2018학년도
석사학위논문

유사도 알고리즘을 활용한
와인 추천 알고리즘 개발 연구

- 빅데이터 분석 기법 중심으로 -

Development of wine recommendation algorithm using similarity algorithm
-Focus on Bigdata analysis techniques-

남서울대학교 (복지경영)대학원

빅데이터전문가학과 빅데이터전문가학전공

조 준 호

2018년 6월

유사도 알고리즘을 활용한
와인 추천 알고리즘 개발 연구

- 빅데이터 분석 기법 중심으로 -

Development of wine recommendation
algorithm using similarity algorithm
-Focus on Bigdata analysis techniques-

지도교수 김 정 범

이 논문을 석사학위논문으로 제출함

2018년 6월

남서울대학교 (복지경영)대학원

빅데이터전문가학과 빅데이터전문가학전공

조 준 호

조준호의 석사학위논문으로 인준함

심사위원장 _____ ㊞

심 사 위 원 _____ ㊞

심 사 위 원 _____ ㊞

남서울대학교 (복지경영)대학원

2018년 6월

목 차

표 목 차

그 림 목 차

Ⅰ. 서 론

최근, 소비자들의 주류에 관한 선호도가 낮은 도수의 주류 및 건강 관련 쪽으로 변해가면서 다양한 도수의 주류와 소비자 기호에 맞춘 칵테일 소주 등이 출시되고 있다.

이러한 영향을 받아 와인 역시 그 소비량이 증가하고 있는데, 한국무역협회에 따르면 국내 와인 수입량은 2015년 36,815톤으로 2013년 32,557톤보다 4,258톤이 증가하면서 지속적인 성장세를 보이고 있다. 특히 최근 한·칠레, 한·미, 한·EU FTA 등의 와인 주요 수출국과의 FTA를 통해 국내 와인 시장의 수입 다변화를 성공시키면서 저가에서부터 고가에 이르기까지 상당히 많은 수의 와인 브랜드가 국내에 자리매김하고 있으며, 그 수입처만 약 35곳, 수입하는 와인 종류만 17,000여 가지에 달한다.[8]

이로 인해 기존의 와인 구매 시 고려 대상이었던 품종과 브랜드, 빈티지뿐만 아니라 가성비 좋은 와인, 브랜드 와인과 유사한 맛의 와인 등 소비자는 자신의 라이프 스타일에 맞춰 와인을 구매하고자 할 때 고려하는 기준에도 많은 변화가 나타났다.

그러나 소비자가 접하는 와인에 대한 정보는 여전히 부족한 실정이며, 다른 주류에 비해 상대적으로 복잡하고 까다로운 전문지식이 필요하다.[7]

예를 들어 와인의 생산지역, 생산국 등에 따라 다양한 종류의 와인으로 나뉠 뿐 아니라 포도의 품종에 따라서 와인의 맛과 향에서 상당한 차이를 보이는데, 와인에 대한 깊은 지식이 없는 와인 초보자와 입문자가 이러한 부가적인 정보들을 이해하고 와인을 선택하기 쉽지 않기 때문에 본인이 마셔봤던 와인 중에 본인의 입맛에 맞는 와인 또는 유사한 맛의 와인을 찾아내기란 어려운 일이다.

따라서 이러한 와인 초보자와 입문자가 마트 또는 와인 샵에서 판매 중인 많은 종류의 와인 중에서, 내 입맛에 맞는 와인을 찾아낼 수 있도록 본인이 마셔 본 와인과 유사한 와인을 추천해 줌으로써 자신의 입맛에 잘 맞는 와인을 선택하는데 도움을 주는 것을 이번 연구의 목적으로 하고 있다.

Ⅱ. 추천 시스템

1. 추천시스템이란

우리는 매일 같이 새로 출시되는 상품, 새로 만들어지는 영화, 새로 창조되는 음식 등 상당히 많은 재화 속에서 살고 있다.

새로운 재화들이 쏟아져 나올 때 마다, 우리는 제품을 구매하기 전 샘플을 구매 해 보거나 광고 또는 블로그, SNS등을 활용하여 사용 후기 등의 의견을 묻는 등 금전적 시간적 비용을 들이게 된다.

하지만 이러한 고민을 조금이나마 해결 해 줄 수 있는 방법이 바로 추천 시스템이다. 추천 시스템이나 추천 알고리즘은 이러한 과정을 자동적으로 처리하여 소비자가 원하는 아이템과 정보를 추천해줌으로써 광고와 블로그 등을 클릭해야 하는 수고로움과 시간적 비용을 줄일 수 있도록 해준다.

2. 추천시스템 종류

추천 시스템에는 여러 가지 방법이 존재하며 이러한 추천 기법은 소비자의 소비 유형에 따라 적절한 방법을 적용해야 한다. 소비자의 소비 성향은 크게 개인적인 소비 성향과 그룹 소비 성향으로 나눌 수 있으며 개인적인 소비 성향의 소비자는 개인의 성향 즉, 성격 및 취향이 구매에 큰 영향을 미치게 된다.

추천 시스템의 연구는 주로 개인적인 소비 성향에 초점을 맞춰 연구되고 있으며, 이러한 추천 시스템은 협업 필터링(Collaborative Filtering) 추천 시스템, 콘텐츠 기반(Content-based) 추천 시스템, 지식 기반(Knowledge-based) 추천 시스템 그리고 하이브리드(Hybrid

System) 추천 시스템으로 나뉜다. 이 중에서 협업 필터링 추천 시스템과 콘텐츠 기반 추천 시스템이 가장 보편적으로 활용되고 있다.

1) 협업 필터링 추천 시스템

협업 필터링 추천 기법은 소비자의 과거 구매 성향이 지속적으로 변하지 않을 것을 가정한 후, 소비자의 선호도를 분석하여 비슷한 유형을 가진 고객들에게 상품을 추천하는 시스템이다.

해당 방법은 가장 대중적인 추천시스템 기법으로 사용자 기반 협업적 필터링(UBCF : User-Based Collaborative Filtering)과 아이템 기반의 협업적 필터링(IBCF : Item-Based Collaborative Filtering)으로 다시 나뉘며,[8] 최근 연구된 다양한 추천 시스템 기법들 중에서 정확도가 높은 편이다. 하지만 고객과 상품의 특성을 무시하고 오직 소비자의 선호도만을 고려하기 때문에 정확도 높은 결과를 얻기 위해서는 많은 데이터가 필요하며, 새로운 소비자 또는 새로운 아이템에 대해서 추천하고자 할 때 제약 사항이 존재한다는 점이 한계점으로 작용한다.[11]

사용자 기반 협업 필터링 추천 시스템(UBCF)은 소비자 간의 선호도가 얼마나 유사한가를 기반으로 예측하는 방법으로 특정 아이템에 대해 새로운 소비자가 생기면 해당 소비자와 유사한 소비 성향을 갖는 소비자를 선별한 후, 그 유사 소비자가 구매하여 좋은 평점을 준 아이템을 추천해주는 방식이다.

해당 필터링에 적용되는 알고리즘은 클러스터링, K-NN(K-최근접 이웃), 베이지안 등이 이용되며, 유사도 점수를 주는 방식으로는 피어슨 또는 스피어만 상관계수, 벡터 유사도가 사용된다. 이러한 사용자 기반 협업적 필터링의 경우 소비자 간의 유사성을 반영하고 있기 때문에 예측 정확도가 높으나, 해당 알고리즘을 적용하기 위해선 둘 이

상의 소비자가 평점 준 특정 아이템이 존재해야 한다는 단점이 있다. 따라서 새로운 사용자가 아직 아이템을 구매하지 않았거나, 새로운 아이템에 대해 어느 누구도 그 아이템을 구매하지 않았다면 해당 알고리즘을 적용할 수 없다.

아이템 기반 협업 필터링 추천 시스템(IBCF)은 아마존이 소비자에게 아이템을 추천할 때 사용하고 있는 방식으로 대부분의 소비자는 과거에 자신이 좋아했던 아이템과 유사한 성향의 아이템을 선호하며, 자신이 싫어했던 아이템과 비슷한 성향의 아이템을 싫어한다는 심리에 주안점을 두고 있다.[11]

따라서 해당 알고리즘은 아이템 간의 유사성에 중심을 두고 있으며 소비자에게서 비슷한 평점을 받은 각각의 아이템들에 대해 얼마나 유사한 지를 측정하여 특정 소비자가 어떤 아이템을 선호 할지를 예측하는 방식이다. 하지만 해당 알고리즘은 소비자 간의 유사도를 전혀 고려하지 않았다는 점에서 특정 소비자의 소비 성향이 전혀 다른 경우 아이템 간의 추천 정확도가 떨어진다는 문제점이 존재한다.

2) 콘텐츠 기반 추천 시스템

콘텐츠 기반의 추천 시스템은 소비자가 평가한 아이템의 세부 속성을 활용하여 아이템과 소비자 간의 유사도를 측정해 소비자에게 추천하는 방식으로, K-NN(K-최근접 이웃)방식을 통해 유사한 소비자와 아이템을 찾는 방식의 협업 필터링 추천 시스템과는 다르게 소비자가 원하는 아이템의 속성 간의 유사도를 통해 추천한다. 예를 들어 콘텐츠 기반의 영화 추천 시스템에서 영화의 속성은 배우, 영화 장르, 감독, 제작사 등이 있다. 이때 사용자 자신이 선호하는 영화를 입력하면, 해당 시스템은 그 영화가 지닌 속성 값의 유사도를 'Bag of the Words Navie Bayesian Test Classifier' 방식을 이용하여 선호도를

파악하고 유사한 속성 값을 갖는 다른 영화를 추천해준다.[12]

이처럼 콘텐츠 기반의 추천 시스템은 해당 사용자의 선호도만을 고려하며, 사용자 주변의 다른 사용자들이 갖는 선호도는 고려하지 않기 때문에 다른 사용자의 특성을 구하기가 어렵다는 단점이 존재한다. 그러므로 추천 대상 아이템이 자주 변경되지 않을 때 사용하기에 적합한 시스템이다.[2]

3. 유사도 계산

추천 시스템은 소비자와 소비자, 아이템과 아이템, 소비자와 아이템 간의 유사도를 점수화 하여 그 값을 고려하여 작동한다.

이러한 추천 시스템의 근간이 되는 유사도 측정 방법에는 유클리디안, 맨하튼 거리, 피어슨 상관계수, 코사인 거리, 타니모토(자카드)거리 측정법 등이 있으며 가장 보편적으로 유클리디안 거리와 코사인 거리, 피어슨 상관계수, 타니모토(자카드) 거리 측정법이 활용되고 있다.

유클리디안 거리는 소비자 또는 아이템 간의 유사도를 계산하는 가장 쉬운 방법으로 가장 직관적이고 직관적인 거리의 개념이다. 일반적으로 2차원 상에서 두 점간의 거리는 피타고라스 정리에 따라 계산된다. 하지만 n차원의 공간에서 두 점간의 거리를 구하기 위해 피타고라스의 정리를 조금 더 확장시킨 것이 유클리디안 거리로 해당 수식은 [수식 1]과 같다.

$$EuclideanD(x,y) = \sqrt{(x_1-y_1)^2+(x_2-y_2)^2+\cdots+(x_n-y_n)^2} = \sqrt{\sum_{i=1}^{n}(x_i-y_i)^2} \quad \text{[수식 1]}$$

여기서 (x, y)는 두 개의 연속적인 데이터이며, n은 데이터 세트의 자료의 개수를 의미한다.

