

종합설계 프로젝트 수행 보고서

프로젝트명	머신러닝 이미지 프로세싱과 협업 필터링을 이용한 와인 인식 및 추천 시스템
팀번호	S5-5
문서제목	수행계획서() 2차발표 중간보고서() 3차발표 중간보고서() 차발표 중간보고서() 최종결과보고서(O)

2021.11.09

팀원 : 채희상 (팀장)
심인영
추연철

지도교수 : 한익주 교수



문서 수정 내역

작성일	대표작성자	버전(Revision)	수정내용	
2020.12.18	채희상(팀장)	1.0	수행계획서	최초작성
2021.02.25	심인영	2.0	2차발표자료	설계서추가
2021.05.07	심인영	3.0	3차발표자료	시험결과추가
2021.06.19	심인영	4.0	4차발표자료	DEMO추가
2021.11.09	채희상	5.0	최종결과보고서	결론추가

문서 구성

진행단계	프로젝트 계획서 발표	중간발표1 (3월)	중간발표2 (5월)	학기말발표 (6월)	최종발표 (10월)
기본양식	계획서 양식	계획서 양식	계획서 양식	계획서 양식	계획서 양식
포함되는 내용	I. 서론 (1~6)	I. 서론 (1~6)	I. 서론 (1~6)	I. 서론 (1~6)	I
	II. 본론 (1~3)	II. 본론 (1~4)	II. 본론 (1~5)	II. 본론 (1~7)	II
	참고자료	참고자료	참고자료	참고자료	III

이 문서는 한국산업기술대학교 컴퓨터공학부의
 “종합설계”교과목에서 프로젝트
 “머신러닝 이미지 프로세싱과 협업 필터링을 이용한
 와인 인식 및 추천 시스템”
 을 수행하는
 (S5-5, 채희상, 심인영, 추연철)들이 작성한 것으로 사용하기
 위해서는 팀원들의 허락이 필요합니다.

목 차

I. 서론

1. 작품선정 배경 및 필요성
2. 기존 연구/기술동향 분석
3. 개발 목표
4. 팀 역할 분담
5. 개발 일정
6. 개발 환경

II. 본론

1. 개발 내용
2. 문제 및 해결방안
3. 시험시나리오
4. 상세 설계
5. Prototype 구현
6. 시험/ 테스트 결과
7. Coding & DEMO

III. 결론

1. 연구 결과
2. 작품제작 소요재료 목록

참고자료

I. 서론

1. 작품선정 배경 및 필요성

작품선정 배경

- 2021년 1월 기준, 와인은 이마트 주류 매출 27.7%으로 1위를 달성하고, 롯데마트 주류 매출 19.8%, 이마트24 편의점의 경우 하루 평균 와인 판매량 4300병을 달성하는 등 우리 사회에 와인의 대중화가 진행되고 있음
- 와인 라벨이 대부분 외국어로 작성되어 있어 언어장벽이 존재함
- 와인의 다양성으로 인한 진입장벽이 존재함

필요성

- 점차 커지는 와인 시장에서 사용자를 대신하여 초심자를 위한 와인에 대한 편리한 정보제공이 필요함.
- 초심자에게 와인 라벨의 정보를 정확하게 제공할 필요가 있음
- 각 사용자들의 취향을 분석, 개인별로 적절하게 사용자에게 추천 와인에 대한 정보를 제공하고자 함.

2. 기존 연구 / 기술 동향 분석

Delectable Wine - Scan & Rate	Delectable Wine - Scan & Rate은 Delectable Inc에서 제공하는 와인 어플리케이션이다. 와인의 레이블을 인식하고 그 정보를 제공하는 기능을 제공하며 개인 와인 저널을 통해 자신의 와인 기록을 남기고 공유할 수 있다. 와인의 판매 및 배달을 제공한다.
Vivino: Buy the Right Wine	Vivino: Buy the Right Wine은 Vivino에서 제공하는 와인 어플리케이션이다. 레스토랑의 메뉴나 여러 와인의 레이블을 인식하여 해당 와인들의 정보를 비교하고 구매처의 정보를 제공한다. 자신의 와인 기록을 추적할 수 있으며, 와인의 판매 및 배달을 제공한다.
CellWine	CellWine은 Good Courage Limited에서 제공하는 와인 어플리케이션이다.. 와인의 레이블을 찍어 자신의 와인 셀러에 기록해 사용자가 수집한 와인의 수량과 가격을 관리할 수 있도록 한다.

3. 개발 목표

- 크롤링을 통해 와인 데이터베이스를 작성
- 머신러닝 이미지 프로세싱과 협업 필터링
- 머신러닝 이미지 프로세싱을 통해 데이터베이스 탐색
- 사용자 기반 협업 필터링 추천 알고리즘을 통해 추천 와인 정보를 제공

4. 팀 역할 분담

	채희상	심인영	추연철
자료수집	관련 기술 및 연구 조사		
설계	서비스 관련 세부 기능을 설계(검색, 추천 등)		
구현	이미지 프로세싱 시스템 구축	데이터베이스 구축 로그인 모듈 검색로그 기록모듈	맞춤 추천 기능 구현 크롤링 및 데이터베이스 구축
테스트	사용자, 판매자, 관리자 별 어플 구동 테스트 추천 알고리즘 동작 테스트		

5. 개발 일정

추진사항	12	1	2	3	4	5	6	7
자료수집	●							
스터디	●	●						
시스템 설계		●	●					
시스템 구현				●				
데모 구현 및 테스트					●	●		
보완						●	●	●
최종 검토 및 발표								●

6. 개발 환경

- 개발 언어 : Java, MySQL, Python
- 주요 라이브러리 : Python String similarity, Tensorflow, Keraes

