

Project

프로젝트	세무 자동화 프로젝트
개발기간	2024.02.29 ~ 2024.03.29 (총 21일, 168시간)
참여인원	5명
담당업무	1. 프로젝트 제안 (주제 선정 및 아이디어 회의 30%) 2. 프로젝트 기획 (AS-IS MAP 제작, TO-BE MAP 제작 25%) 3. 프로그램 구현 (가상데이터 생성 100% 가계별 매출 정보 YoY 100% 관리자/가계별 로그인 아이디/비밀번호 만들기 100% UI 제작 75%)
개발환경	Visual Studio Code, GitHub, StarUML
사용도구	BeautifulSoup, PyQt5, ReportLab, Tkinter
사용기술	Python, Web Crawling



개요

1. 목표 및 문제 정의

- 세무 비용을 아끼려는 소상공인들의 어려움을 해결

소상공인들이 세무 비용을 절감하고 효과적으로 관리할 수 있는 솔루션을 제공하고자 합니다.

많은 소상공인들이 세무 관리에 어려움을 겪고 있으며, 이를 해결할 수 있는 도구가 필요합니다.

2. 주요 기능 및 특징

- 세무 업무

사용자가 수입과 지출 데이터를 홈페이지에 입력하면 매출의 변화(YoY)와 보험료, 세금 등이 자동으로 계산되어 나타납니다.

- 절세 방안 안내

사용자의 데이터에 기반하여 절세 방안을 제시합니다.

- **pdf** 저장 기능

안내 받은 내용은 **pdf** 파일로 저장이 가능합니다.

3. 대상 사용자

- 소상공인

세무 관리에 어려움을 겪고 있는 소상공인들이 주요 사용자입니다.

4. 기술 스택

- 크롤링

직접 제작한 홈페이지에서 가게의 수입과 지출 데이터를 자동으로 수집합니다.

- **UI** 생성

TKinter와 PyQt5를 이용하여 **UI**를 디자인하고 구현합니다.

- **PDF** 기능

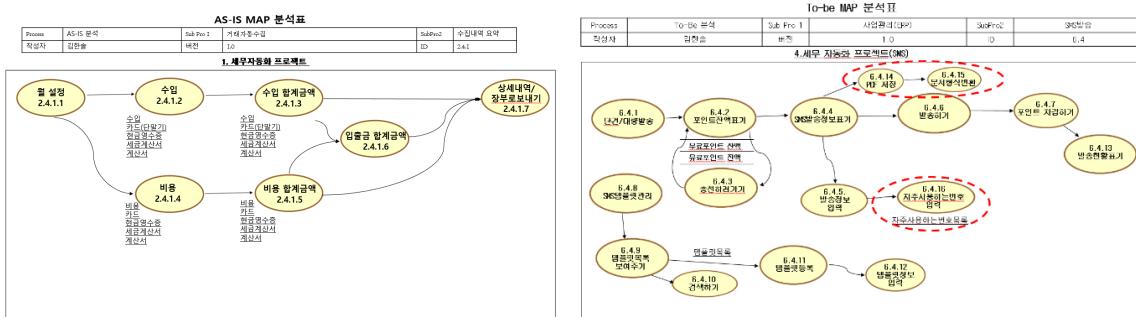
ReportLab을 이용하여 생성된 데이터를 **PDF**파일로 저장하는 기능을 제공합니다.

기획 의도(동기)

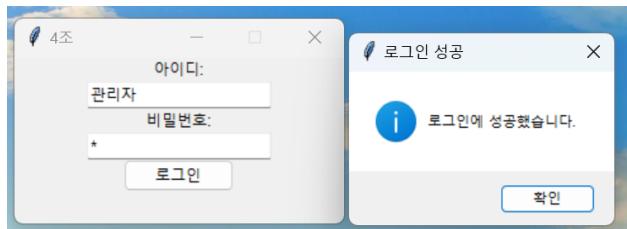
대부분의 소상공인들은 재무 및 회계 분야에 대한 전문 지식이 부족합니다. 또한, 매출 규모가 크지 않거나 재정 상태가 좋지 않아 세무사를 고용하기에는 어려움이 있습니다. 이러한 소상공인들을 돋기 위해, 절세를 위한 실질적인 방안을 제공할 수 있는 프로그램을 개발하였습니다. 이 프로그램은 사용자에게 세무 관련 정보를 제공하고, 수입과 지출 데이터를 분석하여 절세 방안을 제시함으로써 소상공인들이 비용을 절감하고 재무 관리를 효율적으로 할 수 있도록 지원합니다.

