



Cerberus

DA Mini Project 기유진 / 김수빈 / 최은솔

Beer Pairing

맥주 음식 페어링 / 음식 재료 추천 시스템





목차

1. 데이터 생성
2. 데이터 가공 및 모델링
3. 백엔드
4. 웹 구조
5. 프론트엔드
6. 앞으로의 계획 및 추가 기능
7. 사이트 시연





01 | 데이터 소개

- 주요 데이터 소개

1) 맥주 기본 데이터

- 맥주 기본 속성 설명 및 정보 표시



2) 맥주 음식 페어링 데이터

- 페어링할 음식 정보 수집



3) 재료 궁합 데이터

- 재료끼리의 상관도를 알고 추천하는 시스템 구축





01 | 데이터 소개

- 맥주 기본 데이터

1) 맥주 기본 데이터

- 맥주 기본 속성 설명 및 정보 표시



맥주 기본 데이터

- craftbeer.com이라는 웹사이트 사용

텍스트 크롤링

- 기본 텍스트
- Original Gravity (OG):
- Final Gravity (FG):
- Alcohol By Volume (ABV):
- International Bitterness Units (IBUs):
- Bitterness Ratio (BU:GU):
- Volumes of CO₂ (v/v):
- Apparent Attenuation (AA):

데이터 수집 후 dictionary화

*초반에 와인, 맥주 두 주제에 대한 페어링 시스템을 구축하려 했으나
와인별 세부 점수의 부재와 데이터 부족으로 맥주에 중점을 두기로 함





01 | 데이터 소개

- 맥주 기본 데이터

2) 맥주 음식 페어링 데이터

- 페어링할 음식 정보 수집



Beer & Food Pairings

- craftbeer.com
- Brewers Association
- The Hangout, Orange County CA
- Shari's Berries

맥주 음식, 맥주 디저트 등 맥주와 어울리는 기본 속성 파악
후 가장 어울리는 재료를 수동으로 (!!!) 스코어링 (0-10)

완성 데이터셋:

- 19가지 맥주
- 재료 84가지
- 디저트 44가지
- 양념 19가지
- 흄푸드 21가지





01 | 데이터 소개

- 맥주 기본 데이터

3) 재료 궁합 데이터

- 재료끼리의 상관도를 알고 추천하는 시스템 구축



Kaggle “*What's Cooking?*”

- ”id”, ”cuisine”, ”ingredients”로 이루어진 json 파일
- 약 6만개의 재료 조합 데이터
- 재료끼리의 조합을 통해 재료간의 상관도를 유추하여 재료 궁합을 추천하는 데 사용할 수 있다고 판단





02 | 모델링

- 맥주 기본 데이터

1) 맥주 음식 페어링 시스템

- 01-2) 데이터 바탕
- 페어링 할 수 있는 사용자 함수 정의
- 맥주 별 궁합 점수 높은 음식부터 내림차순
- 맥주 선택 시 궁합이 좋은 음식 확인

	Unnamed: 2	Blonde Ale	Hefeweizen	Witbier	Classic Pilsner	British Style Beer	American Wheat Ale	India Pale Ale	Double/ Imperial IPA	Oktoberfest	Pale Ale	Red/ Amber Ale	Barley Wine	Brown Ale	Old/ Strong Ale	Abbey Dubbel	Dark Lager
0	beef	4	4	3	6	6	2	6	7	9	6	4	8	7	9	8	10
1	lamb	3	3	3	5	5	1	3	8	9	6	5	9	7	9	8	8
2	venison	2	2	3	5	6	2	4	7	8	5	4	8	6	9	7	8
3	salami	5	5	3	9	4	3	8	8	8	7	5	8	5	5	6	7
4	prosciutti	7	6	5	8	5	4	6	8	8	7	4	7	4	6	6	7
...
78	quinoa	8	7	6	3	9	7	6	4	8	8	6	4	3	5	6	4
79	brown rice	7	7	7	4	8	7	5	6	5	7	6	5	5	6	6	3
80	sweet potatoes	5	4	2	5	8	2	5	9	5	6	5	3	6	7	5	4
81	taro	4	5	4	4	9	1	4	9	7	5	5	6	5	8	6	5
82	potatoes	4	3	3	5	10	1	3	9	8	4	5	7	5	8	6	5

```
def pairing(alcohol, number = 20):
    df_a = df.loc[:, ['Unnamed: 2', alcohol]]
    df_a = df_a.sort_values(by=[alcohol], ascending=False).values
    df_r = pd.DataFrame(df_a[:number], columns = ['ingredient', alcohol])

    return df_r

pairing('Blonde Ale')
```

	ingredient	Blonde Ale
0	fresh cheese	10
1	chicken	10
2	mozzarella cheese	10
3	lettuce	9
4	salmon	9
5	asparagus	8
6	kale	8
7	green peas	8
8	shrimp	8
9	octopus	8
10	squid	8



02 | 모델링

- 맥주 기본 데이터

2) 재료 궁합 추천 시스템

- 01-3) 데이터 바탕
- 재료 간 co-occurrence 계산

index	id	cuisine	ingredients
0	0	10259 greek	[romaine lettuce, black olives, grape tomatoes...]
1	1	25693 southern_us	[plain flour, ground pepper, salt, tomatoes, g...
2	2	20130 filipino	[eggs, pepper, salt, mayonaise, cooking oil, g...
3	3	22213 indian	[water, vegetable oil, wheat, salt]
4	4	13162 indian	[black pepper, shallots, cornflour, cayenne pe...