II. 본론

1. 개발 내용

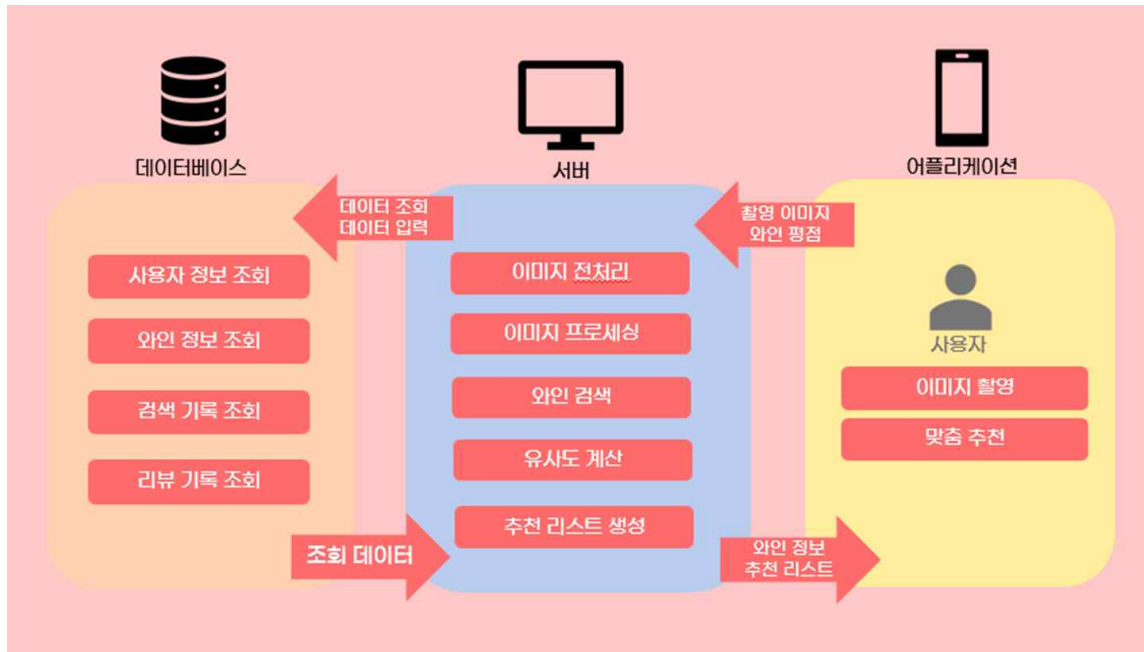
- 와인 라벨을 스마트폰 카메라로 촬영하여 이미지 전처리를 실행한 후 이미지 프로세싱을 통해 이미지를 파싱해서 와인 종류를 특정해내고, 데이터베이스를 탐색해서 정보를 제공함.
- 사용자에게 제공되는 와인에 대한 정보는 와인 명, 생산자, 생산지역, 품종, 종류, 용도, 당도, 산도, 바디, 탄닌, 추천 음식이다.
- 사용자가 검색했던 와인에 대해서 이후 평점 리뷰를 요청, 사용자들의 평점 리뷰를 기반으로 유사도를 측정하여 사용자 협업 필터링 추천 알고리즘을 통해 사용자에게 맞춤 추천을 제공함
- 사용자의 평점 리뷰를 바탕으로 와인의 당도, 산도, 바디, 탄닌에 대해서 취향 그래프를 작성, 사용자 협업 필터링 추천 알고리즘을 통해 알아낸 사용자와 유사한 다른 사용자들의 취향 그래프의 평균과 비교 가능

2. 문제 및 해결방안

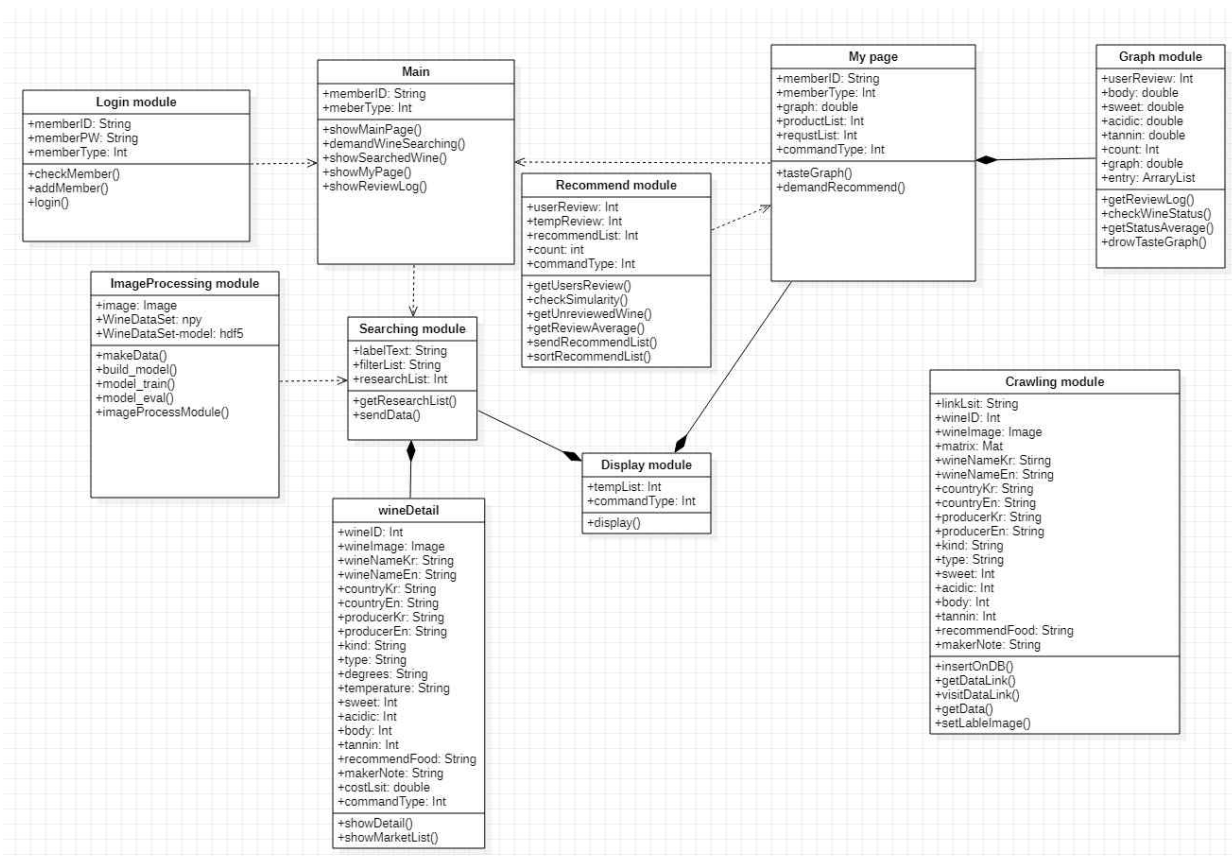
구현을 위한 소프트웨어 테스트를 진행하던 중에 발생한 문제점과 해결 방법은 아래와 같다.

- 기존에 계획했던 OCR 모듈의 테스트에서 문자 인식률이 너무 낮음 -> 와인 인식을 머신러닝 이미지 프로세싱으로 전환
- 로그인 API의 사용의 어려움 -> 자체 회원을 대상으로 하는 로그인 및 회원가입 모듈 개발
- 어플리케이션에서 촬영한 와인 이미지의 서버 전송 -> 기존에 Base64 기반으로 시도하던 것을 Java 기반의 이미지 전송을 위한 스트림 추가해 이미지 자체 전송으로 전환
- 다른 추천 와인 보기를 빠르게 여러번 시도하면 어플리케이션이 종료됨 -> 정확한 데이터를 서버로부터 수신받지 않으면 추천 와인 갱신을 하지 않음
- 이미지 촬영 이후 이미지 프로세싱이 진행되는 동안 어플리케이션에서 서버에 다른 통신을 하면 이후 와인 정보가 표시되지 않음 -> 이미지 프로세싱이 진행되는 동안 로딩 화면을 출력해 사용자의 조작이 불가능하게 함
- 리뷰 기록이 없는 사용자의 취향 그래프를 출력할 때 어플리케이션이 종료됨 -> 리뷰 기록이 없는 사용자가 취향 그래프 출력을 요구할 경우 전체 사용자의 평균 그래프만 출력 (리뷰 기록이 없는 사용자의 그래프는 출력되지 않음)