목표 및 설계

1. 소상공인들을 위한 회계 프로그램 (AS-IS MAP 및 TO-BE MAP)



2. 프로그램 가입자들의 개인정보 보호를 위한 아이디/비밀번호 로그인 시스템



3. 매출액 비교를 위한 YoY 제공

세무/회계 자동화 프로그램																																						
Annual Tax Private Insurance																																						
<p>가계의 수는 웹사이트에서 크롤링 해온 만큼 생성된다</p>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th>1</th> <th>2</th> <th>3</th> </tr> </thead> <tbody> <tr> <td>총 매출</td> <td>180000000</td> <td>120000000</td> <td>+33.33%</td> </tr> <tr> <td>재료비</td> <td>4500000</td> <td>4300000</td> <td>+4.44%</td> </tr> <tr> <td>인건비</td> <td>3000000</td> <td>2600000</td> <td>+13.33%</td> </tr> <tr> <td>소모품</td> <td>350000</td> <td>3400000</td> <td>+2.86%</td> </tr> <tr> <td>주당대</td> <td>24800000</td> <td>22800000</td> <td>+8.06%</td> </tr> <tr> <td>임차료</td> <td>24000000</td> <td>21000000</td> <td>+12.5%</td> </tr> <tr> <td>공과금</td> <td>8000000</td> <td>7000000</td> <td>+12.5%</td> </tr> <tr> <td>기부금</td> <td>10000</td> <td>10000</td> <td>0.0</td> </tr> </tbody> </table> <p>매출(YoY)이 위에 표시된다</p>		1	2	3	총 매출	180000000	120000000	+33.33%	재료비	4500000	4300000	+4.44%	인건비	3000000	2600000	+13.33%	소모품	350000	3400000	+2.86%	주당대	24800000	22800000	+8.06%	임차료	24000000	21000000	+12.5%	공과금	8000000	7000000	+12.5%	기부금	10000	10000	0.0	<p>절제 조언이 아래에 표시된다</p> <p>세금 절약을 위한 조언:</p> <ul style="list-style-type: none"> - 비즈니스 비용을 정확하게 기록하고 가능한 모든 공제를 활용하세요. (예: 업무 관련 비용, 자격 요건에 맞는 공제 항목을 신청하세요) - 세무 전문가와 상의하여 가능한 세액 공제를 모두 활용하는 것이 좋습니다. (예: 세법의 변경사항을 파악하고 세액 공제 가능 여부를 확인하세요) - 소모품 구매 비용에 대한 공제가 가능합니다. - 기부금 공제 가능합니다. - 대출 이자 공제가 불가능합니다. 최대 이자 공제 가능 금액을 초과했습니다. <p>PDF로 저장</p>
	1	2	3																																			
총 매출	180000000	120000000	+33.33%																																			
재료비	4500000	4300000	+4.44%																																			
인건비	3000000	2600000	+13.33%																																			
소모품	350000	3400000	+2.86%																																			
주당대	24800000	22800000	+8.06%																																			
임차료	24000000	21000000	+12.5%																																			
공과금	8000000	7000000	+12.5%																																			
기부금	10000	10000	0.0																																			