# Pick up Example				
cooc_df[cooc_df['ing_x'] == 'chillies'].sort_values('cooc', ascending = False).head(8)				
ing_x	ing_y	x_count	y_count	cooc
3137	chillies	salt	148	22534
3126	chillies	onions	148	10008
3125	chillies	garlic	148	9171
3130	chillies	ginger	148	2190
58906	chillies	garam masala	148	1179
19369	chillies	vegetable oil	148	5516
3124	chillies	water	148	9293
3133	chillies	tomatoes	148	3812



02 | 모델링

- 맥주 기본 데이터

2) 재료 궁합 추천 시스템

- PMI 계산 후 매우 낮거나 음수인 재료쌍 제외
$$PMI(A,B) = \log P(A,B) / (P(A)*P(B))$$
- 최종적인 재료 간 궁합 확인

ing_x	ing_y	x_count	y_count	cocc	PMI
35345	mexican chocolate	pasilla chiles	41	38	12 2.247404
171599	gari	wasabi	57	34	13 2.109193
69460	juniper berries	sauerkraut	30	40	7 1.969489
14516	brown cardamom	green cardamom	55	107	34 1.959854

```
show_most_similar('pork')
```

- Most similar to 'pork'
 - . cabbage head : 1.00
 - . kettle chips : 0.95
 - . pancit : 0.83
 - . fillet red snapper : 0.73
 - . napa cabbage leaves : 0.66
 - . garland chrysanthemum : 0.64
 - . good seasons italian dressing mix : 0.64
 - . egg noodles : 0.62
 - . fresh shiitake mushrooms : 0.61
 - . reduced sodium smoked ham : 0.60



03 | 웹사이트 구조

- 웹 프레임워크 및 주요 구조





04 | 백엔드

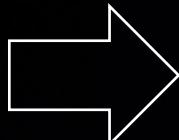
- 웹사이트 백엔드 구성

맥주 세부 Slider

```
def beer_info(request):
    beerList = ['Blonde Ale', 'Hefeweizen', 'Witbier', 'Classic Pilsner',
               'British Style Beer', 'American Wheat Ale', 'India Pale Ale', 'Double/ Imperial IPA',
               'Oktoberfest', 'Pale Ale', 'Red/ Amber Ale', 'Barley Wine', 'Brown Ale', 'Old/ Strong Ale',
               'Abby Dubbel', 'Dark Lager', 'Porter', 'Sweet/ Oatmeal Stout', 'Imperial Stout']

    beerOrigin = ['US', 'Germany', 'Canada', 'Germany', 'US', 'Germany',
                  'US', 'Germany', 'Canada', 'Germany', 'US', 'Germany',
                  'US', 'Germany', 'Canada', 'Germany', 'US', 'Germany', 'US']

    return render(request, "alc_app/templates/beerSelect.html", {"beerList": beerList, "beerOrigin": beerOrigin, "t":
```



```
{% csrf_token %}




<form action="{% url 'showBeer' %}" method="POST" enctype="multipart/form-data" class="form-horizontal">
    {% for beer in beerList %}
        <!-------BEER ITEM 1----->
        <li class="item-a">

            <div class="box">
                <button class="beeritem" type="input" value="{{beer}} name='beerSelect'>{{beer}}</button>
                <a href="{% url 'showBeer' %}">Start Pairing</a>
                
            <p>
```



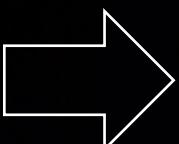


04 | 백엔드

- 웹사이트 백엔드 구성

맥주 선택 정보 내보내기

```
def about_beer(request):  
    return render(request, "alc_app/templates/about.html")  
  
def about_team(request):  
    return render(request, "alc_app/templates/team.html")  
  
def ing_pairing(selected_beer, number = 20):  
    selected_beer = request.GET['select']  
  
    df = pd.read_excel('real_pairing.xlsx')  
  
    df_a = df.loc[:, ['Unnamed: 0', alcohol]]  
    df_a = df_a.sort_values(by=[alcohol], ascending=False).values  
    df_r = df_a[:number]  
  
    for i in range(0,number):  
        i == 0  
        if i < number:
```



```
app_name = 'alc_app'  
urlpatterns = [  
    path('', views.direct_home, name='directHome'),  
    path('about/', views.about_beer, name='beerAbout'),  
    path('team/', views.about_team, name='teamAbout'),  
    path('alc_app/', views.beer_select, name='beerSelect'),  
    path('templates/', views.ing_pairing, name='showBeer')]
```