[수식 1]을 통해 구해진 유클리디안 거리는 거리의 최댓값이 존재하지 않아 해당 거리를 비교할 수가 없다. 따라서 해당 거리 법을 쓰기 위해선 0과 1사이의 값으로 데이터의 정규화가 우선적으로 이루어져야 하며, 정규화 된 거리에서 두 벡터(Vector)가 가까울수록 0에 가깝고 멀수록 1에 가까워지게 된다.[14]

하지만 유클리디안 거리를 통한 유사도 계산 방식은 두 벡터간의 단순한 거리를 계산한다는 점에서 해당 벡터가 같은 방향성을 지니고 있는지를 확인할 수가 없다. 따라서 해당 유사도를 활용할 경우 두 벡터 간의 유클리디안 거리가 같다면 다른 방향성을 갖더라도 유사한 정도가 큰 것으로 나타날 수 있다는 한계점이 존재한다.

코사인 유사도는 내적공간에서 두 벡터 간의 각도를 코사인(Cosine)방식을 이용하여 측정한 값이다. 두 벡터 간의 각도가 0°로 그 방향이 완전하게 같다면 코사인 값은 1, 90°의 각도로 서로 관계가 없다면 0, 180°로 두 벡터간의 방향이 완전히 반대일 때는 -1과 같이 -1과 1의 사이 값을 갖게 되는데, 이때 코사인 유사도의 결과 값은 0과 1 사이의 양수 공간에서 표현되며 이 값은 벡터의 크기가 아닌 두 벡터간의 유사한 정도를 나타낸다.

코사인 거리의 계산식[수식 2][15]은 다음과 같다.

$$x \cdot y = \| x \| \| y \| \cos\theta \qquad \text{[수식 2]}$$

$$CosSim(x,y) = \frac{x \cdot y}{\| x \| \| y \|} = \frac{\sum_{i=1}^{n} x_i \times y_i}{\sqrt{\sum_{i=1}^{n} (x_i)^2} \times \sqrt{\sum_{i=1}^{n} (y_i)^2}}$$

여러 가지 유사도 측정 방법 중 코사인 유사도는 각도 값을 이용하여 유사한 방향으로 뻗어나가는지를 찾기 때문에 모든 벡터가 양수만을 가지고 있다고 간주하며 0에서 1사이의 값만 추출되고, 이는 정규

화가 되어 있는 것으로 볼 수 있어 데이터를 별도로 정규화 시킬 필요가 없다. 또한 벡터 간의 양적 값을 이용해 거리를 계산하는 유클리디안 거리보다 비슷한 성향의 것을 찾아낼 수 있다는 점에서 유사도 측정에 많이 활용되고 있다.

하지만 A, B, C 벡터간의 방향성이 서로 같아 코사인 거리가 0일 때, A가 B와 C중에 어느 벡터와 근접하고 있는지를 판별할 수는 없다는 한계점이 있다.

피어슨 상관 계수를 통한 유사도는 유클리디안, 코사인 거리 측정 방법과 다르게 두 변수 간의 상관관계를 통해 얻어진다. 따라서 사용자 기반 협업 시스템에서 유사도 계산이 되는 대상은 사용자가 되며, 아이템 기반 협업 시스템에서는 두 개의 아이템이 계산 대상이 된다.

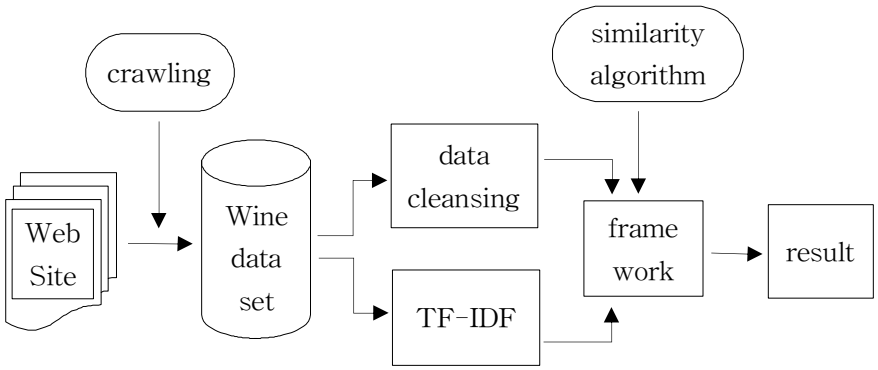
피어슨 상관계수는 두 변수(벡터)간의 공분산 값을 변수들의 표준편차의 곱으로 나눈 값으로 그 공식은 [수식 3]과 같다.[8]

$$PearsonD(r) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \quad \text{[수식 3]}$$

피어슨 상관계수를 통한 유사도의 경우 두 변수간의 상관관계, 즉 두 사용자가 따로 변하는 정도를 두 사용자가 함께 변하는 정도로 나눈 값으로 나타나기 때문에 사용자가 어떠한 아이템에 대해 평점을 높게 측정하여도 해당 점수의 영향을 덜 받으며 해당 값이 1이라면 두 변수가 완전히 동일하다고 보며, 전혀 다르다면 0, 반대방향으로 완전히 동일하면 -1의 값을 갖는다.[16]

Ⅲ. 빅데이터 기반 와인 추천 시스템 구현

본 연구에서 소개하는 빅데이터 기반 와인 추천 시스템은 일반적인 와인 추천시스템에서 생산자, 생산국, 당도, 산도, 바디감, 탄닌과 같이 기본적인 와인선택속성만을 이용하는 것과는 달리 어울리는 음식과 해당 와인의 시음 노트 등 비정형 데이터를 추가적으로 활용하여 추천을 실행한다. 이를 통해 와인의 특성 간의 유사성뿐만 아니라 음식의 유사성과 맛의 유사성 까지 고려하므로 써 소비자에게 가장 만족스러운 맞춤형 추천시스템을 제공해 주는 것을 그 목적으로 하며, [그림 1]과 같은 구조로 빅데이터 기반 와인 추천 시스템이 이루어진다.



[그림 1] 빅데이터 기반 와인 추천 시스템 개념도

1. 와인선택속성에 관한 분류

아이템의 속성이란, 각 아이템이 가지고 있는 고유 또는 다양한 속성 중에서 소비자가 원하는 아이템 기능을 수행하는데 필요로 하는 아이템의 구성요소를 일컫는데, 와인을 선택하는 소비자가 중요하게 여기는 와인의 속성을 와인선택속성이라고 한다.[5]

와인선택속성 크게 두 가지 요인으로 분류 할 수 있으며, 첫 번째

요인은 색상과 포장 용기, 라벨과 같이 주로 와인의 외관상의 분류이고, 두 번째 요인은 와인의 가격, 품질, 종류, 타입, 생산국가와 같이 와인 그 자체에 대한 분류이다.

‘와인선택속성이 고객만족 및 재구매 의도에 미치는 영향’[5]에서 와인 소비자의 와인선택속성을 와인품질, 가격, 국가, 맛, 색깔, 빈티지, 와인종류, 포도품종, 알코올 도수로 구분하였으며, ‘와인선택속성이 만족도 및 재구매의도에 미치는 영향’에서는 국내 와인 소비자에 대한 와인선택속성을 가격, 생산지역, 와이너리, 포도품종, 와인종류, 브랜드, 디자인, 음식과의 조화, 품질, 알코올 도수로 제시하였다.[4] 또한 ‘호텔레스토랑에서 고객의 소비 라이프 스타일에 따른 와인 선택 속성이 고객만족에 미치는 영향’[7]에 관한 연구에서는 와인선택속성에 대한 특성적 요인을 크게 브랜드 및 포도품종, 맛과 향, 와인가격 및 프로모션, 점원의 추천과 와인 색깔 이렇게 4가지로 구분한 후 12가지로 더욱 세분화 시켰다.

해당 연구들을 토대로 하여, 이번 와인 추천시스템 연구에서는 와인의 생산자, 생산국, 종류, 알코올 도수, 당도, 산도, 바디감, 탄닌, 와인의 맛, 음식과의 조화 및 와인의 가격과 함께 해당 와인을 구성하고 있는 포도의 품종을 와인선택속성으로 설정했다. 특히 포도의 품종은 와인에 주로 쓰이는 포도 품종이 존재하기 때문에 해당 품종들을 활용하기 위해서 상위 20개의 포도 품종[1]을 별도로 구분하였다.

2. 와인 및 소비자 데이터 수집

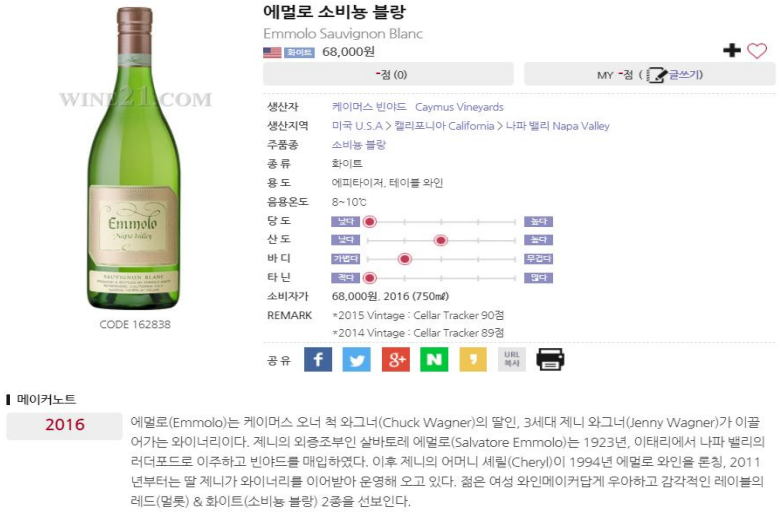
나를 위한, 내 취향에 맞는 와인을 선택하기 위해서 본 연구에서는

1) 20대 와인 품종 : 샤르도네, 리슬링, 쇼비뇽블랑, 게뷔르츠트라미너, 슈냉블랑, 세미용, 뮬러트루가우, 까베르네쇼비뇽, 피노누아, 그르나슈, 메를로, 가메, 시라/쉬라즈, 네비올로, 산지오베세, 템프라니오, 말벡, 진판델, 모스카토, 블랜드

와인21닷컴(www.wine21.com)웹 사이트에서 제공하는 와인에 대한 정보를 수집하여 활용하였다.

와인21닷컴은 1998년에 오픈하여 국내 최대 와인 정보를 보유하고 있는 와인전문 포털 사이트로 각종 와인 정보 및 수입처 정보, 시음회 등의 정보를 제공하고 있으며 와인에 대한 정보는 [그림 2]과 같이 제공하고 있다.[17]

해당 와인을 설명하는 각각의 정보를 살펴보면, 생산자 정보는 해당 와인을 제조하는 와이너리를 나타내며, 생산지역은 생산국과 해당 와인을 생산하는 지역을 나타내고 있다.



