**‘”유한양행”의 새로운 판매전략 수립을 위한 식이보충제 섭취 여부 예측 모델 개발 기획 보고서’**

**(연령대와 섭취여부 데이터를 중심으로)**

**1. Introduction**

*1.1 Reseach Background and Motivation (과제 수행 배경 및 동기)*

2019년 30대 제약회사 매출 대부분이 5% 넘게 증가하였지만 유한양행 사의 영업이익은 8% 이상 감소하였다. 꾸준히 매출에 기여했던 비리어드 간염 치료제의 특허가 만료되면서 약값이 인하되었고, 유한양행 사의 원료의약품 매출이 부진하게 되었다. 그 결과 유한양행 사는 신제품 개발에 대한 필요성이 대두되었다. 연구 및 개발(R&D) 비용을 740억원에서 969억원으로 30% 증가시킴으로써 영업이익이 75%나 줄게 되었다. (영업비용= 판매관리비용 + 감가상각비용 + 연구개발비용) 이러한 영업이익의 감소 만회를 위해, 기존 제품의 고객층 확대 및 신제품의 판매를 위해 효과적인 전략을 수립할 수 있는 데이터와 모델이 필요하게 되었다. 이 때, 모델과 모델의 결과는 어떤 조건 때문에, 또는 어떤 고객군을 대상으로 제품을 판매해야 되는지에 대한 근거를 제공해야 한다. 확보한 데이터는 국민영양조사에 기반한 것이므로 데이터에 적합한 식이보충제 제품군을 target으로 삼고 Data mining을 시행할 것이다.

*1.2 Nature and scope of the problem (문제 정의 및 도출)*

앞서 보았듯 현재 유한양행 사의 문제는 원료의약품 매출 부진으로 인한 영업이익 감소이다. 영업이익의 감소 만회를 위해, 앞서 보았던 문제로부터 도출된 Data Science Problem은 기업의 입장에서 국민건강영양조사 데이터 중 어떤 attribute를 선정해 판매전략을 위해 활용해야 하는 지 알 수 없다는 점이다.

*1.3 Previous works and their problems (기존 연구 및 기술의 문제점)*

1) 전문가 시스템

전문가시스템은 인공지능의 한 분야로서 특정 분야에서 전문가의 축적된 지식과 경험을 시스템화하여, 필요할 때 사용하도록 하는 소프트웨어이다. 이는 1960년대 화합물의 구조를 추정하기 위한 DENDRAL 시스템이 시초가 되어, 이후 의료진단시스템인 MYCIN, 기계고장 진단, 손해배상 판정 등 산업계의 전 분야에서 광범위하게 응용되었다. 그러나 이러한 방식은 기업환경의 급속한 변화에 적용하는 데에 많은 비용이 들었고, 시스템을 통해 해결하고자 하는 문제의 영역이 점차 확대되고 복잡해짐에 따라 해당 분야에 통달한 전문가를 찾기가 어려워지면서 지식 획득에 장기간에 걸쳐 많은 비용을 투자해야 하는 등, 여러 문제를 가지고 있었다.

2) 일반 질의(query)

일반 질의(query)는 데이터베이스의 데이터를 분석하는 데 도움이 되는 도구이다. 쿼리 작성, 쿼리 편집, 검색, 찾기, 보고 및 요약 기능을 제공한다. 쿼리는 데이터베이스에 정보를 요청하여 사실로서의 질문에 대한 사실로서의 답변을 제공받는다. 한편 이러한 쿼리는 실제로 사용하려면 사용자가 원하고자 하는 정보가 무엇인지를 정확하게 알아야 하며, 새로운 목적을 찾아야 하고, 무엇을 찾아야 할 지 모호한 상태일 때에는 사용하기 어렵다는 단점이 있다.

3) 사회과학적 연구

사회과학적 연구는 분석가의 인사이트가 굉장히 중요하다. 특히 국민건강영양조사의 경우 사용하는 속성들이 굉장히 많기 때문에 데이터 마이닝 기법을 사용하지 않으면 분석가의 주관으로 중요할 것 같은 속성들을 추려 내야 한다. 이렇게 하면 만약 분석가가 추려내지 않은 속성들 중에서 중요한 변수가 있을 수도 있기 때문에 타당성이 떨어진다.

1.5 Purposed Method (제안하는 방법의 개요)

기본적으로 우리가 사용하는 국민건강영양조사의 Data set은 그 record 수가 충분하나 결측치가 존재하고 불필요한 attribute가 존재한다. 이에 따라 측정된 변수들의 선형 조합(Linear Combination)에 의해 대표적인 주성분을 만들어 차원을 줄이는 PCA 기법, 파레토 원리 기반 결측치 제거 그리고 휴리스틱에 기반해 속성을 제거하는 등의 차원 축소 기법을 통해 Dimension Reduction을 진행하였다. 또한 그룹화, 정규화, 이상치 탐지 및 형변환을 통해 Feature Engineering을 수행하였다.

이렇게 전처리 된 데이터를 기반으로, 의사 결정 나무(Decision Tree)와 랜덤 포레스트(Random Forest) 기법을 통해 모델링을 수행하였으며, 이후 Confusion Matrix를 통해 해당 모델의 Accuracy를 평가하였다.

*1.6 Principal results (논문 주요 결과의 개요)*

실세계의 모델링에서 가장 적합한 입력만을 선택하는 것은 시스템 성능에 많은 영향을 미친다. 일반적으로 입력변수의 효과적인 선택은 시스템 차원의 감소나 특징추출 등 다양한 용도로 이용된다. 그러나 많은 입력변수들 중에서 모델에 얼마나 많은 또는 어느 입력들이 필요한지 알 수 없으며, 이는 입력차원이 증가할수록 더욱 더 심각하다. 불필요한 입력들은 학습을 복잡하게 하고 과학습 등에 따른 학습성능의 저하도 가져올 수 있다. 입력변수의 잘못된 선택에 여러 가지 문제들이 발생될 수 있다. 먼저, 입력차원의 증가에 따른 계산시간과 메모리의 증가, 다음으로 요구되지 않는 입력들에 의한 학습의 어려움, 추가적인 요구되지 않는 입력에 의한 비수렴과 모델의 정확성의 저하, 그리고 복잡한 모델에 따른 해석의 어려움 등의 제약이 있다.

**2. Method**

*2.1 전처리*

2.1.1 Data Quality

기본적으로 국민건강영상조사(이하 국건영)에서 원시자료를 기본 Dataset으로 한다. Data object의 양은 어느 정도 충분하다. 2018년 자료는 7992개의 Object와 736개의 attribute를 가지고 있고 2017년 Dataset은 8127개의 Object와 834개의 attribue를 가지고 있다. 모델 디자인에 있어서 더 좋은 Dataset을 찾기 위해 2018년 단독 Dataset과 2018년과 2017년 Dataset을 합쳐서 새로운 Dataset를 만들었다. 2018년과 2017년 Dataset의 attribute가 차이가 있었다. 2018년을 기준으로 하고 2017년에만 조사한 Attribute의 경우 모두 제거했다. 최대한 2018년 Dataset과 비슷하게 하면서 Object의 양을 늘리기 위해서이다. 2017년의 LF\_safe변수가 소문자로 저장돼서 2018년의 LF\_SAFE와 합쳐지지 못하는 문제가 발생했다. CSV 파일로 Dataset을 추출하여 엑셀 내에서 이를 결합해줬다. 국건영 자료는 설문조사를 기반으로 생성됐다. 그렇기 때문에 결측치가 존재한다.

2.1.2 Dimension reduction

우선 비해당, 모름, 무응답의 경우 결측치와 다름이 없다고 판단하여 이를 모두 결측치 처리를 해줬다. Data의 양이 충분하다고 판단해서 결측치를 모두 제거하는 것을 최우선으로 생각했다.

2.1.2-1 휴리스틱 기반 속성 제거

주관적으로 Attribute를 제거하는 것은 위험하다. 특히 건강 관련 조사의 경우 의학적인 설문이 많기 때문에 속성을 제거하는 것을 지양했다. 속성 중 가중치에 대한 속성은 모두 제거했다. 가중치는 같은 기에 조사한 자료에 적용하는 것이 아니라 서로 다른 조사 기수 데이터를 통합할 때 사용하는 변수이다. 이 프로젝트에서는 제7기 데이터만을 사용하기 때문에 변수를 삭제해줘도 지장이 없다.

청소년의 경우 경제적 능력이 없어 구매대상으로 적합하지 않다고 판단하여 age를 기준으로 만 18세 미만의 Object를 제거해줬다. 이와 관련하여 소아, 청소년 관련된 질문 항목들을 제거했다. 또한 주관식 설문 문항의 경우 Decision Tree를 사용하기에 적합하지 않기 때문에 제거했다.

2.1.2-2 파레토 원리 기반 결측치 제거

결측치를 처리하는 최적의 해법은 상황과 분석 목적에 따라 달라지기에, 이를 어떻게 처리할 지 결정하는 것은 쉽지 않다. 그럼에도 불구하고 결측치로 인한 정보의 손실이 분석 결과에 미치는 영향을 고려한다면 결측치를 적절하게 처리하는 것은 중요하다. 이에 우리는 데이터에 있는 결측치를 ‘파레토 원리’에 기반하여 제거하고자 하였다. 파레토 원리(또는 희소 인자의 원리)는 전체 결과의 80%는 전체 원인의 20%에서 비롯된다는 것이다. 데이터의 무질서도(Entropy) 또한 20%의 주요 변수들로부터 기인할 것이다. 파레토 원리를 적용한 결측치의 제거 방식은 아래와 같다.

[1] 휴리스틱 기반 속성 제거가 완료된 데이터 셋에서, 행과 열 개수의 각각 20%를 데이터 셋의 임계 개수로 설정한다.

[2] 열을 기준으로 NA값이 많은 20%를 먼저 제거하여, 80%의 유효 열을 남긴다.

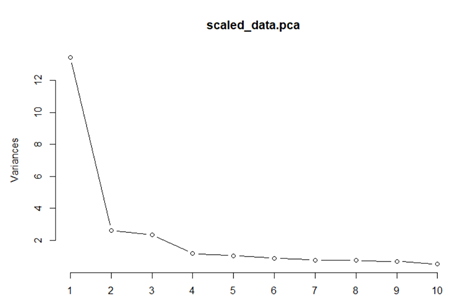
[3] 다시 행을 기준으로 NA값이 많은 20%를 제거하여 80%의 유효 행을 남긴다. 이 때 sort()를 통해 NA 값이 많은 순서대로 정렬하고, 반복문 while()을 통해 모든 행에 대해 해당 과정을 수행하게 한다.

[4] 행이나 열의 개수가 임계 개수보다 작아지기 직전까지 [2]와 [3] 과정을 반복한다.

[5] 행보다는 열이 더 가치 있다는 판단 아래, 남아있는 NA값을 행 제거를 통해 처리한다.

2.1.2-3 PCA

PCA 분석방법은 측정된 변수들의 선형 조합(Linear Combination)에 의해 대표적인 주성분을 만들어 차원을 줄이는 방법이다. 현재 갖고 있는 데이터 중 식품섭취조사 자료인 ‘개인별 24시간 회상자료를 통한 영양소 섭취량에는 식품섭취량, 에너지, 수분 등을 포함하여 총 27가지의 영양소에 관한 연속형 변수들이 있다. 이 27가지의 변수들을 PCA분석을 통해 주요 변수들을 뽑아주었다. 총 27가지의 PC원소들 중 PC4번 변수까지 포함시켰더니 누적분산이 0.7을 넘어 유효한 PC그룹군을 4개까지로 지정하였다. 따라서 27개의 변수를 제거하고 새로운 4개의 PC원소를 추가함으로써 기존 변수들이 갖고 있던 최적의 성질을 유지한 채 차원을 줄여주었다. 이렇게 뽑힌 4가지의 PC원소를 살펴보니 가장 영향력이 큰 PC1변수에 큰 영향을 끼치는 기존 변수들은 인, 에너지, 카로틴으로 인체의 열량 및 활력에 영향을 미치는 요소들로 되어 있었다. 따라서, PC1 변수는 ‘energy’로 재정의 하였다. 두번째 PC2변수는 불포화지방, 지방, 포화지방이 영향력의 비중이 가장 크기 때문에 ‘fat-series’로 재정의하였다. 세번째 PC3변수는 비타민A, 레티놀, 카로틴이 영향을 많이 끼쳤다. 레티놀과 카로틴은 비타민 A의 한 종류이므로 변수를 ‘Vitamin\_A’로 재정의하였다. 마지막 PC4변수는 3계지방산, 다가불포화지방산, 6계지방산이 큰 영향을 끼쳐서 ‘fatty-series’로 재정의하였다. 이처럼 27개의 연속형 변수들을 PCA분석을 통해 기존 분산을 가장 잘 반영하는 축으로 사영하여 주요변수 4개를 뽑아주어 차원을 축소시켜주었다.



2.1.3 Feature engineering

2.1.3-0 EDA

어떤 Feature가 핵심적인 역할을 할 수 있을지 직관에 따라 선택하고 Target Attribue와의 연관성을 알아보기 위해 시각화 해보았다.

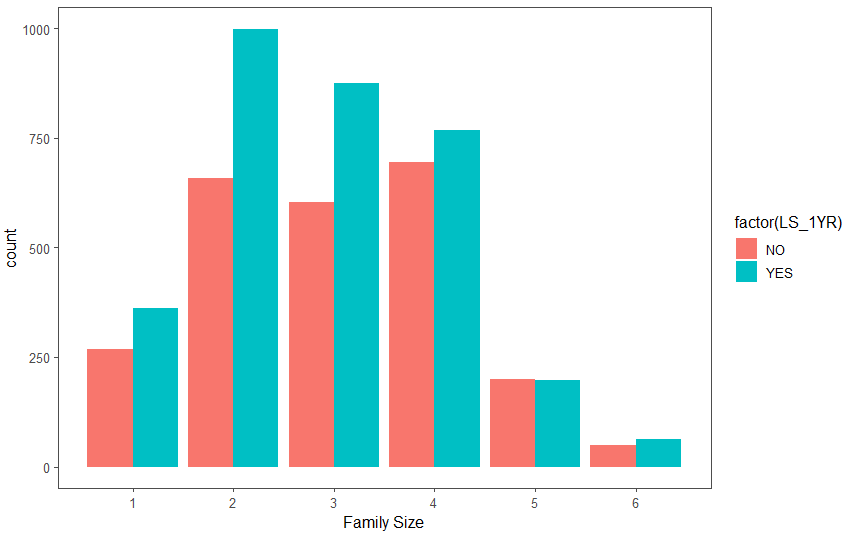
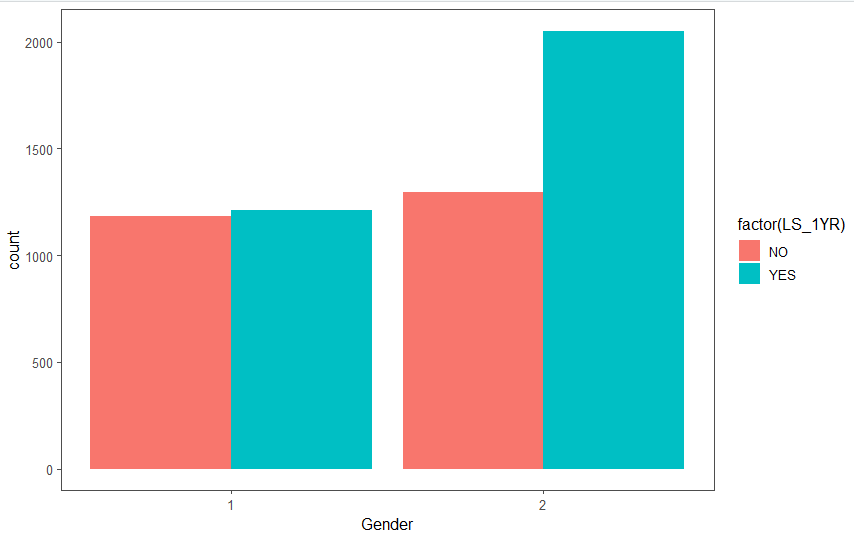
 

