**써브웨이 샌드위치**

**메뉴 조합 추천을 위한**

**어플리케이션**



|  |  |
| --- | --- |
| **과 목 명** | **캡스톤디자인1** |
| **담당교수** | **최진우 교수님** |
| **학 과** | **컴퓨터공학과** |
| **이 름** | **신민식(2017103999)**  **이민영(2018110653)** |
| **제 출 일** | **2022-12-15** |

**목차**

**1.서론**3

1.1 연구배경3

1.2 연구목표4

**2.관련연구7**

2.1 기존의 유사 연구7

2.1.1 써브웨이7

2.1.2 미식큐, 맛있는취향7

2.2 기존 연구의 문제점과 해결 방안9

2.2.1 연구의 문제점9

2.2.2 해결방안9

**3.프로젝트 소개10**

3.1 써브웨이 샌드위치의 주문 방식10

3.2 시나리오12

3.2.1 주 재료 추천12

3.2.2 주 재료에 어울리는 소스 추천 12

3.2.3 어플리케이션 화면에 “주 재료+소스”의 조합을 출력 13

3.3 요구사항12

3.3.1 개발환경12

3.3.2 안드로이드12

3.3.3 사이킷런13

3.3.4 데이터 수집13

3.3.5 코사인 유사도를 이용한 주 재료 선정13

3.3.6 MLP Classifier를 이용한 ‘소스’ 선정13

3.4 시스템 설계14

3.4.1 Use case diagram13

3.4.2 Sequence diagram13

3.4.1 Activity case diagram13

3.4.2 State diagram13

**4. 프로젝트 결과15**

4.1 안드로이드 앱15

4.2 Similarity Model15

4.3 MLP Classification Model15

4.4 성능평가15

**5.결론16**

5.1 기대효과16

5.2 추후 연구방향16

**6.참고문헌17**

**모바일(안드로이드) 앱 개발**

**써브웨이 샌드위치 맞춤형 조합 추천**

신민식(2017103999), 이민영(2018110653) / 지도교수 : 최진우 교수님

**요약**

세계에서 매장 수가 가장 많은 패스트푸드 전문점인 “써브웨이”는 빵, 소스 등의 재료 등을 전부 고객이 직접 선택하게 하는 것이 특징이다. 재료 선택이 복잡해서 불편함을 느끼던 사람들을 위해 사용자 개인의 입맛 취향을 알아내고, 취향에 맞는 샌드위치 재료 조합을 추천해주는 모바일 애플리케이션을 제안한다. 이 때, 유사도 분석과 기계학습을 이용한 학습모델을 통해 재료 조합 추천 기능을 제안하고 구현한다.

1. **서론**
   1. **연구배경**

세계에서 가장 매장 수가 많은 패스트푸드점인 “써브웨이”는 2021년 6월 기준 전 세계 104개국에 36,821개의 매장을 보유하고 있으며, 한국에만 총 504개의 매장이 존재한다 (2022년 9월 15일 기준, 써브웨이 코리아 사이트 참고). 써브웨이만의 가장 큰 차별점은 샌드위치에 들어가는 다양한 주재료, 빵, 소스등을 하나하나 고객이 직접 선택해야 한다는 점이다. 선택할 사항이 매우 많은 점이 써브웨이의 장점이지만, 오히려 이 특징이 단점으로 작용하여 주문 시간이 길어지고, 써브웨이 이용을 기피하는 고객들도 있을 것이라는 생각을 하게 되었다.

이것의 사실여부를 확인하기 위하여 써브웨이의 고객과 비고객(총 32명)을 대상으로 **사전 설문조사(네이버 폼 이용, 9/16~9/23 진행)**를 진행하였다. 우선 사람들의 “써브웨이” 이용 빈도를 조사해서, 적어도 3달에 한번씩 써브웨이를 이용하는 사람들은 고객, 그보다 써브웨이 이용 빈도가 적은 사람들은 비고객으로 간주하였다. 그리고 비고객에게는 “써브웨이 이용 경험이 없거나 적은 이유가 무엇인지”를 질문하였다.

그 결과, 총 답변자수 32명 중 써브웨이의 비고객은 17명, 그 중에서 “메뉴 선택이 복잡해서”를 이유로 꼽은 사람은 12명으로, (나머지 답변으로는 샌드위치 자체를 별로 안 좋아해서:3명, 가격이 비싸서:0명, 기타:2명) 써브웨이의 비고객이 된 가장 주된 이유는 “메뉴선택이 복잡”하기 때문이라는 결과가 나왔다. **[그림1]**

**[그림1]써브웨이 이용관련 사전 설문조사 결과**

위의 조사결과를 통하여, 사람들 각자의 입맛에 맞는 써브웨이 샌드위치 조합을 추천해주는 애플리케이션을 고안하게 되었다. 또한, 이 애플리케이션이 써브웨이의 복잡한 메뉴 선택에 어려움을 겪던 기존 써브웨이 고객층과 비고객층 모두에게 도움을 주기를 기대한다.

* 1. **연구목표**

이 연구의 목표는 써브웨이의 메뉴 조합을 선택할 때, 개인 취향에 맞춰 샌드위치 재료의 조합을 추천해주는 모바일 앱을 구현하는 것이다.

