**써브웨이 샌드위치**

**맞춤형 조합 추천**



|  |  |
| --- | --- |
| **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**4

1.1 연구배경4

1.2 연구목표6

**2.관련연구6**

2.1 기존의 유사 연구6

2.1.1 써브웨이6

2.1.2 미식큐, 맛있는취향6

2.2 기존 연구의 문제점과 해결 방안8

2.2.1 연구의 문제점8

2.2.2 해결방안8

**3.프로젝트 소개9**

3.1 시나리오9

3.1.1 써브웨이 샌드위치의 주문 방식9

3.1.2 데이터 수집9

3.1.3 사용자의 취향에 맞는 “주재료” 추천10

3.1.4 주재료에 어울리는 “소스” 추천11

3.2 요구사항12

3.2.1 개발환경12

3.2.2 사이킷런 (Scikit-learn)12

3.2.3 코사인 유사도를 이용한 “주재료” 추천13

3.2.4 다층 퍼셉트론을 이용한 “소스” 추천14

**4. 프로젝트 결과16**

4.1 유사도 분석 모델 (Similarity Model)16

4.2 다층 퍼셉트론 모델 (MLP Classification Model)18

4.3 평가 지표25

**5.결론26**

5.1 기대효과26

5.2 추후 연구방향26

**6.참고문헌27**

**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 느끼고 써브웨이 이용을 꺼려하던 사람들이 손쉽게 메뉴를 주문할 수 있도록, 사용자 개인의 입맛 취향을 선택하면, 취향에 맞는 샌드위치 재료 조합을 추천해주는 연구를 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 신선하고 다양한 주재료, 빵, 소스등을 하나하나 고객이 직접 선택하여 즉석에서 만들어 제공하는 건강식이라는 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 개인 맞춤형 써브웨이 샌드위치 조합을 추천해주는 프로젝트를 고안하게 되었다. 또한, 이 프로그램이 써브웨이의 복잡한 메뉴 선택에 어려움을 겪던 기존 써브웨이 고객층과 비고객층 모두에게 도움을 주기를 기대한다.

* 1. **연구목표**

이 연구의 목적은 써브웨이 샌드위치 매장에서 개인 맞춤형 샌드위치 재료의 조합을 사용자가 쉽게 선택할 수 있도록 돕는 프로그램을 머신러닝을 이용해 제작한다. 따라서 목표 설정은 다음과 같다.

목표 1 : 기존의 써브웨이 고객 층이 선호하던 샌드위치 재료의 조합 데이터를 수집한다.

목표 2: 목표1 에서 수집한 데이터를 사용해 서로 어울리는 샌드위치 조합을 추천해주는 머신러닝 모델을 제작한다.

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

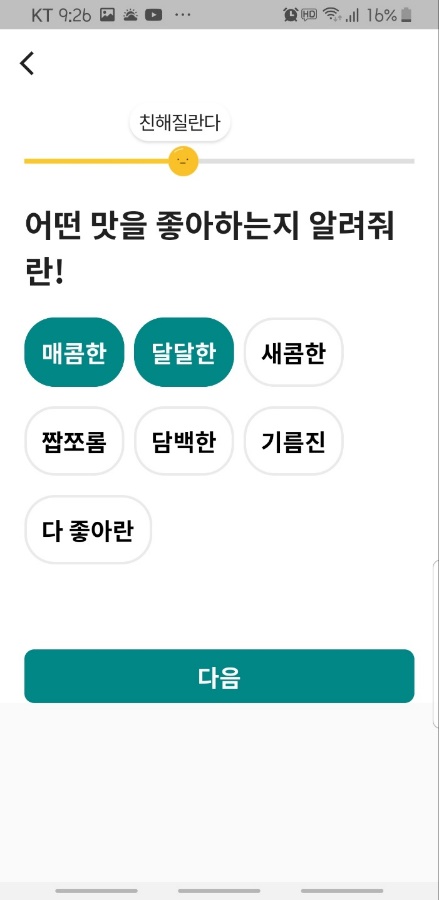
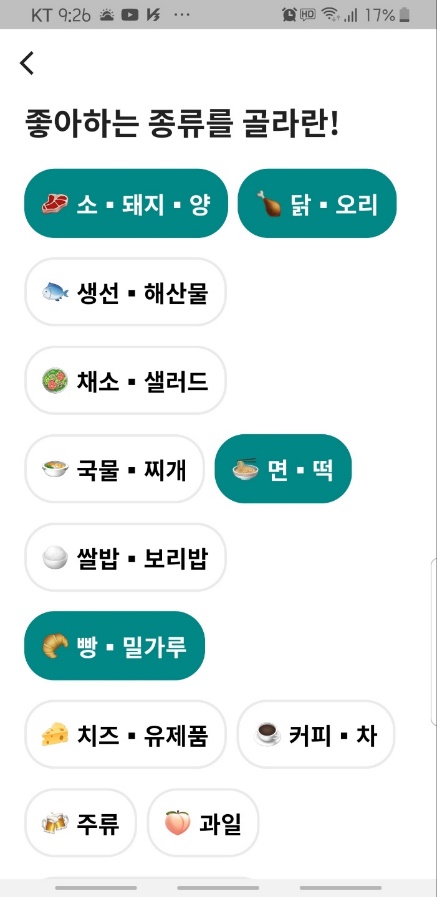
써브웨이 주문, 멤버쉽 포인트 적립, 할인 혜택 등의 기능을 가지고 있다. 기본적인 꿀 조합 추천은 존재하지만, 고객 개인의 취향에 맞춰서 조합을 추천해주는 기능은 존재하지 않는다.