그림 1 : 가족크기와 영양제 복용여부 그림 2 : 성별과 영양제 복용여부

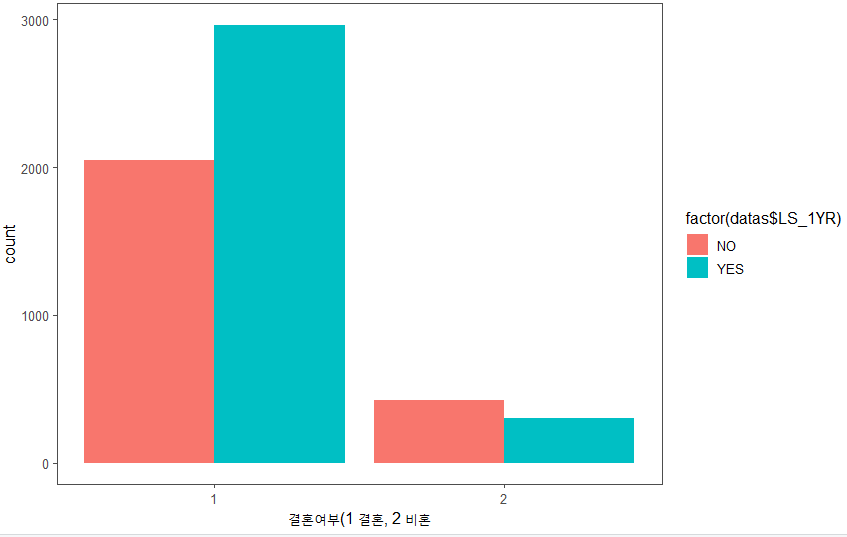
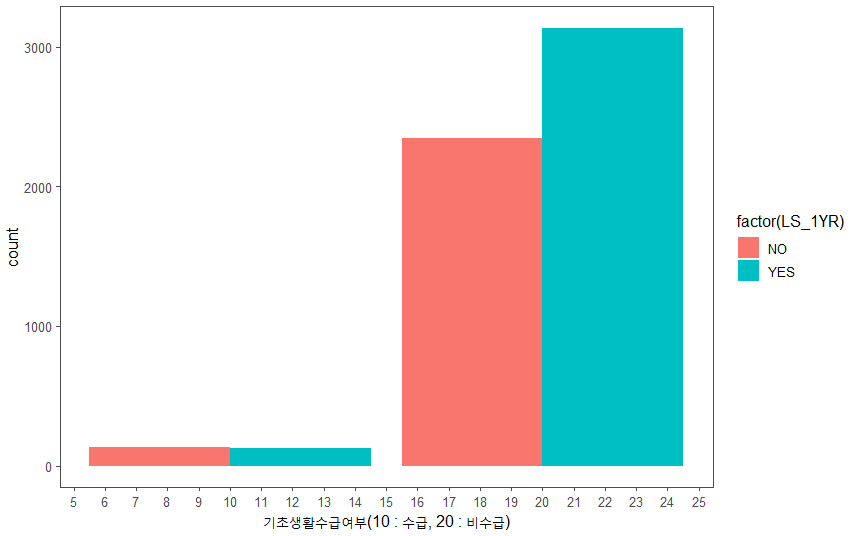
 

그림 3 : 결혼여부와 영양제 복용여부 그림 4 : 기초생활수급과 영양제 복용여부

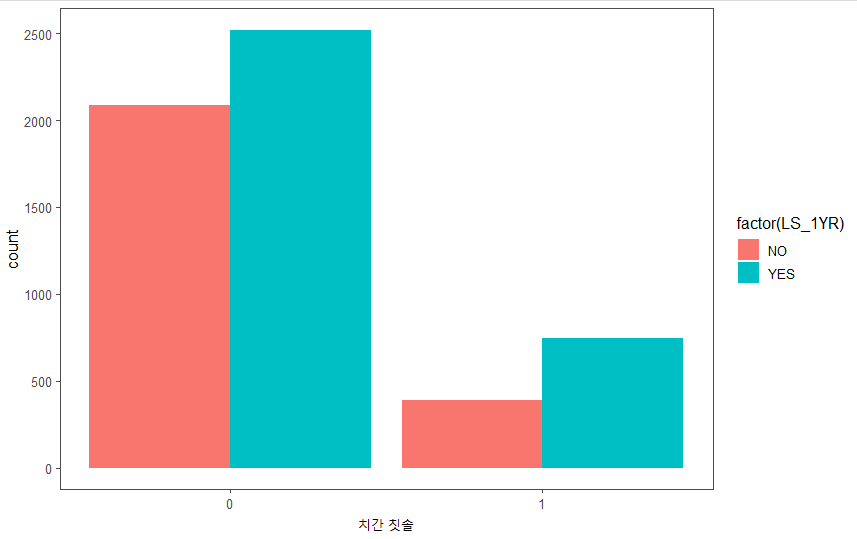
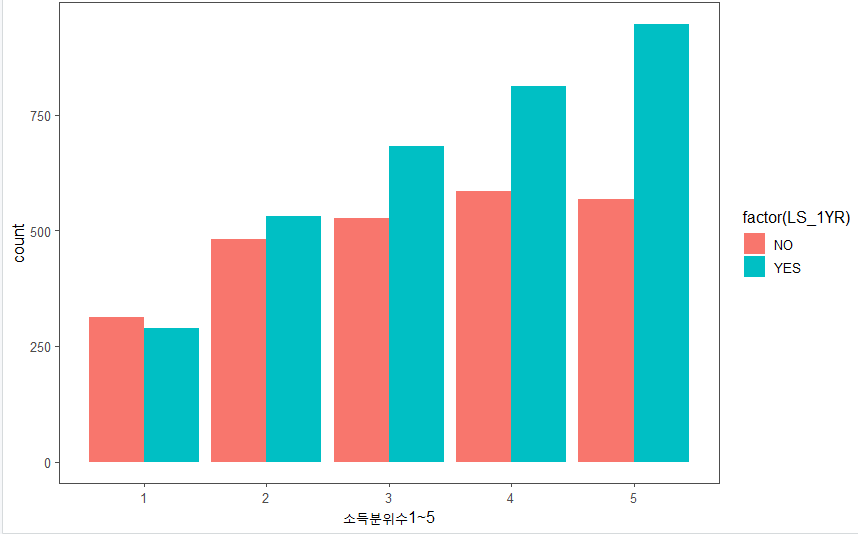
 

그림 5: 치간칫솔과 영양제 복용여부 그림 6: 소득분위와 영양제 복용여부

이렇게 Target Attribute와 직관적으로 관련이 있어 보이는 Attribute들은 여섯 개의 그림은 Modeling 과정에서 실제로 연관이 있는지 검증될 것이다.

2.1.3-1 나이 그룹화

고객군 대상이 될 수 없는 20세 미만의 미성년을 제외한다 하더라도 20세 이상부터 부터 가장 나이가 많은 80살의 사람까지 연속형의 변수는 나름 그 활용성이 높다고 생각하여 나이의 특성을 좀 더 활용하기 위해 그룹을 지어주었다. 20세부터 30세까지의 226명의 사람들은 ‘young’그룹으로 묶어주었고, 그 위로부터 65세 아래까지의 총 2218사람들은 middle로, 그 위의 521명의 사람들은 ‘old’그룹으로 묶어주었다. 따라서 연속형의 나이 변수와 더불어 명목형의 age\_group변수를 추가하여 데이터의 특성을 보강했다.

2.1.3-2 지역 그룹화

나이의 그룹화와 더불어, 지역 또한 17개의 변수들로 나뉘어져 있다. 따라서 그 특성을 좀 더 세분화하고 지역의 특성을 데이터 내에서 좀 더 활용하기 위해, 17개의 지역 중 경기/서울/인천/충북/충남/세종/대전 지역을 합쳐서 윗지방을 의미하는 1로, 그 아래의 강원/전북/전남/광주/제주/경북/경남/대구/부산/울산 지역을 아랫지방을 의미하는 0으로 합쳐주었다. 이 이산화는 명목형 변수들을 큰 틀로 묶어주어 데이터에 대한 특정화를 보장하며 soft한 스플릿을 가능하게 해준다.

2.1.3-3 Normalization

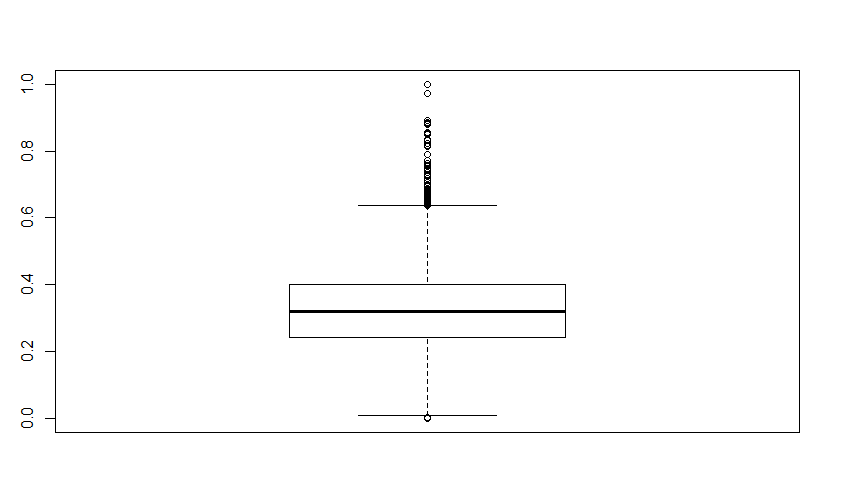
국민건강영양조사 원시 자료에서 애트리뷰트 타입이 등간 속성이나 비율 속성인 경우 변화량의 통일을 위해 데이터를 정규화를 해줄 필요가 있다. 예를 들어 나이와 소득 같은 경우 똑같이 10이 증가했다고 같은 의미를 가지지 않는다. 두 개의 데이터 객체가 있다고 가정해보자. 두 데이터 객체의 나이차가 20살이 나고 소득 차이가 2천만원이 나는데 분석을 할 대 이 변화량을 절대적인 크기로 해석하면 오류를 범할 가능성이 높다. 그 이유는 두 데이터 타입이 가질 수 있는 값의 범위가 다르기 때문인데, 값의 범위를 통일시켜 주는 게 바로 정규화라고 할 수 있다. 정규화의 종류는 MinMax Normalization, Robust Normalization, Standardization 등 굉장히 많지만 우리는 그 중에서 수업시간에서 다룬 MinMax Normalization을 채택했다. MinMax Normalization은 자신의 값과 최소값의 차를 최대값과 최소값의 차로 나눠준 값을 의미한다. 이렇게 하면 정규화를 거친 데이터에 한해서 범위가 0~1로 고정되기 때문에 속성이 다른 데이터끼리 변화량이 같은 의미를 가지게 된다.

2.1.3-4 이상치 탐지

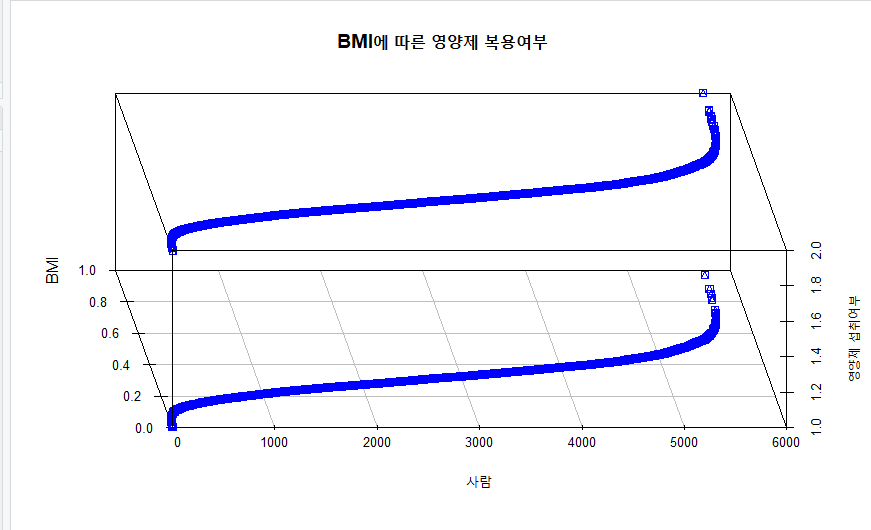
국민건강영양조사 데이터를 분석해서 다른 데이터와 비교했을 때 많이 벗어나 있는 데이터, outlier를 찾아서 제거해내야 한다. 앞에서 전처리한 데이터를 보았을 때 대부분의 데이터는 범주형 데이터이다. 전처리를 통해 범주형 데이터에서 NULL 값이나 모름, 무응답을 전처리 했으므로 연속형 데이터에 이상치가 있을 가능성이 크다. 따라서 대표적인 연속형 어트리뷰트인 1) 월평균 가구 총소득, 2) 체질량 지수 이상치를 제거한다.

1) 월평균 가구 총소득: 먼저 월평균 가구 총소득에 대한 이상치를 탐색하고 시각화해야 한다. 그러나 애초에 원시 데이터에서 소득이 월 평균 1500만원 이상인 사람들은 1500만원으로 바꿔줌으로써 이미 이상치 처리가 되어있었다.

2) 체질량지수: 데이터 표본에 포함된 사람 중에서 정말로 키가 극단적으로 크거나 작고 몸무게가 극단적으로 크거나 작은 사람들이 있을 것이다. 체질량지수에서 이상치를 찾아내서 이런 사람들에 대한 데이터를 한꺼번에 제거할 수 있을 것이다. 메소드는 boxplot을 이용한다.



위 그림을 보면 상자의 크기는 제3사분위수에서 제1사분위수를 나타내게 된다. 이 길이를 L이라고 하자. 상자 위아래의 선분은 최소값, 최대값을 표시하며 1.5L을 넘어가는 부분을 이상치로 표시하게 된다. 이상치가 위 아래로 존재하므로 삭제해야 한다

이상치의 대략적인 양을 알아내기 위해서 3D scatter를 통해 3차원으로 시각화를 해보면 체질량 지수가 대략 0(15)에 가까운 사람과 체질량 지수가 1.0(41)에 가까운 사람이 존재한다. 위로 41 \* 0.8 = 32.8 이상의 체질량 지수를 가진 사람들을 제거해준다. 마찬가지로 아래로는 체질량 지수가 15에 가까운 순으로 10개 정도를 제거해준다.

2.1.3-5 형변환

설문 내용을 데이터 수집가가 스프레드시트에 입력할 때 데이터 타입은 캐릭터형 또는 숫자형 밖에 없다. 하지만 설문 문항은 대부분이 범주형 데이터이다. 원시 데이터에서 확인해봤을 때 문자로 입력된 것들만 범주형 데이터로 분류되었고 나머지는 숫자형 데이터였다. 우리가 쓰는 모델은 분류나무 모델이기 때문에 범주형 데이터로 변환해줘야 할 것은 범주형 데이터로 변환해줘야 한다. 범주형 데이터로 변환하지 않았을 경우에 의사결정나무에서 분류를 할 때 split condition이 어떤 숫자 이상/이하로 설정된다. 예를 들어 Factor level이 3(1, 2, 3)인 경우 split condition 기준이 2.78이 될 수 있다. 이렇게 해도 모델은 잘 돌아가지만 올바르게 분석되었다고 보기는 어렵다. 원시 데이터에서 대부분의 형변환은 int->factor였고 형변환을 할지 말지는 우리의 주관에 따랐다.

앞으로 기술하는 기법들을 적용한 후 is.na() 함수를 이용하여 결측치가 500개 미만이 될 때까지 반복했다. 더 이상 열을 지우지 않고 NA이 하나라도 있는 행을 삭제해 Dataset을 완성했다.

*2.2 Exploratory data analysis*

2.2-1 Cluster Analysis

모델링에 들어가기 앞서 데이터 자체, 즉 고객들이 갖고 있는 자연스러운 특징들(군집, 연관성 등)을 알아보기 위해 2가지 비지도 학습적 데이터 탐색을 시도하였다.

첫 번째로는 군집 분석을 시도하였다. 군집분석을 하는 이유는 데이터마이닝 강의의 내용을 요약해보면 동일한 성격을 가진 여러 개의 그룹으로 데이터를 분류하기 위해 사용한다. 대상 개체를 유사하거나 서로 관련 있는 항목끼리 묶어서 몇 개의 집단으로 그룹화하거나, 각 집단의 성격을 파악해서 전체의 구조에 대한 이해를 돕는다. 군집분석은 종속변수에 대한 독립변수의 영향을 분석한다. 사전에 정의된 특수한 목적은 없다. 데이터 자체에 의존해서 데이터 구조와 자료를 탐색하고 요약하는 기법이다.

1. 국민건간영양조사 데이터와 같이 방대한 데이터의 경우에는 전체에 대한 의미 있는 정보를 얻어낼 수 있다. 전체를 유사한 성질을 지니는 군집으로 구분한다면 군집에 대한 특성을 분석하고 이를 통해 전체 데이터에 대한 직관을 얻고 활용할 수 있을 것이다. 이를 통하여 고객들을 자연스럽게 군집화 하고, 모델링과 평가가 끝나고 실전 배치 단계에서, 해당 군집이 갖고 있는 특성들을 활용하여 더 다양한 고객화 전략 및 마케팅이 가능해진다.
2. 우리가 시도할 군집분석은 수업 시간에 배운 Hierarchical Clustering, K-means clustring이다.  
   계층적 클러스터링은 병합적 방법과 분할적 방법을 바탕으로 군집을 형성시킨다. 계층적 클러스터링은 병합적 방법과 분할적 방법을 바탕으로 군집을 형성시킨다. 한 관찰단위는 한 군집에 속하면 다른 군집에는 다시 속할 수 없다. 군집은 덴드로그램(Dendrogram)이라는 도표를 이용해 군집들 간의 관계를 파악한다.
3. k-means 클러스터링은 사전에 결정된 군집 수 K에 기초하여 전체 데이터를 비슷한 k개의 군집으로 구분하는 방법이다. 상호배타적인 k개의 군집을 형성함으로써 군집들을 형성한다.

K-means Clustering 방법에 있어, 최적의 Centroid를 찾기 위해 일반적으로 사용되는 방법론을 적용하였다. 즉, 초기 Centroid를 random하게 잡은 후, Euclidean distance를 이용해 Closeness를 비교하는 Matrix를 새로 구성한 후, 이를 기반으로 새로운 Centroid를 구성하고 이 과정을 반복적으로 수행하게끔 함수를 만들었다. 새로운 함수를 정의, 선언하여 사용될 수 있게 하였고 이후 kmeans() 등의 외부 라이브러리를 사용해서도 동일 과정을 수행할 수 있었다.