1. **관련연구**
   1. **기존의 유사 연구**
      1. **써브웨이(써브웨이코리아 공식 앱)**

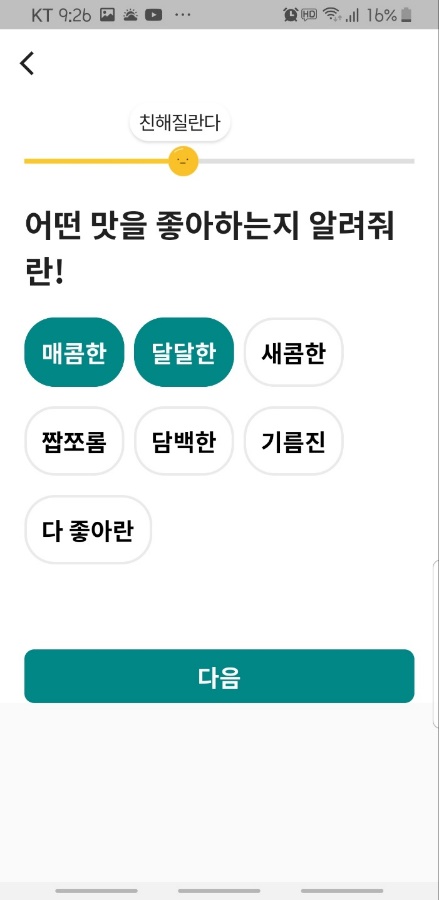
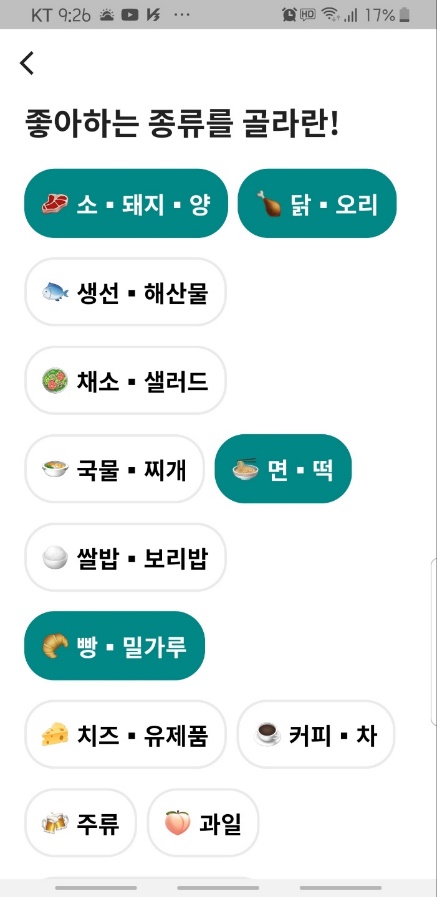
써브웨이 주문, 멤버쉽 포인트 적립, 할인 혜택 등의 기능을 가지고 있다. 기본적인 꿀 조합 추천은 존재하지만, 고객 개인의 취향에 맞춰서 조합을 추천해주는 기능은 존재하지 않는다.

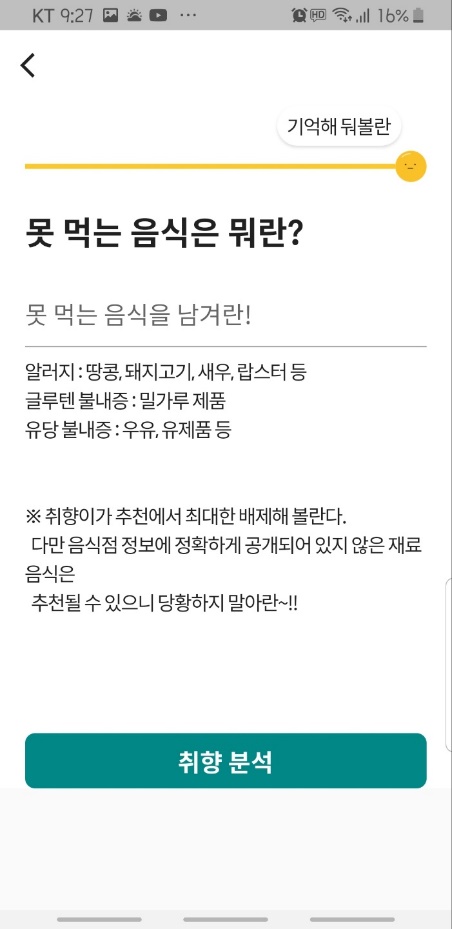
**2.1.2 미식큐, 맛있는취향(취향에 맞는 식사메뉴 추천 앱)**

미식큐**[그림5]**와 맛있는취향**[그림6]**은 일반적인 식사메뉴를 개인의 입맛에 따라 추천해주는 앱이다. 알러지 반응이 있는 음식은 미리 메뉴 추천에서 제외한다. 몇 가지 일반적인 식사메뉴들의 선호도를 아래와 같은 방법으로 조사하고, 그 내용을 토대로 취향에 맞는 메뉴를 추천해준다.



**[그림5]미식큐 입맛 데이터 수집 화면**

****

****

**[그림6]맛있는취향 입맛 데이터 수집 화면**

* 1. **기존 연구의 문제점과 해결 방안**
     1. **연구의 문제점**

기존의 써브웨이 공식 애플리케이션의 경우, 회사 차원에서 제공하는 기본적인 추천 조합이 존재하지만, 고객 “개인”의 입맛에 맞게 조합을 추천해주는 기능이 없다. 미식큐와 맛있는 취향의 경우 일반적인 식사메뉴를 대상으로 하지만, 음식의 종류가 다양하지 않고, GUI에 여백이 많아서 앱 자체가 어색해 보인다. 또한, 한 번 입맛/취향에 관해서 입력이 완료되면 그것을 다시 수정하는 것이 쉽지 않아서 입맛의 변화를 적용하기가 어렵다.

* + 1. **해결방안**

애플리케이션에서 추천하는 음식의 종류에 관해서는, 써브웨이에서 판매하는 모든 종류의 빵, 속재료 메뉴, 치즈, 소스 등을 전부 포함시키려고 한다.

앱 GUI의 경우, 써브웨이 공식 애플리케이션의 디자인을 참고하여 써브웨이의 브랜드 상징색인 vivid green(#008C15), marigold(#FFC600)를 위주로 디자인한다.

구상한 애플리케이션의 기능이 매우 단순하므로, 샌드위치 추천 조합을 하나가 아니라 여러 개(3개 정도) 제시한다. 어울리는 조합을 추천할 때, 보다 신뢰성을 확보하기 위해서 써브웨이 사용자가 선호하는 조합들을 직접 수집한다.

또한, 한 번 입맛에 관해 정보가 저장되었더라도, 원하는 때에 다시 수정할 수 있도록 앱을 제작한다.

그러므로 위의 내용을 포함해서, 기존에 없던 “써브웨이 샌드위치의 재료 조합을 사용자 개인의 취향에 맞춰 추천해주는 앱”을 구현하려고 한다.

1. **프로젝트 소개**
   1. **써브웨이 샌드위치의 주문 방식**

써브웨이 매장에서 샌드위치를 주문하는 순서는 다음과 같다.