성과

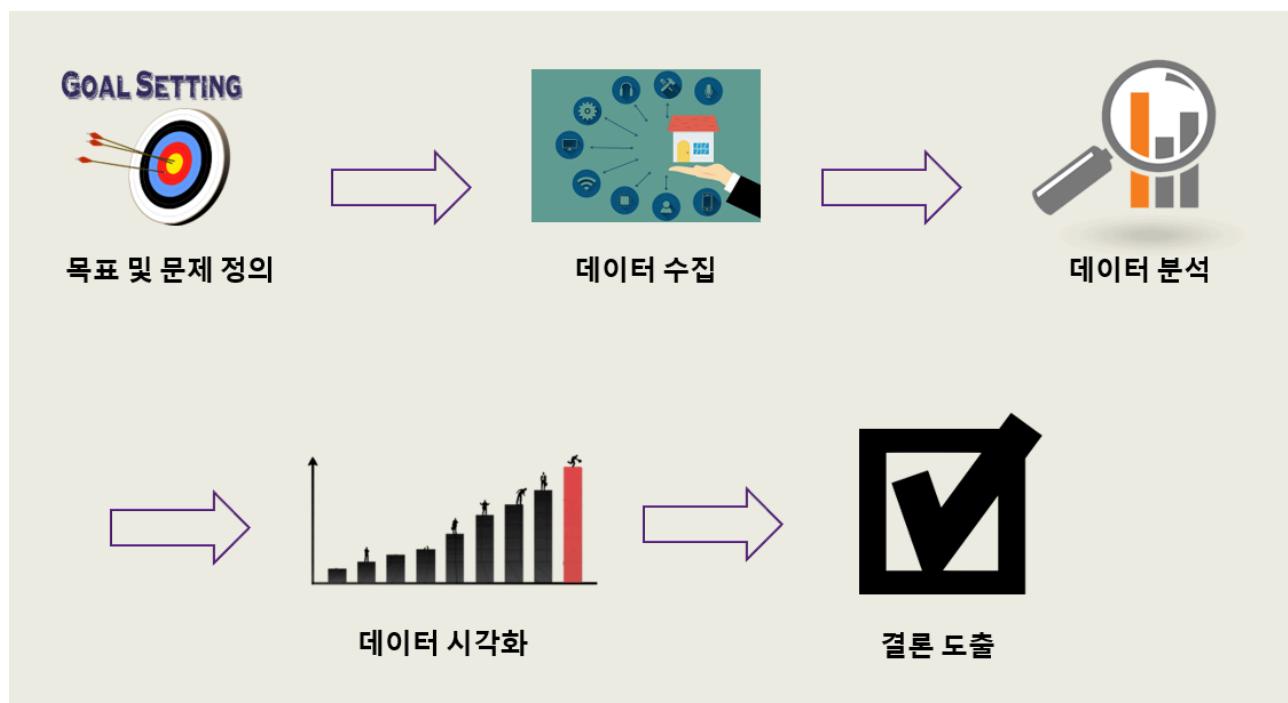
<기술적인 후기>

1. AS-IS MAP과 TO-BE MAP을 활용하여 기존 비즈니스 프로세스를 분석하고 개선할 수 있는 방안을 도출했습니다. 이를 통해, 시스템 개발 및 프로젝트 관리의 기초를 설계할 수 있었으며 효율성을 높이는 개선안을 만들 수 있었습니다.
2. StarUML을 이용하여 USE CASE 다이어그램을 작성하고, 사용자와 시스템간의 상호작용을 명확히 정의하고 시각화하는 방법을 익혔습니다. 이를 통해, 소프트웨어 시스템 구조와 동작을 효과적으로 모델링하고, 프로젝트 요구사항을 체계적으로 분석할 수 있었습니다.
3. Tkinter와 PyQt5를 이용하여 실제 UI를 설계하고 구현하였습니다. 프로젝트 전에 Tkinter에 관해서는 사전지식이 있었기에 먼저 구현하려고 하였지만, 스스로 만족할만한 퀄리티를 내지 못하였습니다. 또한, 기능적 측면에서 부족하다고 느껴 UI 라이브러리들을 검색하였고, PyQt5에 대해 알아보고 프로젝트에 적용을 하였습니다. 이를 통해 프로젝트의 완성도를 높이기 위해 다양한 라이브러리를 찾아보고 익히는 훈련이 스스로 되었으며, 애플리케이션 개발 능력을 향상할 수 있었습니다.