1. 군집분석을 위해 어떤 전처리가 필요한가?   
   가장 우선적으로 k의 개수를 정하는 것이 필요하다. 상관관계분석을 통해 데이터의 이상치를 살펴보고 제거한다. 데이터의 범위가 맞지 않아 정규화를 시켜야 했다. 데이터에서 목적에 가장 부합하는 Atrribute를 선정했다. 결과적으로 활용가능한 총 속성은 나이, 성별, 월평균소득, 결혼여부, 신장, 체중, 체질량지수, 식품섭취량, 에너지, 수분, 단백질, 지방, 포화지방산, 단일불포화지방산, 다가불포화지방산, n-3계 지방산, n-6계 지방산, 콜레스테롤, 탄수화물, 식이섬유, 당, 칼슘, 인, 철, 나트륨, 칼륨, 비타민A, 카로틴, 레티놀, 리보플라빈, 나이아신, 비타민C로 32개가 있다. 여기서 영향이 중복되는 변수는 제거하고 필요하다면 주성분추출을 시행한다.

두 번째 방법은 연관성 분석이다. 연관성 분석을 위해 갖고 있는 데이터셋들을 binarization 및 discretization을 사용하여, 이산적인 transaction data set으로 변형하여 연관성 분석을 시행하였다. 이 또한 모델링이 끝난 후 제품 판매 전략시의 인사이트를 도출할 수 있는데, 예시로 데이터셋에서 칫솔을 사용하고, 아침식사를 하는 고객들은 식생활 형편이 좋다는 연관성을 도출할 수 있었다. 따라서 이러한 연관성을 통하여 더욱 다양한 마케팅 전략을 펼칠 수 있고, 더욱이 데이터들의 속성을 더욱 자세히 조사함으로써 데이터에 대한 이해를 더욱 높일 수 있다는 장점이 있다. 따라서 분류 분석이나 연관성 분석 같은 탐색적 데이터 분석 방법은 당장 직접적으로 모델링의 성과를 높이거나, 데이터셋의 정확도를 향상시키는 효과를 불러올 수는 없지만, 데이터를 탐색함으로써 business problem의 식이 보충제의 판매전략을 다양화할 수 있다.

2.2-2 Association Rule Discovery

규칙(rule)이란 ‘if A(조건) then B(결과)’의 형식으로 표현된다. 연관규칙은 특정 사건이 발생하였을 때 함께 발생하는 또 다른 사건의 규칙을 말한다. 연관분석은 이러한 연관 규칙을 찾아내는 분석 기법이다. 일반적으로 추천 상품 카테고리를 만들고자 할 때 많이 쓰인다. 우리의 목표는 영양제를 먹는 사람이 어떤 사람들인지를 찾는 것이기 때문에 ‘if A then B’에서 A 또는 B가 영양제를 먹는 사람이라면 이에 연결된 변수가 영양제를 먹는 사람들과 어느정도 연관이 있다고 생각할 수 있다. 하지만 연관분석은 비지도 학습이기 때문에 타겟 변수를 따로 설정할 수는 없다. 따라서 연관분석을 실시하는 이유는 타겟 변수와 연관성이 있는 변수를 찾는 것이 아니라 온전히 변수들끼리의 연관성을 파악하기 위함이다. 이를 통해 연관성이 높은 변수들끼리 하나의 변수로 합치는 feature engineering을 추가적으로 실시할 수 있다.

연관분석을 진행하기 위해 원래 데이터를 거래 데이터 형식으로 변환시켜줘야 한다. 우리는 파레토 법칙에 따라 전처리가 완료된 데이터를 먼저 이진화 시켜줬다. 연속형 변수 또는 factor level이 많은 변수들은 우리의 주관적 판단에 따라 과감하게 0과 1로 나누었다. 거래 아이디와 아이템 항목을 속성으로 가지는 빈 데이터프레임을 만들고 이진화된 데이터를 거래 테이블에 넣어주었다. 다시 말해 이진화된 데이터의 행 번호는 거래 데이터의 거래 아이디에 대응되고, 이진화된 데이터의 각 행에서 1로 표시된 데이터는 모두 거래 데이터의 아이템 항목에 담긴다.

연관규칙을 생성하는 알고리즘으로 Apriori 알고리즘을 썼다. 지지도와 신뢰도에 따라 규칙이 달라지기 때문에 많은 시도를 했다. 신뢰도는 최소 0.7 이상으로 설정했으며 지지도는 디폴트값인 0.1보다 크게 설정하였다. 연관 규칙을 생성할 때 특정 속성을 제외시킬 수 있어서 분석 결과에서 당연한 규칙들은 제외시켰다. 예를 들어 분석을 진행하면서 ‘점심을 먹는다 -> 영양제를 먹는다’와 같은 규칙이 나왔었는데 점심을 먹지 않는 사람은 많이 없을 것으로 예상되기 때문에 ‘점심을 먹는다’는 제외시켰다.

*2.3 Modeling*

2.3.1 Decision tree

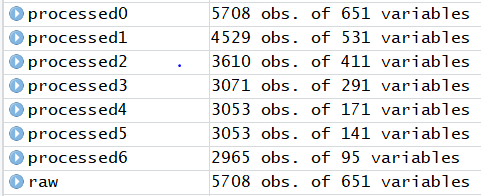
의사 결정 나무(Decision tree)는 데이터를 분석하여 이들 사이에 존재하는 패턴을 예측 가능한 규칙들의 조합으로 나타내며, 그 모양이 나무와 같아 의사결정나무라 불린다. 데이터 마이닝 과정에서 일반적으로 사용되는 방법론으로, 입력 변수를 바탕으로 목표 변수의 값을 예측하는 것을 생성하는 것을 목표로 한다. 결정 트리를 구성하는 알고리즘에는 주로 하향식 기법이 사용되며, 각 진행 단계에서는 주어진 데이터 집합을 가장 적합한 기준으로 분할하는 변수값이 선택된다. 서로 다른 알고리즘들은 분할의 적합도를 측정하는 기준이 있으며, 이 과정에서 지니 불순도, information gain, classification error 등이 사용된다. 우리는 rpart, caret 등의 library를 통해 Decision tree를 구성하였으며, Confusion matrix를 통해 DT의 accuracy를 평가하였다.

2.3.2 Random Forest

Random Forest는 Decision Tree 기반의 예측 모델이다. 무작위의 의사결정트리를 여러 개 만들어서 이로부터 나온 예측 결과들의 평균 또는 다수의 예측 결과를 이용하는 앙상블 기법을 이용한다. 많은 개수의 트리는 보다 높은 성능의 모델을 보장하지만 계산시간이 오래 걸릴 수 있다. 따라서 적절한 개수의 트리를 만들고 사용할 변수의 개수를 정해야 한다. 의사결정나무와 마찬가지로 지니계수, information gain, 예측오차 등을 통해서 트리가 만들어진다. 하지만 랜덤포레스트의 결과물은 위에서 설명했듯이 다수의 의사결정 나무의 결과물들을 앙상블 기법을 이용해 평균을 계산한다. 이 때문에 트리에 대한 직접적인 시각화가 불가능하다. 따라서 랜덤포레스트 모델을 통해서 단지 정확도와 confusion matrix를 계산해보고 전에 만든 Decision Tree보다 좋은 모델이 존재하는지 확인을 하는 용도로 모델을 사용할 것이다.

**3. Results**

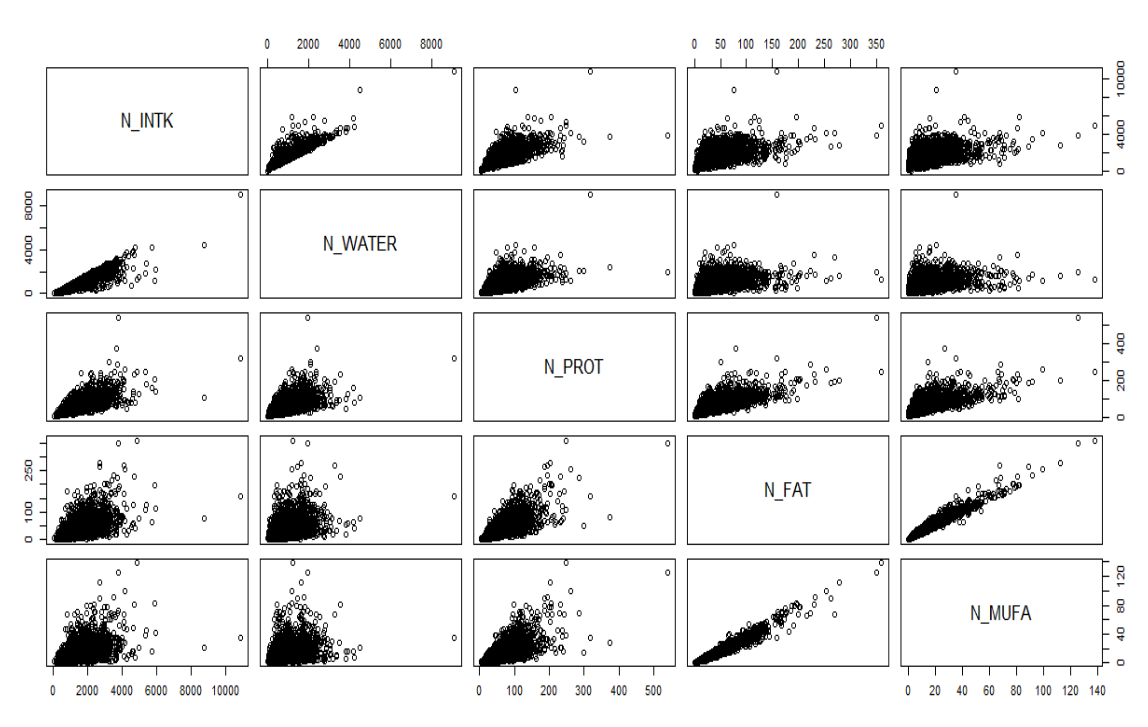
3.1 전처리 결과

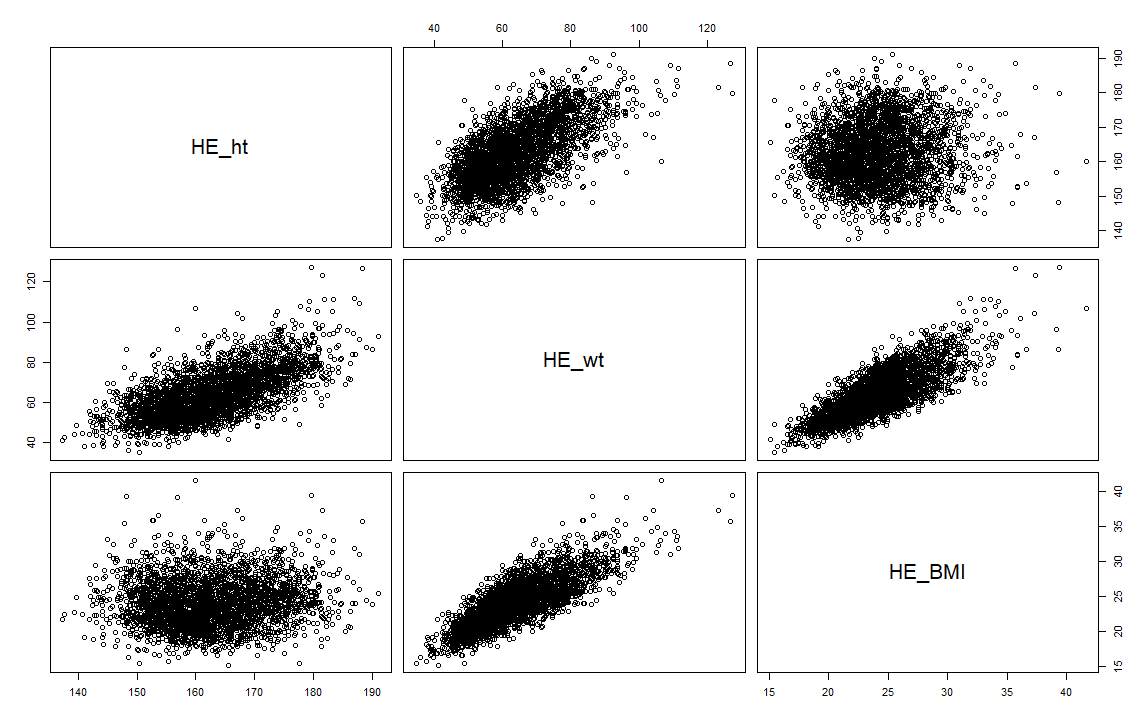


전처리는 위에서 기술한 Method를 그대로 적용했다. raw데이터는 분석 대상에서 제외되는 행, 즉 18세 미만 청소년 행과 휴리스틱 기반 속성 제거를 한 결과이다. Processed1부터 processed5까지는 파레토 법칙에 기반한 Nan값 처리 과정에 대한 결과이고, 마지막 processed6은 Feature engineering을 적용한 결과이다.

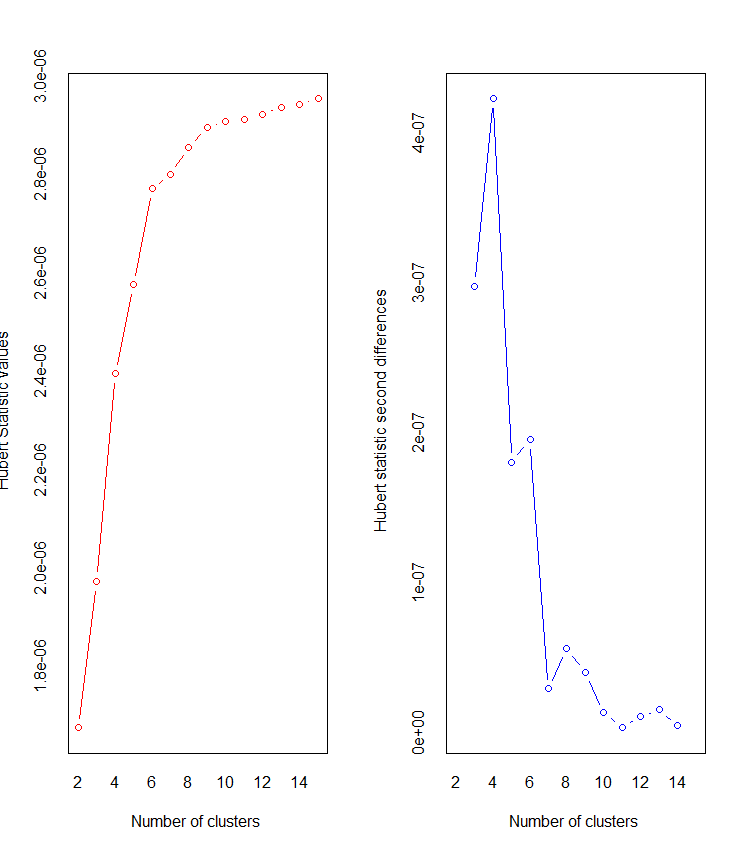
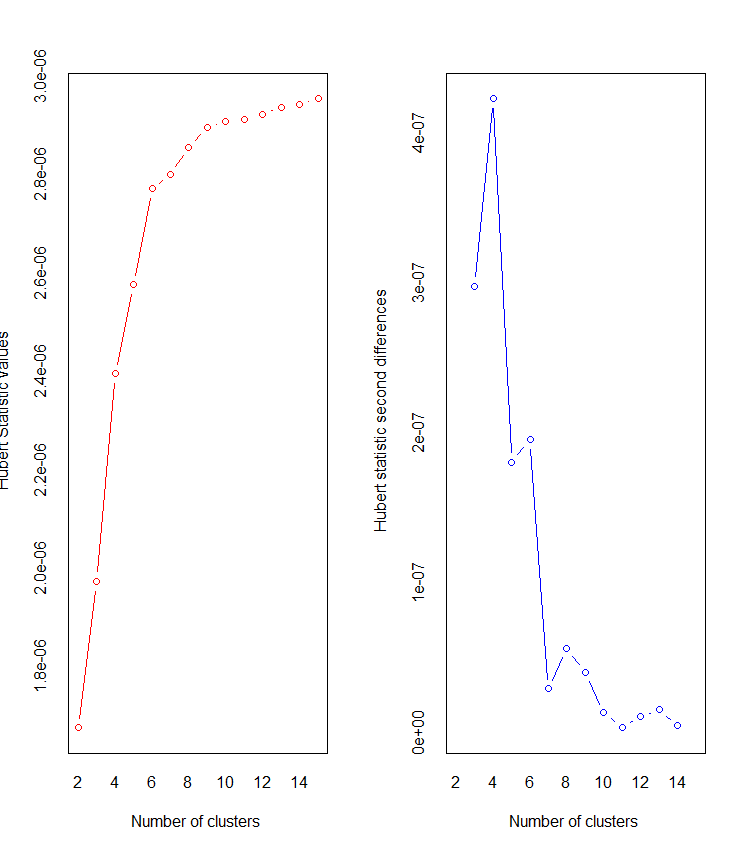
3.2 Cluster Analysis 결과