1) 샌드위치의 주 재료를 선택한다.(18종류 중 택1)

2) 빵의 종류를 선택한다.(6종류 중 택1)

3) 치즈의 종류를 선택한다.(3종류 중 택1)

4) 추가할 재료가 있다면 선택한다.(7종류 중에서 선택)

5) 제외할 야채가 있다면 선택한다.(8종류 중에서 선택)

6) 소스를 선택한다.(15종류 중에서 복수 선택 가능)

* 1. **시나리오**

이 프로젝트의 주요 기능은 애플리케이션 사용자 각자의 입맛에 맞춰 써브웨이 샌드위치의 재료 조합(주 재료 + 소스)을 추천하는 것이다. 앱의 작동 순서는 아래와 같다.

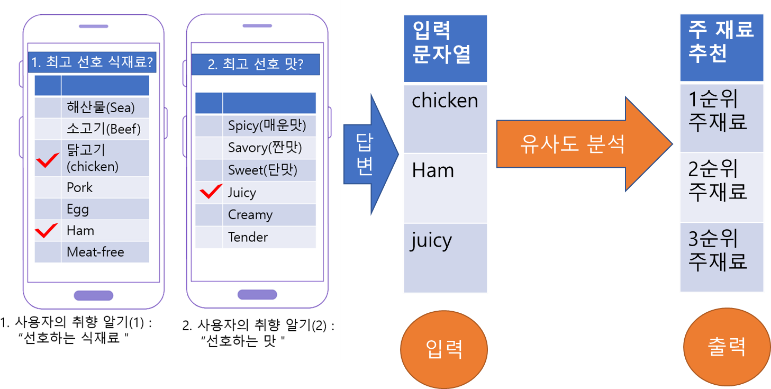
**3.2.1 주 재료 추천**

코사인 유사도 분석을 통해 주 재료를 추천한다. **[그림7]**

첫째, 앱 사용자에게 가장 선호하는 식재료와 가장 좋아하는 맛을 물어보고 답변(문자열 형태)을 받는다.

둘째, 위에서 받은 답변과 샌드위치 주재료(18종) 각각의 유사도를 분석한다.

셋째 , 가장 높은 유사도를 가진 주재료 1,2,3 순위를 뽑는다.



**[그림7]주 재료를 추천하는 과정**

**3.2.2** **주 재료에 어울리는 빵, 소스 추천**

직접 제작한 기계학습(Multi-label Classification)모델을 통해 주재료에 어울리는 소스를 추천한다. **[그림8]**

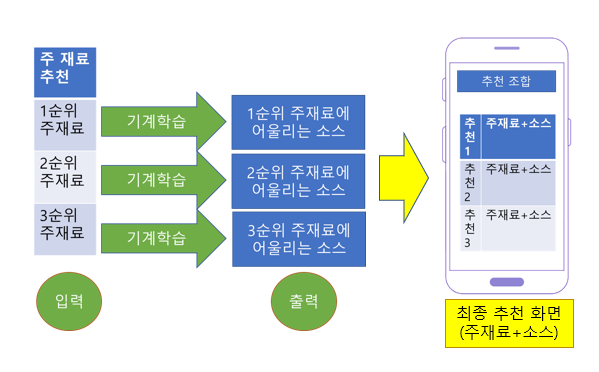
첫째, 3.2.1에서 뽑힌 1순위 주재료를 기계학습 모델에 입력하고, 1순위 주재료에 어울리는 소스를 출력한다.

둘째, 2순위 주재료에 어울리는 소스를 위와 동일한 방식으로 출력한다.

셋째, 3순위 주재료에 어울리는 소스를 위와 동일한 방식으로 출력한다.

**3.2.3 어플리케이션 화면에 “주 재료+소스”의 조합을 출력**

“1순위 주재료 + 어울리는 소스”, “2순위 주재료 + 어울리는 소스”, “3순위 주재료 + 어울리는 소스”를 최종적으로 애플리케이션 화면에 출력한다. **[그림8]**



**[그림8] 주 재료에 어울리는 소스를 추천하고, 최종 추천 조합(주 재료 + 소스)를 화면에 출력하는 과정**

* 1. **요구사항**
     1. **개발환경**

-운영체제: 안드로이드 5.1 (API 22)

-개발언어: JAVA , Python

-개발도구: 안드로이드 스튜디오, 주피터 노트북, 비주얼 스튜디오 코드

* + 1. **안드로이드**

모바일 앱을 구축하는 것이 목표이므로, 일단 모바일 전용 운영체제를 선택해야 한다. 안드로이드는 구글에서 만든 스마트폰용 운영체제로, 2021년 3월 기준 전세계 모바일 운영체제 점유율이 71%에 달하며, ‘오픈소스’라는 특징 덕분에 안드로이드 플랫폼의 단점을 빠르게 개선이 가능하다. 또한 Objective-C 나 Swift를 언어로 사용하는 iOS와 달리, Java를 언어로 사용해서 접근이 더 용이하다. 그리고 안드로이드 플랫폼 자체가 다양한 기능과 컴포넌트를 지원하고, ART라는 우수한 런타임이 탑재되어 있어서 안드로이드를 선택하였다.

* + 1. **사이킷런(Scikit-learn)**

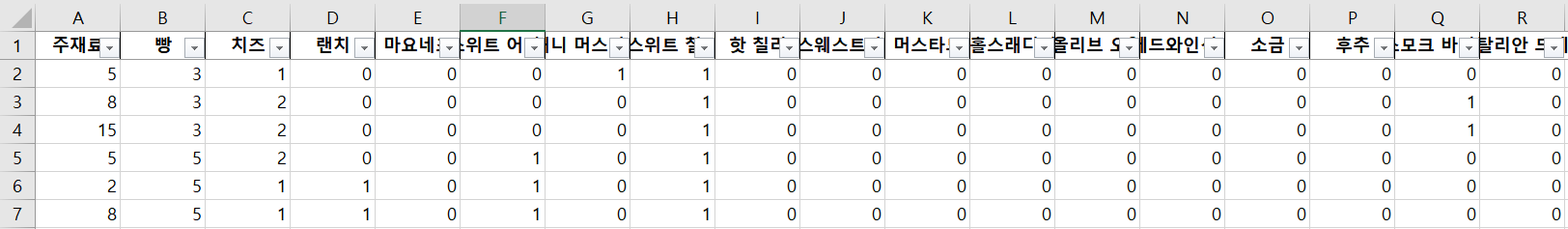
분류, 회귀, 차원 축소, 군집확 등과 관련된 머신 러닝 알고리즘을 제공하는 머신 러닝 라이브러리이다. 데이터 전처리를 위한 모듈, 특징 추출, 초매개변수 최적화, 모델 평가에 대한 모듈을 제공한다. 사이킷런은 파이썬 numpy와 scipy 라이브러리를 이용해 제작되었으며, numpy는 큰 배열과 고차원 행렬을 효과적으로 할 수 있게 파이썬을 확장해준다.

