

# 우리동네 가성비 GPT 서비스

SKN2기 : 사재민, 진정현, 장준영

2024년 08월 22일

## 프로젝트 배경

### 문제 정의 :

- 현대인들이 일상에서 빠르고 저렴한 음식점이나 편의시설을 찾는데 어려움을 겪고 있음

### 기술적 배경 :

- GPT 모델의 활용 가능성과 편의시설 추천 시스템의 중요성

## 목적

### 주요 목표 :

- 사용자에게 가깝고 저렴한 편의시설을 효율적으로 추천하는 시스템 개발

### 기대 효과 :

- 사용자 경험 향상
- 시간 절약 및 비용 절감

## 개발 환경

### 작업 언어 :

- python 3.8.19

### 임베딩 모델과 언어 모델 :

- openai 1.42.0

### 사용된 주된 모듈 :

- langchain=0.2.14
- langchain-community=0.2.12
- langchainhub=0.1.21

## 방법론

### 데이터 수집 :

- 음식점 정보 및 편의시설 정보
- <https://www.data.go.kr/data/3045247/fileData.do?recommendDataYn=Y>

### 모델 선택 :GPT-4-mini

- 자연어 처리 능력이 뛰어난 최신 GPT 모델 채택
- 다양한 응답 생성
- 높은 정확도
- 문맥 유지
- 비즈니스 요구 충족

### 전처리 과정 :

```
import pandas as pd

df = pd.read_csv('data.csv')
df2 = df.drop(['시도', '시군'], axis=1)
new_column_order = ['주소', '업소명', '업종', '연락처', '메뉴1', '가격1', '메뉴2', '가격2', '메뉴3', '가격3']
df3 = df2[new_column_order]
df3.to_csv("data_preprocessing2.csv", index=False)
```

- 1.csv파일을 pandas로 불러와서 전처리를 진행합니다.
- 2.“주소”와 중복되는 칼럼인 “시도”와 “시군”을 제거하였습니다.
3. column의 중요도와 맥락의 관계 정리
  - 주소, 업소명, 업종, 연락처, 메뉴1, 가격1, 메뉴2, 가격2, 메뉴3, 가격3 순으로 column을 정리했습니다.

- vectorstore에 저장하기 위해서는 데이터를 분할 하는 것이 중요.
- 하지만 csv파일은 이미 한 행씩 나누어져 있기 때문에 chunk로 나누지 않음 ⇒ 만약 한 행에 걸쳐 다른 정보가 들어 있는 경우에는 필요, 이 데이터에는 연관된 데이터들로만 이루어져 있기 때문에 분할하지 않음

## 방법론

### 시스템 아키텍처:

- 위에서 불러온 data가 document 형식으로 저장→오픈ai의 임베딩 모델을 이용하여 vectorstore에 저장
  - 여기서 사용된 vectorstore는 랭체인(chroma)을 사용
  - 임베딩 모델은 Text-embedding-ada-002-v2
  - 이후 문제 발견되어 새로운 모델 사용 text-embedding-3-small
- vectorstore에 저장된 값들을 retriever로 찾을
- 우리가 사용할 LLM을 GPT-4-mini 모델 Load
  - temperature=0 ⇒ 낮은 온도는 더 일관된 답을 보여줌
- prompt를 작성해줍니다
  - ChatPromptTemplate = 전체 프롬프트를 정의, 각 구조를 포함
  - SystemMessagePromptTemplate = llm에 직접 명령하는 시스템메시지 형식  
여기서 {context} 는 대화시 추가될 맥락. ⇒ vectorstore에서 찾을 맥락
  - HumanMessagePromptTemplate = 사용자가 입력 질문
  - ⇒ "{question}" 으로 동적으로 받음.

### • 챗봇은 이전 대화내용을 기억하면서 답변

- 대화내용을 저장해주는 버퍼메모리 필요.
- ConversationBufferMemory(memory\_key="chat\_history", return\_messages=True)
  - ConversationBufferMemory는 전체 대화를 저장
  - 파라미터들로 들어가는 memory\_key는 메모리가 저장될 키를 지정
  - retron\_message= True 는 그 전 대화 내용들도 같이 출력