[그림 2] 와인21 웹페이지의 와인 정보

주품종의 경우 와인을 제조할 때 쓰인 포도의 품종을 나타내주고 있으며, 해당 와인이 레드, 화이트, 스파클링 또는 로제와 같이 어떠한 종류의 와인에 해당하는지를 표시해주고 있다. 그와 함께 당도, 산도, 바디, 타닌에 대한 정보를 5점 척도로 책정하여 해당 와인의 특성을 나타내주고 있으며 메이커노트라는 와인 테이스팅 노트가 제공되고

있어 앞서 선정하였던 와인선택속성 변수들과 조건이 부합한다.

데이터를 수집하는 방법은 크롤링, ETL, 로그 수집, HTTP, FTP등으로 구분된다. 특히 외부에 있는 반정형 또는 비정형 데이터를 수집해 오기 위해선 HTTP를 사용하여 파일의 텍스트를 스크래핑 한 후 저장된 메타 정보를 이용하여 파일을 파싱해 원하는 데이터를 얻어야 한다. 이때 필요한 HTTP 사용 기술은 웹에서 텍스트 정보를 가져오는 크롤링 수집기술과 웹을 운영하는 운영 주체가 정보를 제공해주는 Open API 수집기술로 구분되어진다.

Open API 수집 기술은 웹을 운영하는 운영 주체가 정보를 제공해주기 때문에 미리 구현된 DB사용으로 쉬운 DB를 구축할 수 있다. 그에 따라 상당히 쉽게 어플리케이션을 구현할 수 있으며, 정보 제공자와 개발자 간의 상호 연결성이 좋다는 장점을 가지고 있다. 하지만 웹을 운영하는 주체가 정보를 제공하다보니 충분치 못한 라이브러리 구성이라는 단점이 존재한다. 또한 최근 무분별한 API 사용을 방지하기 위해 Open API를 제공하는 네이버, 다음, 트위터, 구글 등에서 인증키(API_Key)를 발행하여 일일 수집 가능한 데이터양을 정해 놓고 있다는 단점이 존재한다.

크롤링은 Open API가 제공되지 않는 사이트의 데이터를 가지고 오기 위한 방법으로 웹 페이지를 읽어와 HTML 또는 CSS 등의 방식으로 파싱하고, 그 중에서 필요한 데이터를 텍스트 형태로 추출하는 기법이다. 해당 방법을 사용하는 경우 대부분의 웹페이지 크롤러를 만들 수 있으나, 법적인 문제 발생 시 모든 책임을 져야한다.

해당 페이지의 경우 웹을 운영하는 운영 주체가 Open API를 제공하지 않고 있어 와인의 정보를 수집하기 위해 크롤링이라는 방법론을 활용하였으며, 크롤링 봇[그림 3]의 제작을 통해 와인 정보를 가지고 왔다.

```
wine_total <- NULL
try(
  for(i in 137204:162838){
    url <- paste0("http://www.wine21.com/13_search/wine_view.html?Idx=",i)
    ie <- read_html(url)

    ko_name <- html_nodes(ie, css = '.name_ko')
    head(ko_name)
    ko_name_1 <- str_replace_all(ko_name[] %>% html_text(), '\n', '')
    ko_name_2 <- as.data.frame(ko_name_1)
    names(ko_name_2) <- c("와인이름_한글")
                    .
                    .
    wine_productor <- html_nodes(ie, css = '.wine_info dd')
    head(wine_productor)
    wine_productor_1 <- str_replace_all(wine_productor[1] %>% html_text(), '\n', '')
    wine_productor_2 <- as.data.frame(wine_productor_1)
    wine_productor_2$wine_productor_1 <- gsub("[A-Za-z]", "", wine_productor_2$wine_productor_1)
    wine_productor_2$wine_productor_1 <- gsub("A\\s+|\\s+$", "", wine_productor_2$wine_productor_1)
    names(wine_productor_2) <- c("생산자")
                    .
                    .
    wine_sweet <- html_nodes(ie, css = '.taste_grap img')
    head(wine_sweet)

    wine_sweet_1 <- as.character(wine_sweet)
    condition <- grep("SWEET", wine_sweet_1)

    if(!is.null(condition) && length(condition) >= 1){
      wine_sweet_2 <- wine_sweet_1[grep("SWEET", wine_sweet_1)]
      if(str_detect(wine_sweet_2, "img_taste_1.jpg")){
      wine_sweet_3 <- as.data.frame(1)
      }else if(str_detect(wine_sweet_2, "img_taste_2.jpg")){
```

[그림 3] 와인 정보 수집을 위한 크롤링 봇

크롤링 봇을 이용하여 16,000여개에 해당하는 와인의 정보를 수집하였으며, 수집된 정보는 최종적으로 wine_total이라는 변수로 들어가 xlsx파일로 저장하였다.

웹페이지에서 추출한 데이터는 총 16개 변수들에 대한 데이터이며, [표 1]은 wine_total 데이터의 각 변수들에 대한 설명이다.

[표 1] wine_total 데이터 변수(컬럼) 설명

영문 이름	변수 설명	영문 이름	변수 설명
ko_name	와인 국문명	sweet	당도
en_name	와인 영문명	acidity	산도
productor	생산자	body	바디감
production	생산국/지역	tannin	탄닌
variety	주품종	food	음식
type	와인종류	price	가격
best	와인용도	volume	용량
alcohol	알코올 도수	tasting	시음 노트

저장된 xlsx파일을 로딩하기 위해 데이터가 위치하고 있는 작업 공간을 설정해 준 후 [표 2]와 같이 해당 데이터를 읽어 들인다.

[표 2] wine_total 데이터 로딩

```
library(xlsx)
wine <- read.xlsx("wine_total.xlsx", sheetIndex = 1)
```

xlsx파일을 불러오기 위해선 xlsx 패키지가 필요하기 때문에 해당

패키지를 로딩 후, xlsx파일의 첫 번째 시트에 위치한 wine_total 데이터를 불러오기 위해 sheetIndex 인자 값을 1로 설정하였으며, 해당 데이터를 wine이라는 변수에 저장하였다.

3. 데이터 전처리

1) 와인 데이터 전처리

총 16개의 변수로 이루어진 wine 데이터[그림 4]에는 각각의 와인에 대한 정보를 담고 있다. 1차적으로 크롤링을 통해 데이터를 수집하면서 html 정보를 제거하였으나, 추천 알고리즘을 만들기 위해선 불용어에 대한 정리가 필요하다.

	ko_name	en_name	productor	production	varity
1	트루이젤 리제르바 까베르네 소비뇽	TroisL Reserva Cabernet Sauvignon	콘차이 토로	칠레 >센트럴 밸리	카베르네 소비뇽 100%
2	트루이젤	Troisl	칠레	프랑스>보르도	메를로,카베르네 소비뇽
3	보사렌 소비뇽	Beauchatel Sauvignon	이봉모	프랑스>서던 프랑스>랑그독-루시옹 -	소비뇽 블랑 100%
4	보사렌 메를로	Beauchatel Merlot	이봉모	프랑스>서던 프랑스>랑그독-루시옹 -	메를로 100%
5	보사렌 까베르네 소비뇽	Beauchatel Cabernet Sauvignon	이봉모	프랑스>서던 프랑스>랑그독-루시옹 -	카베르네 소비뇽 100%
6	비르지니 드 발랑드로	Virgine de Valandraud	위네뱅 꼼베<U+00A0><U+00A0><U+00A0> ()	프랑스>보르도>생-테밀리옹 그랑 크뤼 -	메를로 65%, 카베르네 프랑
7	빈도로 프리미티보 디 만두리아	Vindoro Primitivo di Manduria	산 마르짜노<U+00A0><U+00A0><U+00A0> ()	이탈리아>풀리아	프리미티보 100%
8	비나 코보스, 브라마레 레본 말벡	Vina Cobos, Bramare Rebon Malbec	비나 코보스	아르헨티나>멘도사	말벡 100%
9	비나 코보스, 브라마레 마르치오리 빈야드 말벡	VINA COBOS, Bramare Marchiori Vineyard Malbec	비나 코보스	아르헨티나>멘도사	말벡 100%
10	스프링 밸리 빈야드, 울리아 레드 와인	Spring Valley Vineyard, Uriah Red Wine	샹 미셸 와인 에스테이트<U+00A0><U+00A0><U+00...	미국 ..>워싱턴주>왈라 왈라	메를로 45%, 카베르네 프랑
11	트라피체 싱글 빈야드 말벡 콜레토	Trapiche Single Vineyard Malbec 'Coletto'	트라피체	아르헨티나>멘도사	말벡 100%
12	멜로디아스 와인메이커스 셀렉션 까베르네 소비뇽	Trapiche Melodias Winemaker's Selection Cabernet Sa...	트라피체	아르헨티나>멘도사	카베르네 소비뇽 100%
13	트라피체 폰 데 카브	Trapiche Fond de Cave	트라피체	아르헨티나>멘도사>우코 밸리	말벡 100%
14	트라피체 폰 데 카브	Trapiche Fond de Cave	트라피체	아르헨티나>멘도사>우코 밸리	말벡 100%
15	트라피체 핀카 라스 피에드라스 리미티드 에디션	Trapiche Finca Las Piedras Limited Edition	트라피체	아르헨티나>멘도사	말벡 100%

[그림 4] wine 데이터

본 연구에서는 데이터 전처리를 위해 R에서 제공하는 stringr 패키지와 dplyr 패키지를 활용하였다. 해당 패키지들은 데이터를 전처리하는데 유용한 패키지 중 하나로 dplyr[18]의 경우, R에서 제공하는 또 다른 전처리 패키지인 plyr에 비해 처리 속도가 빠르며, 유연한 데이터 조작 문법을 제공한다는 점에서 빅데이터를 처리하기에 상당히 유용한 패키지이다.

또한 stringr[19]의 경우 문자열을 보다 쉽게 처리하고 작업할 수 있는 패키지로 문자열 안에 위치한 개별 문자를 조작할 수 있으며, 전체적인 문자열을 합치거나 분리하여 출력한다.

[표 3] productor(생산자) 변수 전처리

```
s <- '\U00A0'
wine1$productor <- gsub(s, "", wine1$productor)
wine1$productor <- gsub("'", "", wine1$productor)
wine1$productor <- gsub("\\(", "", wine1$productor)
wine1$productor <- str_trim(wine1$productor)
```

[표 3]은 productor(생산자) 변수에서 유니코드 문자와 괄호를 제거한 후 공백을 제거하기 위한 명령어이다.

production(생산국) 변수의 경우 생산국과 생산지역에 관한 내용이 담겨있으나, 해당 문자열이 분리가 되어있지 않아 필요한 변수인 생산국을 얻기 위해 생산국, 생산지역을 분리해 줄 필요가 있다. 따라서 불필요한 문자열을 제거한 후, 해당 문자열에서 생산국과 생산지역을 구분 해주는 문자열 '>'을 기준으로 문자열을 분리했다.