* + 1. **데이터 수집**

써브웨이 이용자들이 추천하는 “주 재료 + 소스”의 조합들을 데이터로 수집한다. 주재료/빵/치즈/소스의 종류 각각에 번호를 매기고**[그림9]** , 수집한 데이터는 엑셀로 정리해 기계학습 모델(주재료에 어울리는 소스추천)에 사용한다. **[그림10]**



**[그림9] 주 재료, 소스 각각에 번호를 매긴 모습**



**[그림10] “주재료 + 소스” 추천 조합** **데이터를 엑셀로 정리한 모습**

* + 1. **코사인 유사도를 이용한 ‘주 재료’ 선정**

**• TF-IDF 벡터화**

TF-IDF(Term Frequency – Inverse Document Frequency) 인코딩은 단어의 개수를 그대로 세지 않고 모든 문서에 공통적으로 포함되어 있는 단어의 경우, 문서 구별 능력이 떨어진다고 판단해 오히려 가중치를 축소하는 방법이다. 따라서 TF-IDF 값이 낮으면 중요도가 낮은 것이며, TF-IDF 값이 크면 중요도가 큰 것이다.

아래와 같은 예시가 있다고 가정해보자.

***문서1 : 먹고 싶은 사과  
문서2 : 먹고 싶은 바나나  
문서3 : 길고 노란 바나나 바나나  
문서4 : 저는 과일이 좋아요***

그렇다면 일반적인 DTM(Document-Term Matrix, 문서 단어 행렬)은 다음과 같다. **[그림11]**



**[그림11] 예시 문서에 대한 DTM**

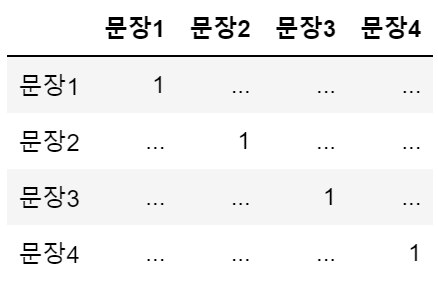
TF-IDF계산 결과는 다음과 같다.



**[그림12] 예시 문서에 대한 TF-IDF**

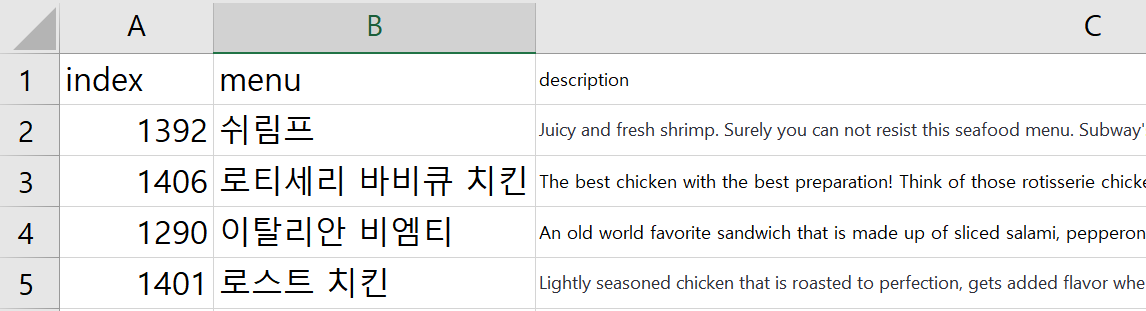
이를 구현하기 위해 Scikit-learn API의 TfidfVectorizer를 이용한다.

TF-IDF Matrix가 생성되면 Scikit-learn API의 linear\_kernel을 이용하여 코사인 유사도를 계산할 수 있게 된다. 결과는 다음과 같은 형태를 가진다. **[그림13]**



**[그림13] 코사인 유사도 예시**

이 프로젝트에서는 사용자의 취향을 문자열 형태로 입력 받아서 각 주재료에 대한 설명 문자열**[그림14]**과 비교한 후, 유사도가 가장 높은 주재료를 선정한다.



**[그림14] 각 주재료(메뉴)에 대한 설명 문자열**

**3.3.6 Multi-label classification(다중 레이블 분류)**

머신 러닝 모델의 입력은 주재료(메뉴)이고, 이에 대한 출력은 소스이다. 하지만 하나의 메뉴에도 여러 종류의 소스가 들어갈 수 있다. 따라서 우리는 Multi-label classification을 할 필요가 있다. 한 영화가 액션, 어드벤처, 로맨스 등 여러 개의 장르에 동시에 해당될 수 있는 것처럼 Multi-label classification은 결과 값이 여러 label에 해당될 수 있다.

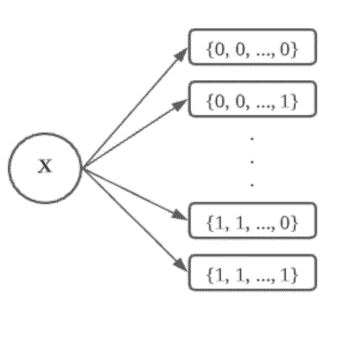
Multi-label classification는 여러 가지 방법으로 구현이 될 수 있는데, 대표적으로Binary Relevance, Classifier chain, Label Powerset 등이 있다.

**• Binary Relevance**

Binary Relevance는 결과에 해당하는 label들을 분리하여 각각의 single class로 표현한다. 그 다음, 입력에 대해 각 label 별로 single-class classification을 한 후에 이를 합치게 된다. 하지만 이 방법의 문제점은 각 label이 독립적이지 않을 수 있음에도 이를 무시한다는 것이다. 이를 보완하기 위한 방법으로 label powerset, classifier chain 등이 있다.