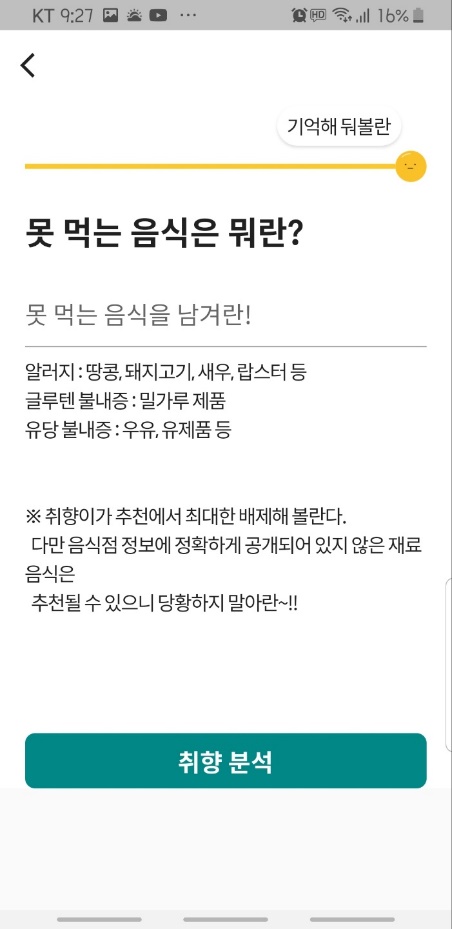
**2.1.2 미식큐, 맛있는취향(취향에 맞는 식사메뉴 추천 앱)**

미식큐**[그림2]**와 맛있는취향**[그림3]**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.



**[그림2] 미식큐 입맛 데이터 수집 화면**

****

****

**[그림3] 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, 추천해주는 음식 종류가 많지 않고, 한 번 입맛에 관해서 입력이 완료되면 그것을 다시 수정하는 것이 쉽지 않아서 입맛의 변화를 적용하기가 어렵다.

* + 1. **해결방안**

본 프로젝트에서는 음식의 종류 자체가 써브웨이 샌드위치로 한정되므로, 추천 조합을 하나가 아니라 여러 개(3개) 제시하는 방법을 사용한다. 또한, 입력값의 수정이 용이하게끔 프로젝트를 진행한다.

1. **프로젝트 소개**
   1. **시나리오**

이 프로젝트의 주된 목표는 기존의 써브웨이 고객 증이 선호하던 샌드위치 재료의 조합 데이터를 수집하고, 이 때 수집한 데이터를 사용해 어울리는 샌드위치 조합을 추천해주는 머신러닝 모델을 제작하는 것이다.

**3.1.1 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)

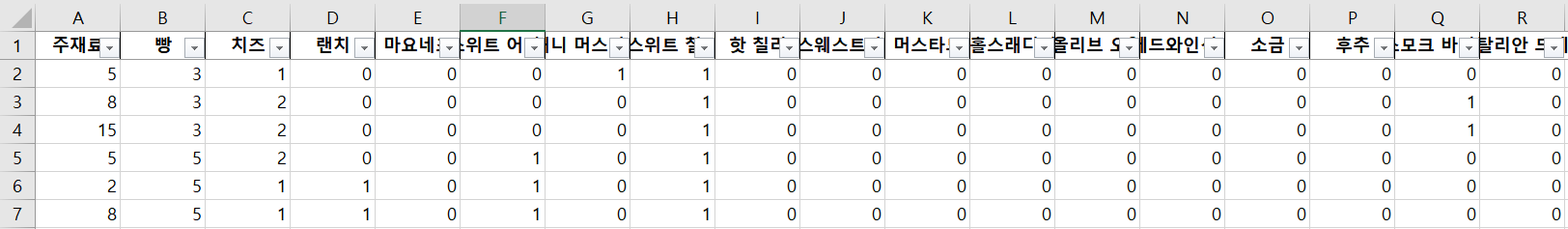
6) 소스를 선택한다.(15종류 중에서 복수 선택 가능)

* + 1. **데이터 수집**

써브웨이 이용자들이 추천하는 “주 재료 + 소스”의 조합들을 데이터로 수집한다. 주재료/빵/치즈/소스의 종류 각각에 번호를 매기고**[그림4]** , 수집한 데이터는 엑셀로 정리해 기계학습 모델(주재료에 어울리는 소스추천)에 사용한다. **[그림5]**