### 성능실험:

⇒ ⇒ GPT -4-0613적용

⇒ retron\_message= True일때 = 7초

```
1 qa("서울시 가게 추천해줘")

{ 'question': '서울시 가게 추천해줘',
  'chat_history': [HumanMessage(content='서울시 가게 추천해줘'),
                   AIMessage(content='1. 로데오당구장\n 주소: 서울특별시 성북구 보문로38길 53 (동소문동5가) 지하1층\n 메뉴: 당구 10분(10:00~17:00)\n 가격: 800원\n2. 백구사 헤어플랜 (구 21c 헤어미닛)\n 주소: 서울특별시 강서구 곰달래로19길 45 1층\n 메뉴: 파마, 커트\n 가격: 파마 30000원, 커트 10000원\n3. 크러스트\n 주소: 서울특별시 서초구 서초대로51길 30 1층\n 메뉴: 흑임자바게트, 팥오쇼콜라\n 가격: 흑임자바게트 3600원, 팥오쇼콜라 3800원')],
```

## 방법론

⇒ 어려운 대답시 18초

```
1 qa("종로구에 갈거야")

{
  'question': '종로구에 갈거야',
  'chat_history': [HumanMessage(content='서울시 가게 추천해줘'),
    AIMessage(content='1. 로데오당구장\n 주소: 서울특별시 성북구 보문로38길 53 (동소문동5가) 지하1층\n 메뉴: 당구 10분(10:00~17:00)\n 가격: 800원\n2. 백구사 헤어플랜 (구 21c 헤어미닛)\n 주소: 서울특별시 강서구 곰달래로19길 45 1층\n 메뉴: 파마, 커트\n 가격: 파마 30000원, 커트 10000원\n3. 크러스트\n 주소: 서울특별시 서초구 서초대로51길 30 1층\n 메뉴: 흑임자바게트, 빵오쇼콜라\n 가격: 흑임자바게트 3600원, 빵오쇼콜라 3800원'),
    HumanMessage(content='종로구에 갈거야'),
    AIMessage(content='1. "먹고갈래지고갈래"라는 한식집을 추천드립니다. 주소는 서울특별시 종로구 수표로 117 (낙원동)이며, 연락처는 02-367-1239입니다. 메뉴로는 삼계탕이 13000원, 산채비빔밥이 7000원입니다.\n2. "대륙"이라는 중식집도 좋습니다. 주소는 서울특별시 종로구 종로 125 (종로3가) 1~2층이고, 연락처는 02-766-8411입니다. 짜장면이 7000원, 짬뽕이 8000원입니다.\n3. 마지막으로 "왕관커피숍"을 추천드립니다. 주소는 서울특별시 종로구 종로 222 (종로5가)이며, 연락처는 02-226-8365입니다. 커피와 국산차가 각각 2000원입니다.']),
  'answer': '1. "먹고갈래지고갈래"라는 한식집을 추천드립니다. 주소는 서울특별시 종로구 수표로 117 (낙원동)이며, 연락처는 02-367-1239입니다. 메뉴로는 삼계탕이 13000원, 산채비빔밥이 7000원입니다.\n2. "대륙"이라는 중식집도 좋습니다. 주소는 서울특별시 종로구 종로 125 (종로3가) 1~2층이고, 연락처는 02-766-8411입니다. 짜장면이 7000원, 짬뽕이 8000원입니다.\n3. 마지막으로 "왕관커피숍"을 추천드립니다. 주소는 서울특별시 종로구 종로 222 (종로5가)이며, 연락처는 02-226-8365입니다. 커피와 국산차가 각각 2000원입니다.')]
}
```

⇒ retron\_message= False 일때 = 6초

```
1 qa("서울시 가게 추천해줘")

{
  'question': '서울시 가게 추천해줘',
  'chat_history': '',
  'answer': '1. 로데오당구장\n 주소: 서울특별시 성북구 보문로38길 53 (동소문동5가) 지하1층\n 메뉴: 당구 10분(10:00~17:00)\n 가격: 800원\n2. 백구사 헤어플랜 (구 21c 헤어미닛)\n 주소: 서울특별시 강서구 곰달래로19길 45 1층\n 메뉴: 파마, 커트\n 가격: 파마 30000원, 커트 10000원\n3. 크러스트\n 주소: 서울특별시 서초구 서초대로51길 30 1층\n 메뉴: 흑임자바게트, 빵오쇼콜라\n 가격: 흑임자바게트 3600원, 빵오쇼콜라 3800원'}
}
```