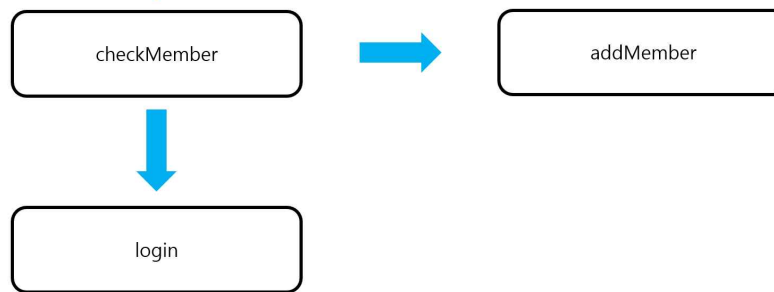
3. 시험 시나리오



4. 상세 설계

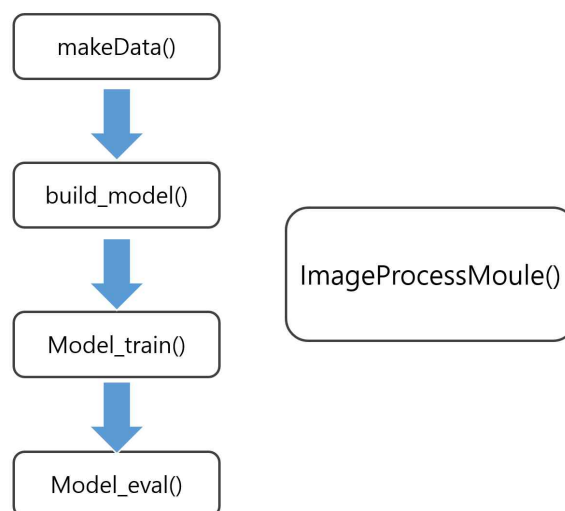


1. Login module



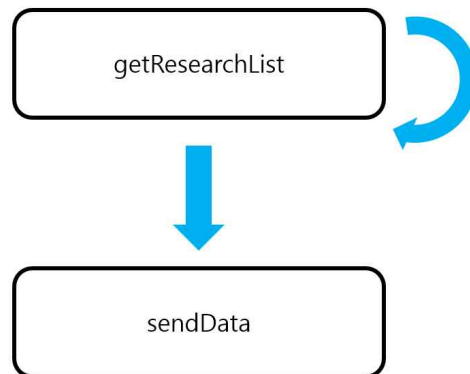
- checkMember : 해당 어플리케이션의 회원인지를 체크함
- addMember : 비회원의 경우 회원가입을 진행시킴
- login : 로그인을 진행함

2. ImageProcessing module



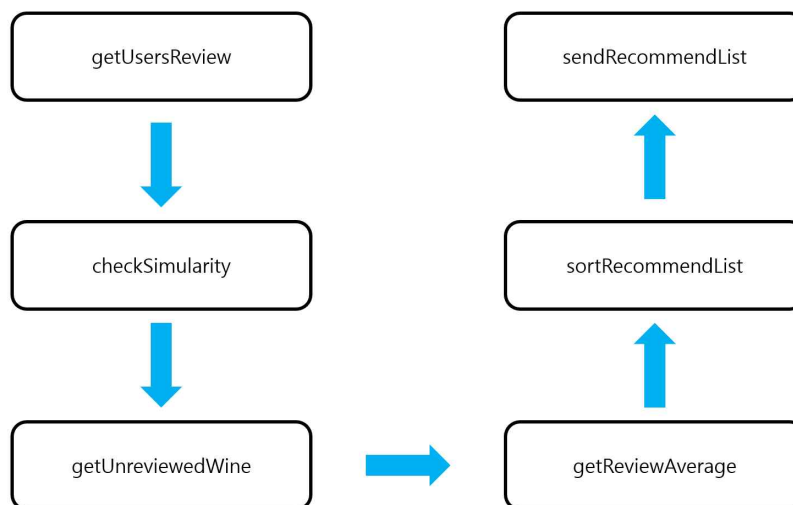
- makeData : 이미지 전처리를 한다.
- build_model : 모델을 제작한다.
- model_train : 모델을 학습시킨다.
- model_eval : 모델을 평가한다.

3. Searching module



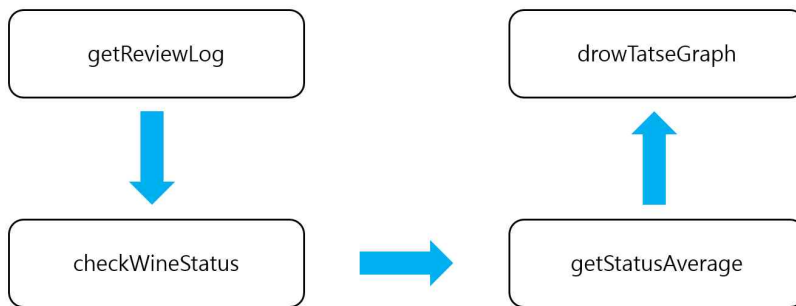
- getResearchList : 탐색한 리스트를 가져옴
- sendData : 리스트 데이터를 전송함

4. Recommend module



- getUsersReview : 추천을 요청한 사용자의 리뷰를 가져옴
- checkSimilarity : 유사도를 분석함
- getUnreviewedWine : 사용자가 리뷰하지 않은 와인을 식별함
- getReviewedAverage : 사용자가 리뷰했던 와인의 별점 평균을 가져옴
- sendRecommendList : 추천 리스트를 전송함
- sortRecommendList : 추천 리스트를 정렬함

5. Graph module



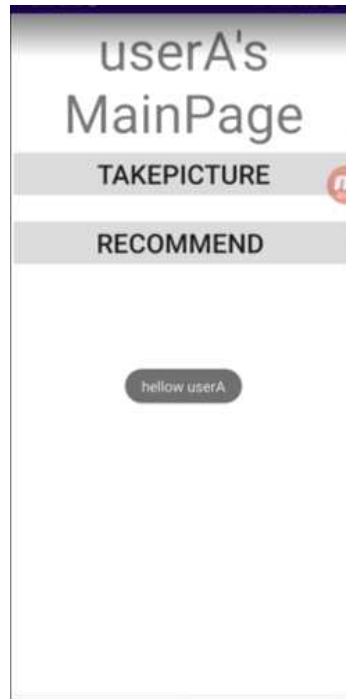
- getReviewLog : 리뷰로그를 가져옴
- checkWineStatus : 리뷰한 와인의 와인의 평가항목을 리뷰 평점에 의해 변수에 더하여 저장함
- getStatusAverage : 평가 항목의 평균을 연산함
- drawTastGraph : 통계 기록을 바탕으로 방사형 그래프를 화면에 출력함