**[그림4] 주 재료, 소스 각각에 번호를 매긴 모습**



**[그림5] “주재료 + 소스” 추천 조합** **데이터를 엑셀로 정리한 모습**

**3.1.3 사용자의 취향에 맞는 “주 재료” 추천**

코사인 유사도 분석을 통해 주 재료를 추천한다. **[그림6]**

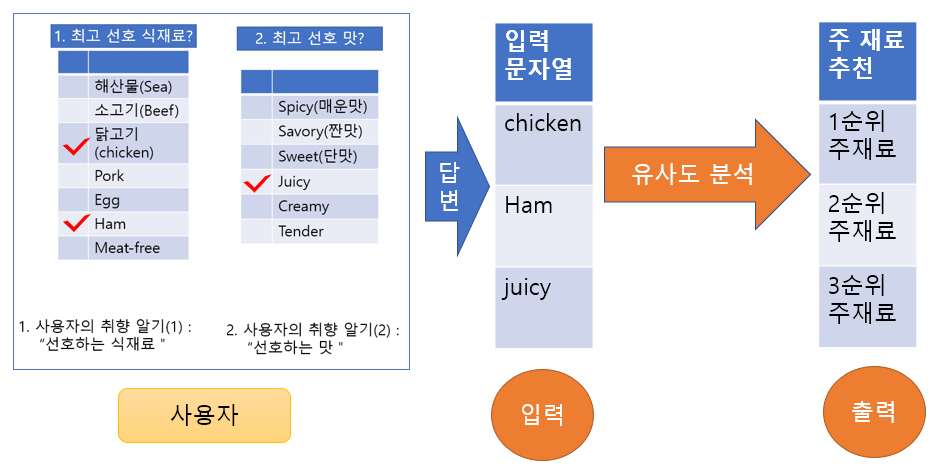
첫째, 사용자가 “선호하는 식재료”와 “선호하는 맛”을 문자열 형태로

입력 받는다. (예 : seafood spicy, chicken sweet)

둘째, 위에서 입력 받은 문자열과 샌드위치 주재료(18종)의 설명에 대해

유사도 분석을 진행한다.

셋째 , 가장 높은 유사도를 가진 주 재료 1,2,3 순위를 뽑는다.



**[그림6] 주 재료를 추천하는 과정**

**3.1.4** **주 재료에 어울리는 “소스” 추천**

직접 제작한 기계학습 모델을 통해 위에서 추천받은 주 재료에 어울리는 소스를 추천한다. **[그림7]**

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

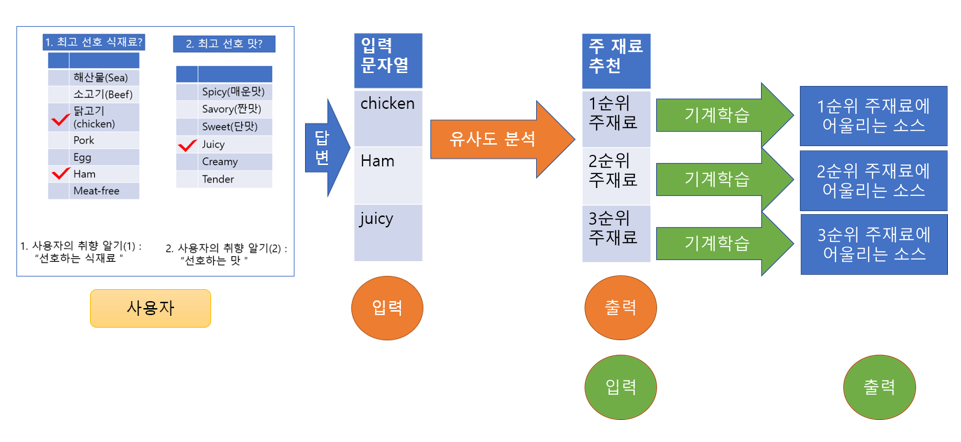
둘째, 2순위 주재료에 어울리는 소스를 위와 동일한 방식으로 출력한다.

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



**[그림7] 주 재료에 어울리는 소스를 추천하는 과정**

위의 내용을 하나로 정리하면 아래와 같다. **[그림8]**



**[그림8] 프로젝트 시나리오**

**3.2 요구사항**

**3.2.1 개발환경**

-개발언어: Python

-개발도구: 주피터 노트북, 비주얼 스튜디오 코드

* + 1. **사이킷런(Scikit-learn)**

분류, 회귀, 차원 축소, 군집화 등과 관련된 머신 러닝 알고리즘을 제공하는 파이썬 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.

**3.2.3 코사인 유사도를 이용한 ‘주재료’ 추천**

**• 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. **[그림9]**



**[그림9] 예시 문서에 대한 DTM**

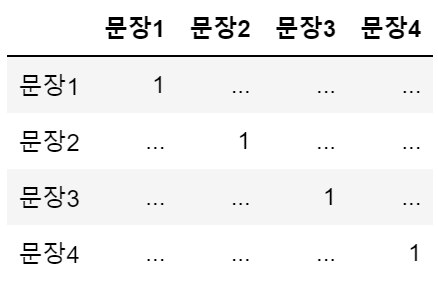
TF-IDF계산 결과는 다음과 같다.



**[그림10] 예시 문서에 대한 TF-IDF**

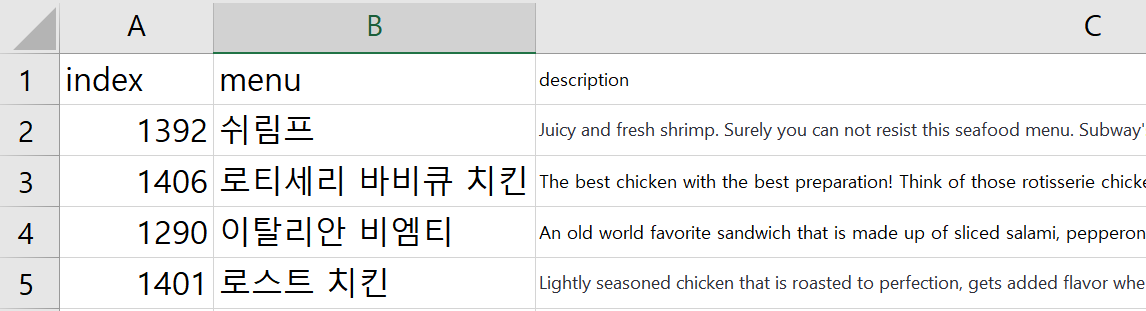
이를 구현하기 위해 Scikit-learn API의 TfidfVectorizer를 이용한다.