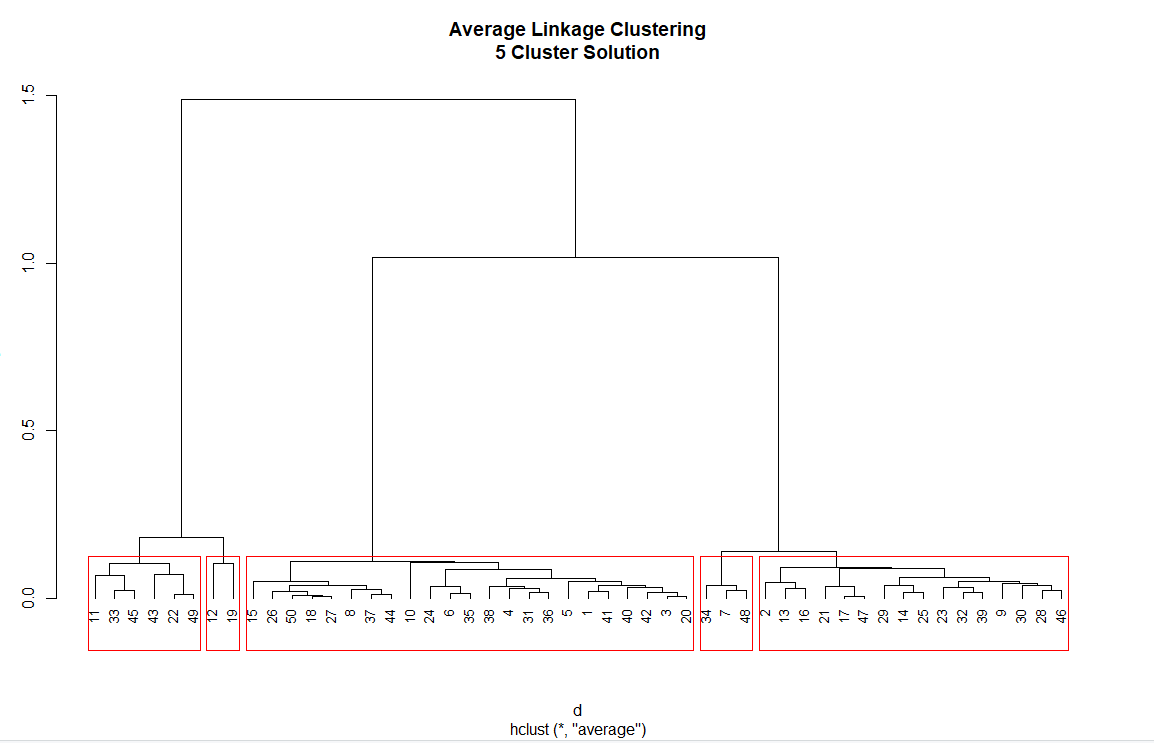
3.2-1 상관분석 :   
전처리리를 마무리 했던 데이터에서는 연속형 Attribute가 많지 않았다. 그래서 추가적인 군집분석을 시행하기 위해서 CRISP-DM의 과정을 따라 다시 데이터 전처리로 돌아갔고 연속형 Attribute를 찾아보았다. 연속형 Attribute 간에 상관이 높은 Atrribute를 분석하고 제거하였다.

  
예시1 : N\_FAT(지방)과 N\_MUFA(단일불포화지방산)은 상관관계가 높으므로 N\_FAT하나만 군집분석에 사용한다.



예시2: HE\_wt (몸무게)와 HE\_BMI (체질량지수)는 상관관계가 높으므로 HE\_BMI 하나만 군집분석에 사용한다.

3.1-2 최적 군집 수의 파악  
Nbclust 라이브러리를 통해 데이터 속에서 최적의 군집 수를 파악할 수 있다. 분석 결과, 4에서 6까지의 군집 수가 최적 군집수로 파악되었다. 따라서 4에서 6사이의 수를 군집 개수로 지정하여 군집분석을 시행한다.



K-means를 사용한 분할적 군집분석(Partitional Clustering)이 정당한 군집으로 나뉘었다는 근거를 얻기 위해서 계층적 군집분석(Hierarchical Clustering)을 추가적으로 시행하였다. 이는 단지 초기 군집을 잡을 때 사용한 centroid들과 결과로써 나온 군집들이 타당한 군집인지를 판별하기 위해 사용하였다. 계층적 군집 분석은 너무 많은 양의 데이터를 포함하기 어렵기 때문에 대표적으로 100개 정도를 랜덤하게 뽑아서 사용하였다.

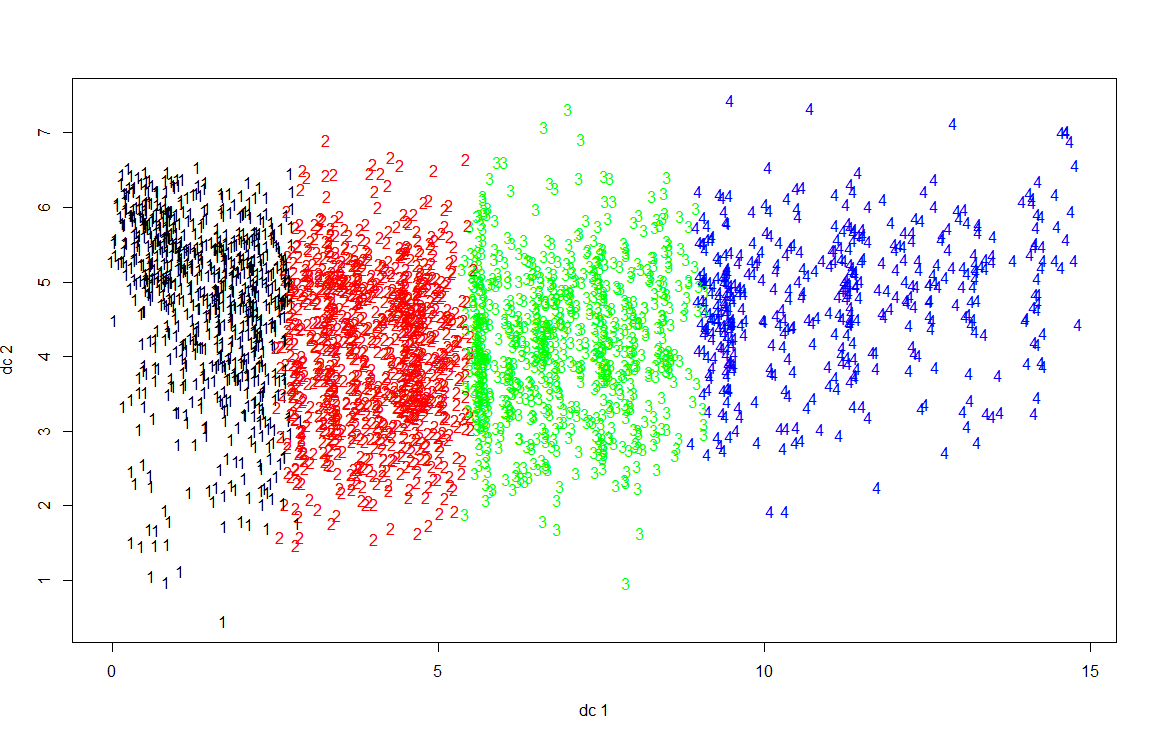
3.1-3 K-means Cluster Analysis  
직관적으로 군집이 존재할 만한 데이터끼리 군집분석을 시행해보았다.  


그림 : 나이, 소득분위에 대한 군집분석

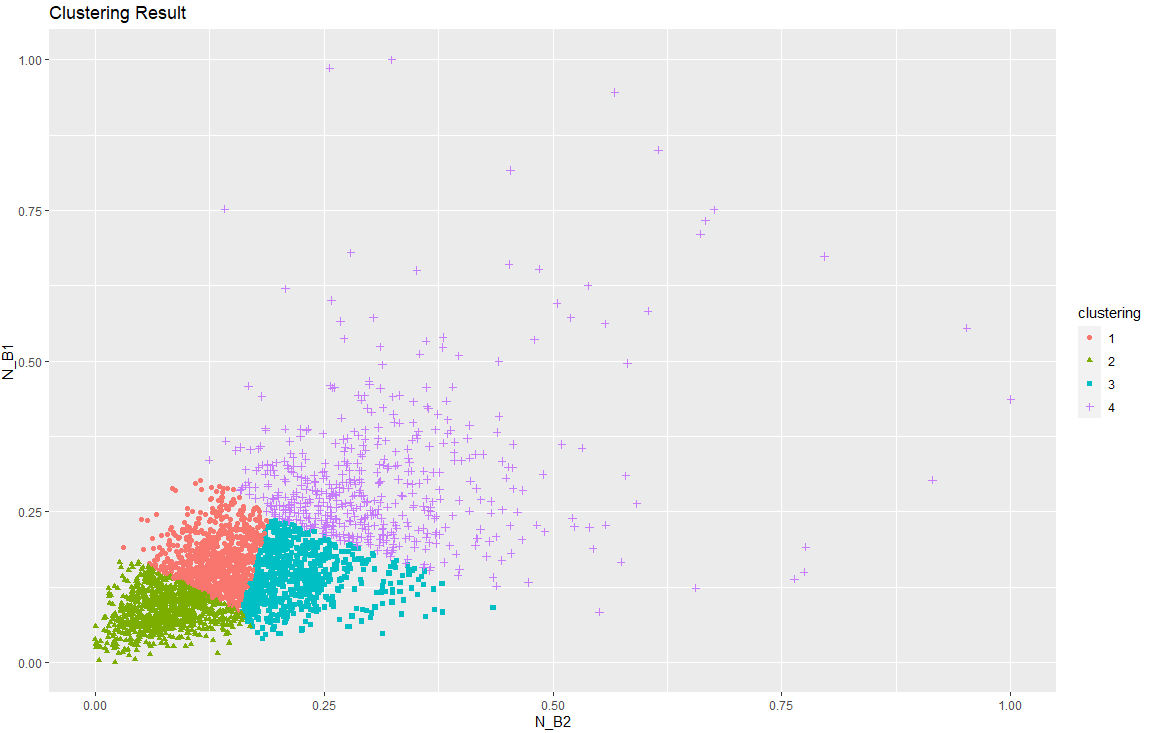


그림 : 비타민A와 비타민 C에 대한 군집분석

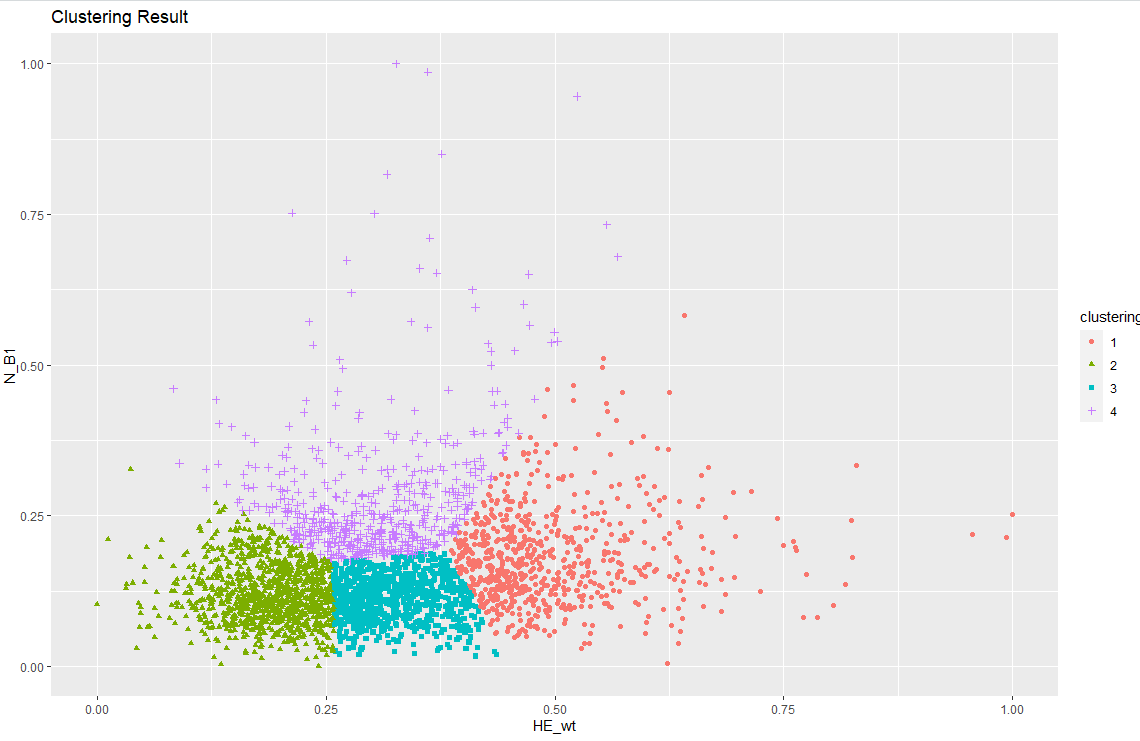
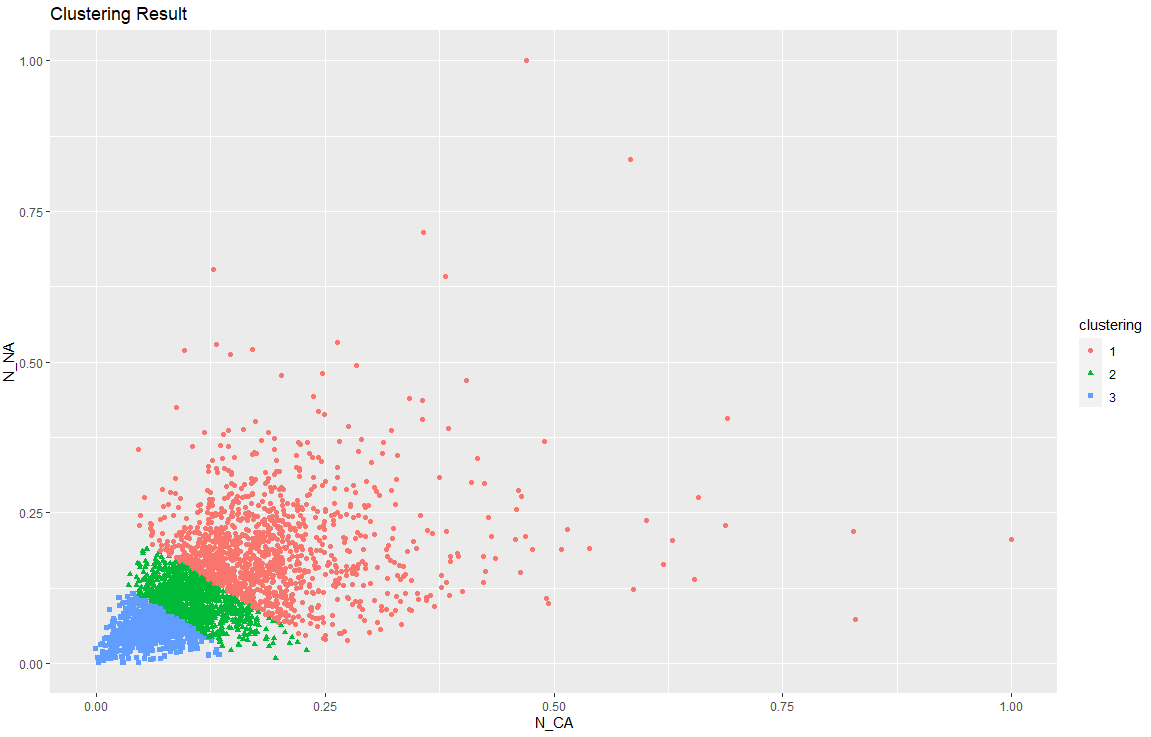
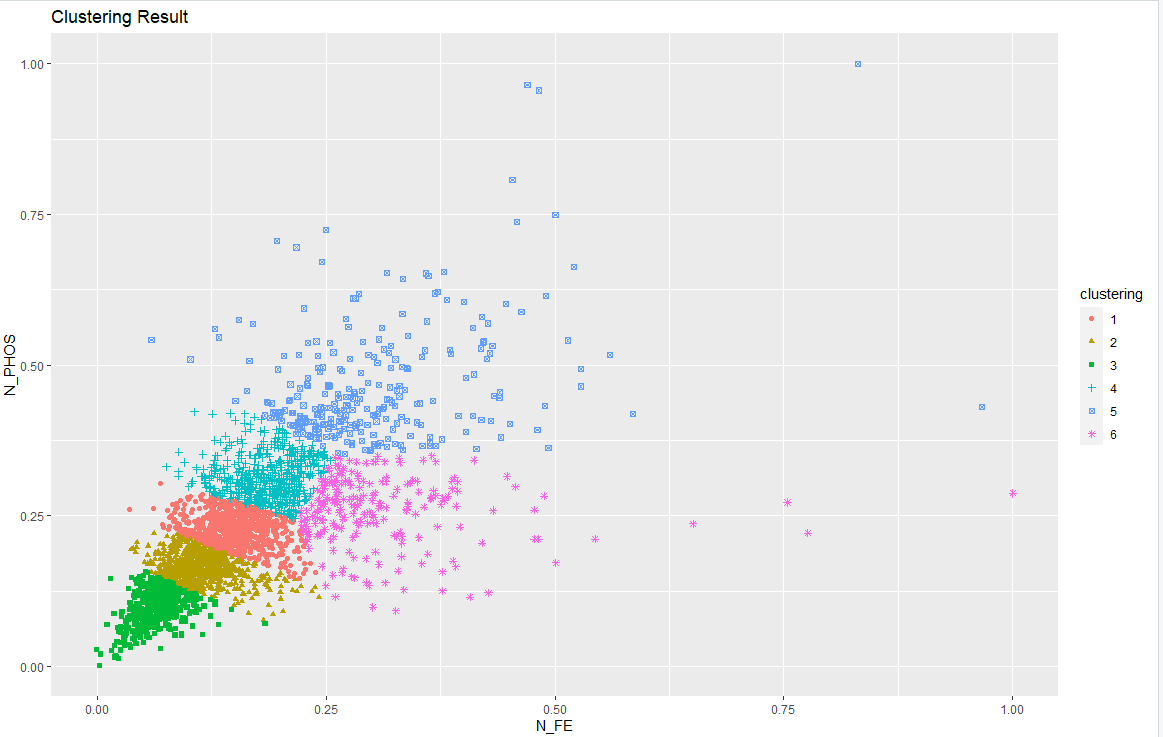
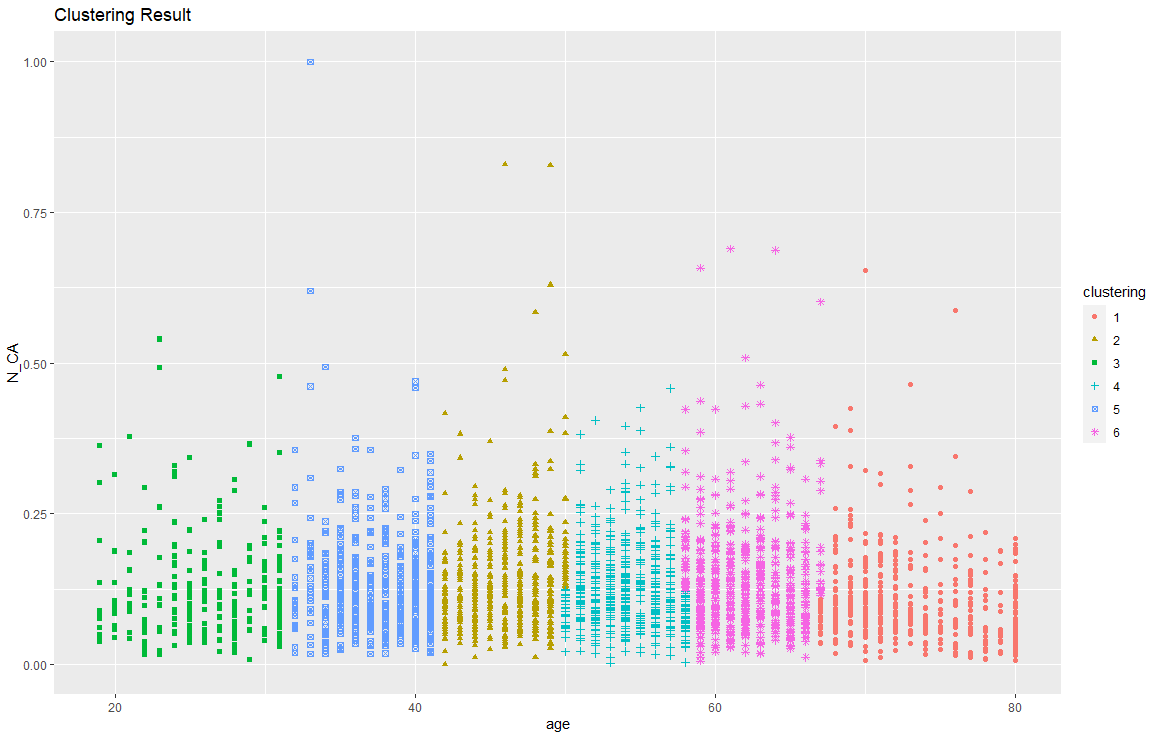