**• Label Powerset**

Label Powerset은 label의 가능한 모든 조합들을 각각 하나의 class로 치환하여 classification을 수행한다. 이를 그림으로 표현하면 다음과 같다.[그림15]

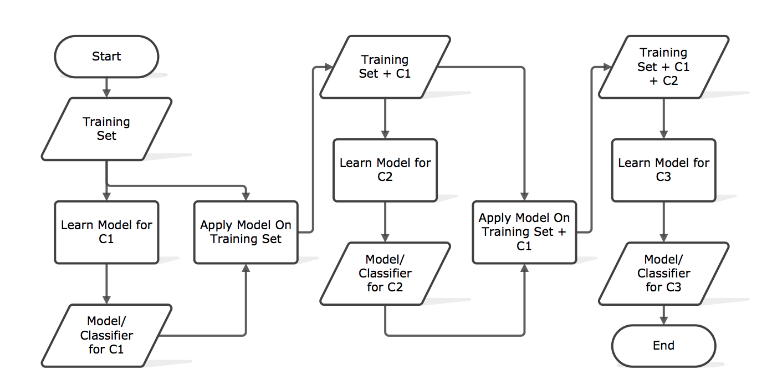


**[그림15] Label Powerset**

이렇게 label 집합을 예측하면 label들 간의 관계도 포함하여 예측이 가능하다는 장점이 있다. 하지만 공간 복잡도가 높다는 단점이 있다. Label이 d개 있다면 가능한 label 집합은 2d개가 된다.

**• Classifier Chain**

Classifier Chain에서는 여러 개의 classifier을 사용한다. 순서에 따라 레이블들을 예측하는데 한 레이블을 예측할 때, 이전에 예측한 레이블들까지 포함하여 예측에 활용한다. 이를 그림으로 표현하면 다음과 같다.[그림16]



**[그림16] Classifier Chain**

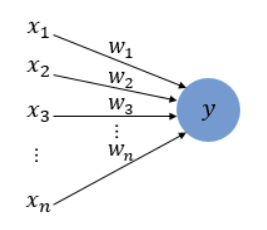
이 방법은 레이블 간의 관계도 담아내면서도 공간 복잡도를 줄일 수 있다는 장점이 있다. 하지만 label 예측 순서에 따라 성능이 달라질 수 있다는 단점이 있다.

* + 1. **MLP(Multi-layer Perceptron) Classifier를 이용한 ‘소스’선정**

Scikit-learn에서 제공하는 MLP Classifier는 다중 레이블 분류를 지원하므로, 이를 기계학습 모델에 활용한다.

**• 퍼셉트론**

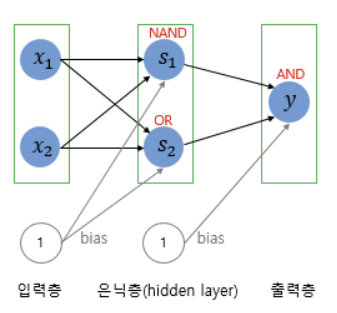
퍼셉트론(Perceptron)은 초기 형태의 인공 신경망으로 다수의 입력으로부터 하나의 결과를 내보내는 알고리즘이다. 다수의 입력을 받는 퍼셉트론의 모습은 아래와 같다.[그림17] x는 입력값을 의미하며, w는 가중치(Weight), y는 출력값이다. 각각의 입력값에는 각각의 가중치가 존재하는데, 이때 가중치의 값이 크면 클수록 해당 입력 값이 중요하다는 것을 알 수 있다.



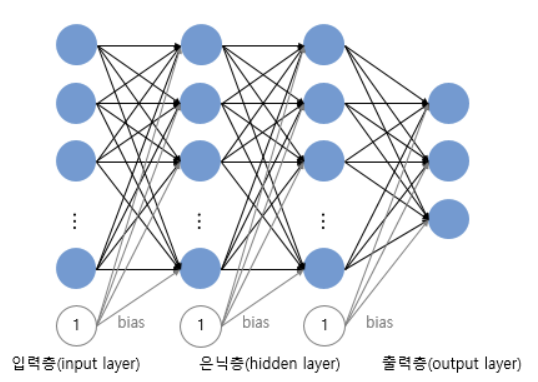
**[그림17]퍼셉트론**

**• 다층 퍼셉트론(MLP)**

다층 퍼셉트론(multi-layer perceptron, MLP)는 퍼셉트론으로 이루어진 층 (layer) 여러 개를 순차적으로 붙여놓은 형태이다. **[그림18]** 단층 퍼셉트론은 입력층과 출력층만 존재하지만, 다층 퍼셉트론은 중간에 층을 더 추가하며, 이렇게 입력층과 출력층 사이에 존재하는 층을 **은닉층(hidden layer)**이라고 하는데, 은닉층은 여러 개일 수도 있다.**[그림19]** 이때, 은닉층은 훈련 데이터에서는 관측하지 못했던 잠재변수를 나타낸다. 일반적으로 각 계층은 그 후속 계층과 완전연결 관계에 있으며, 각 계층에서의 출력은 그 다음 계층의 입력이 된다.



**[그림18]다층 퍼셉트론의 예시**



**[그림19]은닉층이 여러 개인 다층 퍼셉트론**

이전 단계에서 유사도분석을 통해 선정된 주재료(메뉴)에 어울리는 소스를 추천해주는 기계학습 모델을 scikit-learn 의 Multi-layer Perceptron을 사용해 제작한다.