[표 4] production(생산국) 변수 전처리

```
wine1$production <- gsub("-", "", wine1$production)
temp <- as.data.frame(do.call(
          rbind, strsplit(wine1$production, '>', perl=TRUE)))
temp <- as.data.frame(temp[,1])
colnames(temp) <- "production"
```

문자열을 분리하게 되면 각각의 리스트(list)형태로 분리되는데, 이를 데이터 프레임(data frame)형태로 만들기 위해 as.data.frame함수를 이용하였고, 각각의 데이터 프레임을 행(row)단위의 데이터 프레임으로 묶기 위해 rbind함수와 do.call 함수를 이용했다. 이렇게 만들어진 temp라는 변수와 wine1데이터를 합쳐 와인 생산국 정보를 갖는 production(생산국)변수를 생성하였다.

variety(주품종)는 와인에 들어간 포도의 품종을 나타내는 변수로, 각각의 포도 품종은 자신만의 고유의 특색이 존재하므로 해당 변수를 통해 와인의 맛을 미리 예상할 수 있다.[20]

따라서 와인 맛의 유사성을 나타내기 위해 와인에 주로 쓰이는 20개의 포도 품종에 대하여 해당 품종을 포함하면 1, 포함하지 않으면 0으로 데이터를 변환했다.

[표 5]는 데이터 변환하기 위한 명령어이다.

[표 5] variety(주품종)에 대한 변환

```
wine1$샤르도네 <- ifelse(str_detect(
                  wine1$variety, "샤르도네"), yes = 1, no = 0)
wine1$슈냉블랑 <- ifelse(str_detect(
                  wine1$variety, "슈냉블랑"), yes = 1, no = 0)
```

[그림 5]는 [표 5]의 명령어를 실행한 후 variety(주품종)변수가 변환되어진 데이터의 일부이다.

샤르도네	리슬링	쇼비뇽블랑	게뷔르츠트라미너	슈냉블랑	세미용	몰러트루가우	까베르네쇼비뇽	피노누아	그르나슈	메를로	가메	시라_쉬라즈	네비올로	산지오베제	템프라니오	말벡	진판델	모스카토	블렌드
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0

[그림 5] variety(주품종) 변수의 변환 결과

2) TF-IDF를 활용한 전처리

wine1 데이터에서 food(어울리는 음식)와 tasting_note(시음노트)는 음식의 종류와 와인의 맛을 나타내며, 해당 변수는 [그림 6]와 같이 문장 형태의 텍스트 형식으로 구성되어 있다.

	food	tasting
1	육류와 치즈 등과 잘 어울린다	깊고 짙은 레드 컬러로 믿을 수 없을 정도로 진한 체리 블랙 ...
2	육류와 치즈 등과 잘 어울린다	블랙 커런트와 정인 체리와 같은 아주 짙은 아로마를 느낄 ...
3	샐러드나 해산물 등과 잘 어울린다	눈부신 페일 옐로우 계열 아몬드 계열의 컬러감을 느낄 수 ...
4	돼지고기와 바베큐와 잘 어울린다	검은 자두 블랙 베리 강초의 풍부한 향과 더불어 블랙 후추...
5	스테이크나 치즈와 잘 어울린다	매우 진한 라즈베리 빛깔로 라즈베리 카시스 블랙베리 류의 ...
6	스테이크와 육류 및 치즈와 잘 어울린다	진한 다크 레드 컬러로 블루베리를 비롯한 과일 향과 오크 ...
7	양고기와 야생고기 및 숙성한 치즈와 잘 어울린다	루비 레드빛의 잘 익은 체리 자두 코코아 바닐라 향 느낄수 ...
8	스테이크와 치즈에 잘 어울린다	진한 다크 레드 컬러로 블루베리를 비롯한 과일 향과 오크 ...
9	스테이크와 치즈에 잘 어울린다	자주빛의 와인으로 팔레트에서 매우 잘 짜여진 구조감이 느...
10	스테이크와 육류 및 치즈와 잘 어울린다	년 빈티지는 올리아 와인의 번째 빈티지이다 개의 다양한 레...
11	양고기와 스테이크 및 치즈와 잘 어울린다	달콤한 검은 과일의 아로마와 스모키한 오크향 스파이시한 ...
12	스테이크와 치즈와 잘 어울린다	블랙베리 블랙체리 삼나무와 스파이시한 향신료 등 다양한 ...
13	스테이크와 육류 및 치즈와 잘 어울린다	진한 보랏빛이 감도는 레드 컬러 잘 익은 붉은 과일과 약간...

[그림 6] food(음식)과 tasting_note(시음노트) 변수 테이블

추천 알고리즘을 만들기 위해선 각각의 와인들이 어떠한 음식과 잘 어울리는지 또는 어떠한 맛과 향이 나는지에 대해 점수화 되어야 한다. 하지만 음식의 종류가 매우 다양하고, 와인의 맛과 향에 대해 조금씩 표현하는 방식이 다르므로 해당 변수들을 특정 단어만을 선택하여 점수화 할 수 없다. 따라서 두 변수는 텍스트 마이닝과 정보 검색 등에서 이용되는 TF-IDF방식을 활용하여 전처리를 실시하였다.

TF-IDF(Term Frequency - Inverse Document Frequency)는 문서 간의 유사도를 계산하기 위해 문서 내에 들어있는 단어들의 빈도수와 가중치를 통해 어떠한 수치 값을 부여해 놓은 방법이다.[21] TF는 해당 문서에서 특정한 단어가 얼마나 자주 나타나는지를 카운트한 값으로 해당 값이 높을수록 중요한 단어라 판단하며, 해당 문서(d_j)에서 단어(t_i)의 중요도는 [수식 4]과 같다.

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad \text{[수식 4]}$$

여기서 $n_{i,j}$는 해당 문서(d_j)에서 특정한 단어(t_i)가 출현하는 횟수를 의미하며, $\sum_k n_{k,j}$는 해당 문서(d_j)에서 출현하는 모든 단어 수를 말한다.

하지만 TF값이 높다고 그 단어를 해당 문서의 키워드로 판단 할 수는 없다. 문장을 구성하다보면 단어와 단어를 연결해주는 조사들의 비중이 높아지게 되거나 흔한 단어들이 해당 문장의 키워드로 떠오를 수 있기 때문이다. 이를 방지하고 핵심 단어를 찾기 위해 IDF(역문서 빈도)를 사용하게 되는데, IDF[수식 5]는 특정 단어가 몇 개의 고유한 문서에서 나왔는지를 카운트 한 수의 역수이며 로그를 취한 값이다.

$$IDF_i = \log \frac{|D|}{1 + |\{d_i : t_i \in d_j\}|} \quad [수식\ 5]$$

여기서 $|D|$는 전체 문서의 개수이며, $|\{d_i : t_i \in d_j\}|$는 특정한 단어(t_i)가 나오는 문서의 개수를 의미한다.[10]

이 두 방식을 통해 최종적으로 TF-IDF 값을 구하게 되며, 해당 문서 내에 특정 단어 빈도가 높을수록, 전체 문서 중 특정 단어를 포함한 문서가 적을수록 TF-IDF[수식 6]값은 높이지게 된다.

$$TF-IDF_{i,j} = TF_{i,j} \times IDF_i \quad [수식\ 6]$$

TF-IDF를 통해 특정 점수를 부여하기 위해선 단어 뭉치(Corpus)를 만들고, 이를 단어들의 출현 빈도로 정리한 TermDocument Matrix가 필요하다. 이를 위해 유사한 의미를 갖는 단어들을 하나로 묶어줄 필요가 있는데, 유사한 의미를 갖는 단어들을 묶어 주지 않은 채 분석할 경우 잘못된 방향으로 분석이 될 수 있기 때문이다.

유사 의미의 단어를 묶음 처리했으며 묶음 처리 한 단어들은 [표 6]과 같다.

[표 6] 유사한 의미의 단어 묶음

처리 전	처리 후
해물 요리/해산물 요리	해산물 요리
에피타이저/애피타이져/전체요리	애피타이저
바비큐/바베큐	바베큐
거위간 요리/프와그라	프와그라
회/생선회	생선회
소고기/쇠고기	쇠고기
타바스/타파스	타파스
아페리티프/식전주	식전주
하얀색의 육류	흰살 육류
맷돼지/멧돼지	멧돼지
카시스/까시스	까시스
까베르네/카베르네	카베르네
쇼비뇽/소비뇽	소비뇽
모스까또/모스카토	모스카토

단어 묶음 이후 단어 뭉치(Corpus)를 만들기 위해 한글 텍스트 마이닝에 적합한 KoNLP 패키지를 이용하였다.

KoNLP를 통해 형태소 분석을 하기 위해서는 참고 할 사전이 필요하다. 사전의 경우 시스템에서 제공하는 사전과 세종단어 사전이 있으며, 이번 연구에서는 세종단어사전을 참고하였다. 또한 세종단어사전에 들어있지 않은 와인 및 음식에 관한 단어는 KAIST 품사 태그셋을 참고하여 새로운 단어 사전에 추가하였다.[22](추가된 단어사전은 부록을 참고할 것)

TermDocument Matrix는 TermDocument Matrix()라는 함수를 이용하여 만들 수 있으며, 함수 안에 'control = list()'라는 벡터를 만들고 'tokenize = '옵션을 설정해 주면 문장을 원하는 단위로 분리할 수 있다.[22]

[표 7] weight_Tf-Idf 함수로 말뭉치 분리

```
noun_tokenizer <- function(doc){extractNoun(doc)}
tdmat <- TermDocumentMatrix(docs.corp,
                control = list(tokenize = noun_tokenizer,
                weighting = function(x) weightTfIdf(x, TRUE),
                wordLengths = c(1,Inf)))
```

명사 추출 함수를 만든 후 해당 함수를 TF-IDF함수와 함께 TermDocument Matrix 함수 안에 넣어주면 명사 형태로 분리된 말뭉치를 얻게 되며, 이렇게 얻은 말뭉치를 Matrix형태로 변경해주면 [그림 7]과 같은 결과물이 나오게 된다.

docs	doc1	doc2	doc3	doc4	doc5	doc6	doc7	doc8	doc9
육류와 치즈 등과 잘 어울린다	0.29574643	0.29574643	0.29574643	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
육류와 치즈 등과 잘 어울린다	0.02817251	0.02817251	0.02817251	0.03521563	0.03521563	0.02347709	0.02012322	0.03521563	0.03521563
붉은 육류, 치즈 등과 잘 어울린다	1.12973144	1.12973144	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
샐러드나 해산물 등과 잘 어울린다	0.03894014	0.03894014	0.03894014	0.04867517	0.04867517	0.03245011	0.02781438	0.04867517	0.04867517
돼지고기와 바베큐와 잘 어울린다	0.33275281	0.33275281	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
스테이크나 치즈와 잘 어울린다	0.00000000	0.00000000	1.24672394	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
스테이크와 육류 및 치즈와 잘 어울린다	0.00000000	0.00000000	0.67512774	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
양고기와 야생고기 및 숙성한 치즈와 잘 어울린다	0.00000000	0.00000000	0.00000000	1.66216429	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
스테이크와 치즈에 잘 어울린다	0.00000000	0.00000000	0.00000000	1.80840492	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
스테이크와 치즈에 잘 어울린다	0.00000000	0.00000000	0.00000000	0.00000000	2.05840492	0.00000000	0.00000000	0.00000000	0.00000000
스테이크와 육류 및 치즈와 잘 어울린다	0.00000000	0.00000000	0.00000000	0.00000000	0.74642304	0.49761536	0.42652745	0.00000000	0.00000000
양고기와 스테이크 및 치즈와 잘 어울린다	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.66428203	0.56938459	0.00000000	0.00000000
스테이크와 치즈와 잘 어울린다	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.84394911	0.00000000	1.26592367	1.26592367
스테이크와 육류 및 치즈와 잘 어울린다	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.39182944	0.00000000	0.00000000	0.00000000
스테이크와 육류 및 치즈와 잘 어울린다	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	1.17623138	0.00000000	0.00000000

[그림 7] TermDocumentMatrix을 통해 얻은 Matrix

해당 Matrix는 각 행(Row)을 한 개의 문서로 보고, 문서 내의 단어들의 빈도수를 TF-IDF로 계량화 점수이다.