TF-IDF Matrix가 생성되면 Scikit-learn API의 linear\_kernel을 이용하여 코사인 유사도를 계산할 수 있게 된다. 결과는 다음과 같은 형태를 가진다. **[그림11]**



**[그림11] 코사인 유사도 예시**

이 프로젝트에서는 사용자의 취향을 문자열 형태로 입력 받아서 각 주재료에 대한 설명**[그림12]**과 비교한 후, 유사도가 가장 높은 주재료를 선정한다.



**[그림12] 각 주재료(메뉴)에 대한 설명 문자열**

**3.2.4 다층 퍼셉트론(MLP)을 이용한 ‘소스’ 추천**

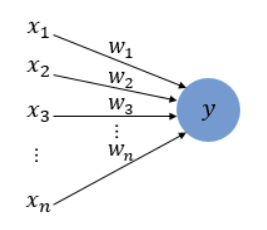
Scikit-learn에서 제공하는 MLP Classifier는 다중 레이블 분류(Multi-label Classification)를 지원하므로, 이를 기계학습 모델에 활용한다.

**• 다중 레이블 분류(Multi-label Classification)**

입력값을 x, 출력값을 y로 했을 때, 다중 레이블 분류는 인스턴스(x)가 여러 개의 레이블(y)과 연결될 수 있는 지도 학습 문제이다. 즉, 여러 개의 클래스가 동시에 정답이 될 수 있는 것을 의미한다. y의 여러 클래스에 대해 0 또는 1의 값을 할당하게 되는데, 본 프로젝트에서는 y의 각 클래스를 소스의 종류(15가지)로 두고, 추천 조합에 포함되는 소스는 1, 포함되지 않는 소스는 0으로 표시한다.

**• 퍼셉트론(Perceptron)**

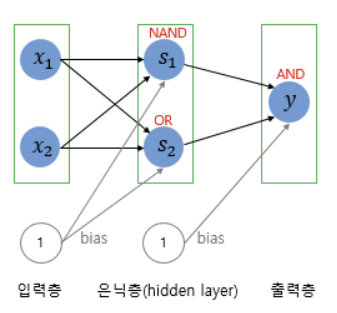
퍼셉트론(Perceptron)은 초기 형태의 인공 신경망으로 다수의 입력으로부터 하나의 결과를 내보내는 알고리즘이다. 다수의 입력을 받는 퍼셉트론의 모습은 아래와 같다**.[그림13]** x는 입력 값을 의미하며, w는 가중치(Weight), y는 출력 값이다. 각각의 입력 값에는 각각의 가중치가 존재하는데, 이때 가중치의 값이 크면 클수록 해당 입력 값이 중요하다는 것을 알 수 있다.



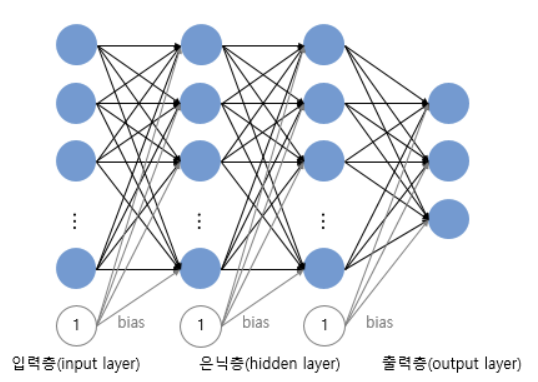
**[그림13]퍼셉트론**

**• 다층 퍼셉트론(MLP)**

다층 퍼셉트론(multi-layer perceptron, MLP)는 퍼셉트론으로 이루어진 층 (layer) 여러 개를 순차적으로 붙여놓은 형태이다. **[그림14]** 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.**[그림15]** 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.