그림 : 몸무게와 비타민에 대한 군집분석

  
그림 : 칼슘과 나트륨대한 군집분석

  
그림 : 철과 인에 대한 군집분석

  
그림 : 나이와 칼슘에 대한 군집분석

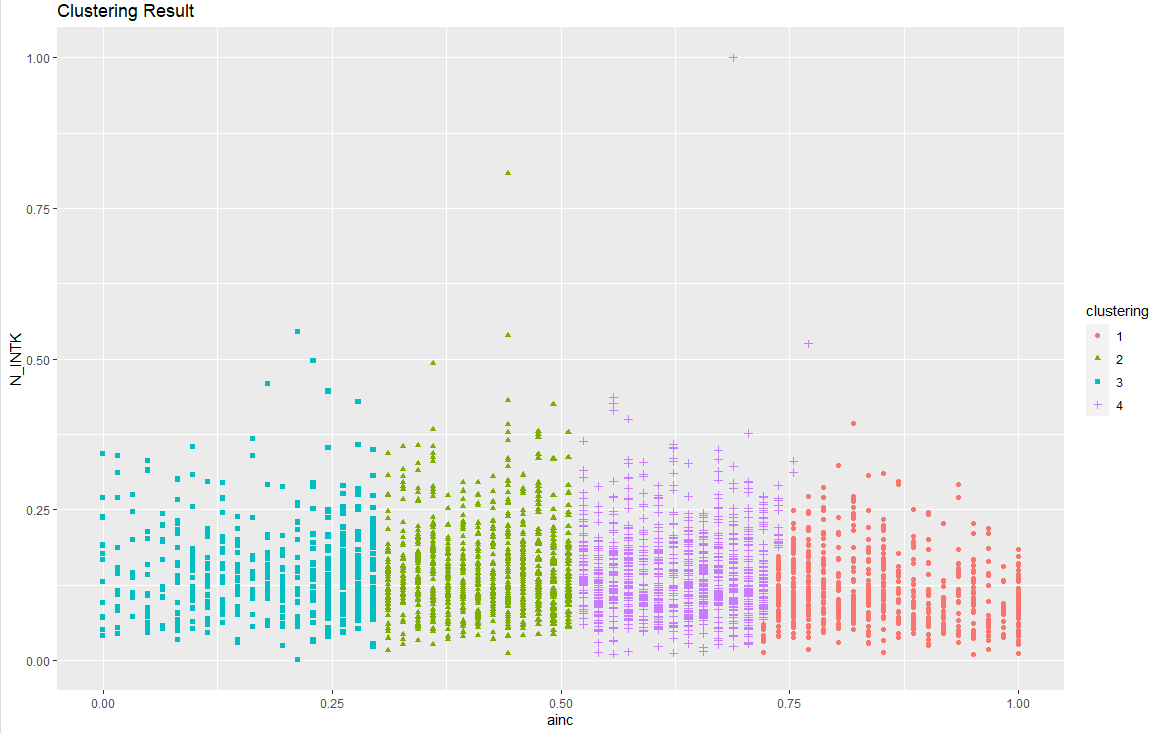


그림 : 월 평균 소득과 식품섭취량에 대한 군집분석

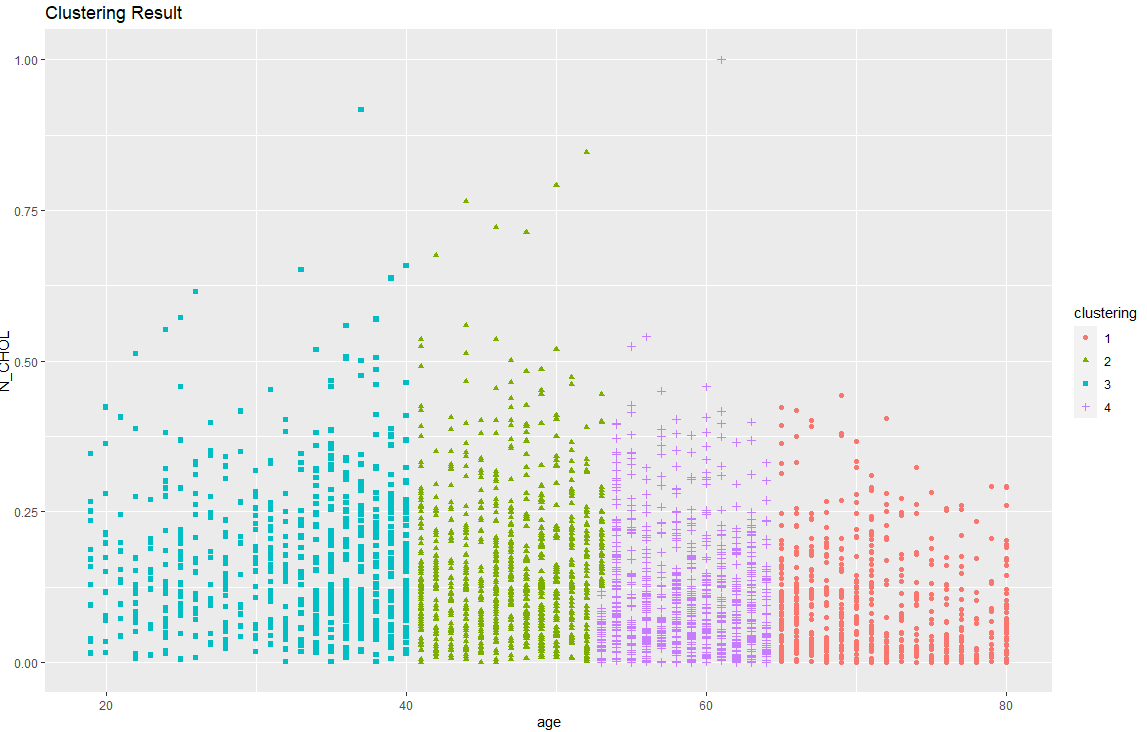
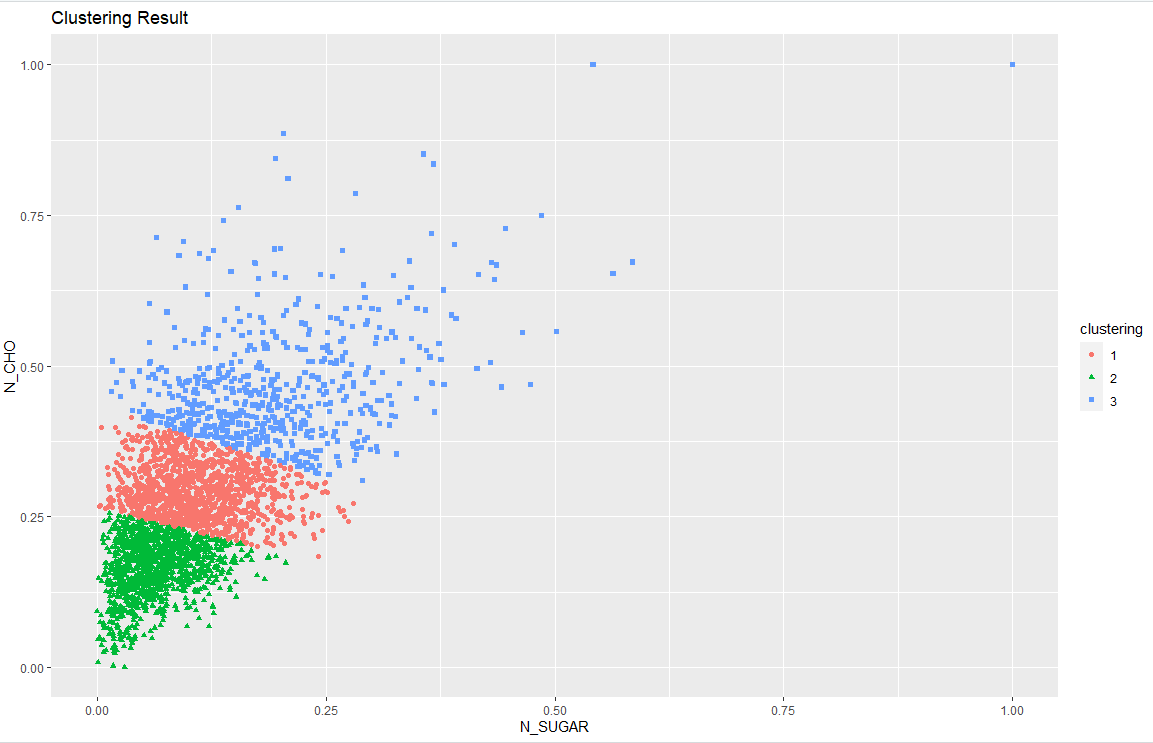


그림 : 나이와 콜레스테테롤에 대한 군집분석

  
그림 : 설탕과 탄수화물에 대한 군집분석

또한 3차원 형태로도 시각화를 수행하였다. 3개 Attribute 이상, 즉 3차원 이상의 데이터에 대해 군집을 분석할 때 2차원 상에서는 직관적으로 이를 구분하기 어렵다. 따라서 이를 3차원 이상으로 새로이 시각화할 수 있는 방법이 필요했다. 우리는 K-means clustering에 있어 kmeans() 라이브러리 내장 함수를 이용하여 clustering을 수행하였고, clustered 된 데이터를 scatterplot3d라는 3차원 시각화 라이브러리를 통해 시각화할 수 있었다. 아래는 나이(datas2$age), BMI(datas2$HE\_BMI), 월간 소득(datas2$earn\_month)의 세 가지 attribute를 통해 군집을 시각화한 예시이다.

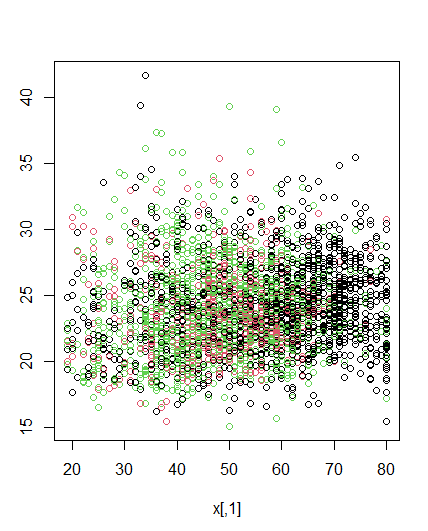


그림 : 나이, 체질량지수, 월평균소득에 대한 군집분석 시각화(2차원)

위처럼, 다양한 군집을 2차원 평면 상에서 표현할 경우 이를 제대로 구분할 수 없고, x,y,z의 세 개 축을 각각 원하는 Attribute로 지정하여 군집이 3차원 상에서 어떻게 형성되는 지 확인할 수 있었다.

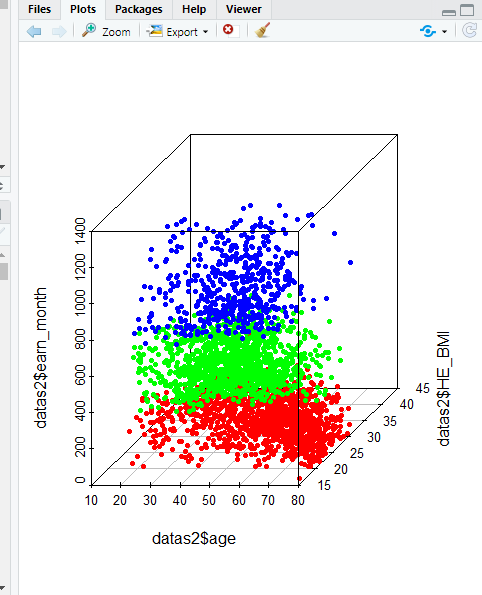


그림 : 나이, 체질량지수, 월평균소득에 대한 군집분석(3차원)

3.2-4 Multiple runs

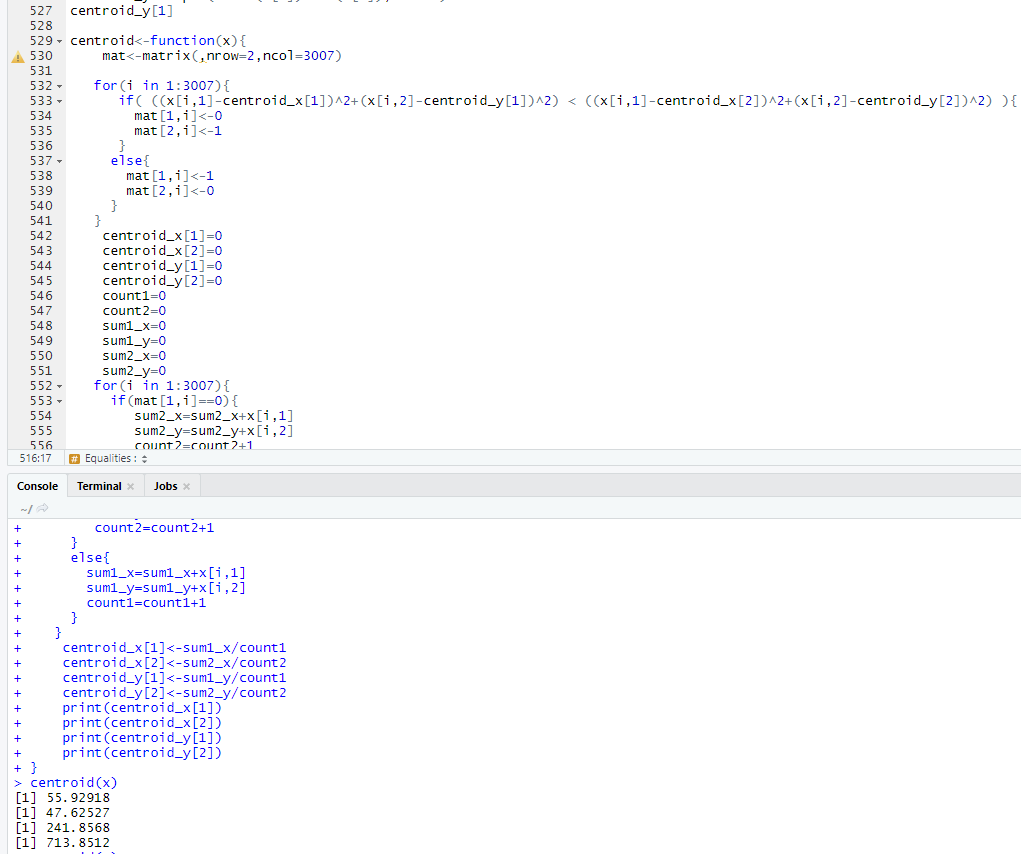
갖고 있는 데이터들 중 clustering하기에 적합한 연속형 데이터를 전처리하고 통합하니, 나이, 월 소득, 몸무게, 키, 전 일 섭취 영양소들(특히 PCA를 통해 나온 PC1요소인 make energy series: [에너지, 단백질, 인]의 요소를 대표로 사용하였다.) 등이 사용하기 좋은 형태로 나와 있어서, 정규화를 통해 k-means clustering을 다시 시행하였다.