이 점수 행렬을 표준 값이 1이 되도록 정규화[표 8] 한 뒤 행렬 곱을 이용하여 문서간의 유사도를 구하게 되면 와인간의 어울리는 음식 또는 유사한 시음 노트를 갖는 데이터를 얻을 수 있다.

[표 8] 표준 값이 1이 되도록 정규화 및 유사도 계산

```
norm_vec <- function(x) {x/sqrt(sum(x^2))}
tdmatmat <- apply(tdmatmat, 2, norm_vec)
docord <- t(tdmatmat[,1:16700]) %*% tdmatmat[,1:16700]
```

[그림 8]은 행렬 곱을 이용하여 각 와인간의 유사한 음식 값을 구한 결과이다.

	docs	scores.doc1	scores.doc2	scores.doc3	scores.doc4	scores.doc5	scores.doc6
doc1	육류와 치즈 등과 잘 어울린다	1.0000	1.0000	0.0510	0.0010	0.0011	0.0013
doc2	육류와 치즈 등과 잘 어울린다	1.0000	1.0000	0.0510	0.0010	0.0011	0.0013
doc49	스테이크, 양고기, 육류와 잘 어울린다	0.7583	0.7583	0.0011	0.0008	0.0010	0.0011
doc55	양고기, 스테이크, 립, 육류와 잘 어울린다	0.5249	0.5249	0.0008	0.0006	0.0007	0.0008
doc121	붉은 육류와 향료가 첨가된 요리와 잘 어울린다	0.3707	0.3707	0.0006	0.0004	0.0005	0.0005
doc166	고급스러운 디너에 어울리는 와인이다 붉은 육류와 매운 스...	0.2873	0.2873	0.0146	0.0003	0.0003	0.0004
doc272	해산물, 파스타, 치즈 등과 잘 어울린다	0.1705	0.1705	0.3890	0.0012	0.0014	0.0016
doc273	쇠고기, 파스타, 치즈 등과 잘 어울린다	0.1561	0.1561	0.0586	0.0011	0.0012	0.0015
doc108	붉은 육류, 치즈 등과 잘 어울린다	0.1546	0.1546	0.0581	0.0011	0.0012	0.1399
doc117	육류, 피자, 파스타, 치즈 등과 잘 어울린다	0.1427	0.1427	0.0536	0.0010	0.0011	0.1290
doc202	쇠고기, 가금류, 치즈 등과 잘 어울린다	0.1401	0.1401	0.0526	0.0010	0.0011	0.0013
doc268	쇠고기, 가금류, 치즈 등과 잘 어울린다	0.1401	0.1401	0.0526	0.0010	0.0011	0.0013
doc204	붉은 육류, 치즈, 파스타 등과 잘 어울린다	0.1385	0.1385	0.0520	0.0010	0.0011	0.1253
doc71	스테이크, 치즈 와 잘 어울린다	0.1178	0.1178	0.0020	0.0015	0.0017	0.0020
doc73	스테이크, 치즈 와 잘 어울린다	0.1178	0.1178	0.0020	0.0015	0.0017	0.0020
doc98	치즈, 스테이크 와 잘 어울린다	0.1178	0.1178	0.0020	0.0015	0.0017	0.0020

[그림 8] 행렬 곱을 이용한 와인간의 유사 음식 값

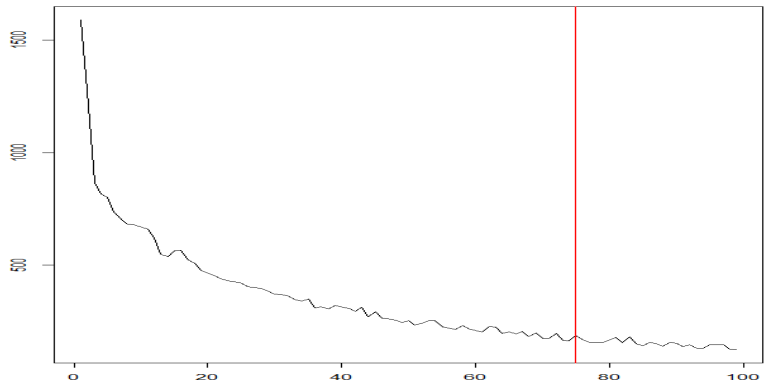
해당 값이 비슷할수록 특정 음식과 어울리는 와인이라고 볼 수 있기 때문에 K-means방법론을 이용하여 와인을 군집화 시켰다.

클러스터의 개수를 설정하기 위해 클러스터의 개수에 따른 내부분산을 확인한 결과 75개 정도의 클러스터가 적합하다고 판단된다.

[표 9] K-means 클러스터 개수에 따른 내부분산 확인

```
for(i in 2:100) {set.seed(1234)
  eval(parse(text=paste
        ("result",i,"<- kmeans(food_score[,-1],",i,");",sep="")))
  eval(parse(text=paste
        ("visual[",i,"] <- result",i,"$tot.withinss",sep="")))  }
```

[표 9]은 K-means 클러스터 개수에 따른 내부분산을 확인하기 위한 알고리즘이며, [그림 9]는 클러스터 개수에 따른 내부분산이 줄어드는 결과이다.



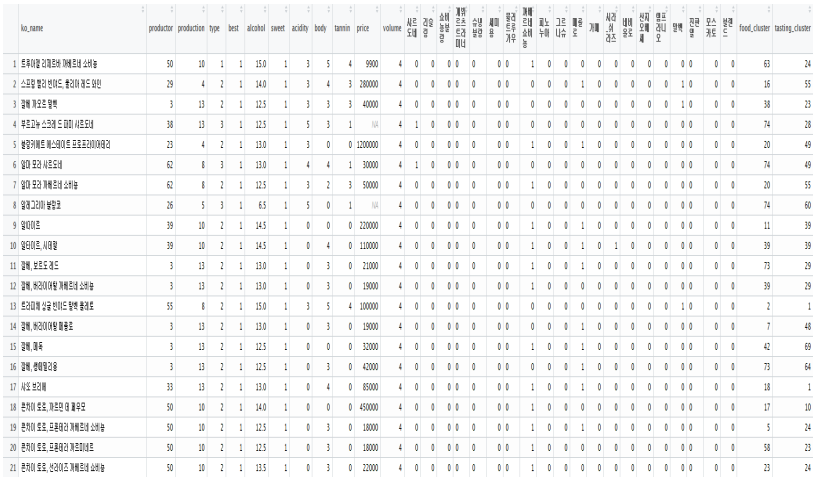
[그림 9] 클러스터 개수에 따른 내부분산

75개의 클러스터 중심점을 가지고 군집한 결과는 [그림 10]과 같다.

	docs	cluster.food.kmeans$cluster
doc113	가금류 고기 구이, 스테이크, 버섯 구이와 잘 어울린다	14
doc144	가금류 요리, 볶은 생선, 베이컨, 하몽, 살라미, 바베큐 요리 …	47
doc181	가금류 요리, 볶은 생선, 프와그라, 부드러운 치즈 등과 잘 …	47
doc258	가금류 요리, 볶은 생선, 프와그라, 부드러운 치즈 등과 잘 …	47
doc158	가금류, 고다치즈, 부드러운 소스의 중국음식, 일본퓨전 음…	75
doc270	가금류, 돼지고기, 스테이크, 치즈 등과 잘 어울린다	74
doc111	가금류, 볶은 육류, 고다치즈, 살라미, 양고기 등과 잘 어울린다	36
doc151	가금류, 스테이크 요리, 돼지갈비, 갈비찜, 튀김, 한식 요리 …	34
doc152	가금류, 스테이크 요리, 한식 요리 등과 잘 어울린다	47
doc156	가금류, 스테이크 요리, 한식 요리 등과 잘 어울린다	47
doc127	가금류, 흰살 육류, 해산물, 동양 요리와 잘 어울리며, 대부…	9
doc116	가벼운 서양 요리 등과 치즈와 잘 어울린다	72
doc191	가벼운 스택, 핑거 푸드, 치즈 등과 잘 어울린다	55
doc141	가벼운 애피타이저, 굴 요리, 샐러드 등과 잘 어울린다	52
doc149	각종 그릴 요리, 로스트 비프, 바베큐, 치즈 등의 음식등과 …	56
doc150	각종 그릴 요리, 로스트 비프, 바베큐, 치즈 등의 음식등과 …	56

[그림 10] 유사 음식으로 와인 클러스터링

TF-IDF와 클러스터링을 통한 전처리 이후 해당 변수를 병합하고, 각각의 변수들을 숫자 형태의 데이터로 코딩하여 데이터 셋[그림 11]을 구성하였다.