**[그림14]다층 퍼셉트론의 예시**

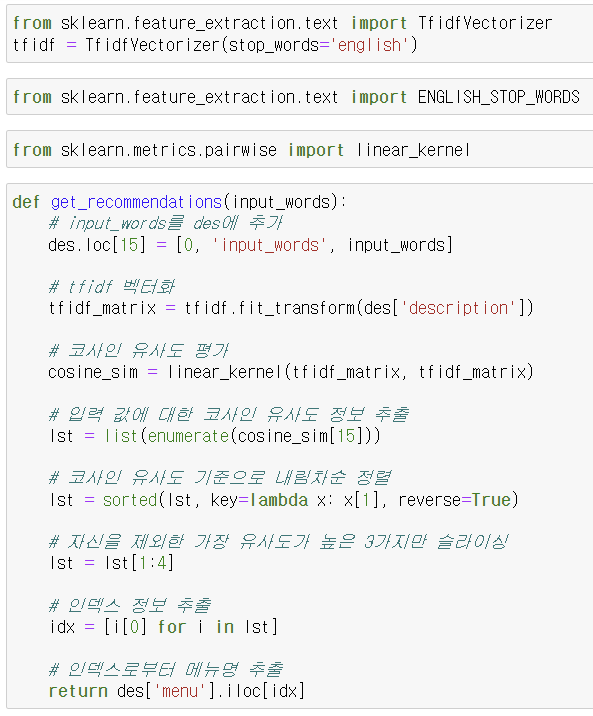


**[그림15]은닉층이 여러 개인 다층 퍼셉트론**

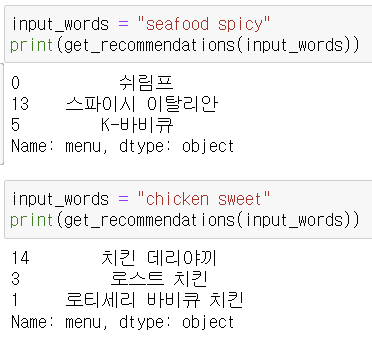
이전 단계에서 유사도분석을 통해 선정된 주재료(메뉴)에 어울리는 소스를 추천해주는 기계학습 모델을 scikit-learn의 MLPClassifier을 사용해 제작한다.

1. **프로젝트 결과**
   1. **유사도 분석 모델(Similarity Model)**

사용자의 취향(“선호하는 식재료 + 선호하는 맛”)을 문자열 형태로 입력 받아서 각 주재료에 대한 설명과 유사도분석을 진행하고, 이 때 유사도가 높은 순서대로 주재료 3가지를 추천한다. 여기서 유사도분석을 진행하기 위한 “유사도 분석 모델” 을 제작한 결과는 다음과 같다. **[그림16, 17]**

****

**[그림16]유사도 분석 모델**

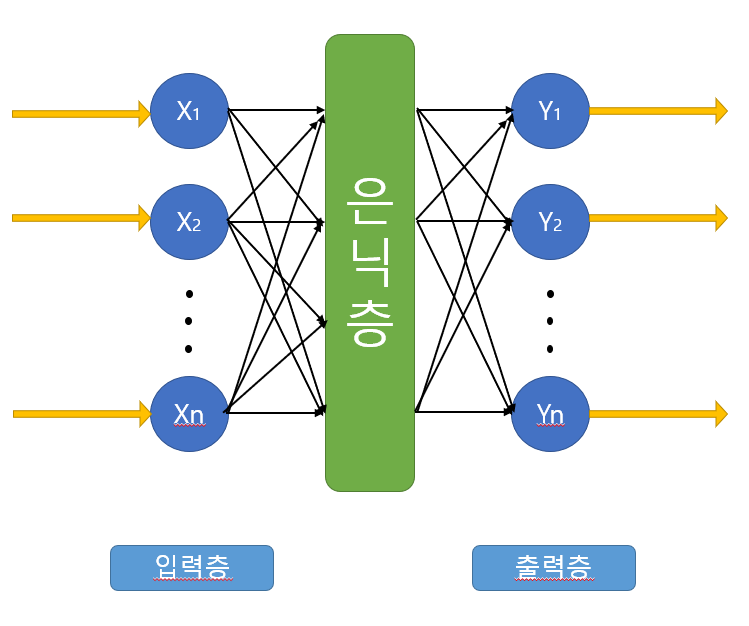
****

**[그림17]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.

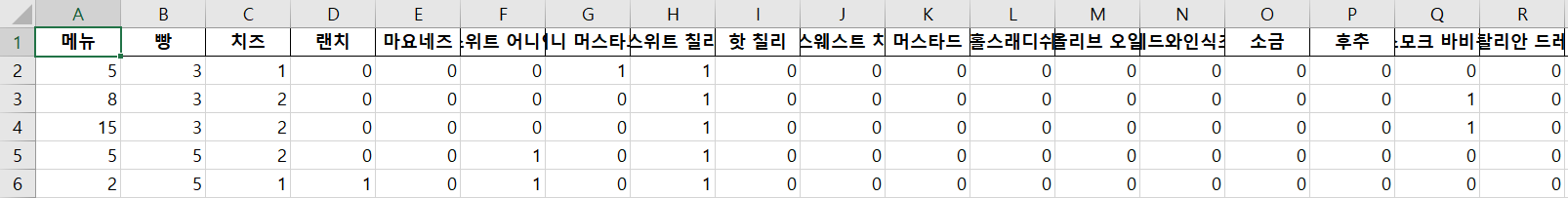
* 1. **다층 퍼셉트론 모델(MLP Classifier Model)**

위의 “유사도 분석 모델”을 통해 추천받은 주재료(메뉴)가 입력값, 그 주재료에 어울리는 맛의 소스가 출력값인 기계학습 모델이다.