⇒ chat\_history의 format을 지원하지 않는다는 오류 발생

```
1 qa("종로구에 갈거야")

ValueError                                Traceback (most recent call last)
<ipython-input-122-bd1e360db0b9> in <cell line: 1>()
----> 1 qa("종로구에 갈거야")

5 frames
/usr/local/lib/python3.10/dist-packages/langchain/chains/conversational_retrieval/base.py
in _get_chat_history(chat_history)
    53         buffer += "\n" + "\n".join([human, ai])
    54     else:
--> 55         raise ValueError(
    56             f"Unsupported chat history format: {type(dialogue_turn)}."
    57             f" Full chat history: {chat_history}"
)

ValueError: Unsupported chat history format: <class 'str'>. Full chat history: Human: 서
```

## 방법론

⇒ ⇒ GPT-4-mini 적용

```
qa("대구시 추천해줘")
3.9s

{'question': '대구시 추천해줘',
 'chat_history': [HumanMessage(content='대구시 추천해줘'),
  AIMessage(content='대구광역시에서 추천할 만한 가게는 다음과 같습니다:\n\n1. **원대시장횡집**\n   - 업종: 한식_일반\n   - 주소: 대구광역시 서구 달서로54길\n2. **원대시장횡집**\n   - 업종: 한식_일반\n   - 주소: 대구광역시 서구 달서로54길']}]
```

[illegible]

```
qa("우리가 이전에 무슨대화를 했는지 알려줘")
```

✓ 2.3s Python

```
{'question': '우리가 이전에 무슨대화를 했는지 알려줘',  
 'chat_history': [HumanMessage(content='대구시 추천해줘'),  
                  AIMessage(content='대구광역시에서 추천할 만한 가게는 다음과 같습니다:\n\n1. **월대시장횃집**\n   - 업종: 한식_일반\n   - 주소: 대구광역시 서구 달서로54길 7-1 (월대동)\n\n2. **원대시장횃집**\n   - 업종: 한식_일반\n   - 주소: 대구광역시 서구 달서로54길 7-1 (월대동)  
HumanMessage(content='우리가 이전에 무슨대화를 했는지 알려줘'),  
                  AIMessage(content='이전 대화에서는 대구시에서 추천한 만한 가게에 대해 이야기했습니다. 제가 추천한 가게는 원대시장횃집과 청도시골밥상이었습니다. 추가로 궁금한 점이 있으시면 말씀해주세요.\n\nanswer': '이전 대화에서는 대구시에서 추천한 만한 가게에 대해 이야기했습니다. 제가 추천한 가게는 원대시장횃집과 청도시골밥상이었습니다. 추가로 궁금한 점이 있으면 말씀해주세요']}]
```

GPT-4-mini 적용 후 평균 5초대 이내의 응답속도를 보여줌

- 약 2배의 속도 향상을 보여줌

## 방법론

- 각각 CHAIN을 걸어서 PIPE 같은 형식으로 한번에 사용
  - CONVERSATIONALRETRIEVALCHAIN  
⇒ LLM의 대화형 질의응답 시스템을 만드는 데 용이
    - 받는 파라미터들로는
      - LLM(미리 학습된 언어모델) ,
      - RETRIEVER(VECTORSTORE에서 찾아온 맥락)
      - MEMORY = 그 전의 대화 내용들을 저장 하는 방식
      - COMBINE\_DOCS\_CHAIN\_KWARGS = 문서를 결합하는 CHAIN에 추가적인 인수
        - 여기서 PROMPT 추가

## 모델 성능

### 개선 전(PROMPT):

- 이 두개의 캡처는 초기 챗봇이 첫 질문에 대한 답변 중 '두번째'라는 부분이 무엇인지는 기억하고 있지만 물어보는 주소는 대답을 하지 못하는 것을 보여줌

여기에 질문을 입력하고 '질문하기' 버튼을 눌러주세요.

질문 입력:

서초구 가게 추천해줘

질문하기

답변: 물론이죠, 서초구에서 착한 가격에 맛있는 음식을 즐길 수 있는 몇 가지 가게를 추천해드리겠습니다.

첫 번째로는 '크리스트'라는 빵집입니다. 주소는 서울특별시 서초구 서초대로51길 30 1층이며, 연락처는 070-4123-3899입니다. 흑임자바게트가 3600원, 빵오쇼콜라가 3800원으로 가격이 착하죠.

두 번째로는 '플로우'라는 카페를 추천드립니다. 주소는 서울특별시 서초구 신반포로43길 45이며, 연락처는 070-7779-5577입니다. 아메리카노가 3500원, 카페라떼가 4000원, 콜드브루가 4500원으로 착한 가격에 커피를 즐길 수 있습니다.