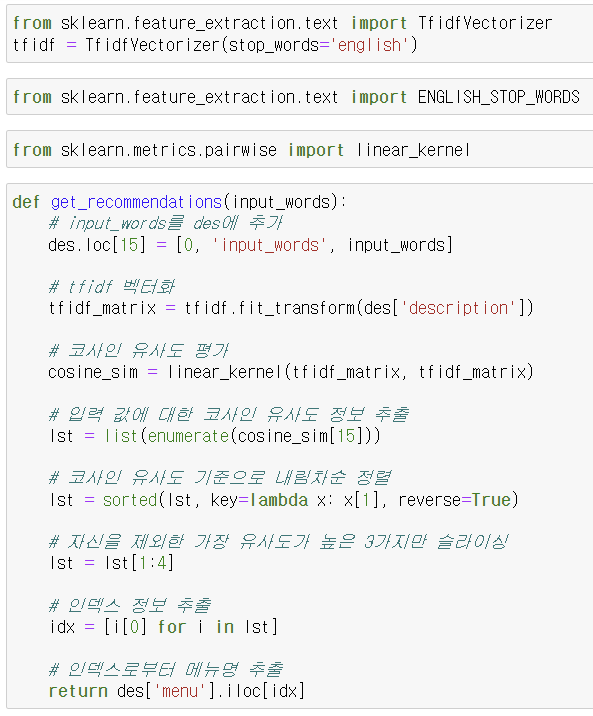
* 1. **시스템 설계 🡪멘토님께 질문함**
     1. **Use case diagram**
     2. **Suequeunce diagram**
     3. **Activitiy diagram**
     4. **State diagram**

1. **프로젝트 결과**
   1. **안드로이드 앱**

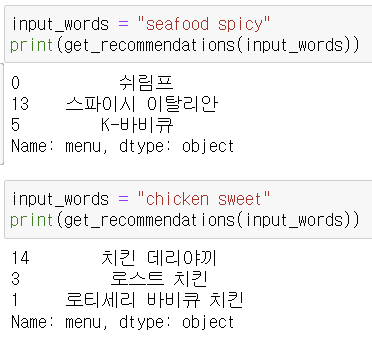
|  |  |  |
| --- | --- | --- |
| **레이아웃** | **앱 출력 화면** | **설명** |
| **1** |  | **(🡨 이미지 수정 예정)**  **표지의 역할을 하는 시작화면이다.**  **시작하기 버튼을 눌러서 다음 레이아웃으로 전환이 가능하다.** |
| **2** |  | **시작하기 버튼을 누른 후에 나오는 화면의 모습이다. 앱 사용자가 선호하는 식재료를 선택하도록 한다. 여기서 한 선택이 추천할 ‘주재료(메뉴)’를 좌우한다. 선택지에는 복수 선택이 가능하도록 체크박스를 적용했다. 이전화면 버튼과 다음화면 버튼을 통해 레이아웃 이동이 가능하다.** |
| **3** |  | **2페이지에서 다음화면 버튼을 누른 후에 나오는 화면의 모습이다. 앱 사용자가 선호하는 맛을 선택하도록 한다. 여기서 한 선택이 추천할 ‘주재료(메뉴)’를 좌우한다. 선택지에는 복수 선택이 가능하도록 체크박스를 적용했다. 이전화면 버튼과 다음화면 버튼을 통해 레이아웃 이동이 가능하다.** |
| **4** |  | **(🡨 이미지 수정 예정)**  **3페이지에서 다음화면 버튼을 누른 후에 나오는 화면의 모습이다. 주재료 + 소스의 추천 조합 1,2,3 순위를 결과로 제시한다.** |

**[그림17] 안드로이드 앱 출력 화면**

* 1. **Similarity Model**

****

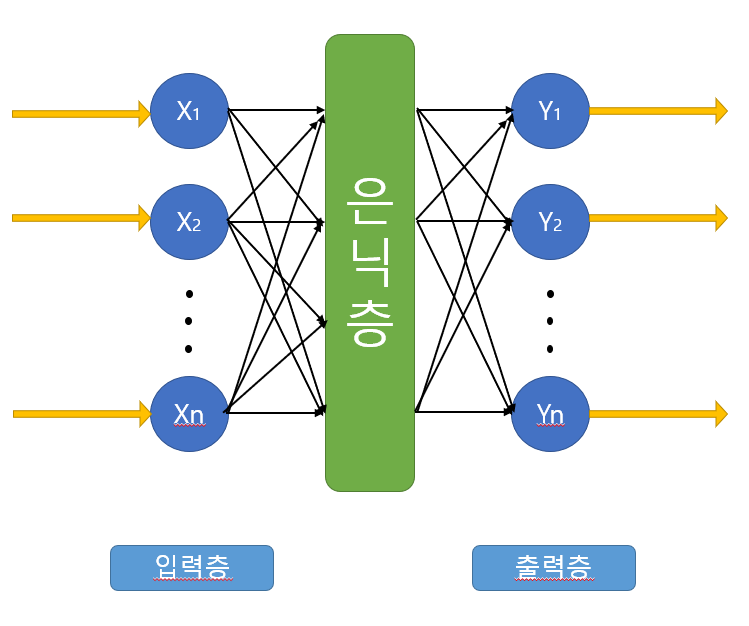
**[그림18]유사도 분석 모델**

****

**[그림19]유사도 분석 모델의 입력과 출력**

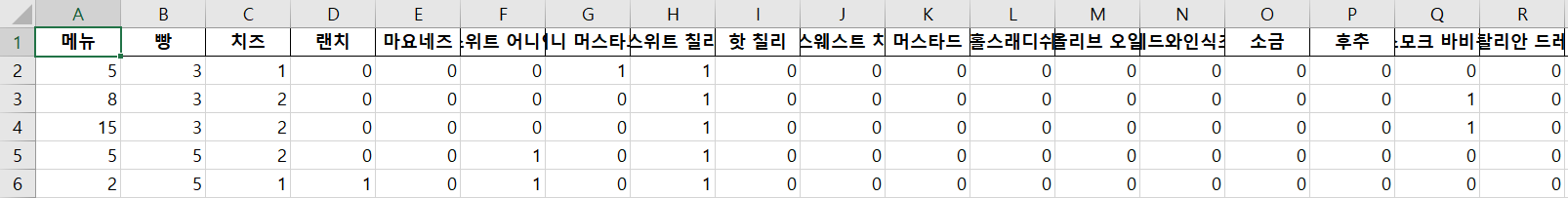
주재료(메뉴)에 대한 설명을 TF-IDF 벡터화하고[그림 18], 예를 들어 입력값을 “seafood spicy” 처럼 “식재료 + 맛”의 조합으로 입력하면 유사도분석을 통해 가장 연관도가 높은 주재료 1,2,3 순위를 추천해준다.