5. Prototype 구현

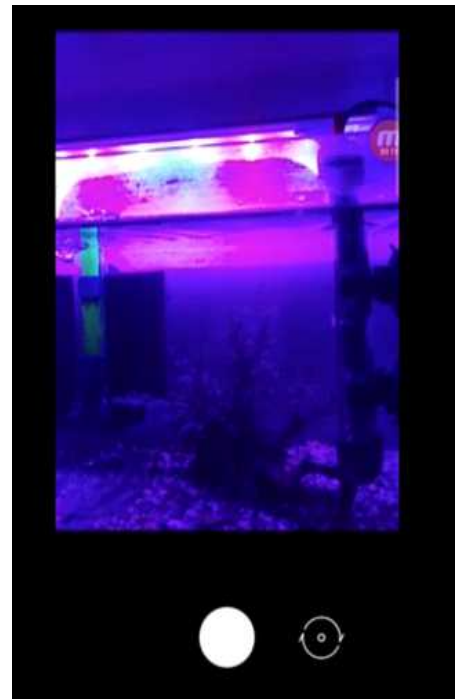
5.1) 어플리케이션 화면



[로그인 화면]



[메인 페이지 화면]



[카메라를 통한 캡처]

6. 시험/테스트 결과

6.1) 이미지 프로세싱 모듈



[샘플 이미지]

```
Run: test
2021-05-07 22:33:26.278761: W tensorflow/core/common_runtime/gpu/gpu_device.cc:1757] Cannot dlopen some GPU libraries. Please make sure the missing libraries
Skipping registering GPU devices...
2021-05-07 22:33:26.279169: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2021-05-07 22:33:26.279628: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1261] Device interconnect StreamExecutor with strength 1 edge matrix:
2021-05-07 22:33:26.279731: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1267]
2021-05-07 22:33:26.279783: I tensorflow/compiler/jit/xla_gpu_device.cc:99] Not creating XLA devices, tf_xla_enable_xla_devices not set
2021-05-07 22:33:26.358175: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:116] None of the MLIR optimization passes are enabled (registered 2)
duriman.jpg
caldirola, durimaan
Process finished with exit code 0
```

[이미지 프로세싱 결과]

6.2) 맞춤 추천 모듈

id	review
userA	{"1": 5, "2": 4, "3": 4, "4": 3, "5": 0}
userB	{"1": 1, "2": 0, "3": 1, "4": 0, "5": 4}
userC	{"1": 4, "2": 0, "3": 4, "4": 4, "5": 2}
userD	{"1": 4, "2": 2, "3": 3, "4": 0, "5": 1}
userE	{"1": 0, "2": 0, "3": 0, "4": 0, "5": 0}

[샘플 데이터]

```
Run: recommendtest x
D:\WWworkspace\WineWhat\recommend\venv\Scripts\python.exe D:\WWworkspace\WineWhat\recommend\recommendtest.py
userA
추천 와인 ID
[5]
Process finished with exit code 0
```

[맞춤 추천 모듈 결과, 리뷰 기록이 있고 유사한 사용자가 있는 경우]

```
Run: recommendtest x
D:\WWworkspace\WineWhat\recommend\venv\Scripts\python.exe D:\WWworkspace\WineWhat\recommend\recommendtest.py
userE
추천 와인 ID
유사한 사용자 없음
[5, 4, 3, 2, 1]
Process finished with exit code 0
```

[맞춤 추천 모듈 결과, 리뷰 기록이 없는 경우]

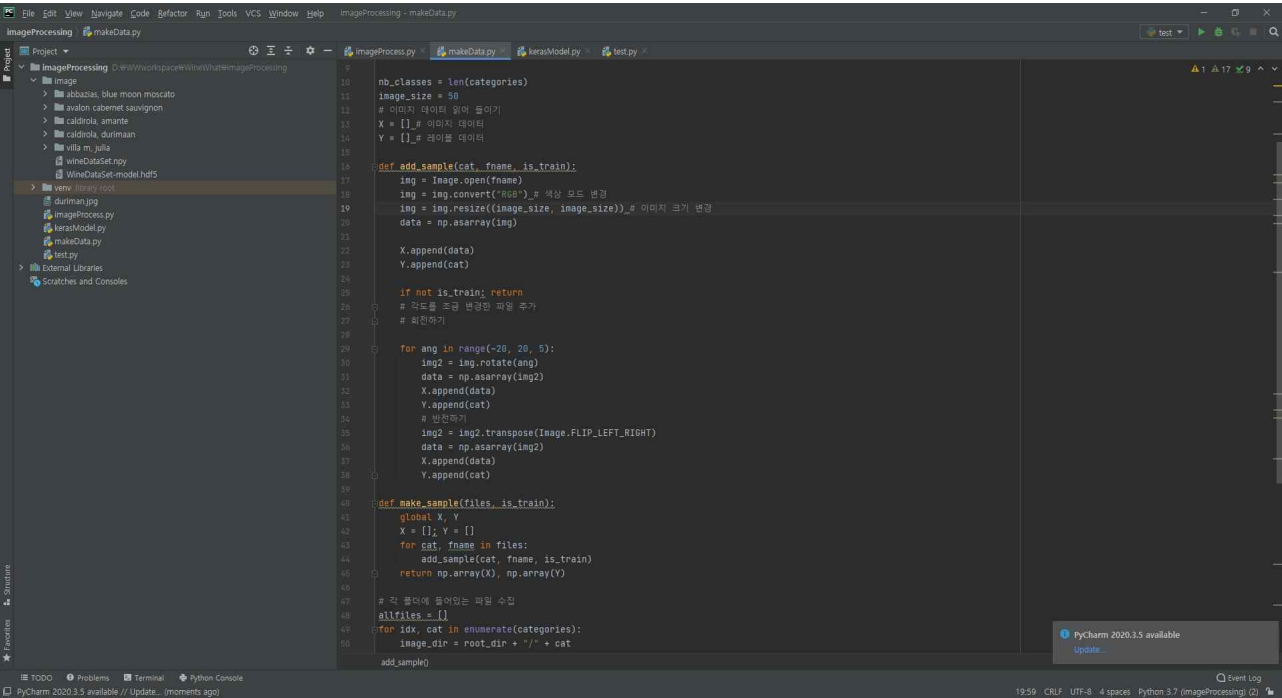
```
Run: recommendtest x
D:\WWworkspace\WineWhat\recommend\venv\Scripts\python.exe D:\WWworkspace\WineWhat\recommend\recommendtest.py
userB
추천 와인 ID
유사한 사용자 없음
[4, 2]
Process finished with exit code 0
```

[맞춤 추천 모듈 결과, 리뷰 기록이 있고, 유사한 사용자가 없는 경우]