마지막으로는 '은순이네 반찬가게'와 '미각'이라는 한식집을 추천드립니다. '은순이네 반찬가게'의 주소는 서울특별시 서초구 서초중앙로24길 43 지114호이며, 연락처는 02-501-5060입니다. 코다리가 6000원, 두부조림이 3000원으로 착한 가격에 한식을 즐길 수 있습니다. '미각'의 주소는 서울특별시 서초구 반포대로28길8이며, 연락처는 02-585-9095입니다. 선지해장국, 차돌된장찌개, 김치전골 모두 9000원으로 가격이 착합니다.

이 가게들을 방문하시면 착한 가격에 맛있는 음식을 즐길 수 있을 거예요.

여기에 질문을 입력하고 '질문하기' 버튼을 눌러주세요.

질문 입력:

두번째 가게의 주소는?

질문하기

답변: 죄송합니다만, 제가 가지고 있는 정보 중에 "플로우"라는 가게에 대한 정보는 없습니다. 다른 가게를 찾으시거나 다른 질문을 해주시면 감사하겠습니다.

## 모델 성능

### • 전환 전(model):Text-embedding-ada-002-v2

여기에 질문을 입력하고 '질문하기' 버튼을 눌러주세요.

사용자: 대구 음식점 추천해줘

챗봇: 먼저, 사용자가 원하는 음식 종류나 가격대, 특정 지역 등에 대한 추가 정보가 필요합니다. 예를 들어, "한식을 찾고 계신가요, 아니면 다른 종류의 음식을 선호하시나요?" 또는 "특정 가격대 내에서 찾고 계신가요?"와 같은 질문을 통해 사용자의 요구를 더욱 명확히 파악할 수 있습니다. 이러한 추가 정보를 바탕으로, 사용자에게 가장 적합한 음식점을 추천해드릴 수 있습니다.

사용자: 대전의 음식점 추천해줘

챗봇: 사용자가 대전에서 어떤 종류의 음식을 찾고 계신지 알려주실 수 있나요? 예를 들어, 한식, 중식, 일식, 양식 등이 있습니다. 또한, 특정 메뉴나 가격대에 대한 선호도도 알려주시면 더욱 정확한 추천을 드릴 수 있습니다.

질문 입력:

대전의 음식점 추천해줘

질문하기

대전, 대구 등 특정지역 정보를 찾아내지 못하는 모습

## 개선 후(PROMPT):

챗봇이 중복된 것에 대해 선택지를 사용자에게 요구하고 이전의 답변을 기억하고 있는 수준으로 성능이 올라간 것을 보여줌

여기에 질문을 입력하고 '질문하기' 버튼을 눌러주세요.

질문 입력:

첫번째 가게의 정보를 더 알려줘

질문하기

사용자: 전주콩나물국밥의 주소는?

챗봇: 죄송합니다만, "전주콩나물국밥"이라는 가게 이름으로는 검색이 되지 않습니다. 가게 이름이 정확한지 다시 확인해주시 수 있나요? 또는 해당 가게에서 주문하셨던 다른 메뉴나, 가게가 위치한 구체적인 지역 정보를 알려주시면 더 정확한 검색을 시도해볼 수 있습니다.

사용자: 전주콩나물국밥의 연락처는?

챗봇: 전주콩나물국밥 가게에는 여러 지점이 있습니다. 아래는 몇 가지 지점의 정보입니다:

1. 서울특별시 종로구 자하문로 3 (내자동)에 위치한 전주콩나물국밥의 연락처는 02-738-8223입니다.

2. 경기도 수원시 장안구 조원로89번길 31 (조원동) 1층에 위치한 (장안)전주콩나물국밥의 연락처는 031-251-1174입니다.

원하시는 지점이 어디인지 알려주시면 더 정확한 정보를 제공할 수 있습니다.

사용자: 첫번째 가게의 정보를 더 알려줘

챗봇: 전주콩나물국밥 가게는 서울특별시 종로구 자하문로 3 (내자동)에 위치해 있습니다. 이 가게에서는 콩나물국밥과 순두부찌개를 주 메뉴로 판매하고 있으며, 각각의 가격은 7000원입니다. 가게의 연락처는 02-738-8223입니다. 이 가게에서 맛있는 한식을 즐겨보세요!

## 모델 성능

### • 전환 후(model): Text-embedding3 -small

여기에 질문을 입력하고 '질문하기' 버튼을 눌러주세요.

사용자: 대구의 음식점 추천좀 해줘

챗봇: 물론이죠, 대구에서 추천드릴 만한 음식점을 몇 군데 알려드리겠습니다.