* 1. **MLP Classifier Model**



**[그림20] MLP Classifier Model의 동작**

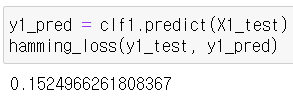
초기의 목표였던 “입력 X : [주재료] 🡪 출력 y : [소스]” 모델을 먼저 제작한 뒤, 모델의 성능을 좀 더 높이기 위해서 “입력 X : [주재료,빵,치즈] 🡪 출력 y : [소스]” 모델과 “입력 X : [주재료,빵,치즈,주재료에 대한 설명(td-idf)] 🡪 출력 y : [소스]” 모델을 추가로 진행했다.

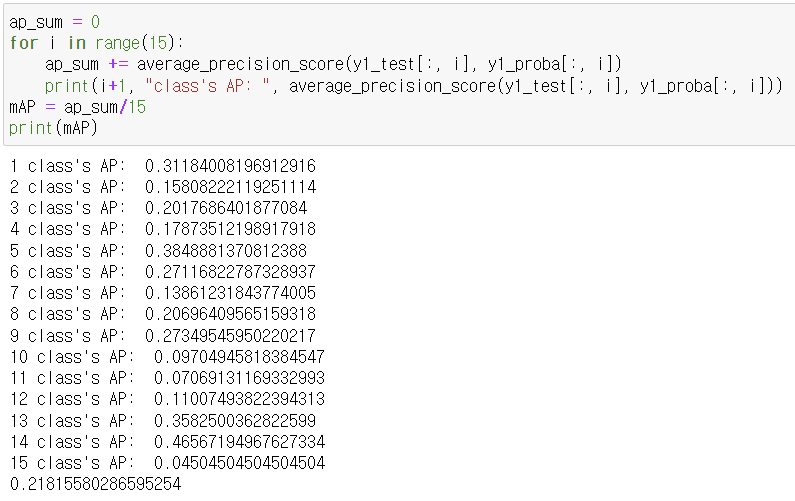


[그림21]입력 데이터

**• 입력 X : [주재료] 🡪 출력 y : [소스]**





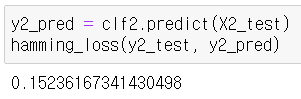


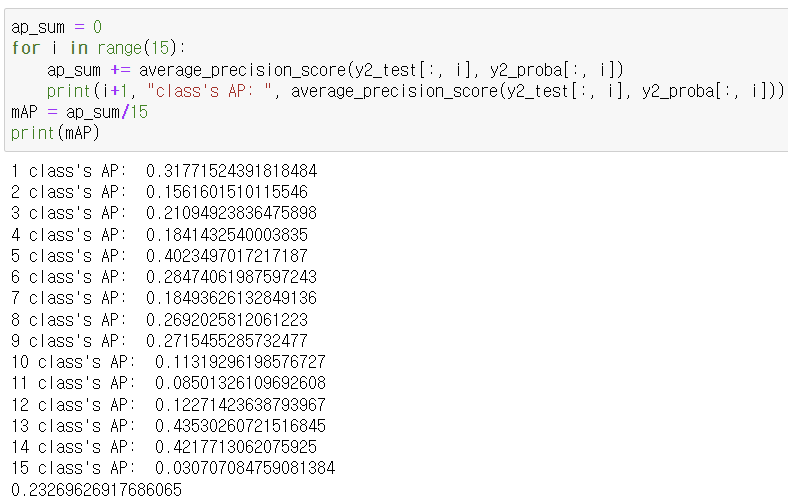
* **Hamming loss는 0.1524966261808367,**

**AP는 0.21815580286595254 가 나왔다.**

**• 입력 X : [주재료,빵,치즈] 🡪 출력 y : [소스]**



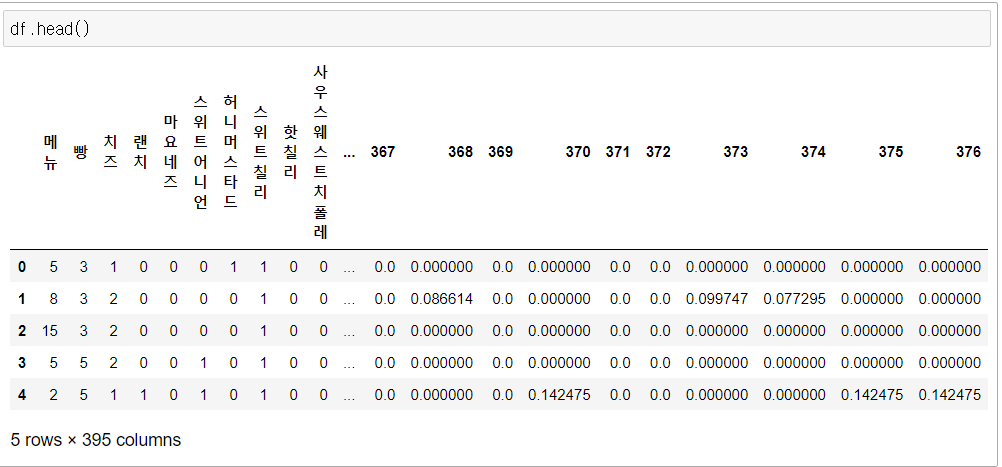




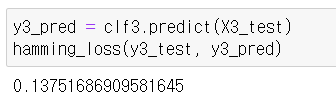
* **Hamming loss는 0.15236167341430498**

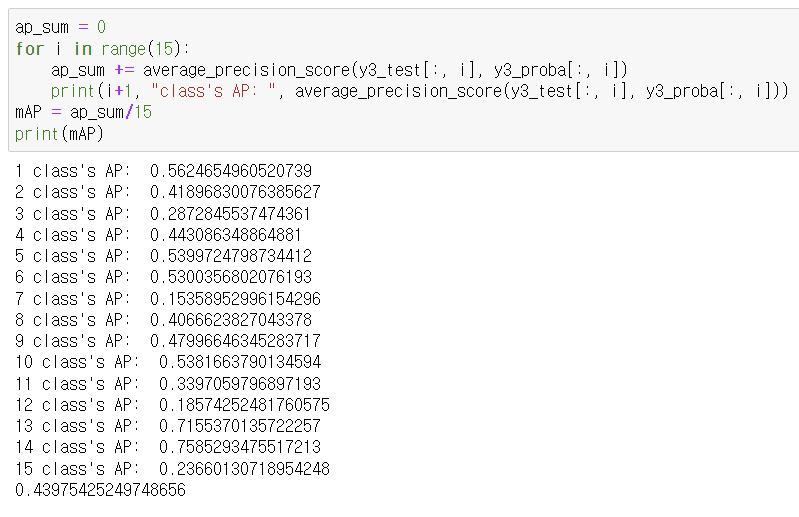
**AP는 0.23269626917686065가 나왔다.**

**• 입력 X : [주재료,빵,치즈,주재료에 대한 설명(td-idf)] 🡪 출력 y : [소스]**



[그림22] make\_input\_data.jpynb **으로 [그림21]입력데이터에 “주재료에 대한 설명”을 추가한다.**





* **Hamming loss는 0.13751686909581645,**

**AP는 0.43975425249748656가 나왔다.**

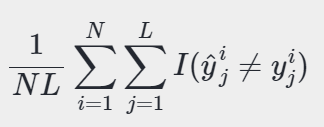
**결론적으로, 입력값이** [주재료], [주재료, 치즈, 빵], [주재료, 치즈, 빵, 설명] 일때

뒤로 갈수록 성능이 우수하게 나왔다.

**\*hamming loss 와 ap에 대한 설명은 위치를 옮기는 것을 고려중**

**• Hamming Loss**

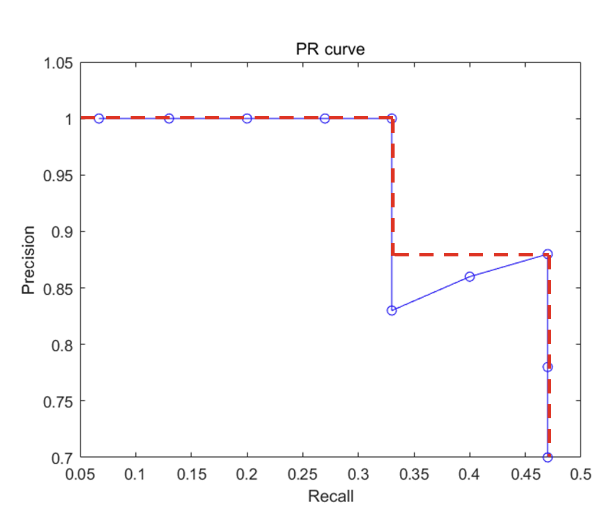
전체 label 중 잘못 분류된 label을 의미한다.



[그림19]hamming loss

**• Average Precision**

Average Precision은 인식 알고리즘의 성능을 하나의 값으로 표현한 것으로 precision-recall 그래프 선 아래 쪽의 면적으로 계산 된다. **[그림20]** Average Precision이 높으면 높을 수록 그 알고리즘의 성능이 전체적으로 우수하다는걸 의미한다.



**[그림20]** precision-recall 그래프

1. **결론**