7. Coding & DEMO

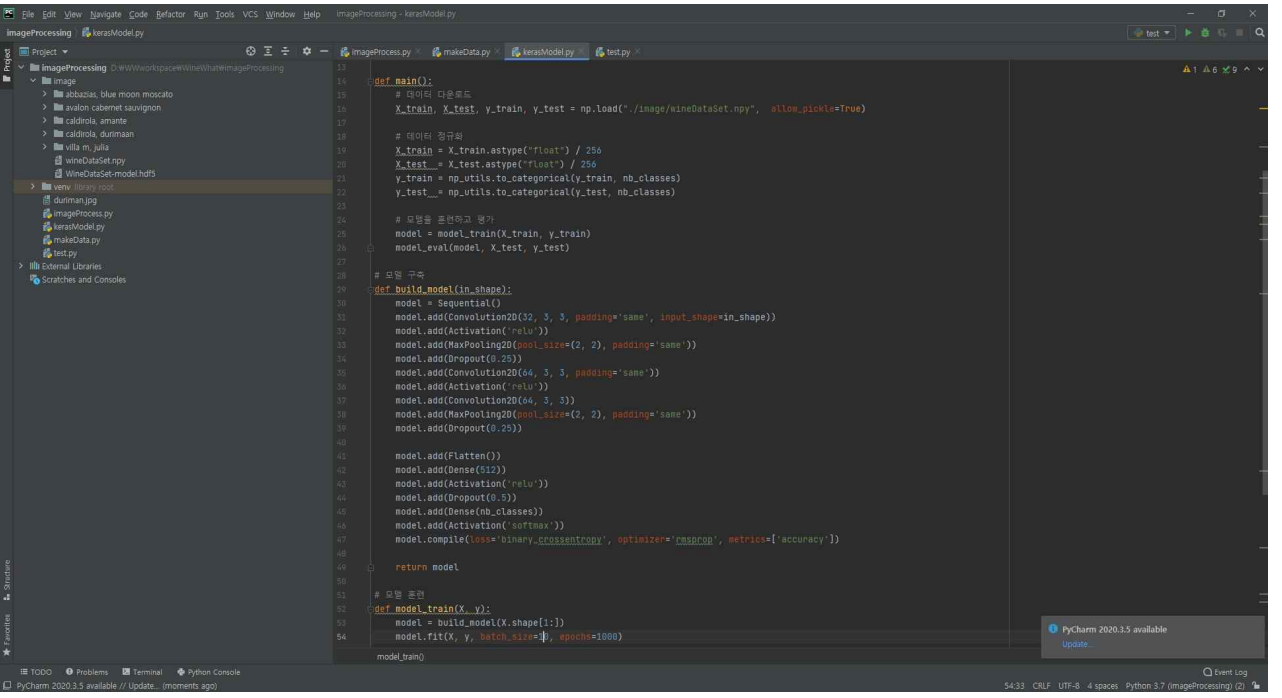
7.1) 이미지 프로세싱 모듈

[이미지 전처리]



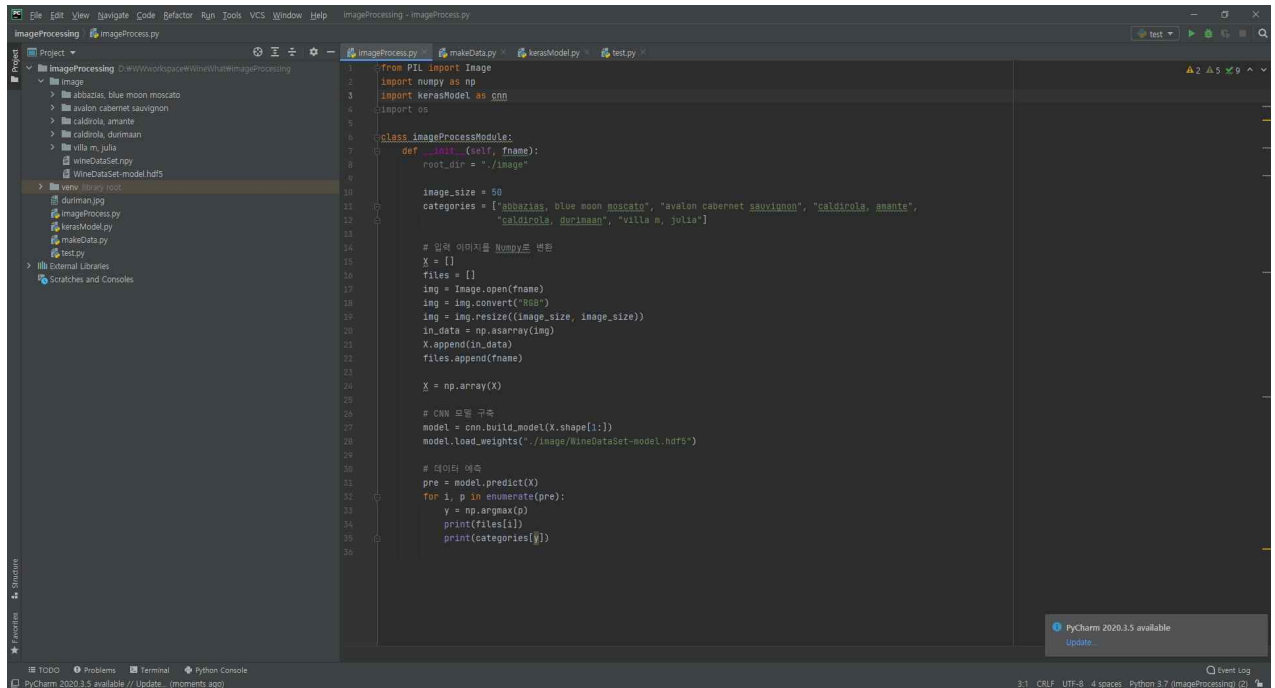
```
1 nb_classes = len(categories)
2 image_size = 50
3 # 이미지 데이터 읽어들이기
4 X = [] # 이미지 데이터
5 Y = [] # 레이블 데이터
6
7 def add_sample(cat, fname, is_train):
8     img = Image.open(fname)
9     img = img.convert('RGB') # 색상 모드 변경
10    img = img.resize((image_size, image_size)) # 이미지 크기 변경
11    data = np.asarray(img)
12    X.append(data)
13    Y.append(cat)
14
15    if not is_train: return
16    # 각도를 조금 변경한 파일 추가
17    # 회전하기
18    for ang in range(-20, 20, 5):
19        img2 = img.rotate(ang)
20        data = np.asarray(img2)
21        X.append(data)
22        Y.append(cat)
23        # 반전하기
24        img2 = img2.transpose(Image.FLIP_LEFT_RIGHT)
25        data = np.asarray(img2)
26        X.append(data)
27        Y.append(cat)
28
29 def make_sample(files, is_train):
30     global X, Y
31     X = []; Y = []
32     for cat, fname in files:
33         add_sample(cat, fname, is_train)
34     return np.array(X), np.array(Y)
35
36 # 각 폴더에 들어있는 파일 수집
37 allfiles = []
38 for idx, cat in enumerate(categories):
39     image_dir = root_dir + "/" + cat
40     add_sample()
```

[CNN 모델 구축]