분할적 군집분석(partitioning clustering)을 실시한 이유는 계산비용 측면에서 초반에 다양한 시도를 하기 위해서, 특히 객체 별로 exclusive하게 어떤 군집에 속할지를 명시적으로 할당하기위해 hard clustering을 해주었다. Fuzzy나 가중치를 활용한 soft clustering은 비용적인 문제로 선택하지 않았고, 다만 결과로 나온 군집들을 평가하기 위해, 그래프를 활용하여 단일 연결을 기반으로 한 응집형 계층적 군집화를 통하여, 그 합리성을 판단하였다. (계층적 군집 분석은 ‘average’ 를 기반으로 거리계산을 하였다.) 밀도나 분포 기반의 군집화는 데이터 셋의 형태에 최적화된 기법이 아니라 판단되어 평균(Mean)이나 중앙값(Median)을 기준으로 잡은 최소 거리를 기준으로 clustering을 하였다.

가장 타당한 군집의 개수는 “fviz\_nbclust” 함수를 통하여 정해주었는데, 이는 Intra-cluster 거리는 최소화하면서, Inter-cluster 거리를 최대화해주는 군집의 개수를 알려주는 함수이다. 이후 데이터가 정규 분포를 띄지 않는 특징을 반영하여 ‘pearson’ 상관계수를 사용하여 다시 한번 군집화를 시도하였다. nstart = 25로 설정하여 예측력을 높이고, 초기 centroid가 랜덤하게 잡히다 보니 결과로 나온 cluster들이 상당히 합리적이지 않은 결과들이 종종 도출되기도 하였다. 따라서 적절한 Centroid를 찾아 업데이트하고, ‘좋은 군집’을 찾기 위해 여러 가지의 시도를 할 수 있었다.

초기 Centroid를 random하게 잡은 후, Euclidean distance를 이용해 Closeness를 비교하는 Matrix를 새로 구성한 후, 이를 기반으로 새로운 Centroid를 구성하고 이 과정을 반복적으로 수행하게끔 함수를 만들었다. 이후 여러 번의 시도를 통해서 초기 centroid를 바꾸면서 군집을 나눌 수 있었다.

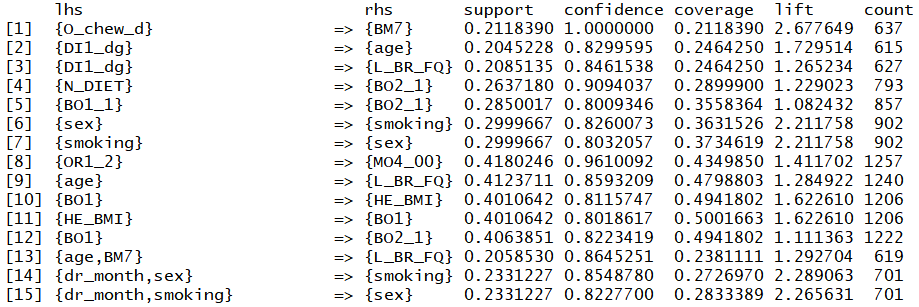
하지만 이 결과는 다양한 그룹들을 가져오긴 하지만 나온 그룹이 정말 근거가 있는 그룹인지를 확인할 수 없고, 다만 분석 목적에 합당한 그룹을 뽑는다는 점에서 근거가 부족하여 기각하였다. 물론 결과로 나온 군집 중에는 정말 합리적인 군집이 있을 수 있지만, 그렇다면 다른 과정을 통해서도 도출 가능하기 때문에, 단지 검증의 용도로만 사용하였다.



예시 : 먼저 initial centroid를 기존의 방식으로 도출하기 위해 함수 centroid(x)를 구현하였고, 이를 통해 첫 번째 centroid를 도출한 모습이다.

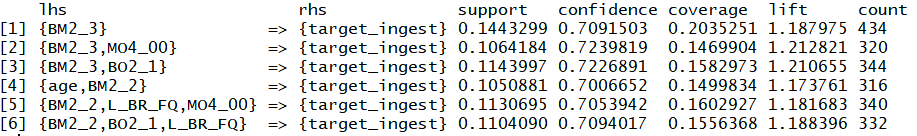
**3.3 연관분석 결과**

<아무런 설정을 하지 않은 경우>



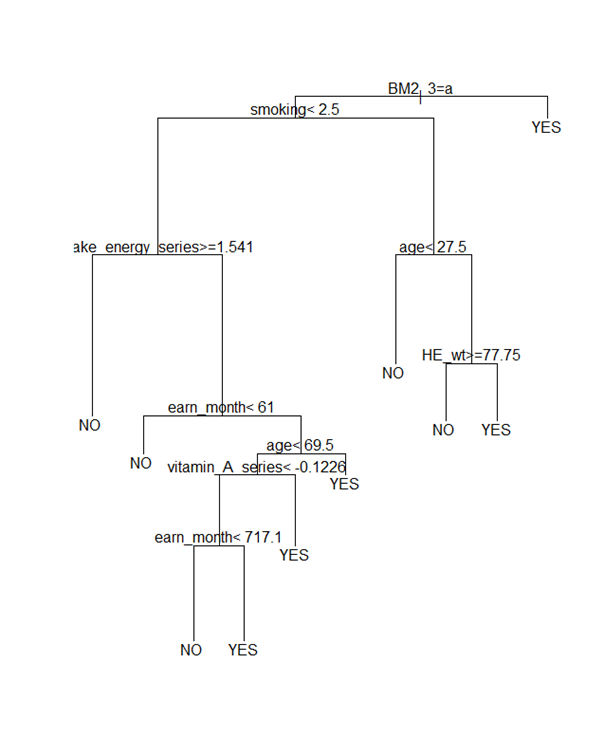
(지지도 = 0.2, 신뢰도 0.8)로 설정해서 모델을 돌린 결과이다. 상위 다섯 개의 규칙은 아래와 같다. 단, ‘점심을 먹는다’, ‘저녁을 먹는다’ 속성은 제외시키고 돌린 결과이다.

<우변 상수를 ‘영양제 복용’으로 적용한 경우>

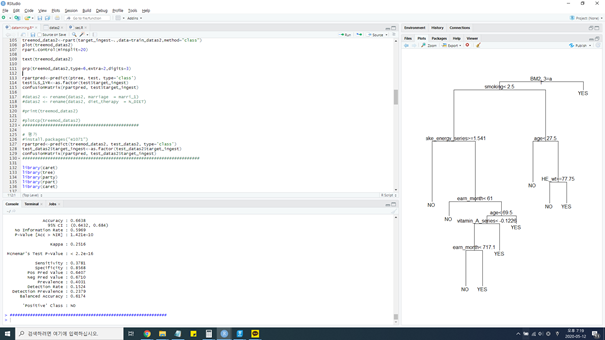


(지지도 = 0.1, 신뢰도 = 0.7)로 잡았다. 우변상수를 결정했을 때 신뢰도가 0.7보다 크면 규칙이 생성되지 않았다. 생성된 규칙은 아래와 같다. BM2\_3은 치간칫솔을 사용하는 사람이다. 우변상수를 설정한 경우 신뢰도를 0.7 이상으로 잡으면 규칙이 생성되지 않는다.

4) 의사결정나무 분석 결과



Model 생성 결과, 위와 같은 모델이 생성되었다. BM2\_3 -> 칫솔 외, 치간칫솔의 사용 여부



Model의 평가 결과, 해당 Model의 Accuaracy는 66.38%로 측정되었다.

(5) 랜덤포레스트 분석 결과

앞에서 전처리한 데이터는 96개의 Attribute와 2965개의 Record를 가졌다. 계산 시간을 줄이고 데이터셋 마다 모델 출력결과를 확인하기 위해 전체 데이터셋을 4개로 나누었다. 이를 위해 R코드를 작성하여 각 데이터셋을 학습하기 위해 train set과 test set으로 7:3으로 나누었다. 그리고 개별적으로 Train set으로 랜덤 포레스트 모델을 돌려본다. 모델의 파라미터는 1000, 변수의 개수는 6개이다. 모델의 결과는 각각 다음과 같다.

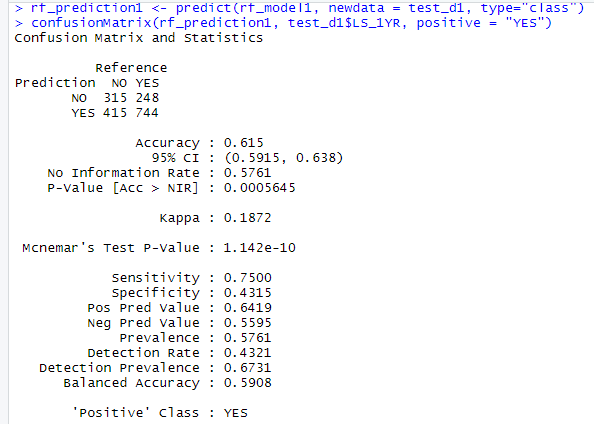


그림 2 : Data1에 대한 모델학습결과

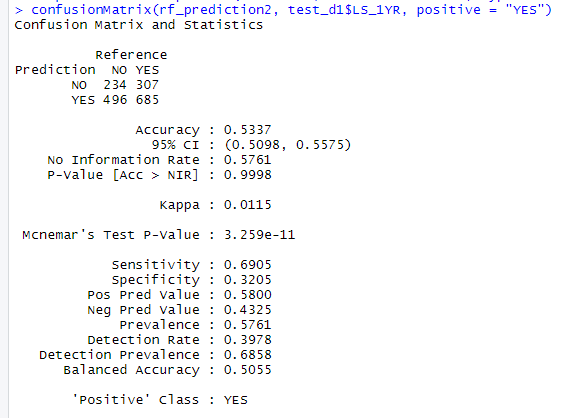


그림 2 : Data2에 대한 모델학습결과

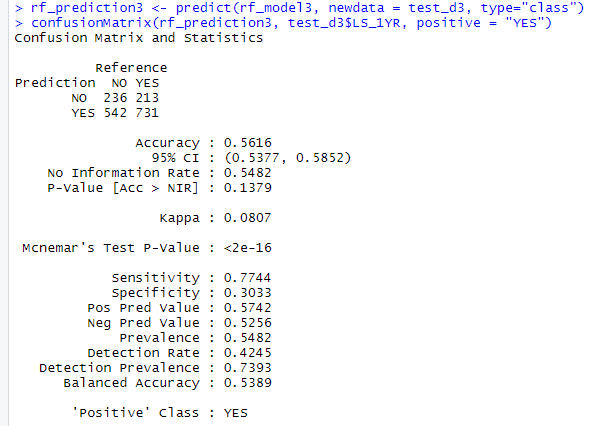


그림 3: Data3에 대한 모델학습결과

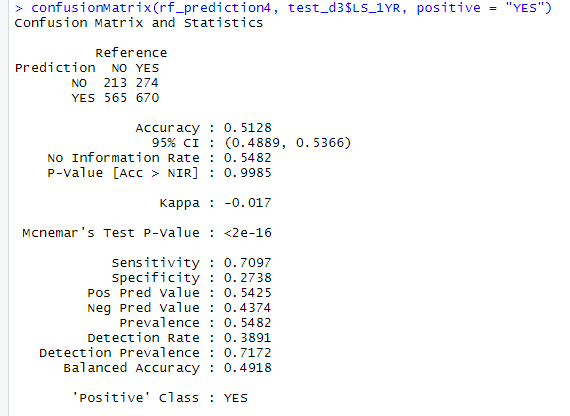


그림 4: Data4에 대한 모델학습결과

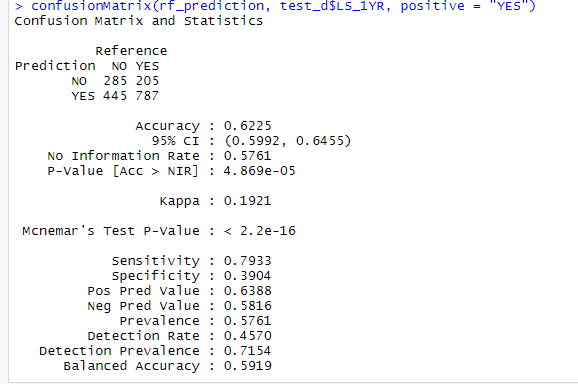


그림 5: 전체 Data에 대한 모델학습결과

**4. Discussion**

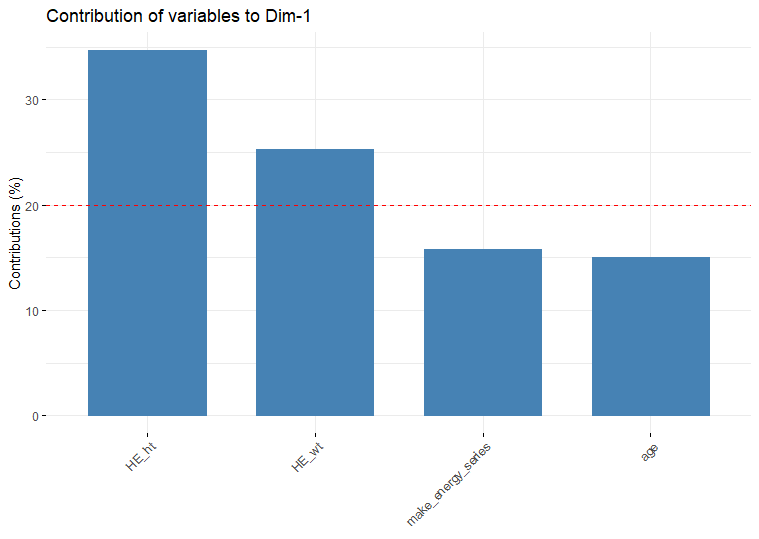
1) 전처리 분석 결과

파레토 법칙에 기반한 Nan값 처리를 먼저 살펴보자(processed0~processed5). 먼저, 파레토 법칙이란 ‘전체 결과의 80%가 전체 원인의 20%에서 일어나는 형상’을 가리킨다. 이를 Nan값에 대응시켜 생각해보면 모든 속성을 Nan값의 개수로 내림차순 정렬했을 때 상위 20%가 전체의 ‘원인’에 해당하므로 지워주면 된다. 이 과정은 Nan값에 가장 크게 기여하는 상위 20%를 지운다는 점에서 의미 있다. 이렇게 행과 열 기준으로 파레토 법칙을 계속 적용하면 Nan값이 없는 데이터프레임을 얻을 수 있다.

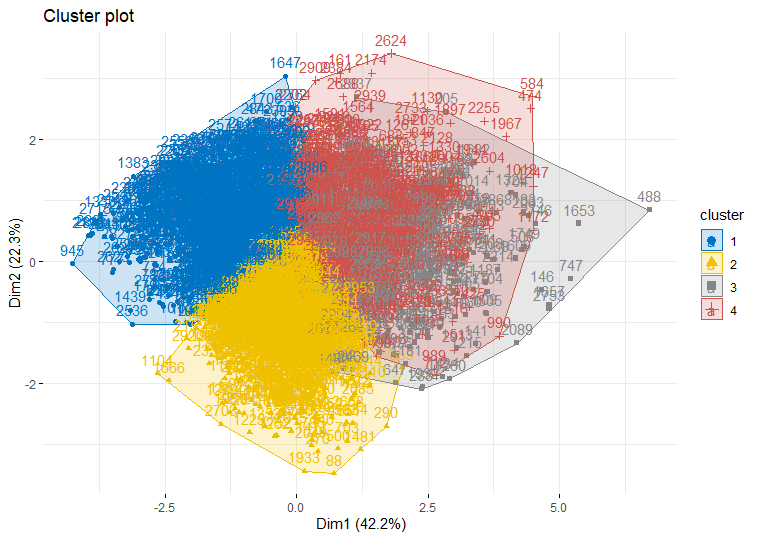
processed5~processed6에서 그룹화, 이상치 제거, 정규화, 형변환은 각각이 의미를 갖는다. 각각이 가지는 의미는 Method의 feature engineering과 겹치므로 생략하겠다. 하나만 예를 들어 보자면, 지역 그룹화는 factor level이 17인데 이를 그대로 분석에 사용하면 split 개수가 17개나 되므로 모델의 성능에 악영향을 끼칠 수 있다. 17레벨을 중부와 남부 지역으로 binarization 시켜준 것은 의사결정 나무에 적합한 데이터 속성으로 변환시켜준 것이라고 할 수 있다.