	ko_name	productor	production	type	best	alcohol	sweet	acidity	body	tannin	price	volume	사과 오레	감귤	소비 뇽향	계부 르스 즈산 미디	슈냥 뇽향	체미	광과 르무 가우	체베 르에 소비 뇽	꽃노 누베	그르 니슈	메락 오	계베	사과 _외 관즈	네배 유오	헌치 오메	캠프 리니 오	탱백	관련 뇽	모스 처트	뇽향 드	food_cluster	tasting_cluster
1	트루이알 리제르와 꺼베르네 소비뇽	50	10	1	1	15.0	1	3	5	4	9900	4	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	63	24
2	스프랑 뽈리 빈이드 _ 론긴이 레드 와인	29	4	2	1	14.0	1	3	4	3	280000	4	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	16	55
3	갤베 개오트 탕백	3	13	2	1	12.5	1	3	3	3	40000	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	58	23
4	부르고뉴 스크와 드 더미 시르드네	38	13	3	1	12.5	1	5	3	1	NA	4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	74	28
5	봉랑커베트 메스와이트 프로프리이와대긴	23	4	2	1	13.0	1	3	0	0	1200000	4	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	20	49
6	알마 모간 시르드네	62	8	3	1	13.0	1	4	4	1	30000	4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	74	49
7	알마 모간 꺼베르네 소비뇽	62	8	2	1	12.5	1	3	2	3	50000	4	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	20	55
8	알제그리아 블랑꼬	26	5	3	1	6.5	1	5	0	1	NA	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	74	60
9	알타이트	39	10	2	1	14.5	1	0	0	0	220000	4	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	11	39
10	알타이트_시라멜	39	10	2	1	14.5	1	0	4	0	110000	4	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	39	39
11	갤베_보르도 레드	3	13	2	1	13.0	1	0	3	0	21000	4	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	73	29
12	갤베_꺼리어대탕 꺼베르네 소비뇽	3	13	2	1	13.0	1	0	3	0	19000	4	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	39	29
13	트리더제 상슈 빈이드 탕백 롱개론	55	8	2	1	15.0	1	3	5	4	100000	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	2	1
14	갤베_꺼리어대탕 메롱르	3	13	2	1	13.0	1	0	3	0	19000	4	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	7	48
15	갤베_꺼루	3	13	2	1	12.5	1	0	0	0	32000	4	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	42	69
16	갤베_샌메일리옹	3	13	2	1	12.5	1	0	3	0	42000	4	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	73	64
17	시도 브리베	33	13	2	1	13.0	1	0	4	0	85000	4	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	18	1
18	론자이 또도_꺼르한 대 블랑꾸	50	10	2	1	14.0	1	0	0	0	450000	4	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	17	10
19	론자이 또도_프로와리 꺼베르네 소비뇽	50	10	2	1	12.5	1	0	3	0	18000	4	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	5	24
20	론자이 또도_프로와리 꺼르미네르	50	10	2	1	12.5	1	0	3	0	18000	4	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	58	23
21	론자이 또도_빈리어즈 꺼베르네 소비뇽	50	10	2	1	13.5	1	0	3	0	22000	4	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	23	24

[그림 11] 유사도 알고리즘을 적용하기 위한 데이터 셋

3) 차원 축소

이렇게 구성된 wine1 데이터 셋은 총 34개의 변수들로 이루어진 다차원으로 구성되어 있다. 하지만 데이터의 차원이 커지게 되면 차원 내의 부피는 기하급수적으로 증가하며, 데이터가 없는 빈 공간도 많아지게 되어 모델의 예측력이 현저하게 낮아지는 '차원의 저주' 라는 문제점이 빠지게 된다. 이러한 '차원의 저주'의 문제점과 데이터 간의 다중 공선성 문제를 해결하기 위해 차원 축소를 실시한다.

차원 축소를 위해 각 독립변수들 간의 공분산 분석을 통해 해당 독립변수들이 상관성이 존재하지 않는지를 확인하였다.

공분산 분석은 두 개의 확률변수의 상관정도를 나타내는 값으로 공분산 값이 양수이면 두 개의 변수는 같은 방향으로 변하는 상관관계를 갖으며, 해당 값이 음수이면 두 변수는 다른 방향으로 변하는 상관관계를 갖는다. 하지만 사실 방향은 알 수 있어도 x와 y의 단위가 다르면 표준화되지 않아서 얼마나 상관이 큰지를 알 수 없다. 따라서

공분산을 표준화한 상관계수[그림 12]를 가지고 각 데이터 간의 상관성을 확인하였다.

	productor	production	type	best	alcohol	sweet	acidity	body	tannin	price	volume
productor	1.000000000	-0.142428947	0.11932235	-0.028311935	0.16331103	-0.21054540	0.02575139	0.01052487	0.009554503	0.02127452	-0.03227997
production	-0.142428947	1.000000000	-0.07785779	-0.004853468	0.06764009	-0.11124753	-0.04657463	0.09148146	-0.055155189	0.11069167	0.08704768
type	0.119322347	-0.077857788	1.00000000	0.524809223	-0.40838731	0.36044423	0.17011915	-0.15070090	-0.069852963	-0.12359378	-0.15298266
best	-0.028311935	-0.004853468	0.52480922	1.000000000	-0.50674279	0.53081382	-0.13570644	-0.04050006	-0.162951784	-0.03247198	-0.26443961
alcohol	0.163311035	0.067640090	-0.40838731	-0.506742793	1.00000000	-0.64207995	-0.04441042	0.34525706	0.213474680	0.43812706	0.17386014
sweet	-0.210545404	-0.111247531	0.36044423	0.530813816	-0.64207995	1.00000000	-0.07303557	-0.06072038	-0.070509988	-0.31165330	-0.22102899
acidity	0.025751390	-0.046574635	0.17011915	-0.135706437	-0.04441042	-0.07303557	1.00000000	-0.08585420	0.638764089	-0.21845758	-0.06189103
body	0.010524867	0.091481461	-0.15070090	-0.040500057	0.34525706	-0.06072038	-0.08585420	1.00000000	0.229782774	0.29710910	0.01338449
tannin	0.009554503	-0.055155189	-0.06985296	-0.162951784	0.21347468	-0.07050999	0.63876409	0.22978277	1.000000000	-0.01422180	-0.02703882
price	0.021274524	0.110691670	-0.12359378	-0.032471984	0.43812706	-0.31165330	-0.21845758	0.29710910	-0.014221799	1.00000000	0.16417502
volume	-0.032279971	0.087047677	-0.15298266	-0.264439605	0.17386014	-0.22102899	-0.06189103	0.01338449	-0.027038818	0.16417502	1.00000000

[그림 12] 와인 데이터 셋의 독립변수 간의 상관계수

각각의 독립 변수들 간의 상관계수를 살펴보면 전반적으로 하나의 독립변수가 증가 또는 감소함에 따라 다른 독립변수 역시 증가 또는 감소하는 성향을 보이는 것을 알 수 있다. 따라서 해당 변수들의 다중 공선성 문제를 해결하기 위해 차원 축소를 실시 할 필요가 있으며, 이번 연구에서는 차원 축소를 위해 주성분 분석을 실시하였다.

주성분 분석은 변수들 간에 중요성을 갖고, 변수들 간의 순서가 주어졌을 때 활용되는 기법으로 이를 통해 차원을 축소하고, 다중 공선성 문제를 해결할 수 있다.

주성분 분석은 선형적으로 m차원의 입력 공간을 n차원의 출력 공간으로 변환하여 (m-n)만큼의 차원을 포기하면서 발생하는 분산 값과 정보의 손실을 최소화하는 것을 목표로 한다.[11] 이를 통해 첫 번째 주성분이 가장 큰 분산 값을 갖도록 하며, 다음 구성 요소는 첫 번째 주성분과 상관관계가 적고 첫 번째 주성분과 직교를 이룬다.

주성분 분석에서 주성분의 개수를 결정하는 기준은 통상적으로 분산의 누적 합계가 70~90% 인 것까지를 사용하며, 개별 고유 값의 분

산 값이 1이상인 것을 선택한다.[13]

[표 10] 주성분 분석

```
wine.pca <- prcomp(wine2_2, scale. =T)
wine.pca
```

[표 10]는 wine1 데이터에 대한 주성분 분석을 실시하기 위한 명령어이며, 각 데이터들의 크기가 다르기 때문에 해당 주성분 분석을 위해 정규화 작업을 함께 실시한다.

```
                PC1         PC2         PC3         PC4         PC5         PC6         PC7         PC8         PC9         PC10        PC11
productor -0.0728698886 -0.086060188 0.0771723030 -0.76019717 -0.18829586  0.17685250 -0.45395336  0.33159045 -0.13834182  0.02791989  0.027428589
production -0.0896504553  0.169079362 0.1071046696  0.46797781 -0.57152452  0.57062008 -0.23311986  0.09715338 -0.08673290 -0.09560809 -0.014339427
type        0.3748972421 -0.096902701 0.2540721310 -0.24541541 -0.44558310 -0.12088096  0.13437911 -0.59623490 -0.15994871 -0.25254085 -0.221832773
best        0.4251849770  0.144459223 0.4002199993 -0.04510810 -0.13699791 -0.03596997  0.08801340  0.26353932  0.69831248  0.07634451  0.213091724
alcohol    -0.5054379537  0.004240018 0.1441762419 -0.13076652  0.05264353  0.06980863  0.10835727 -0.19610797  0.21893568 -0.62585458  0.457336651
sweet       0.4626723269  0.020837420 0.1467418383  0.21006289  0.20510594 -0.20189240 -0.23735352  0.26876971 -0.45883356 -0.41968484  0.348961456
acidity    -0.0005625882 -0.680233248 -0.0005920662  0.08971607 -0.24362216  0.00599356  0.13861485 -0.04443228 -0.10934812  0.36119339  0.553389038
body       -0.2155908248  0.032291733 0.6279275885  0.14477929  0.26056044 -0.10672538 -0.48224768 -0.34758608 -0.02660713  0.32211223  0.009927427
tannin     -0.1443529499 -0.614579334 0.2602488576  0.17453605  0.02384368 -0.11306475 -0.01158953  0.33383373  0.14181442 -0.30710247 -0.513719257
price      -0.2842367880  0.275620518 0.4409585431 -0.06496086 -0.17497344 -0.20050301  0.53857617  0.32756256 -0.39026370  0.15475223 -0.012072206
volume     -0.2183605167  0.125938799 -0.2430165887  0.12615273 -0.46618039 -0.71914169 -0.32206190  0.05858299  0.12873201 -0.01295202  0.055696393
Importance of components%s:
                          PC1    PC2    PC3    PC4     PC5     PC6     PC7     PC8     PC9     PC10    PC11
Standard deviation     1.698 1.3235 1.1375 1.1003 0.98516 0.93007 0.83868 0.68148 0.58091 0.56343 0.45089
Proportion of Variance 0.262 0.1592 0.1176 0.1101 0.08823 0.07864 0.06394 0.04222 0.03068 0.02886 0.01848
Cumulative Proportion  0.262 0.4213 0.5389 0.6490 0.73718 0.81582 0.87976 0.92198 0.95266 0.98152 1.00000
```

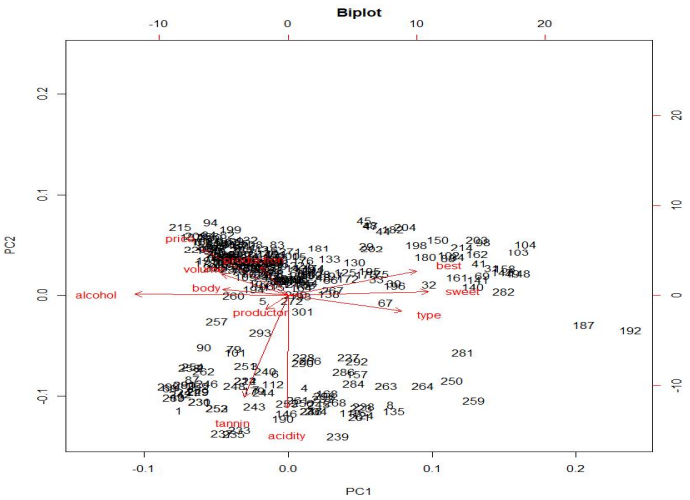
[그림 13] 변수간의 주성분 분석 결과

[그림 13]는 주성분 분석의 결과를 나타내고 있으며, Standard deviation(표준편차)의 제곱이 1 이상이고, Cumulative Proportion(분산의 누적 합계)가 70~90% 이상인 주성분이 없어 해당 값과 가장 근접한 PC4까지로 선택하였다.