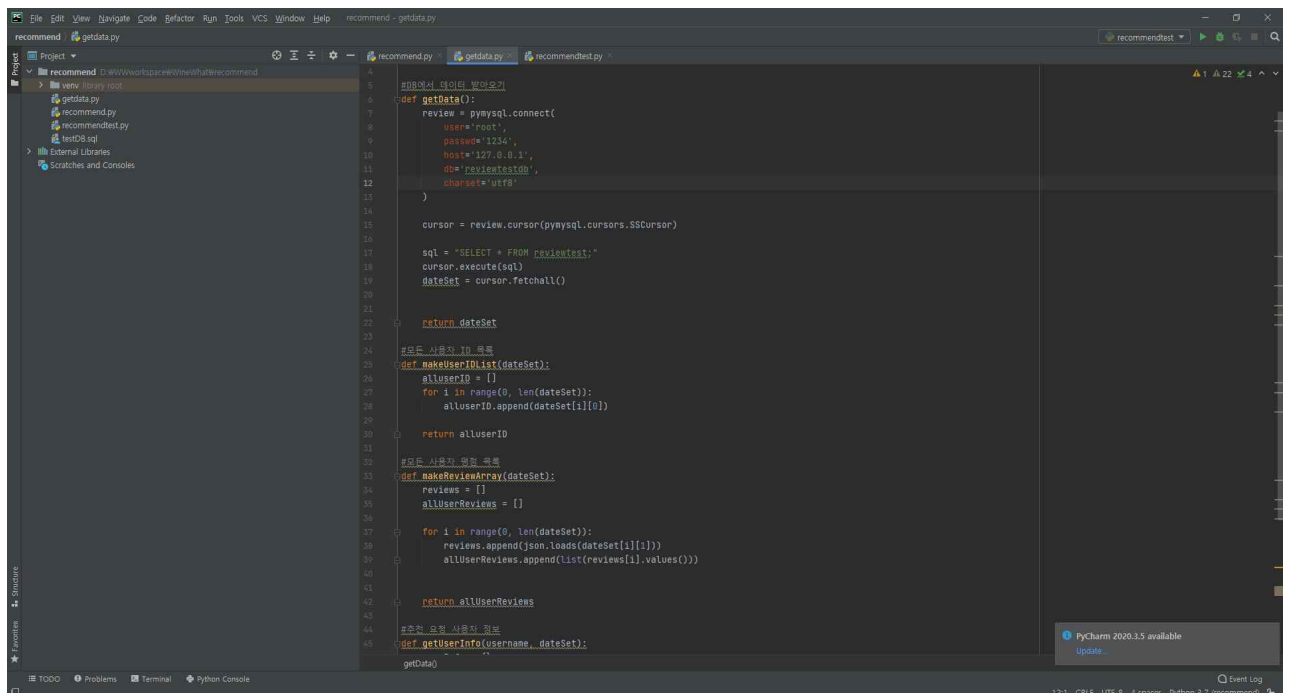
```
13 def main():
14     # 데이터 다운로드
15     X_train, X_test, y_train, y_test = load("../image/wineDataSet.npy", allow_pickle=True)
16
17     # 데이터 정규화
18     X_train = X_train.astype("float") / 256
19     X_test = X_test.astype("float") / 256
20     y_train = np_utils.to_categorical(y_train, nb_classes)
21     y_test = np_utils.to_categorical(y_test, nb_classes)
22
23     # 모델을 훈련하고 평가
24     model = model_train(X_train, y_train)
25     model_eval(model, X_test, y_test)
26
27 # 모델 구축
28 def build_model(in_shape):
29     model = Sequential()
30     model.add(Convolution2D(32, 3, 3, padding='same', input_shape=in_shape))
31     model.add(Activation('relu'))
32     model.add(MaxPooling2D(pool_size=(2, 2), padding='same'))
33     model.add(Dropout(0.25))
34     model.add(Convolution2D(64, 3, 3, padding='same'))
35     model.add(Activation('relu'))
36     model.add(Convolution2D(64, 3, 3))
37     model.add(MaxPooling2D(pool_size=(2, 2), padding='same'))
38     model.add(Dropout(0.25))
39
40     model.add(Flatten())
41     model.add(Dense(512))
42     model.add(Activation('relu'))
43     model.add(Dropout(0.5))
44     model.add(Dense(nb_classes))
45     model.add(Activation('softmax'))
46     model.compile(loss='binary_crossentropy', optimizer='rmsprop', metrics=['accuracy'])
47
48     return model
49
50 # 모델 훈련
51 def model_train(X, y):
52     model = build_model(X.shape[1:])
53     model.fit(X, y, batch_size=32, epochs=1000)
54
55     model_train()
```

[이미지 프로세싱]

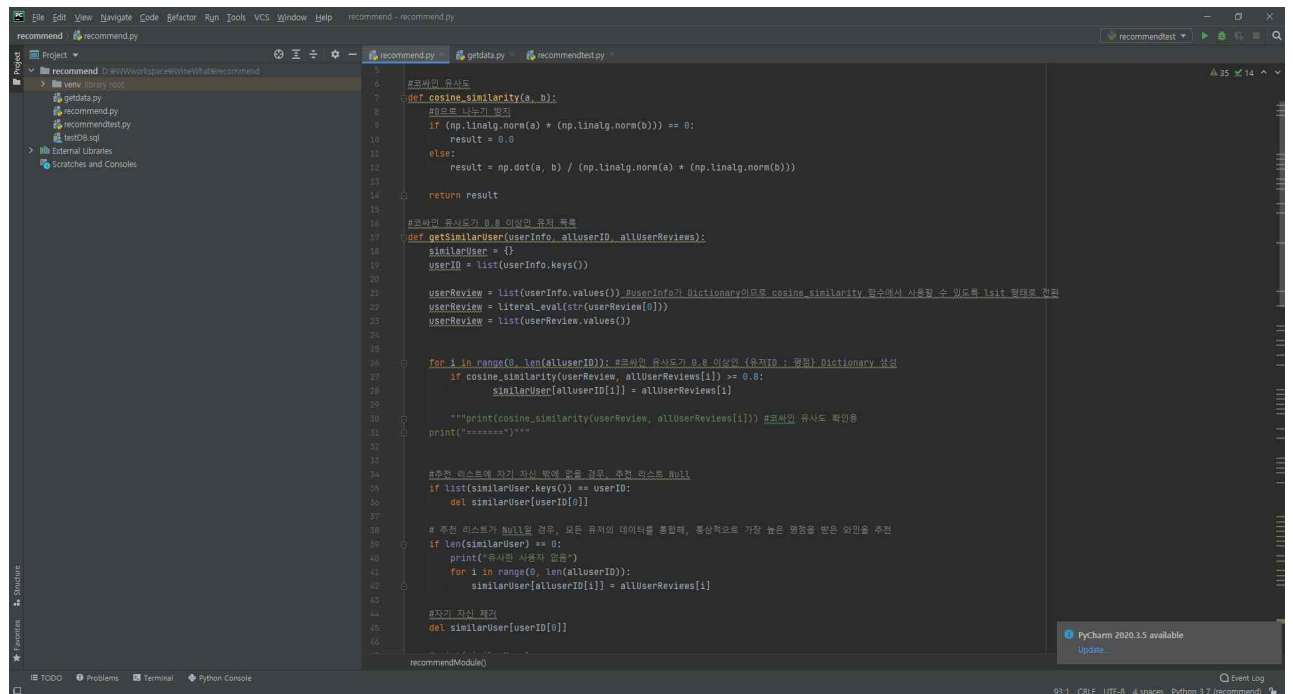


7.2) 맞춤 추천 모듈

[사용자 리뷰 데이터 전처리]