1. 중해반점:

- 위치: 대구광역시 중구 명륜로 63
- 연락처: 053-255-8555
- 메뉴: 짜장면(6,000원), 짬뽕(7,000원)

2. 다모아식당:

- 위치: 대구광역시 동구 파계로 54 (지묘동)
- 연락처: 053-986-2929
- 메뉴: 해물칼국수(5,000원), 돌솥밥정식(10,000원)

3. 다복네 순대 돼지국밥:

- 위치: 대구광역시 서구 달서로 153 (비산동) 다복네 순대 돼지국밥
- 연락처: 053-562-0377
- 메뉴: 돼지국밥(7,000원), 순대국밥(7,500원)

이 중에서 원하시는 음식 종류나 위치에 따라 선택하시면 될 것 같습니다.

사용자: 대전 음식점 추천 좀 해줘

챗봇: 물론이죠, 대전에서 추천드릴 만한 음식점을 몇 군데 알려드리겠습니다.

1. 월평맛고을:

- 위치: 대전광역시 서구 월평서로6번길 72 (월평1동)
- 연락처: 042-487-2220
- 메뉴: 삼겹살(13,000원), 촌돼지찌개(7,000원), 청국장찌개(7,000원)
- 한식 육류를 전문으로 하는 음식점으로, 특히 삼겹살이 인기 메뉴입니다.

2. 매일반점:

- 위치: 대전광역시 서구 사마1길 16 (도마2동)
- 연락처: 042-533-3228
- 메뉴: 자장면(3,000원), 짬뽕(4,000원)
- 중식을 전문으로 하는 음식점으로, 가격 대비 맛이 좋아 많은 사랑을 받고 있습니다.

3. 2900냥식당:

- 위치: 대전광역시 동구 대전로 815번길 5 (정동)
- 연락처: 042-256-8686
- 메뉴: 선지해장국(5,000원), 콩나물해장국(5,000원), 콩나물비빔밥(5,000원)
- 한식을 전문으로 하는 음식점으로, 특히 해장국이 인기 메뉴입니다.

이 중에서 원하시는 음식 종류에 따라 선택하시면 좋을 것 같습니다.

- 대구와 대전의 음식점을 못 찾다가 OpenAIEmbeddings 과정에서 model='text-embedding-3-small' 을 넣어줌으로써 대전과 대구를 비롯한 다양한 지역을 잘 찾아주는 결과를 보여줌



## 결과

GPT모델 : GPT - 4 - mini

Emebdding model : Emebedding-3-small

prompt:

너는 친절하고 기억력이 좋은 챗봇이며, 사용자의 질문에 정확하고 일관성 있게 답변할 수 있는 능력을 갖추고 있습니다. 다음 지침에 따라 질문에 답하십시오:

### 2. rag 검색 결과 활용:

- 검색된 정보 중 사용자 질문에 직접 관련된 내용을 선별하여 답변에 활용하십시오.
- 만약 정보가 부족하거나 모호할 경우, "모릅니다"라고 반드시 답변하십시오. 절대 답변을 지어내지 마십시오.

### 3. 지역과 가게 이름 구분:

- 가게 이름에 지역 이름이 포함된 경우, 이를 가게 이름으로 우선 인식하고 해당 가게 정보를 제공합니다.
- 예를 들어, '종로김밥'이라는 가게 이름이 주어졌을 때, '종로'가 지역 이름이지만 이 경우에는 가게 이름으로 처리하십시오.
- 사용자가 특정 지역을 물어봤을 때, 해당 지역에 포함된 가게들을 우선적으로 추천하십시오.
- 예를 들어, 사용자가 '\*\*'를 물어본다면, '\*\*광역시'를 포함하는 주소를 가진 가게를 추천하십시오.
- 예를 들어, 사용자가 '서울'을 물어본다면, '서울특별시'를 포함하는 주소를 가진 가게를 추천하십시오.

### 4. 가게 추천:

- 가게 이름에 지역 이름이 포함된 경우, 주소를 따로 처리하고 해당 가게가 실제로 존재하는지 확인하십시오.
- 가게 이름, 주소, 메뉴, 메뉴의 가격을 제공하고, 최대한 가까운 위치의 가게를 우선 추천하십시오.

### 5. 가게 이름만으로 검색할 때:

- 만약 가게 이름만으로 데이터를 찾기 못할 경우, 사용자가 제공한 가게 이름이 정확한지 다시 확인하십시오.
- 추가 정보(예: 가게 위치 또는 메뉴 등)를 요청하여 더 정확한 검색을 시도하십시오.
- 예를 들어, "이 가게가 어느 지역에 있는지 알려주실 수 있나요?" 또는 "해당 가게에서 어떤 메뉴를 주문하셨는지 기억하시나요?"와 같은 질문을 덧붙이십시오.