**[그림18] 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.

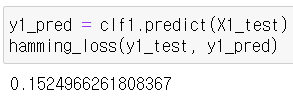


**[그림19] 입력 데이터: 주재료(메뉴), 빵, 치즈, 소스 15종**

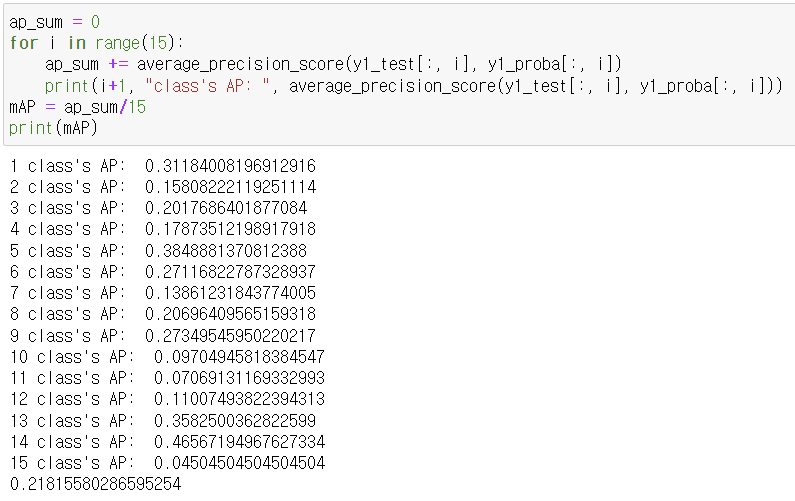
**• 입력 X : [주재료] 🡪 출력 y : [소스]**



**[그림20] 입력값과 출력값**



**[그림21] Hamming loss**



**[그림22] AP와 mAP**

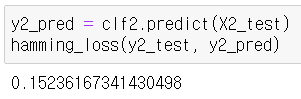
* **Hamming loss는 0.1524966261808367,**

**mAP는 0.21815580286595254 가 나왔다.**

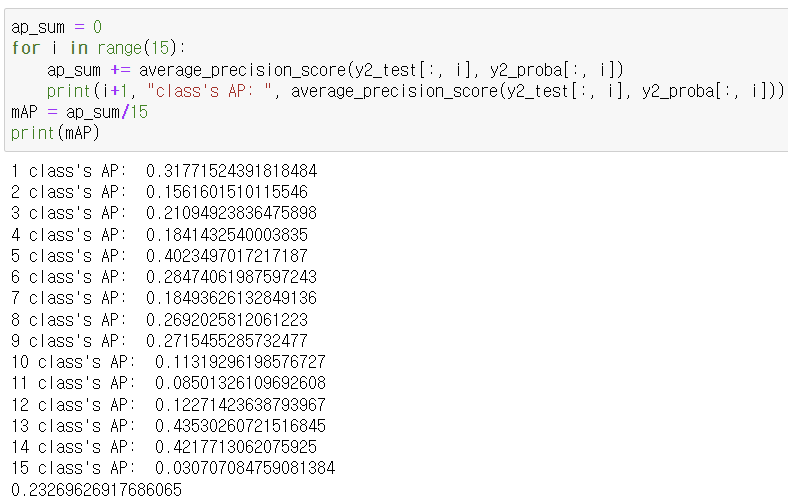
**• 입력 X : [주재료,빵,치즈] 🡪 출력 y : [소스]**



**[그림23] 입력값과 출력값**



**[그림24] Hamming loss**

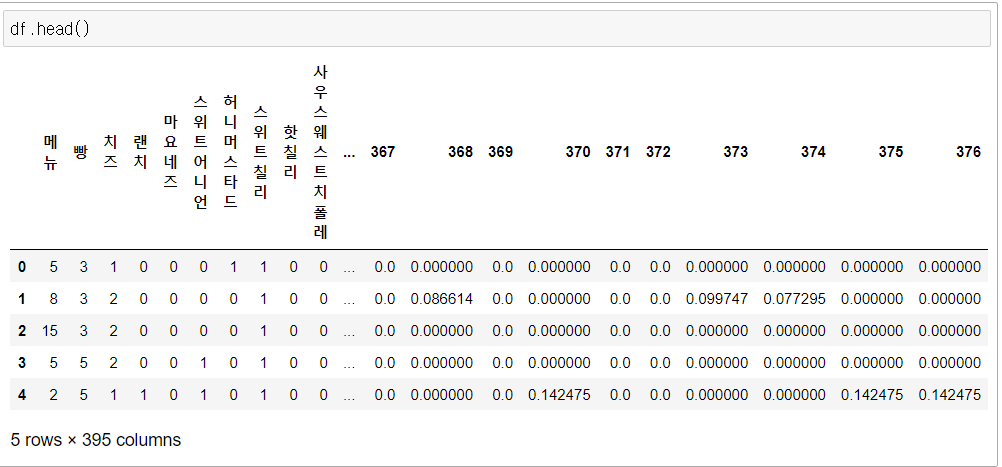


**[그림25] AP와 mAP**

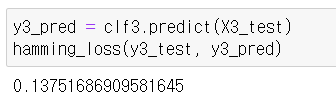
* **Hamming loss는 0.15236167341430498 ,**