2)군집 분석 결과

군집분석을 전체 데이터에 대해 최초로 시행하고 유의미한 결과를 얻지 못했다. 다른 군집을 더 찾아보기 위해 영양소의 연속형 데이터들 중에서 2개씩만을 산출하여 2차원의 군집분석을 추가적으로 시행했지만, 이에 따른 군집이 초기 centroid를 잡은 그대로만 형성되어 유의미한 군집을 획득하지는 못하였다. 이 문제를 해결하고자 계층적 군집화와 초기 centroid의 직접적인 변형을 시도해 보았지만, 결과로 큰 차이를 만들지 않았다. 이에 따라 변수들의 상관 분석을 해보았더니, 영양소의 데이터들은 상당히 상관계수가 높아서 군집을 형성하기에 적합하지 않은 변수들이었다는 판단을 내릴 수 있었다. 결론적으로 시각화된 결과를 보았을 때 유의미한 군집을 발견할 수 없었다. 그림에 나타난 군집은 최적의 k에 따른 가상의 군집이다. 이는 데이터를 생성하기 위한 표본추출이 아주 잘 시행되었다고 볼 수 있다. 단지 두 개의 데이터만으로는 군집을 찾을 수 없을 것 같아서 변수를 통합하고 PCA 기법을 통해 영양소 데이터를 PC1으로 축소해서 다시 군집분석을 시행하였다.

  
그림 : 차원1에 대한 변수의 기여도 분석

입력한 여섯 가지 연속형 데이터들 중에서 PCA로 요약된 두 가지 차원에서 각 차원별로 가장 중요한 영향을 끼치는 두 요소는 키와 월 간 소득이었다. 군집 간의 거리를 최대화하고, 군집 내의 거리를 최소화하는 평가기준으로 적정한 군집 수를 찾아보니 우리의 데이터에서는 4개의 군집이 가장 적합하다는 결과가 나왔었다. 군집분석 시작 시 정규화를 선행하여 편차를 맞춰주었기 때문에 각 평균을 기준으로 총 4가지의 군집이 나왔다.

  
그림 : 주성분이 추출된 데이터를 바탕으로 시행한 군집분석

4 가지의 군집 특성을 모두 이용할 수 있지만, 우리가 관심있게 본 그룹은 키가 작고, 월 소득이 높은 집단이었다. 특히 해당 집단은 밀도가 상당히 촘촘하여 유의미한 타겟이 될 수도 있다는 판단을 하였다. 따라서 해당 그룹의 집단들을 고객으로 규정한다면, 그에 적합한 마케팅적 접근을 하기 위해서는 가격을 낮추는 전략보다는 키에 관한 부분을 자극하여, 허리나 골격에 좋은 영양제에 관한 판매 전략을 수립하여, 좀 더 고객 친화적인 제품 전략을 취할 수 있다.

이 군집 분석은 그룹의 초기 centroid를 랜덤하게 잡아 주기 때문에 이에 대한 적절한 신뢰성을 부여하기 위하여 시드를 고정하지 않고 그룹을 몇 번 더 잡아보았다. 그래도 같은 형태의 그룹이 자주 반복적으로 나타나는 것으로 보아 잡혀 진 군집이 상당히 신뢰할 수 있다고 결론지었다.

3)연관 분석 결과

[1] 저작 불편 호소 -> 씹기 불편

[2] 고혈압 진단 받음 -> 54세 이상

[3] 고혈압 진단 받음 -> 아침을 자주 먹는다

[4] 식단 조절을 한다 -> 1년간 체중변화 시도를 함

[5] 1년간 체중 변화가 있다 -> 1년간 체중변화 시도를 함

연관 분석의 평가 척도로 신뢰도를 사용하겠다. 신뢰도는 높을수록 그 규칙이 유용할 가능성이 높다는 뜻이다. 먼저 지지도와 신뢰도 외에 아무런 설정을 하지 않은 경우 ‘저작 불편 호소 -> 씹기 불편’이 신뢰도가 1로 가장 높았다. 이는 설문 문항에서 ‘저작 불편 호소’를 ‘예’라고 대답한 사람은 모두 ‘씹기 불편’을 ‘예’로 답했다는 뜻이다. 모델을 돌릴 때 최소 신뢰도는 0.8로 지정했기 때문에 위의 규칙들은 모두 신뢰도가 0.8 이상이다. 그만큼 유용한 규칙들이며 모두 눈으로 봤을때 납득이 가는 사실이다. 저작 불편 호소와 씹기 불편은 거의 같은 항목이므로 신뢰도가 1인게 말이 된다. 두 번째 규칙도 대체적으로 나이가 많은 사람들이 고혈압에 걸릴 확률이 크기 때문에 말이 된다. 연관 분석이 제대로 시행됐음을 알 수 있다. 우변 상수를 타겟 변수로 지정해서 모델을 돌려 어떤 규칙이 나오는지 살펴보았다. 결과는 아래와 같다. 상위 다섯 개의 규칙들만 가져왔다.

[1] 치간 칫솔 사용 -> 영양제 복용

[2] 치간 칫솔 사용, 치과 병원 이용 -> 영양제 복용

[3] 치간 칫솔 사용, 1년간 체중 변화 시도를 함 -> 영양제 복용

[4] 54세 이상, 구강 세정액 사용 -> 영양제 복용

[5] 구강 세정액 사용, 아침 자주 먹음, 치과 병원 이용 -> 영양제 복용

분석 결과 치간 칫솔을 사용하는 사람이 영양제를 복용할 가능성이 높다는 사실을 알 수 있다. 신뢰도는 0.7로 설정했다. 흥미로운 점은 상위 다섯 개의 규칙이 모두 이빨과 관련 있다는 사실이다. 이는 두 가지 의미로 해석할 수 있다. 먼저 치과에 다니는 사람 중에 잠재고객이 있다는 사실로 해석할 수 있다. 또는 치실을 사용하는 사람일수록 자기 관리를 열심히 하고, 자신의 건강을 챙기기 위해 영양제를 복용하는 것으로 해석할 수 있다. 이를 통해 다음과 같은 인사이트를 발굴할 수 있다.

-치과에 영양제 홍보를 간접적으로라도 한다면 마케팅 효과를 기대할 수 있다

-비타민처럼 자주 먹는 영양제는 마트에서 치간 칫솔 또는 구강 세정액과 가까운 곳에 위치시킨다

4) 의사결정나무 분석 결과

의사결정나무 모델 생성을 위해 rpart, tree, caret 등의 R library를 사용하였다. 또한 Random seed, 즉 난수를 생성하여 최적의 모델을 찾을 수 있도록 하였다. 도출된 의사결정나무 모델을 분석한 결과, 식이보충제 섭취 여부를 결정하는 예측요인은 치간칫솔의 사용 여부(BM2\_3), 평생흡연 여부(smoking), 인, 에너지, 칼륨, 단백질에 대한 24시간 내 섭취량(make\_energy\_series), 나이(age), 몸무게(HE\_wt), 월 소득(earn\_month), 비타민 A, 레티놀, 카로틴에 대한 24시간 내 섭취량 (vitamin\_A\_series)등으로 나타났다.

이를 구체적으로 살펴보면, 먼저 식이보충제 섭취 여부를 결정하는 결정적인 요인은 정말 의외의 요인으로, 구강건강 관리를 위해 치간칫솔을 사용하는 지 여부였다. 즉, 치간칫솔을 사용하는 사람은 다수가 식이보충제를 섭취한다는 것이다. 구강 관리를 세심하게 하는 사람일수록 식이보충제를 섭취하는 등, 건강 관리에 더 유념한다는 뜻으로 해석될 수 있는 대목이다. 한편 담배를 피우지 않을수록(smoking<2.5는 흡연자, smoking>=2.5는 비흡연), 나이가 27.5세 이상일수록 식이보충제를 더 많이 섭취하는 것으로 확인할 수 있었다. 한편 해당 모델의 정확도를 평가하기 위해 Confusion Matrix를 사용하였다. 전체 중에서 올바르게 예측한 데이터가 얼마나 되는지, 즉 TP + TN / TP + TN + FP + FN으로 계산되는 Accuracy를 통해 평가하였으며 Accuracy가 66.38%임을 확인하였다. 100%에 가까울수록 좋은 Model임을 의미한다.

5) 랜덤 포레스트 분석 결과

각 데이터셋을 모델에 돌린 출력결과인 Result의 그림1~4를 보면 정확도가 높지 않다는 것을 알 수 있다. 전체 데이터에 대한 모델의 결과는 Result 항목의 그림5를 통해 확인 가능하다. 그림을 통해 알 수 있듯이 전체 데이터셋을 학습시켰을 때 모델의 정확도가 더 높다. 즉, 랜덤으로 생성되는 의사결정 나무에서 더 좋은 결과를 내는 의사결정나무가 존재한다는 것이다. 이는 데이터셋에서 Feature Selection, Feature, Deletion, Feature Engineering을 통해 모델의 성능을 더욱 개선할 수 있음을 시사한다. 결론적으로 CRISP-DM의 과정에 따라 다시 앞의 과정으로 돌아가서 데이터에 대한 작업을 지속적으로 해야 한다.

**5. Limitation**

가장 큰 문제는 정확도가 낮다는 것이다. 데이터셋에서 식이 보충을 섭취한다는 사람의 비율과 그렇지 않은 사람의 비율이 약 6:4 정도이다. 이 때 모델의 정확도가 65% 정도면 랜덤으로 뽑은 사람이 식이 보충제를 섭취한다고 예상하고 그것이 맞을 확률과 식이 보충제를 섭취하지 않는다고 예상하고 그것이 맞을 확률을 결합한 것과 크게 차이가 없다. 즉 모델이 유의미한 성능을 가졌다고 말할 수 없다. 앞서 실행한 전처리 방법과 분석 방법의 한계점을 파악하여 모델 성능 개선이 필요하다.

(1) 전처리 방법의 한계점

파레토 법칙에 기반해 Nan값을 처리하는 전 단계에서 타겟 변수와 상관없을 것 같은 속성들을 제거할 때 분석가의 주관이 들어갔다는 점이다. 하지만 주관에 따라 제거한 속성에는 근거가 있다. 예를 들어 전처리 결과에서 factor level이 너무 많은 속성들은 모델의 성능에 악영향을 끼칠 수 있기 때문에 제거해주었다. 또한 데이터 속성 중에서 주관식 문항들은 분석 목적에 맞지 않으므로 제거해주었다. ‘소아청소년관련 질환응답자’문항은 만 18세 이하만 응답하는 문항이므로 제거해주었다.

파레토 기법을 이용하여 전처리를 실행하면 NA 값의 개수를 기준으로 속성을 지우기 때문에 NA을 가지더라도 분석에 있어서 좋은 속성을 지우게 된다. 때로는 행을 많이 제거하더라도 속성을 남겨둬야 하는 경우가 있다.

(2) 연관 분석의 한계점

연관 분석을 위해서는 데이터를 거래 데이터 형식으로 바꿔 줘야 한다. 이 부분에서 연속형 데이터를 0, 1 형태로 바꿔야 하기 때문에 DT(의사 결정 나무)에 있어서는 데이터가 더 나은 형태가 되지만 object가 가진 값의 특성을 세세히 나타내진 못한다. 예를 들어 소득이 세세하게 나눠져 있음에도 중앙값을 기준으로 0, 1 값을 할당한다고 하자. 중앙값 근처에 속하는 object들은 값의 큰 차이가 없음에도 부여 받는 값은 0, 1 완전히 구분된다. 이 때문에 분석 결과에 오차가 발생할 가능성이 존재한다.

국민 건강영양조사의 원시자료는 속성이 많다. 연관 분석에 있어서 속성이 많으면 계산 과정이 많이 필요하다. 이 계산 시간을 최대한 단축시키기 위해 기준을 가지고 속성들을 제거해줬다. 이 과정에서 자칫 연관분석에 있어서 좋은 결과를 낼 수 있는 속성을 제거할 위험이 있다.

‘조건 -> 반응’ 으로 표현되는 연관성 분석의 표현에서 ‘반응’에 해당 하는 부분에 target\_ingest를 넣어주지 않으면 유용한 분석 결과를 도출하는 것이 쉽지 않다. 예를 들어 반응 부분을 제한해주지 않는 경우 점심 식사 여부-> 저녁 식사 여부 와 같은 식으로 결과가 도출되는데 이는 전혀 유의미한 insight를 제공해주지 못한다. 나이가 많고 소득이 많으면 영양제를 복용할 것이라는 예측 하에서 분석을 진행하여 나이 -> 소득 의미 있는 신뢰도와 지지도를 가진 관계를 발견했다 하더라도 이가 영양제 복용과 연관이 없다면 분석 목적에 맞는 결과를 얻었다고 할 수 없다.

(3)군집 분석의 한계점

군집분석을 시행하기 위해서는 분석 시행자가 적정한 군집의 개수처럼 몇 가지 매개변수를 명시적으로 지정해 주어야 한다. 분석 자체적으로 군집을 분할하는 매커니즘이 미흡하고, 분석의 목적에 따라 휴리스틱한 관점을 합쳐서 군집의 개수 및 초기 centroid를 정해줘야 이 군집 분석의 한계점 중 하나이다.

또한 유형 별 데이터의 분산이 비슷하고, 구형으로 분포되어 있는 경우가 아니라면, 부적절한 결과가 도출될 수 있다. 우리의 데이터 셋 역시 대다수의 정규화 된 변수들 사이에서 상관관계가 높게 나타나기 때문에 적합하지 않은 데이터 셋이었다. 이에 따라 우리는 각 변수들을 PCA의 두 차원으로 분리하여 상관계수를 낮춘 상태에서 진행하였다.

군집 간의 경계가 모호하다면, 그 분석이 무의미해질 수 있다는 한계가 있다. 애초에 군집이 형성되지 않을 정도의 데이터 셋에서 억지로 군집을 형성하려 한다면, 군집이 일그러질(distorted) 수 있고, 그렇지 않다면 군집이 의미가 옅을 것이다. 즉, 군집이 그저 가상으로 형성되었을 뿐이다. 따라서 군집 분석에 앞서서 데이터셋의 군집 가능성을 여러 확률적인 검증이나 판단 기준을 통해서 얻고자 하는 군집이 형성될만한 지에 대한 사전 지식을 얻고 시작해야 한다.

(4) 의사결정나무의 한계점

앞서 만든 의사결정나무는 아직 일부 한계점이 존재한다. 가장 큰 한계점은 분석의 정확도가 66.38%로 아직 완전한 분류 모델이 아니라는 점이다. 여기에는 다양한 이유가 존재한다. 먼저 데이터가 특정 변수에 대해 비슷한 수준의 information gain을 가질 때, 약간의 다른 변수가 선택되면 이후의 트리 구성이 크게 달라질 수도 있기 때문이다. 또한 데이터 수에 약간의 변화만 가해져도 전혀 다른 트리가 생성되는 등 다른 결과가 나올 수 있었다. 또한 결정 트리는 Greedy 방식을 사용하기에, 일반적인 Greedy 방식처럼 항상 최적해를 보장하는 것이 아니기 때문이기도 하다. 그리고 데이터의 특성이 특정 변수에 대해 수직 또는 수평적으로 구분되지 못할 때, 정확도가 떨어지고 트리가 복잡해지는 문제가 발생하였다.

의사결정나무는 정확도 이외에도, 한 번에 하나의 변수를 처리해야만 하기에 변수 간의 상호작용을 Capture하는 데에 한계가 존재했다. 예를 들어 국민건강영양조사 Data에서 개인과 개인 간의 가족 관계를 보여주는 fam\_rela라는 attribute 등이 의사결정나무 안에서는 제대로 반영되지 않았다. 이 부분은 향후 연관성을 보이는 변수를 적절히 결합 또는 변환하거나, 새로운 모델 구현 방식을 통해 해결해야 할 것이다.

(5) 랜덤 포레스트의 한계점

RandomForest모델은 앙상블 기법을 통해 Decision Tree보다 더 나은 성능을 제공해준다. 하지만 보다 정확한 예측을 위해서는 많은 수의 트리를 필요로 하고 생성하게 된다. 이에 의한 RandomForest의 한계점은 알고리즘을 느리고 효과적이지 못하게 만들 수 있다는 것이다. 실시간으로 모델을 사용해야하는 상황에서는 Runtime이 매우 느리므로 사용하기 어려운 점이 있다. 게다가 모델의 하이퍼 파라미터를 고정해야만 동일한 결과를 얻을 수 있다. 텍스트와 같은 차원이 높은 데이터는 처리할 수 없어서 전처리를 요구하게 된다. 또한 Random Forest는 예측 모델링 도구이지만 설명도구는 아니다. 데이터를 자세히 추정할 수가 없어서 모델의 설명력이 떨어진다. 데이터에서 관계에 대한 설명을 하기 위해서는 다른 접근 방식을 이용해야 할 것이다. 결과적으로, 프로젝트 제한조건에 의해 이 모델을 실제 사용하기보다는 Decision Tree를 더 개선할 수 있을지에 대해 intuition을 얻는 용도로 사용해야 할 것이다.