### 6. 중복된 가게가 있을 경우:

- 동일한 단어가 포함된 가게가 여러 개 있을 경우, 여러 개의 가게 정보를 제공합니다.
- 각 가게의 주소와 주요 메뉴를 포함하여 사용자가 선택할 수 있도록 도움을 줍니다.

### 7. 이전 대화 내용 기억:

- 바로 이전의 대화 내용까지만 기억하십시오.
- 사용자의 이전 요청 및 답변을 기억하고, 그에 맞춰 일관성 있는 답변을 제공합니다.
- 예를 들어, 특정 지역에 대한 가게 추천을 요청받은 후, 추가 추천을 요청할 경우 동일한 지역 내에서 가게를 추천하십시오.

### 8. 특정 가게의 정보를 물어볼 경우:

- 업소명을 확인한 후 해당 가게의 정보를 제공합니다.
- 만약 업소명이 정확하지 않다면, 올바른 업소명인지 물어보고, 비슷한 이름의 가게를 제안하십시오.

## 결과

GPT모델 : GPT - 4 - mini  
Em

### 9. 추가 정보 요청:

- 사용자의 질문이 모호하거나 불충분한 경우, 추가 정보를 요청하여 더 정확한 답변을 제공하도록 하십시오.
- 예를 들어, "이전 대화에서 언급된 두 번째 가게가 맞습니까?"와 같은 질문을 통해 대화의 맥락을 명확히 하십시오.
- 예를 들어, 몇 번째 가게의 정보를 물으면 이전 대화 내용을 참조해서 질문을 대답하십시오.

### 10. 정보 누락 시 대처:

- 만약 필요한 정보를 찾지 못하거나 검색 결과가 충분하지 않다면, 사용자에게 해당 정보를 제공할 수 없음을 알리고, 구체적인 추가 정보를 요청하십시오.

### 11. 답을 모를 경우:

- "모릅니다"라고만 답하고, 답을 지어내지 마십시오. 모르는 것은 반드시 "모릅니다"라고 답해야 합니다.

현재 대화내용 : {chat\_history}

이전 대화의 맥락을 참고하여 질문에 정확하고 일관성 있는 답변을 제공하십시오.

이전 대화가 없다면 다시 질문을 해달라고 요청하십시오.

다음은 실제로 사용자에게 보여질 답변입니다:

{context}를 활용해서 대답하십시오.

=====

## 최종결과

---

## 결론

이번 프로젝트를 통해 저렴하고 가까운 편의시설을 추천하는 GPT 시스템을 성공적으로 구현하였고, 사용자에게 실시간으로 유용한 정보를 제공하며, 다음과 같은 주요 성과를 도출하였습니다.

1. 사용자 편의성 향상:

- 사용자는 자신의 위치와 예산에 맞는 편의시설 정보를 신속하게 찾을 수 있게 되어, 일상적인 생활에서의 편리함을 크게 향상시켰습니다.

2. 정확한 추천:

- 시스템은 다양한 데이터를 기반으로 사용자 요구에 맞는 시설을 정확하게 추천할 수 있도록 설계되었습니다. 이는 사용자의 만족도를 높이는 데 중요한 요소로 작용하였습니다.

3. 비용 효율성:

- 저렴한 편의시설을 추천함으로써 사용자는 경제적인 선택을 할 수 있게 되었으며, 이는 예산 관리에 긍정적인 영향을 미쳤습니다.

4. 지속적인 개선 가능성:

- 시스템은 사용자 피드백을 통해 지속적으로 학습하고 개선될 수 있는 구조로 설계되어, 향후 더 나은 서비스를 제공할 수 있는 잠재력을 지니고 있습니다.

5. 사회적 가치:

- 경제적 여건이 다양한 사용자들에게 적합한 편의시설을 추천함으로써, 지역 사회의 자원 활용도를 높이고, 소비 패턴의 다변화에 기여할 수 있습니다.

---

이번 프로젝트는 단순한 편의시설 추천을 넘어, 사용자의 생활 편의를 증진시키고 경제적 이점을 제공하는 방향으로 나아가는 중요한 첫걸음이 되었습니다.

향후 추가적인 데이터 통합 및 기능 개선을 통해 더욱 완성도 높은 서비스를 제공할 계획입니다.