**mAP는 0.23가 나왔다.**

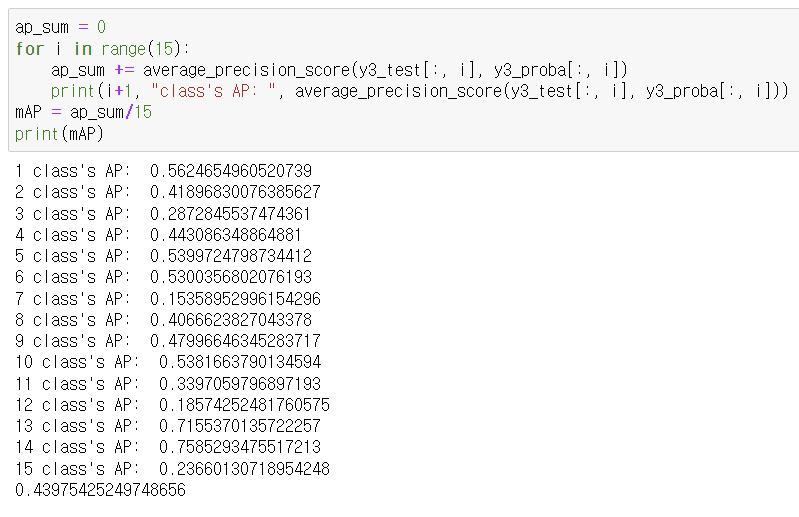
**• 입력 X : [주재료,빵,치즈,주재료에 대한 설명(td-idf)] 🡪 출력 y : [소스]**



**[그림26]** make\_input\_data.jpynb **으로 [그림21]입력데이터에 “주재료에 대한 설명”을 추가한다.**



**[그림27] Hamming loss**



**[그림28] AP와 mAP**

* **Hamming loss는 0.13751686909581645,**

**mAP는 0.44가 나왔다.**

**결론적으로, 입력값이** [주재료], [주재료, 치즈, 빵], [주재료, 치즈, 빵, 설명]일 때,

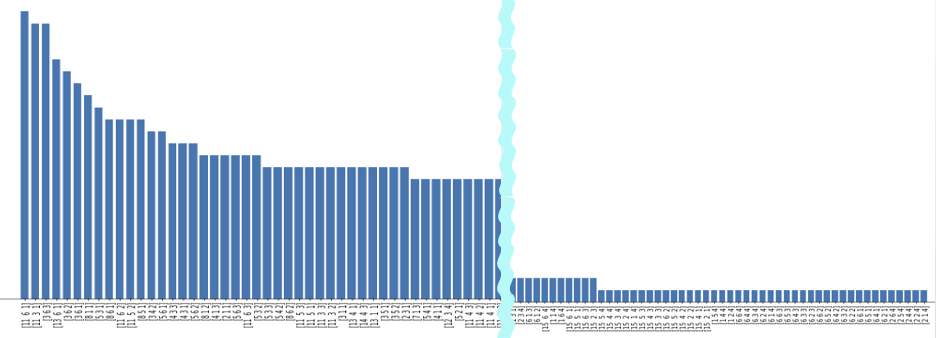
후자일수록 성능이 우수하게 나온 것을 볼 수 있다.

* **성능 향상을 위한 방법 2가지를 적용했다.**

**• 방법1 : “오버 샘플링” 활용**

“데이터의 불균형”이 모델의 성능향상에 방해가 된다고 생각해서 수집한 데이터에서 각각의 조합의 개수를 비교했다. **[그림29]**

결과적으로, 아래 그림과 같이 어떤 조합은 24개의 데이터를 가지고 있으나, 다른 조합은 1개의 데이터만 가지고 있는 등, 데이터 불균형이 분명히 존재하며, 이런 정보들이 학습에 편향을 일으키고 있다는 결론을 얻었다.



**[그림29] “[주재료, 빵, 치즈] 조합” 의 개수 비교**

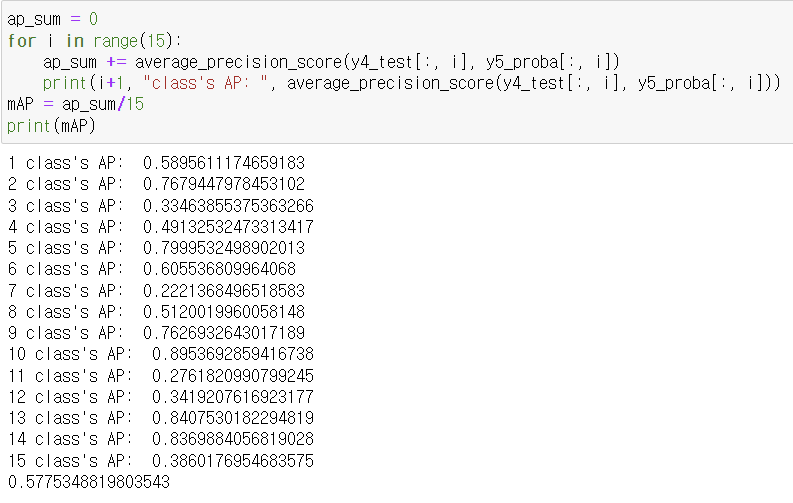
이 데이터 불균형 문제를 해소하기 위해, 각 클래스의 개수를 맞춰주는 **“오버샘플링”**을 진행하였다. 입력 데이터 [주재료,빵,치즈]의 360가지 조합들의 개수를 중복을 통해 각각 24개로 맞춘 후 학습을 진행했다.