따라서 해당 주성분의 경우 PC1은 와인의 종류, 용도, 당도로 구분

되어 지며, PC2는 산도와 탄닌 PC3는 바디감, 가격, 알코올, PC4는 생산자로 나눠진다.

[그림 14]은 Biplot으로 확인한 주성분 분포로 각 개체의 관찰 값은 주성분 점수로 하여, 각 변수와 주성분과의 관계를 나타내는 주성분 계수를 동시에 나타낸다.



[그림 14] Bipolt으로 본 주성분 분석 결과

해당 주성분 분석을 통해 생산지역, 와인용도, 알코올, 탄닌 4가지의 변수를 축소한 후 상관계수를 확인한 결과 각 변수들 간의 상관성이 크게 존재하지 않는 것으로 나타났다.

	productor	type	sweet	acidity	body	price	volume
productor	1.00000000	0.11932235	-0.21054540	0.02575139	0.01052487	-0.04634628	-0.03227997
type	0.11932235	1.00000000	0.36044423	0.17011915	-0.15070090	-0.08498949	-0.15298266
sweet	-0.21054540	0.36044423	1.00000000	-0.07303557	-0.06072038	-0.17637748	-0.22102899
acidity	0.02575139	0.17011915	-0.07303557	1.00000000	-0.08585420	-0.03027955	-0.06189103
body	0.01052487	-0.15070090	-0.06072038	-0.08585420	1.00000000	0.14449579	0.01338449
price	-0.04634628	-0.08498949	-0.17637748	-0.03027955	0.14449579	1.00000000	0.08015366
volume	-0.03227997	-0.15298266	-0.22102899	-0.06189103	0.01338449	0.08015366	1.00000000

[그림 15] 변수 축소 후 독립변수 간의 상관계수

변수 축소 이후 최종적으로 유사도 알고리즘을 적용하기 위한 wine2 데이터 셋은 28개의 변수로 이루어졌다.

4. 유사도 알고리즘을 적용한 추천시스템 구현

와인선택속성 데이터를 기반으로 소비자가 입력하는 와인 A에 대해 해당 와인과 가장 유사한 와인을 추천해 주는 시스템을 구현하고자 한다.

본 논문 2-3 유사도 계산에 관한 연구에 나타나 있듯이, 유사성을 측정하는 방법에는 유클리디안, 맨하탄, 민코우스키, 코사인 등 다양한 방법이 존재한다. 이 중에서 두 벡터(Vector)간의 거리를 계산하여 유사도를 구하는 유클리디안 거리 유사도 방식과 내적공간에서 두 벡터(Vector)간의 각도를 코사인(Cosine)방식을 통해 측정한 값을 활용하는 코사인 유사도가 가장 많이 활용되고 있다.

유클리디안 거리는 다차원에서의 두 지점간의 상대거리를 계산하는 방식으로, 각 아이템 간의 각도 값만을 이용하여 방향성만을 확인하는 코사인 유사도보다 다차원(속성)의 값들로 이루어진 와인간의 유사성을 표현하기에 더 적합하다고 판단된다. 따라서 이번 연구에서는

유클리디안 거리 유사도를 선택하였고 이를 이용하여 와인별 유사도를 계산하였다.

와인 1개를 하나의 벡터로 인식하고 그 벡터의 구성요소를 이루고 있는 와인의 속성 값은 [수식 7]과 같이 정규화 될 수 있으며, [표 11]는 해당 데이터를 정규화한 데이터의 일부이다.

$$\frac{52}{\sqrt{(52^2+8^2+1^2+\cdots+58^2)}} = 0.805885 \text{ [수식 7]}$$

[표 11] 정규화한 데이터

	productor	type	sweet	acidity
트루아젤리제르바 카베르네 소비뇽	0.805885	-1.922998	-0.441546	1.137026
스프링 밸리 빈야드 울리아 레드 와인	-0.330281	-0.544849	-0.441546	1.137026
깔베 까오르 말벡	-1.724667	-0.544849	-0.441546	1.137026
알레그리아 블랑코	0.134514	0.833299	-0.441546	2.279746

이후 유클리디안 거리 유사도 행렬을 구하기 위해 정규화한 데이터를 dist함수에 넣고 method를 유클리디안으로 설정하였다.

[표 12] 유클리디안 유사도 알고리즘 적용

```
euclidean_wine <- scale(wine3)
temp1 <- wine2$ko_name
rownames(euclidean_wine) <- temp1

euclidean_wine <- dist(euclidean_wine, method = "euclidean")
euclidean_wine1 <- as.data.frame(as.matrix(euclidean_wine))
```

유클리디안 유사도 알고리즘을 적용한 후 출력 알고리즘을 위해선 Matrix 형태의 데이터가 data.frame형태로 변형이 되어야 한다. 따라서 해당 데이터를 data.frame 형태로 변환하였으며 해당 값은 [표 13]과 같다.

[표 13] 유클리디안 유사도를 대입한 결과

	트루아젤 리제르바 까베르네 소비뇽	스프링 밸리빈야 드울리아 레드 와인	깔베 까오르 말벡	부르고뉴 스크레 드파미 샤르도네	알마 모라 샤르도네	알마 모라 까베르네 소비뇽	알레 그리아 블랑코	알따이르
트루아젤 리제르바 까베르네 소비뇽	0	6.617836	6.057041	5.606532	5.408156	3.911853	5.866099	6.202858
스프링 밸리 빈야드 울리아 레드 와인		0	4.672844	7.255647	7.178388	6.505268	7.104268	5.912118
깔베 까오르 말벡			0	6.634612	6.366978	5.737012	5.88507	7.389772
부르고뉴 스크레 드파미 샤르도네				0	3.179921	6.093251	4.138585	7.183153
알마 모라 샤르도네					0	4.957546	5.611097	7.597673
알마 모라 까베르네 소비뇽						0	5.393681	5.237113
알레 그리아 블랑코							0	6.001297
알따이르								0

[표 13]에서 유클리디안 유사도 행렬 값은 대각선을 중심으로 대칭을 이루고 있으므로 한쪽 면만을 참고하기 위해 대칭행렬의 하단 부분은 삭제하였다.

[표 14] 출력 알고리즘을 위한 행렬 데이터 변환

```
euclidean_wine2$names <- rownames(euclidean_wine2)

euclidean_wine2 <- reshape2::melt(euclidean_wine2,
                                  id.vars = "names")
euclidean_wine2 <- plyr::arrange(euclidean_wine2,
                                 variable, plyr::desc(names))
```

출력 알고리즘을 구현하기 전 행렬 형태도 되어 있는 데이터 구조를 변환할 필요가 있다. 따라서 [표 14]와 같이 reshape함수를 이용하여 해당 행렬구조를 와인 이름 별 유클리디안 값을 갖는 데이터 형태로 변환시켰다.

이후 유클리디안 유사도 결과를 토대로 소비자가 작성한 와인의 데이터를 입력받아 와인 별 선택 속성이 유사한 와인을 5개씩 출력해주는 알고리즘[표 15]을 구현하였다.

[표 15] 유사도에 따른 추천 와인 출력 알고리즘

```
x <- c("알따이르", "간치아 모스카토 다스티")
qwe = function(x){
  wine_N <- as.data.frame(x)
  colnames(wine_N) <- "wine_name"
  wine_N$wine_name <- as.character(wine_N$wine_name)
  wine_B <- c()
  for(i in 1:length(wine_N$wine_name)){
    wine_A <- head(filter(euclidean_wine2,
                          variable == wine_N$wine_name[i]) %>%
                           select(names, variable, value) %>%
                           arrange(variable, value), 5)
    wine_B <- cbind(wine_B, wine_A)}}
```

소비자가 입력한 와인과 유사한 추천 와인 5개를 데이터 프레임(data frame)형태로 출력한 결과 [표 16], [표 17]와 같다.

[표 16] 알따이르와 유사 와인 추천 리스트

names	variable	value
에토스 리저브 카베르네 소비뇽	알따이르	1.480498
아르젠티에라 빌라 도노라티코	알따이르	3.054457
스프링 밸리 빈야드 더비 카베르네 소비뇽	알따이르	3.145879
스프링 밸리 빈야드 프레데릭 레드	알따이르	3.183806
죠단 카베르네 소비뇽	알따이르	3.396034

[표 17] 모스카토 다스티와 유사 와인 추천 리스트

names	variable	value
간치아 프리미엄 모스카토 다스티	간치아 모스카토 다스티	1.566198
간치아 아스티	간치아 모스카토 다스티	1.955716
미켈레 끼아를로 가비 레 마르네	간치아 모스카토 다스티	2.129593
페스타 비앙코	간치아 모스카토 다스티	2.673181
로까 세리나 모스카토 다스티	간치아 모스카토 다스티	2.796701

해당 값들은 위에서 적용된 출력 알고리즘과 유사도 값에 따라 오름차순으로 정렬되어 사용자가 입력한 와인에 대해 유클리디안 거리가 가장 낮은 상위 5개의 와인의 값을 출력하게 된다.

특히 value 변수에 담긴 값은 정규화 된 유클리디안 행렬 값을 이용하여 유클리디안 거리를 구한 값으로 해당 와인과 가장 유사한 특징을 가질 때 0에 가까워진다. 또한 거의 유사한 값을 가지고 있는 와인의 경우 해당 값을 절사하게 되면 순위 변동 및 수치의 변화가 생길 수 있어 해당 값들의 절사는 하지 않았다.

따라서 [표 16]의 알따이르 와인과 유사한 값을 갖는 상위 5개의 와인의 값은 '에토스 리저브 카베르네 소비뇽', '아르젠티에라 빌라 도노라티코', '스프링 밸리 빈야드 더비 카베르네 소비뇽', '스프링 밸리 빈야드 프레데릭 레드', '죠단 카베르네 소비뇽' 라는 결과를 볼 수 있다. 해당 와인들의 특징을 살펴보면 와인 종류는 레드와인, 용도는 테이

블 와인, 평균 알코올 도수는 13도이며, 까베르네쇼비뇽과 메를로가 들어가 있다. 또한 어울리는 음식의 경우 그릴에 구운 육류와 스테이크로 비슷한 성향을 보여주는 것을 볼 수 있다.

[표 17]의 간치아 모스카토 다스티와 유사한 값을 갖는 상위 5개으 와인 값은 '간치아 프리미엄 모스카토 다스티', '간치아 아스티', '브라운 브라더스 모스카토', '페스타 비앙코', 로까 세리나 모스카토 다스티' 이며, 이 와인들은 화이트 와인, 테이블 또는 디저트 와인, 평균 당도 4.5점, 평균 알코올 도수 7도, 플로랄 계열의 향과 청포도 맛을 공통된 특징으로 갖는다.

Ⅴ. 결론 및 향후 연구과제

이번 연구를 추진하게 된 주요 배경은 와인 초보자와 입문자가 시중에 판매되고 있는 다양한 와인 중에 본인의 입맛에 잘 맞는 와인 또는 유사한 맛의 와인을 추천받게 함으로써, 와인 구매에 소요되는 시간적인 비용을 줄이고 소비자가 가격 대비 만족스러운 구매를 할 수 있도록 빅데이터를 활용한 유사도 알고리즘의 와인 추천 시스템을 개발 하였다.