**5.1 기대효과**

써브웨이는 신선한 재료와 다양한 선택이 가능한 건강한 패스트푸드의 대명사나 다름없다. 그 결과로 써브웨이는 전 세계에서 가장 많은 수의 점포를 가진 프랜차이즈 식당으로 자리잡았다. 한국에서 전국에 있는 써브웨이 매장 개수가 500개 이상임에도 불구하고, 샌드위치 조합을 주문하는 방식이 복잡해 써브웨이 이용을 꺼리는 사람들이 많이 있다. 본 프로젝트는 이러한 사람들의 불편함을 해소할 뿐만 아니라, 개인의 입맛에 딱 맞는 조합을 추천하여 사람들이 자신의 입에 가장 맛있는 샌드위치를 맛보길 기대한다.

**5.2 추후 연구방향**

써브웨이 공식 애플리케이션과 연계하여, 추천된 샌드위치 조합을 곧바로 주문할 수 있도록 추진한다면 더욱 좋을 것이다. 추천된 메뉴의 조합이 2가지 이상이기 때문에 애플리케이션의 추천 조합을 모두 한번씩 맛을 보기가 쉽다. 그 조합들 중에서 가장 본인의 입맛에 맞는 조합을 찾고, 그 조합을 애플리케이션에 저장하도록 만든다면, 나중에 다시 주문할 때 하나하나 고를 필요 없이 저장된 조합을 그대로 주문 가능하도록 하면 편리할 것이다.

1. **참고문헌**

<논문 및 관련서적>

[1] 창의적 설계 프로젝트 (조진성 지음)

[2] 모바일 웹 및 오픈소스 SW 기반의 기초 프로그래밍 (문현준 지음)

[3] 단계별로 배우는 안드로이드 프로그래밍 (김선우 지음)

[4] Do it! 안드로이드 앱 프로그래밍 (정재곤 지음)

[5] 작게 시작하여 크게 성공하라 (프레드 드루카 외 지음)

[6] 이것이 빅데이터 기업이다 - 현실화되고 있는 미래기업들 (함유근 지음)

[7] Scikit-learn으로 머신 러닝 마스터-2ed (개빈 해클링 지음)

[8] 파이썬 머신러ㅣㄴㅇ 완벽가이드 (권철민 지음)

[9]

<애플리케이션>

[1] 써브웨이 : <https://play.google.com/store/apps/details?id=kr.co.subway.preorder>

[2] 맛있는취향 : <https://play.google.com/store/apps/details?id=com.mytaste>

[3] 미식큐 : <https://play.google.com/store/apps/details?id=kr.misiq>

<인터넷 사이트>

[1] 전 세계에 존재하는 써브웨이 매장 수(2022-09-15) :

<https://www.subway.com/en-US/ExploreOurWorld>

[2] 써브웨이 코리아 공식 사이트(2022-09-15) : <https://www.subway.co.kr/>

[3] 안드로이드 점유율:

<https://gs.statcounter.com/os-market-share/mobile/worldwide>

[4] 써브웨이 프로세스:

<https://www.slideshare.net/afzalusmani1/subway-55134668>

[5] 써브웨이 브랜드 칼라: <https://usbrandcolors.com/subway-colors/>

[6] 무료 스마트폰 이미지소스(픽사베이): <https://pixabay.com/>

[7] 머신러닝 관련: <https://ikkison.tistory.com/51?category=785277>

[8] average precision : <https://lapina.tistory.com/98>

[9] scikit-learn : https://scikit-learn.org/