참고문헌

# 1) 강민지. 한국인의 주요 식이보충제 영양소 데이터베이스 구축 및 식이보충제를 포함한 영양소 섭취 평가. Diss. 서울대학교 대학원, 2015.

# 2) 김미정, and 이경혜. "중년 여성의 식이보충제 섭취 수준에 따른 식습관 및 갱년기 증상에 관한 연구." 한국식품영양과학회지 42.7 (2013): 1054-1064.

# 3) <http://www.consumernews.co.kr/news/articleView.html?idxno=602686>

# 4) 김은미, et al. "초등학생들의 일반적 특성에 따른 영양보충제 섭취실태 분석." 한국식품조리과학회지 24.5 (2008): 672-681.

# 5) 김정남. (2008). 한국 비타민 영양제의 40년 리더 유한양행 삐콤씨. 마케팅, 42(1), 79-82.

# 6) 김진숙, et al. "50 세 이상 성인 및 노인의 특수영양 및 건강보조식품의 섭취 실태 (A Study on Supplements Use in the Middle-Aged and Elderly)." 대한지역사회영양학회지 6.5 (2001): 798-808.

# 7) 유양자, 홍완수, and 최영심. "서울지역 성인들의 비타민ㆍ무기질 보충제 섭취 실태에 관한 연구." *한국식품영양과학회지* 30.2 (2001): 357-363.

# 8) 이여옥, and 송윤주. "우리나라 성인의 비타민. 무기질 보충제 섭취여부에 따른 인구. 사회학적, 생활습관 및 영양섭취상태에 관한 연구." 韓國食生活文化學會誌 25.4 (2010): 480-486.

# 9) 이윤정, et al. "한국 성인의 성별에 따른 식이보충제 섭취 및 관련 요인에 대한 연구: 제 4 기 국민건강영양조사 (2007-2009) 자료를 이용하여." 대한지역사회영양학회지 22.4 (2017): 347-355.

# 10) 이주은. "체육교육전공 대학생들의 단백질 보충제 섭취실태." 한국식품영양과학회지 43.10 (2014): 1607-1613.

# 11) 이지연, 김도희, 이윤나, 고은미, 장영수, 이혜영, ... & 김초일. (2011). 우리나라 어린이 및 청소년의 식이보충제 섭취에 영향을 미치는 요인. 대한지역사회영양학회지, 16(6), 740-750.

# 12) 이현숙, 한지혜, and 김선효. "한국 일부 청소년의 건강기능식품용 비타민· 무기질 보충제 섭취 실태 및 관련 요인 조사." 韓國食生活文化學會誌 28.4 (2013): 415-423.

# 13) 이훈호 外 , (2009), 한국인은 어떤 식이보충제를 복용하고 있는가? -2005년 국민건강영양조사를 중심으로- ,가정의학회지 Vol. 30 No. 12, 936 – 952

# 14) 장혜순, and 김미라. "전북 일부지역 중년의 건강 자가인식도와 식생활 및 질병 관련인자가 건강식품 섭취에 미치는 영향 (Effect of Self-Perception of Health and Related Factors of Food Life and Disease on Health Foods Intakes among the Middle Aged in the Jeonbuk Region)." *대한지역사회영양학회지* 6.5 (2001): 744-754.

# 15) 전려홍, and 서미혜. "건강관련 대인커뮤니케이션과 경제수준이 신체적 정신적 건강증진행위에 미치는 영향: 객관적 및 주관적 경제수준을 중심으로." *한국언론정보학보* 91 (2018): 288-318.

########################################################

#데이터를 transaction 형태로 변환 위한 전처리

rm(list = ls())

new <- read.csv("newnew.csv")

new <- new %>% select(-X, -town\_t, -ho\_incm5, -cfam, -allownc, -house, -live\_t,-marri\_1, -tins, -npins, -BD1, -BD7\_5,-BA2\_22, -BP6\_2 , -BP6\_31, -BP7, -mh\_stress, -BS9\_2, -BS13, -BS12\_1, -sm\_presnt, -HE\_wt, -HE\_ht, -BM1\_0, -BM2\_4, - BM2\_5, -BM13, -BM8, -BM14, -T\_ex, -L\_BR\_TO, -L\_LN\_TO, -L\_DN\_TO, -LK\_EDU, -LK\_LB\_CO, -N\_DUSUAL, -make\_energy\_series, -fat\_series, -vitamin\_A\_series, -fatty\_acid\_series,-LF\_BUYER, -LK\_LB\_IT, -LQ2\_ab )

new$earn\_month <- ifelse(new$earn\_month < 410.1667, 0, 1)

new$genertn <- ifelse(new$genertn %in% c(4,5),1 , 0) #미혼자녀 + 부모(편부모) : 1, 그 외 0

new$BO1 <- ifelse(new$BO1 %in% c(4, 5),1 , 0) # 자기체형을 뚱뚱하다고 인식 1, 마르다고 인식 0

new$BO1\_1 <- ifelse(new$BO1\_1 %in% c(2, 3),1 , 0) # 1년간 체형변화 있음:1, 없음:0

new$BO2\_1 <- ifelse(new$BO2\_1 %in% c(1, 2, 3),1 , 0) # 체중변화 시도:1, 없음:0

new$BP1 <- ifelse(new$BP1 %in% c(1, 2),1 , 0) # 스트레스 많음:1, 스트레스 조금:0

new$smoking <- ifelse(new$smoking %in% c(1, 2),1 , 0) # 흡연 5갑이상:1, 흡연 경험 5값 이하:0

new$HE\_BMI <- ifelse(new$HE\_BMI < 23.86631, 0, 1) # bmi 중간아래 : 0, bmi 중간값 이상:1

new$BM7 <- ifelse(new$BM7 %in% c(1, 2, 3),1 , 0) # 씹기문제있음:1, 문제 없음:0

new$L\_BR\_FQ <- ifelse(new$L\_BR\_FQ %in% c(1),1 , 0) # 주 아침식사 5회이상:1, 문제 없음:0

new$L\_LN\_FQ <- ifelse(new$L\_LN\_FQ %in% c(1),1 , 0) # 주 점심식사 5회이상:1, 문제 없음:0

new$L\_DN\_FQ <- ifelse(new$L\_DN\_FQ %in% c(1),1 , 0) # 주 저녁식사 5회이상:1, 문제 없음:0

new$L\_OUT\_FQ <- ifelse(new$L\_OUT\_FQ %in% c(1,2,3,4),1 , 0) # 주 3,4회 이상 외식:1, 주1,2회 이하:0

new$target\_ingest <- ifelse(new$target\_ingest == "NO",0 , 1) # 타겟인제스트 복용:1, 비복용:0

new$N\_DIET <- ifelse(new$N\_DIET %in% c(2),0 , 1) # 식단조절함:1, 조절안함:0

new$LF\_SAFE<- ifelse(new$LF\_SAFE %in% c(1),1 , 0) # 식생활 좋음:1, 안좋음:0

new$region <- ifelse(new$region %in% c(1,4,6,8,9,11,12),1 , 0) #윗지방 : 1, 아랫지방:0

new$age <- ifelse(new$age > 53 ,1 , 0) #평균이상나이 : 1, 평균이하나이:0

new$sex < - ifelse(new$sex == "2" ,0 , 1) #남자 : 1, 여자:0

new

df <- data.frame(ID=integer(), Item=character()) # 데이터프레임 껍데기 생성

table(df)

col <- length(colnames(new)) # test의 열 수

row <- length(row.names(new)) # test의 행 수

col\_names <- colnames(new)

# test[1,2] 1행 2열 값 추출

new

count <- 1

for(i in 1:row){

for(j in 1:col){

if(new[i,j] == 1){ # 만약 값이 1이면 df에 (id, input\_colnames)삽입

input\_colnames <- col\_names[j] # 대입하는 속성 이름

newrow = data.frame(ID=count, Item=input\_colnames) # 아래와 함께 행 결합

df <- rbind(df, newrow)

}

}

count = count + 1

}

df

df$Item <- as.character(df$Item) # 형변환

str(df)

# id에 대한 구매 데이터 완료

# 진짜 거래 데이터로 변환

# 먼저 id별 구매 데이터 분리

library("arules")

df.list <- split(df$Item, df$ID)

df.list

# 거래 데이터로 변환

df.trans <- as(df.list, "transactions")

df.trans

image(df.trans)

summary(df.trans)

df.rules <- apriori(df.trans)

summary(df.rules)

rules <- apriori(df.trans, parameter=list(support=0.1, confidence=0.9,

target="rules"))

summary(rules)

inspect(rules)

######### 타켓값 지정 가능, 원하지 않는 At 제거 가능############

rule\_s <- apriori(df.trans, parameter=list(support=0.1,confidence=0.6),

appearance=list(rhs="target\_ingest",default='lhs', none=c("L\_LN\_FQ","L\_DN\_FQ")))

inspect(rule\_s)

##################

install.packages("arulesViz")

library(arulesViz)

plot(rule\_s)

plot(rule\_s, method = "paracoord", control = list(reorder = TRUE))

plot(rule\_s, method = "graph", control = list(type = "items"))

plot(rule\_s, method = "graph")

plot(rule\_s, method = "grouped")

################################################################3

# 군집 분석

## 연속형 변수 추출 및 정규화

start <- read.csv("newnew.csv")

k <- start %>% select(earn\_month, age, make\_energy\_series,HE\_ht,HE\_wt)

g <- k %>% head(100)

gg <- scale(g)

k <- scale(k)

## Distance measures 분석

res.dist <- get\_dist(k, stand = TRUE, method = "pearson")

fviz\_dist(res.dist,

gradient = list(low = "#00AFBB", mid = "white", high = "#FC4E07"))

## Partitioning clustering 최적의 클러스터링 개수 파악

fviz\_nbclust(k, kmeans, method = "gap\_stat")

## 클러스터링 계산

set.seed(123)

km.res <- kmeans(k, 4, nstart = 25)

# Visualize

fviz\_cluster(km.res, data = k,

ellipse.type = "convex",

palette = "jco",

ggtheme = theme\_minimal())

##요인분석

pca.res<-PCA(k, graph=FALSE)

fviz\_contrib(pca.res, choice="var", axes=1, top=4)

fviz\_contrib(pca.res, choice="var", axes=2, top=4)

raw<-read.csv("newnew.csv")

x <- raw$make\_energy\_series

y<- raw$age

xy <- data.frame(cbind(x,y))

plot(xy, pch = 19, xlab = c("x coordinate"), ylab = c("y coordinate"),

xlim = c(-5, 5), ylim = c(20,80),

main = "scatter plot of xy")

# adding student label

text(xy[,1], xy[,2], labels = abbreviate(rownames(xy)),

+ cex = 0.8, pos = 1, col = "blue") # pos=1 : at the bottom

# adding dotted line

abline(v=c(3), col = "gray", lty = 2) # vertical line

abline(h=c(3), col = "gray", lty = 2) # horizontal line

hc\_cent <- hclust(dist(xy)^2, method="centroid")

hc\_cent

my\_par = par(no.readonly = TRUE)

par(oma = c(0, 0, 1, 0))

par(mfrow = c(1, 2))

plot(hc\_cent)

plot(hc\_cent, hang = -1) # hang = -1 : line from the bottom

#######################################################################

# 정규분포 파괴된 데이터 생성

str(raw)

age<-raw$age

age

hist(age, breaks=10) # 모든 나이 100명씩 샘플링 하면 된다...

# 방법 - 1. filter를 써서 20~30 / 30 ~ 40 / 40 ~ 50 / 50 ~ 60 / 60 ~ 70 / 70 ~ 80 를 나눈다

# 2. 각각에서 100개씩 샘플링 한다

# 3. 샘플링 한걸 합친다

# 4. 합친걸로 돌려본다

two\_3<- raw %>% filter(age >= 20 & age < 30) # 20 ~ 30

three\_4<- raw %>% filter(age >= 30 & age < 40) # 30 ~ 40

four\_5<- raw %>% filter(age >= 40 & age < 50) # 40 ~ 50

five\_6<- raw %>% filter(age >= 50 & age < 60) # 50 ~ 60

six\_7<- raw %>% filter(age >= 60 & age < 70) # 60 ~ 70

seven\_8<- raw %>% filter(age >= 70 & age < 80) # 70 ~ 80

# 랜덤 샘플링

two3 <- two\_3[sample(nrow(two\_3), 60),]

three4 <- three\_4[sample(nrow(three\_4), 60),]

four5 <- four\_5[sample(nrow(four\_5),60),]

five6 <- five\_6[sample(nrow(five\_6), 60),]

six7 <- six\_7[sample(nrow(six\_7), 60),]

seven8 <- seven\_8[sample(nrow(seven\_8), 60),]

random <- rbind(two3, three4, four5, five6, six7, seven8)

random

# 군집분석 실시

kmeans()

{

require(graphics)x<-cbind(datas2$age,datas2$HE\_BMI,datas2$earn\_month)

colnames(x) <- c("x", "y")

(cl <- kmeans(x, 2))

plot(x, col = cl$cluster)

points(cl$centers, col = 1:2, pch = 8, cex = 2)

# sum of squares

ss <- function(x) sum(scale(x, scale = FALSE)^2)

fitted.x <- fitted(cl); head(fitted.x)

resid.x <- x - fitted(cl)

cbind(cl[c("betweenss", "tot.withinss", "totss")], # the same two columns

c(ss(fitted.x), ss(resid.x), ss(x)))

stopifnot(all.equal(cl$ totss, ss(x)),

all.equal(cl$ tot.withinss, ss(resid.x)),

all.equal(cl$ betweenss, ss(fitted.x)),

all.equal(cl$ betweenss, cl$totss - cl$tot.withinss),

all.equal(ss(x), ss(fitted.x) + ss(resid.x))

)

kmeans(x,1)$withinss # trivial one-cluster, (its W.SS == ss(x))

(cl <- kmeans(x, 3, nstart = 16))

plot(x, col = cl$cluster)

points(cl$centers, col = 1:5, pch = 8)

# }

#install.packages("scatterplot3d")

library(scatterplot3d)

scatterplot3d(x=datas2$age,y=datas2$HE\_BMI,z=datas2$earn\_month,pch=20,color=rainbow(3)[cl$cluster])

# Centroid 개선 하기

#Centroid 함수 - 초기 Centroid의 x, y좌표값을 sample()을 통해 random하게 받음

centroid\_x=sample(x=min(x[1]):max(x[1]),size=2) #random값의 x값이 data의 최소값과 최대값 안에 오게끔 함

centroid\_x[2]

centroid\_y=sample(x=min(x[2]):max(x[2]),size=2)#random값의 y값이 data의 최소값과 최대값 안에 오게끔 함

centroid\_y[1]

centroid<-function(x){#함수의 선언

mat<-matrix(,nrow=2,ncol=3007)#centroid와 Euclidean distance를 계산하고, 비교한 값을 저장하는 matrix를 선언

#이 때 3007은 data set의 개수, nrow=2는 축의 수로 분석 목적에 따라 조정 가능함

for(i in 1:3007){#for문을 통해 모든 data가 해당 과정을 반복

if( ((x[i,1]-centroid\_x[1])^2+(x[i,2]-centroid\_y[1])^2) > ((x[i,1]-centroid\_x[2])^2+(x[i,2]-centroid\_y[2])^2) ){

#if()문을 통해 데이터와 두 Centroid 간의 Euclidean distance를 서로 비교 수행

mat[1,i]<-0#2번째 센트로이드와 거리가 더 가까울 경우

mat[2,i]<-1#행렬의 1행에 0을, 2행에 1을 대입.

}

else{#반대의 경우도 마찬가지로 동작

mat[1,i]<-1

mat[2,i]<-0

}

}

centroid\_x[1]=0 #Update에 앞서 값을 초기화

centroid\_x[2]=0

centroid\_y[1]=0

centroid\_y[2]=0

count1=0

count2=0

sum1\_x=0

sum1\_y=0

sum2\_x=0

sum2\_y=0

for(i in 1:3007){#데이터의 수만큼 반복하여 Centroid를 update

if(mat[1,i]==0){

sum2\_x=sum2\_x+x[i,1]

sum2\_y=sum2\_y+x[i,2]

count2=count2+1

}

else{

sum1\_x=sum1\_x+x[i,1]

sum1\_y=sum1\_y+x[i,2]

count1=count1+1

}

}

centroid\_x[1]<-sum1\_x/count1#합을 Count로 나누어 최종적으로 centroid를 업데이트

centroid\_x[2]<-sum2\_x/count2

centroid\_y[1]<-sum1\_y/count1

centroid\_y[2]<-sum2\_y/count2

print(centroid\_x[1])

print(centroid\_x[2])

print(centroid\_y[1])

print(centroid\_y[2])

}