) Decision Tress - ctree(R) Variable Selection(SAS) 최종 보고서 작성
김수정 (201323081)	데이터 전처리(crossTable) Decision Tress - CART(R)

	Random Forest(R) 최종 보고서 작성
안채림 (201420176)	데이터 전처리(mice, SAS) Logistic Regression(stepwise)(R) 발표자료 작성 최종 보고서 작성

7. 목표연관성

임동희: 데이터 분석 분야에서 내 능력을 펼치기를 희망하는 데 산업공학종합설계 프로젝트를 통해서 전반적인 데이터 분석 과정을 심도 있게 경험 해볼 수 있어서 뜻 깊었다. 특히 의료라는 도메인의 데이터를 접할 수 있었던 것은 소중한 기회였다.

치매 진단 예측의 데이터를 분석하는 과정에서 결측률이 높고 예측 변수들이 대부분 범주형으로 주어져, 결측치를 대체하고 이상치를 제거하는 방향을 정하는 데 어려움이 많았다. 새롭게 교차표를 분석하는 방법을 익히고, 카이스퀘어 테스트를 이용해 데이터 전처리를 시도한 경험들은 앞으로 다시 범주형 데이터를 접하였을 때 많은 도움이 될 것이라 생각한다.

이번 프로젝트를 통해 분류를 목적으로 하는 분석의 기법을 온전히 경험 한 점, 전체적인 데이터 마이닝 프로세스를 익히고 적용 시킨 점, 프로젝트 진행 간에 팀원과의 의사소통을 통해 조율을 함으로써 문제를 해결하고 분석의 방향을 결정 한 것은 큰 소득으로 남아 앞으로 이 분야에서 일을 하는 데 큰 도움이 될 것이다.

또한 예측 변수에 대한 히스토그램이나 모델의 예측력 비교를 통해 유의미한 파생 변수를 만들고 결정하는 것에서 나아가서 다양한 통계적 근거를 바탕으로 효과적인 파생변수를 만들고 통계적 모순이 없는지 검정 할 수 있는 능력을 키워야겠다는 구체적인 목표를 이번 프로젝트를 마치면서 정할 수 있었다.

손소희: 산업공학종합설계 프로젝트 주제의 데이터에 대한 파악부터, 전처리, 탐색, 분석, 결론에 이르기까지 전 과정에서 산업공학적인 사고를 가지고 접근하여 모델링하는 능력을 발휘해볼 수 있었다. 처음에는 치매진단이라는 주제가 생소하게 느껴졌고, 62개의 변수를 이해하고 연속형 변수에 대해서는 50%가 넘는 결측치를 처리하는 데 어려움이 있었다. 하지만 계속적인 아이디어 회의를 통해 다양한 방법을 시도하고, 데이터의 본질을 변화시키지 않는 방향으로 전처리를 하고자 노력했다. 또한 R프로그래밍에 있어 익숙지 않아 초반에 어려움을 많이 느꼈지만, 지금은 편안하게 느껴질 정도로 R코딩 능력을 습득할 수 있었다. 어려움에 봉착하였을 때 의견을 주고받으면서 문제에 접근하고 해결하는 능력을 기를 수 있었고 그 과정에서의 협력이 매우 의미 있었다. 본 프로젝트 경험은 향후 사회생활에 있어 문제를 해결하고, 의사결정을 하는 데 있어 큰 밑거름이 될 것이다.

김수정: 본 프로젝트를 진행하며 그동안 배운 전공지식을 많이 활용할 수 있었고, 데이터 분석 과정에 있어서 다양한 각도로 생각할 수 있는 힘을 기를 수 있었다. 치매 예측이라는 주제가 정해진 후, 변수 명부터 난항을 겪었다. 그러나 차근차근 하나씩 찾아가며 의미를 파악하였고, 서로의 의견을 주고받으며 위기를 극복할 수 있었다. 이와 마찬가지로 결측치 처리, 이상치 처리, 유의변수 도출, 결론에 이르기까지 많은 시행착오가 있었지만 끝까지 해결하려는 의지와 열띤 토론을 통해 기간 내에 완수할 수 있었다. 결과적으로 본 프로젝트를 통해 익숙하지 않았던 R, SAS 프로그램을 잘 다룰 수 있게 되었고, 문제해결과정에 있어서 협업의 중요성을 깨달을 수 있었다. 향후 Data scientist로서의 길로 한걸음 더 가까워 질 수 있었던 뜻깊고 소중한 프로젝트였다.

안채림: 산업공학종합설계를 진행하며 그동안 배운 교과목의 개념을 가능한 많이 사용하려고 노력하였다. 교과목 수업에서 배운 통계, R 프로그래밍, 데이터 분석 등을 큰 데이터로 활용해 볼 수 있었다. 기존 데이터에 결측률이 컸기 때문에 결측치와 이상치를 처리하는 데에 많은 시간을 할애하였다. 의사결정을 단순화하기보다 산업공학적인 생각을 가지고 접근하여 해결하려 하였다. 다양한 분석 방법을 공부하고 데이터 분석을 하여 결과를 도출하였다. 치매 진단을 예측하는 방법론과 가능성을 제시할 수 있었다. 향후 본 프로젝트를 더욱 발전시켜 더 많은 변수들을 이용하여 더 고도화된 모델을 만들고 싶다.

※ 참고 문헌

1. 강미애·백용매, “주관적 기억 장애와 경도 인지 장애의 신경인지기능 특성 비교”, Korean Geriatric Society, 2014
2. 강연옥·진주희·나덕렬, “숫자 외우기 검사(Digit Span Test)의 노인 규준 연구”, 한국심리학회지, 제21권 제4호. 2002
3. Bang SJ, Son SJ, Roh HW, Lee SY, Lee KW, Hong CH, Shin HJ, *Quad-phased data mining modeling for dementia diagnosis.* 2017
4. 신준현, “치매의 진단: 신경심리검사”, 가정의학회지 Vol3 No,4, 2010
5. 최성혜, “치매의 임상적 진단”, The Journal of Korean Diabetes, 2012
6. 고길곤· 탁현우, “설문자료의 결측치 처리방법에 관한 연구: 다중대체법과 재조사법을 중심으로”, 행정논총 제54권 제4호, 2016
7. chosun, 고령사회, http://news.chosun.com/site/data/html_dir/2017/09/04/2017090400203.html, 2017.09.04.
8. 이데일리, 치매 노인 환자 증가율, http://www.edaily.co.kr/news/news_detail.asp?newsId=01367766616061760&mediaCodeNo=257&OutLnkChk=Y, 2017.09.19.
9. chosun, 고령화 속도, http://news.chosun.com/site/data/html_dir/2017/02/22/2017022200221.html, 2017.02.22.
10. 네이버 블로그, 결측치 처리, http://blog.naver.com/tjdudwo93/220976082118, 2017.04.05.
11. BIRC, Decision Tree,

http://www.birc.co.kr/2017/01/11/%EC%9D%98%EC%82%AC%EA%B2%B0%EC%A0%95%EB%82%98%EB%AC%B4decision-tree/, 2017.01.11.
12. Tistory, 추론통계분석-교차분석 및 카이제곱 분석, http://dbrang.tistory.com/1067, 2016.12.11.
13. 네이버 블로그, 분류 알고리즘, http://blog.naver.com/PostView.nhn?blogId=soccerball1&logNo=150187128099&parentCategoryNo=&categoryNo=22&viewDate=&isShowPopularPosts=true&from=search,, 2014,03,18
14. 네이버 블로그, R tree모형(ctree), https://blog.naver.com/liberty264/221015964454, 2017.06.27