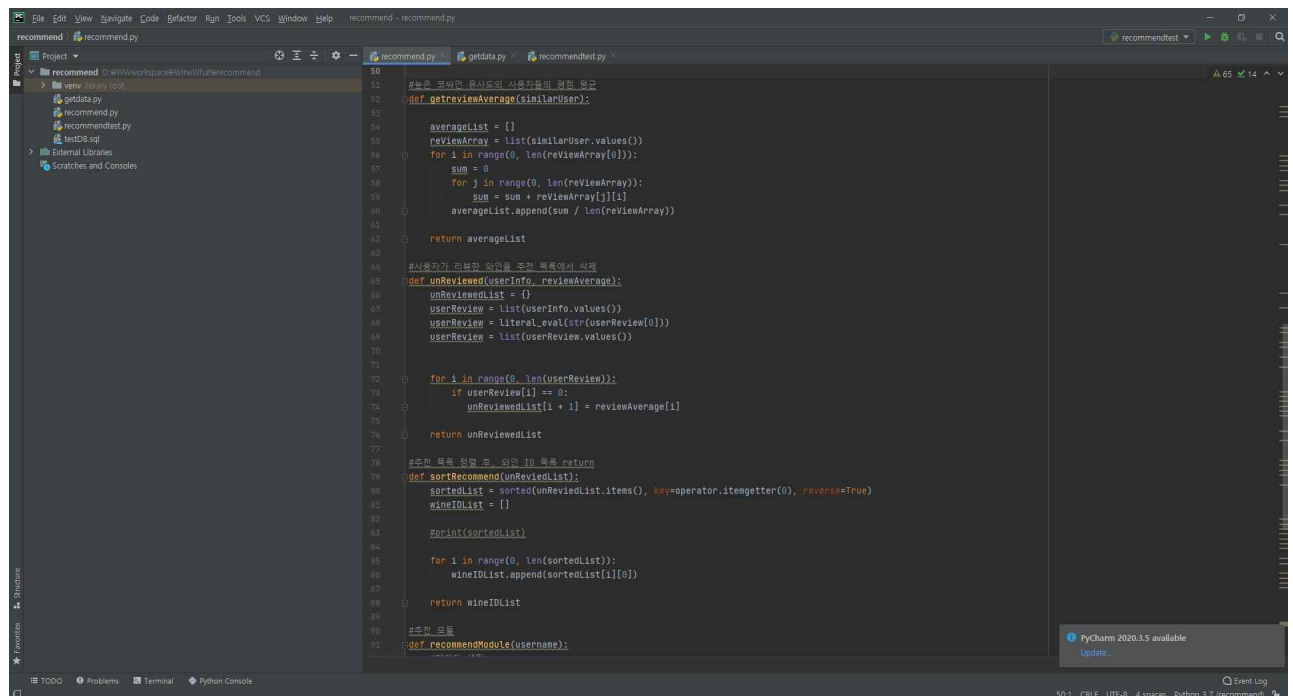


[유사한 사용자 추출]



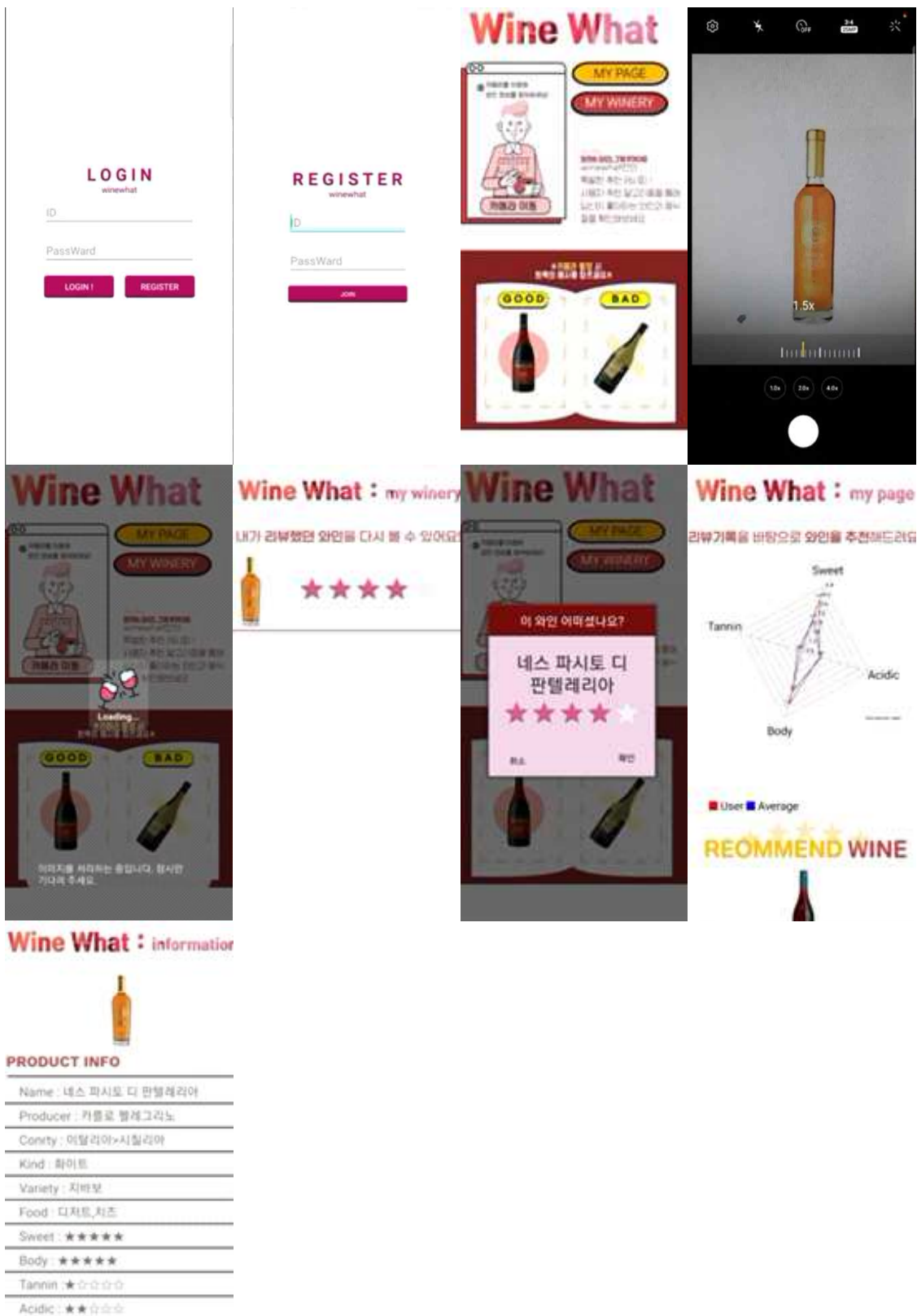
```
5
6 #유사한 사용자
7 def cosine_similarity(a, b):
8     #유사도를 나누기 방지
9     if (np.linalg.norm(a) * (np.linalg.norm(b))) == 0:
10         result = 0.0
11     else:
12         result = np.dot(a, b) / (np.linalg.norm(a) * (np.linalg.norm(b)))
13
14     return result
15
16 #유사한 사용자와 유사도 이상의 유사도 목록
17 def getSimilarUser(userInfo, allUserID, allUserReviews):
18     similarUser = {}
19     userID = list(userInfo.keys())
20
21     userReview = list(userInfo.values()) #userInfo가 Dictionary이므로 cosine_similarity 함수에서 사용할 수 있도록 list 형태로 변환
22     userReview = literal_eval(str(userReview[0]))
23     userReview = list(userReview.values())
24
25
26     for i in range(0, len(allUserID)): #유사한 사용자와 유사도 {유사도 : 평균} Dictionary 생성
27         if cosine_similarity(userReview, allUserReviews[i]) >= 0.8:
28             similarUser[allUserID[i]] = allUserReviews[i]
29
30     """print(cosine_similarity(userReview, allUserReviews[i])) #유사한 사용자 확인용
31     print("*****")"""
32
33
34     #추천 리스트에 자기 자신 밖에 있을 경우, 추천 리스트 Null
35     if list(similarUser.keys()) == userID:
36         del similarUser[userID[0]]
37
38     # 추천 리스트가 Null일 경우, 모든 유저의 데이터를 통째로, 통상적으로 가장 높은 평점을 받은 와인을 추천
39     if len(similarUser) == 0:
40         print("유사한 사용자 없음")
41         for i in range(0, len(allUserID)):
42             similarUser[allUserID[i]] = allUserReviews[i]
43
44     #자기 자신 제거
45     del similarUser[userID[0]]
46
47     return similarUser
```