04 | Cytoscape.js

- Node.js 기반 마인드맵 툴

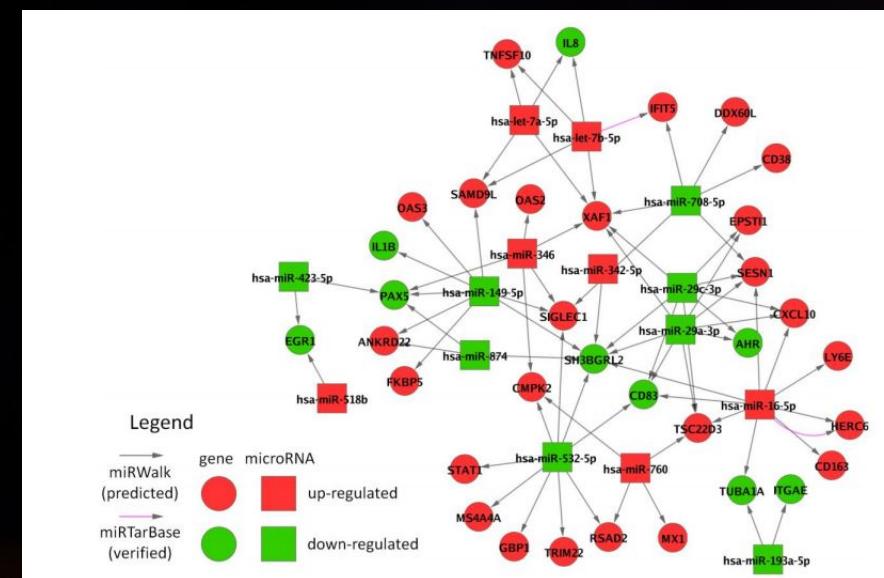
페어링 모델 실현

목표

- 가장 잘 어울리는 재료 20개와
- 각 재료에서 뺀어나온 해당 재료와 가장 잘 어울리는 재료 5개를 마인드맵의 형식으로 화면에 도출

노드의 크기

- 1차 : 술과의 pairing 점수
- 2차 : 각 재료와의 pairing 점수에 비례





04 | Cytoscape.js

- Node.js 기반 마인드맵 툴

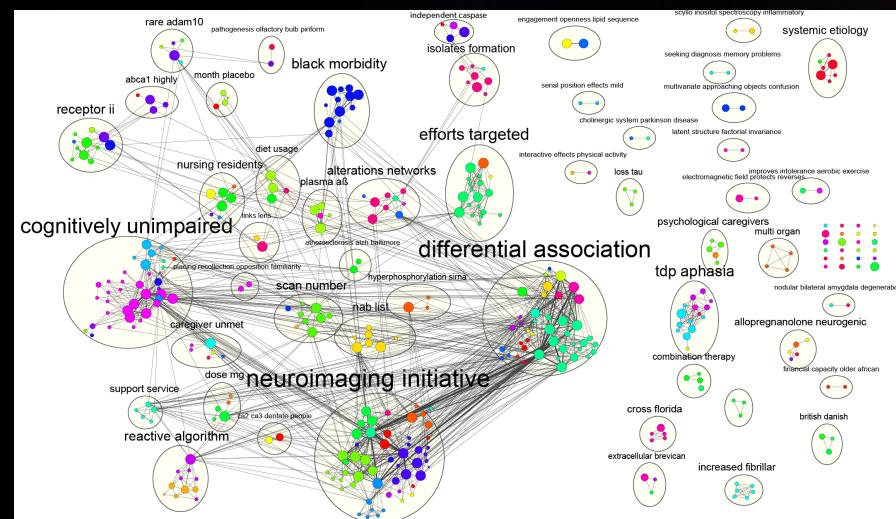
페어링 모델 실현

Json 파일 생성 + 연결

- node와 edge data로 이루어진 json 파일로 만드는 코딩을 하여 필요한 전체 데이터를 확보

사용자 UI 설정

- 사용자가 관심있는 재료 위 hover -> 재료 이름 강조
- 해당 재료와 연결되어있는 술/어울리는 재료 -> 위계별로 색상 설정





05 | 프론트엔드

- 웹사이트 프론트 소개

프론트엔트 작업

- 단순하고 직관적인 디자인
- 맥주를 시각적으로 표현
- base color: yellow, black
- 흘에서 화면 전환 시 물결치는 효과





06 | 앞으로의 계획

- 앞으로의 거창하고 갈 길의 먼 계획

- 1) 주종 추가
- 2) 술 특성에 대한 사용자 기호 입력 후 맥주 추천 알고리즘 구성
- 3) Django Node.js **백엔드 통일**
- 4) 사이트 배포



07 | 사이트 구

• 앞으로의 거창하고 갈 길의 먼 계획



Click