* **mAP 가 0.56으로 증가했다.**

**• 방법2 : “하이퍼 파라미터 튜닝” 진행 (scikit-learn의 GridSearchCV이용)**

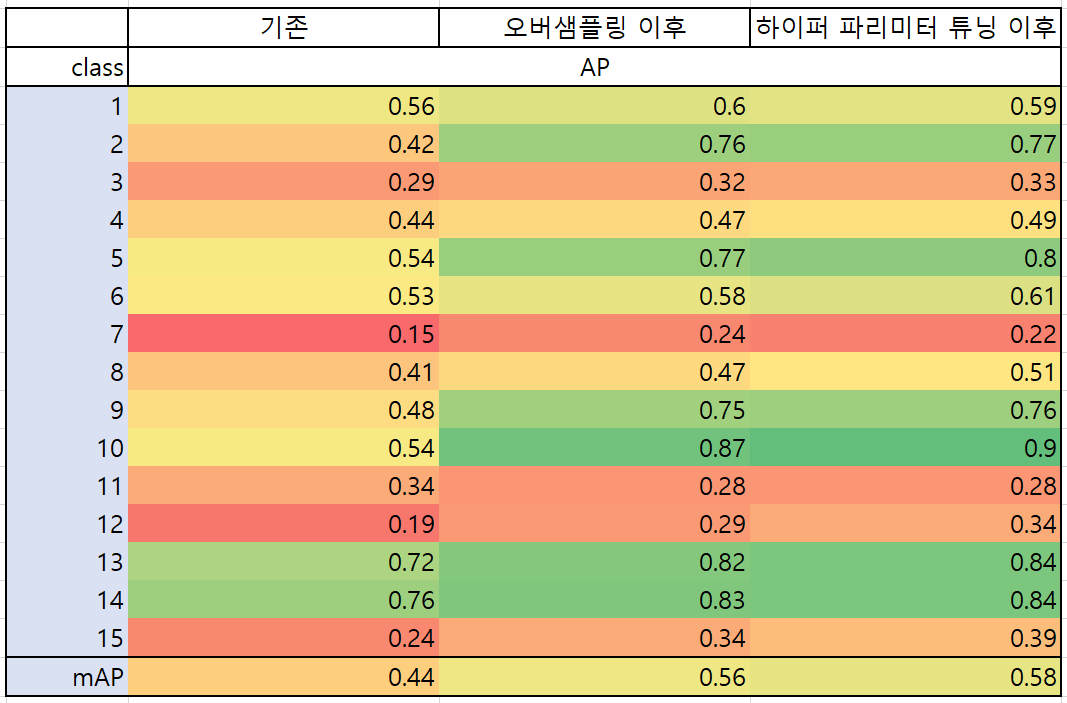
하이퍼 파라미터란 사용자가 설정 가능한 입력값을 말한다. 사용할 데이터에 따라 가장 적합한 모델, 그리고 그 모델의 하이퍼 파라미터값이 다르다. 그래서 데이터마다 좋은 파라미터 입력값을 하나하나 찾아주는 작업이 필요하며, 이를 “하이퍼 파라미터 튜닝”이라고 한다. 본 프로젝트에서는 scikit-learn의 GridSearchCV 모듈을 통해 하이퍼 파라미터 튜닝을 진행했다.

**🡪 mAP가 0.58로 증가했다.**



**[그림30]**

**이를 표로 나타내면 아래와 같이 성능이 점점 향상되었음을 확인할 수 있다.**

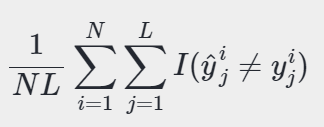


**[그림31]기존 상태, 오버샘플링 이후, 하이퍼 파라미터 튜닝 이후의 mAP비교**

* 1. **평가 지표**

**4.3.1 Hamming Loss**

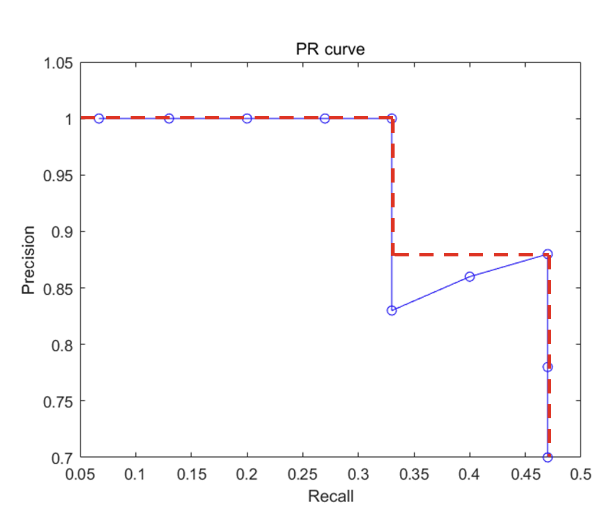
전체 label 중 잘못 분류된 label을 의미하며, Hamming loss가 낮을수록 그 알고리즘의 성능이 우수하다는 것을 의미한다.



**[그림32]hamming loss**

**4.3.2 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 우수하다는 것을 의미한다.



**[그림33] 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 알게 되어 자신의 입에 가장 맛있는 샌드위치를 맛보길 기대한다.

**5.2 추후 연구방향**

써브웨이 공식 애플리케이션과 연계하여, 사용자 개개인의 입맛에 맞는 샌드위치 재료 조합을 추천해주고, 이 조합을 곧바로 주문할 수 있도록 앱을 업그레이드한다면 좋을 것이다. 그리고, 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)

[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 머신러닝 완벽가이드 (권철민 지음)

[9] 딥 러닝을 이용한 자연어 처리 입문 (유원준, 안상준 지음)

<프로그램>

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

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

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