[추천 와인 리스트 생성]



```
50
51 #유사한 사용자에 사용자의 사용기록의 평균 평균
52 def getReviewAverage(similarUser):
53
54     averageList = []
55     reviewArray = list(similarUser.values())
56     for i in range(0, len(reviewArray[0])):
57         sum = 0
58         for j in range(0, len(reviewArray)):
59             sum = sum + reviewArray[j][i]
60         averageList.append(sum / len(reviewArray))
61
62     return averageList
63
64 #사용자가 리뷰한 와인을 추천 목록에서 삭제
65 def unReviewed(userInfo, reviewAverage):
66     unReviewedList = []
67     userReview = list(userInfo.values())
68     userReview = literal_eval(str(userReview[0]))
69     userReview = list(userReview.values())
70
71
72     for i in range(0, len(userReview)):
73         if userReview[i] == 0:
74             unReviewedList[i + 1] = reviewAverage[i]
75
76     return unReviewedList
77
78 #추천 목록 정렬 후, wine ID 목록 return
79 def sortRecommend(unReviewedList):
80     sortedList = sorted(unReviewedList.items(), key=operator.itemgetter(0), reverse=True)
81     wineIDList = []
82
83     #print(sortedList)
84
85     for i in range(0, len(sortedList)):
86         wineIDList.append(sortedList[i][0])
87
88     return wineIDList
89
90 #추천 모듈
91 def recommendModule(username):
```

7.3) DEMO



Ⅲ. 결론

1. 연구결과

본 연구는 와인을 처음 접하는 사람들을 주요한 대상으로 와인 라벨의 언어적인 장벽이나 종류의 다양성, 와인의 정보를 쉽게 얻을 수 있는 매체의 부족으로 인한 와인의 진입장벽 문제를 해결하고자 하였다.

이러한 문제를 해결하기 위해 사용자가 라벨을 읽지 않고도 와인의 종류를 파악할 수 있고, 해당 와인의 정보를 제공하는데 초점을 맞추고 사용자가 쉽게 접근할 수 있게 어플리케이션으로 개발하기로 하였다. 머신러닝(CNN) 모듈을 기반으로 와인의 이미지를 통해 와인의 종류를 식별하고, 와인 데이터 베이스를 탐색해 사용자에게 정보를 제공하였다. 또한 정보를 제공한 와인을 대상으로 사용자에게 별점 리뷰를 요청해 리뷰 데이터 베이스를 작성하고, 리뷰 데이터를 분석하여 사용자 기반 협업 필터링 알고리즘을 통해 추천 와인을 제공하였다. 결과적으로 와인을 처음 접하는 사람들이 쉽게 와인의 정보를 접하고 자신의 취향에 맞는 와인에 대한 정보를 추가적으로 얻을 수 있는 머신러닝 이미지 프로세싱과 사용자 기반 협업 필터링을 이용한 와인 인식 및 추천 시스템을 완성하였다.

본 연구는 종래의 유사한 서비스가 대부분 해외에서 개발되어 영문으로 제공되는 서비스인 점, 리뷰 기록을 기반으로 사용자의 취향을 시각적으로 표현하고 있다는 점에서 차별점을 가지고 있다. 또한 빠르게 확장되고 있는 국내 와인 시장에 있어 새로운 소비자의 유입을 가능케하고, 이들이 계속해서 자신의 입맛에 잘 맞는 와인을 선택할 수 있도록 도와주는 부분에 있어 의의를 가진다.

향후 더 발전된 서비스를 위해서는 연구 중 가장 큰 문제였던 와인의 인식률 문제와 와인의 다양성에 대한 문제를 해결하기 위한 연구가 진행되어야 한다.

2. 작품제작 소요재료 목록

이름	내용
이미지 프로세싱 모듈 (CNN)	와인 이미지를 통해 종류 식별
사용자 기반 협업 필터링 알고리즘	리뷰 기록을 기반으로 와인 추천
MPAndroidChart	리뷰 기록을 기반으로 취향 그래프 작성

IV. 참고자료

- [1] 쿠지라 하코우즈쿠에, 「파이썬을 이용한 머신러닝, 딥러닝 실전 개발 입문」, 위키북스, 2019
- [2] 우재남, 박남길, 「Android Studio를 활용한 안드로이드 프로그래밍 5판」, 한빛아카데미, 2020
- [3] 홍의경, 「오라클을 기반으로 하는 데이터베이스 배움터 – 개정3판」, 생능, 2014
- [4] 윤성우, 「열혈 TCP/IP 소켓 프로그래밍」, 오렌지 미디어, 2020
- [5] scvgoe, 협업 필터링 추천 시스템 (Collaborative Filtering Recommendation System) 세상 간단한 추천 시스템, 2017,
[https://scvgoe.github.io/2017-02-01-%ED%98%91%EC%97%85-%ED%95%84%ED%84%B0%EB%A7%81-%EC%B6%94%EC%B2%9C-%EC%8B%9C%EC%8A%A4%ED%85%9C-\(Collaborative-Filtering-Recommendation-System\)/](https://scvgoe.github.io/2017-02-01-%ED%98%91%EC%97%85-%ED%95%84%ED%84%B0%EB%A7%81-%EC%B6%94%EC%B2%9C-%EC%8B%9C%EC%8A%A4%ED%85%9C-(Collaborative-Filtering-Recommendation-System)/)
- [6] MPAndroidchart, <https://github.com/PhilJay/MPAndroidChart>
- [7] 이미경, 와인, '국민 주류'로 자리할까...맥주 제치고 매출 1위, 한경 경제, 2020,
<https://www.hankyung.com/economy/article/2020122192827>