다양한 종류의 추천 알고리즘이 존재하지만 본 논문에서는 아이템 기반의 유사도 알고리즘을 이용한 추천시스템에 대해 다루었다.

해당 시스템은 각각의 와인을 특징 짓는 와인선택속성들이 지닌 값을 코사인 유사도 알고리즘을 통해 계산함으로써 각 와인 간의 비슷한 정도를 측정하였고 이 값이 높은 상위 5개의 와인을 추천 해 주는 방식으로 이루어져 있다. 또한 비정형 데이터로 구성된 음식과 시음 속성을 문서 간 단어들의 출현 빈도수를 통해 유사도를 측정하는 방식인 TF-IDF방식으로 처리함으로써 해당 와인과 어울리는 음식 및 와인의 맛의 유사성까지 고려하였다.

이를 통해 다양한 와인의 특성 중 하나의 카테고리에 초점을 맞춘 추천 시스템이 아닌 와인선택속성이 전반적으로 비슷한 와인을 추천하는 시스템이라는 점에서 소비자의 입맛과 유사한 와인을 더 정확하게 추천해 줄 수 있다고 판단된다.

그러나 와인의 경우 소득, 성별, 이벤트성, 함께 마신 사람 등 소비자의 개인화된 특성에 따라서도 와인을 선택하는 성향이 바뀌게 되는데, 소비자의 개인화된 특성까지 고려하지 못하였다는 점이 이번 연구의 한계점이다.

따라서 향후 관련 연구에서는 위에서 개발한 추천 시스템을 기반으로 소비자의 개인화 된 특성이 고려된 구조적 와인 추천 알고리즘에

대한 연구가 이루어져야 하며, 이를 토대로 소비자 특성을 고려한 추천 알고리즘에 대한 발전된 연구가 이루어져야 할 것이다.

참 고 문 헌

[1]고재윤, "국내 와인소비자의 웰빙인식과 와인구매선택속성간의 관계", 『한국호텔외식관광경영학회』 제16권 제1호, 2007.

[2]문정훈, 장익훈 외 3명, "빅데이터 기반 소비자 유형별 농식품 추천시스템 구축 사례", 『The Journal of Korean Institute of Communications and Information Sciences』, 제05권 제40호, 2015. p.904

[3]방진식, 전진화, "와인 소비자 분류에 따른 와인 선호도에 관한 연구", 『한국조리학학지』 제11권 제2호, 2005. p2.

[4]송경숙, "와인선택속성이 만족도 및 재구매의도에 미치는 영향", 『한국콘텐츠학회논문지』 제12권 제12호, 2012. p.433

[5]김영자, 김동진, "와인선택속성이 고객만족 및 재구매 의도에 미치는 영향", 『한국 외식산업학회지』 제8권 제3호, 2012. p60-62

[6]이승민, "통계패키지 R과 유사도 알고리즘을 활용한 추천시스템 구현", 숭실대학교 대학원, 석사학위 논문, 2014.

[7]남승민, 윤승희, 신홍철, "호텔레스토랑에서 고객의 소비 라이프 스타일에 따른 와인선택속성이 고객만족에 미치는 영향", 『한국호텔관광학회』 제17권 제5호, 2015. p.181

[8]이성현, "R 패키지 Recommenderlab을 이용한 추천시스템 성능평가", 동국대학교 학사 졸업논문, 2015. p4-7

[9]안두철, "TF-IDF 기반 Aggregation Strategy을 이용한 그룹 추천 시스템", 충남대학교 대학원, 석사학위 논문, 2015.

[10]박종영, 서충원 "TF-IDF 가중치 모델을 이용한 주택시장의 변화 특성 분석", 『한국부동산학회』 제63권, 2015. p.51-52

[11]수레시 고라칼라 & 미셸 우수엘리, 『Building a Recommendation System with R』 에이콘. 2017. p.26-37

[12]Mooney, Raymond J & Roy Loriene, "Content-Based Book Recommending Using Learning for Text Categorization", 『Graduate School of Library and Information Science University of Texas』, 2000. p.1-2

[13]Jolliffe I, 『Principal Component Analysis, Second Edition』, Springer, 2002. p.112-115
[14]http://egloos.zum.com/metashower/v/9957577
[15]https://ko.wikipedia.org/wiki/%EC%BD%94%EC%82%AC%EC%9D%B8_%EC%9C%A0%EC%82%AC%EB%8F%84
[16]http://bigBigdata.tistory.com/99?category=529087
[17]http://www.wine21.com/main.html
[18]https://cran.r-project.org/web/packages/dplyr/vignettes/dplyr.html
[19]https://cran.r-project.org/web/packages/stringr/vignettes/stringr.html
[20]http://blog.daum.net/musiker/18150028
[21]https://ko.wikipedia.org/wiki/TF-IDF
[22]https://github.com/haven-jeon/KoNLP/blob/master/etcs/KoNLP-API.md

부 록

<부록 1> 음식 및 시음노트 단어(명사)사전

<부록 1> 음식 및 시음노트 단어(명사) 사전

번호	단어	품사기호	번호	단어	품사기호
1	육류	ncn	39	스모키	ncn
2	치즈	ncn	40	탄닌	ncn
3	샐러드	ncn	41	르꼬끄	ncn
4	해산물	ncn	42	오크통	ncn
5	돼지고기	ncn	43	로스트비프	ncn
6	바베큐	ncn	44	까르미네르	ncn
7	스테이크	ncn	45	까시스	ncn
8	양고기	ncn	46	트라피체	ncn
9	야생고기	ncn	47	쉬라즈	ncn
10	크림	ncn	48	샤르도네	ncn
11	피자	ncn	49	리슬링	ncn
12	토마토	ncn	50	소비뇽	ncn
13	파스타	ncn	51	슈냉블랑	ncn
14	쇠고기	ncn	52	세미용	ncn
15	닭고기	ncn	53	카베르네	ncn
16	샌드위치	ncn	54	피노누아	ncn
17	립	ncn	55	그르나슈	ncn
18	과일	ncn	56	메를로	ncn
19	케이크	ncn	57	네비올로	ncn
20	생선	ncn	58	산지오베세	ncn
21	스프	ncn	59	말벡	ncn
22	타파스	ncn	60	진판델	ncn
23	등심	ncn	61	모스카토	ncn

24	안심	ncn	62	블랜드	ncn
25	오리꽁피	ncn	63	템프라니오	ncn
26	비스킷	ncn	64	게뷔르츠 트라미너	ncn
27	디저트	ncn	65	이스까이	ncn
28	카나페	ncn	66	강성	ncn
29	애피타이저	ncn	67	연성	ncn
30	푸와그라	ncn	68	고다	ncn
31	식전주	ncn	69	프레쉬	ncn
32	라자냐	ncn	70	와이너리	ncn
33	바다가재	ncn	71	가금류	ncn
34	캐비어	ncn	72	커런트	ncn
35	육두구	ncn	73	쌉싸름	ncp
36	미트볼	ncn	74	인상적	mma
37	브리또	ncn	75	매혹적	mma
38	핑거푸드	ncn	76	섬세	mma

국 문 초 록

유사도 알고리즘을 활용한
와인 추천 알고리즘 개발 연구

-빅데이터 분석 기법 중심으로-

조준호
빅데이터전문가 학과 빅데이터전문가학 전공
남서울대학교 (복지경영)대학원

이번 연구는 와인 초보자 및 와인 입문자가 자신의 입맛에 맞는 와인을 쉽게 선택할 수 있도록 도움을 주는 것을 목적으로 하고 있다.

추천 시스템은 다양한 추천 알고리즘을 통해 사용자가 원하는 정보 또는 제품을 추천해줌으로써 사용자가 들여야 하는 시간적, 물질적 비용을 줄이고 투자 대비 만족감을 극대화 시킬 수 있도록 도움을 주는 시스템이다. 그 중에서 유사도 알고리즘을 기반으로 한 추천시스템은 소비자가 현재 사용하고 있는 아이템 또는 정보와 비슷한 특성의 다른 정보 및 아이템을 추천해 준다는 점에서 효율적인 추천 시스템이라고 할 수 있다.

이에 이번 연구에서는 추천 시스템의 유형과 추천 시스템에 활용되는 알고리즘에 대한 유용성 및 한계점에 대해 정성적으로 분석하고, 더 나아가 빅데이터 분석 기법을 중심으로 한 유사도 알고리즘 기반의 와인 추천 알고리즘을 연구하였다.

해당 추천 시스템을 위해 크롤링 봇을 제작하여 와인선택속성을 기준으로 데이터를 수집하였고, 해당 값 중에 비정형 데이터로 이루어

진 어울리는 음식과 시음 노트 변수를 문서 간 단어들의 출현 빈도수를 통해 유사도를 측정하는 방식인 TF-IDF방식으로 처리함으로써 비정형 데이터를 정형화 시켰다. 이를 통해 각 와인들 간에 어울리는 음식의 유사성과 맛의 유사성까지 고려하여 해당 추천 시스템의 정확도를 높였다.

해당 유사도 알고리즘 기반의 와인 추천 시스템을 통해, 소비자 자신이 과거 시음했던 와인 중 자신의 마셔 본 와인과 가장 유사한 종류의 와인을 추천해줌으로써 와인 초보자 또는 와인 입문자가 와인에 대해 조금 더 친숙하게 다가갈 수 있을 것으로 본다.

주제어(키워드, 색인어)
빅데이터, 추천 시스템, R프로그램, 유사도 알고리즘, TF-IDF, 와인

ABSTRACT

Development of wine recommendation algorithm using similarity algorithm

-Focus on Bigdata analysis techniques-

Jo Jun-Ho
Bigdata Specialist Dept., Bigdata analysis Major
The Graduate School of Namseoul University

The study aims to help ensure that wine novices and wine beginners can easily choose a wine to suit your taste.

The recommendation system is a system that helps users to maximize the satisfaction of investment by reducing the time and material cost that users need by recommending the information or product they want through various recommendation algorithms. Recommendation system based on the similarity algorithm is an effective recommendation system in that it recommends other information and items similar in characteristics to items or information currently used by the consumer.

In this study, I have analyzed qualitatively the types of recommendation systems, the usefulness of the algorithms that constitute them, and their limitations. Furthermore, we developed a wine recommendation algorithm based on the similarity algorithm based on Big Data Analysis.

The crawling bots were created for the recommendation system and the data were collected based on the wine selection attributes. Among the corresponding values, the matching food and tasting note variables, which are atypical data, were measured using TF -IDF method to form the unstructured data. In this way, the accuracy of the recommendation system is improved by taking into account the similarity of taste and similarity of food among the wines.

The wine recommendation system based on the similarity algorithm can recommend the wine of the kind most similar to one of the wines that the consumer has tasted in the past so that the wine novice or the beginner can approach the wine more familiarly I think there will be.

Key word(Guide word)
Bigdata, R program, Similarity algorithm, Recommendation system, TF-IDF, Wine