<의사소통 후기>

1. 프로젝트의 성공을 위해 주제 선정부터 시작하여 사전 회의를 통해 목표와 전략을 명확히 설정했습니다. 또한 매일 작업의 진척도에 대한 회의를 하며, 코딩을 하다가 어려운 부분이 있으면 함께 고민해주고 해결하려고 노력하는 모습들을 통해 강력한 팀워크를 형성할 수 있었습니다.
2. GitHub를 활용한 협업에서 효율적인 버전 관리를 통해 코드의 품질을 높이고 작업의 일관성을 유지할 수 있었습니다. 처음으로 사용하는 GitHub였지만, Git과 GitHub에 대한 내용과 설치법, 활용법 등을 팀원들과 함께 내용을 공유하며 프로젝트를 진행하였습니다. 이를 통해 GitHub의 사용법을 익히는 과정에서 브랜치 관리의 중요성을 이해할 수 있었습니다.

프로젝트	데이터 분석 프로젝트
개발기간	2024.03.28 ~ 2024.04.17 (총 15일, 120시간)
참여인원	3명(팀장)
담당업무	프로젝트 제안(100%) 프로젝트 기획(100%) 데이터 수집(50%) : OECD 홈페이지 등을 통해 데이터 수집 데이터 전처리 및 분석(50%) : Pandas 라이브러리를 통한 전처리 및 분석 데이터 시각화(50%) : Matplotlib을 통한 시각화 앱 개발 및 배포(100%)
개발환경	Jupyter Notebook, GitHub, Visual Studio Code
사용도구	Matplotlib, Pandas, Streamlit
사용기술	Python, Data Visualization, Data Analysis, Web Application Development



개요

1. 목표 및 문제 정의

- 정책 타당성 검증

대한민국 정부의 의대생 정원 증원 정책에 대한 타당성을 검증하기 위한 분석을 수행합니다.

- 현재 의료 상황 분석

한국의 현재 의료 상황과 미래 인구 추세를 파악하고 정책의 필요성과 효과를 평가합니다.

- 국제 비교 및 적합성 평가

다른 국가의 의료제도 및 의사 수와 비교하여 이 정책의 적합성을 평가하는 것을 목표로 합니다.

2. 주요 기능 및 특징

- 데이터 수집

의사 수, 인구 추세, 고령화 비율, 경제적 요소 등 관련 데이터를 수집합니다.

- 데이터 전처리 및 분석

수집된 데이터를 기반으로 필요한 정보를 쉽게 알 수 있도록 전처리를 진행한 후 분석을 수행하여 가설을 검증합니다.

- 결과 시각화

분석 결과를 그래프, 차트 등으로 시각화하여 이해를 돋습니다.

3. 대상 사용자

- 정책 결정자

정부 관계자 및 정책 입안자 등이 포함됩니다.

- 공공기관 관계자

보건복지부, 건강보험공단 등의 관계자들이 해당됩니다.

- 연구자 및 의료전문가

의료 및 보건 관련 분야 연구자 및 실무자들이 대상입니다.

4. 기술 스택

- 데이터 분석 및 시각화

Matplotlib과 Pandas를 활용하여 수행합니다.

- 웹 애플리케이션 개발

Streamlit을 사용하여 웹 애플리케이션을 개발하고 배포합니다.

기획 의도(동기)

대한민국 정부는 2006년 이후 의대생 증원을 하지 않고 있었습니다. 그러나 최근 매년 2,000명의 의대생 정원을 늘리는 정책을 내놓았습니다. 이에 대한의사협회와 의대생들은 큰 반발을 하였고 의료 교육 질 저하, 의료비 증가, 의료 환경 악화 등의 우려를 내놓았습니다.

이 프로젝트는 이러한 많은 우려들을 바탕으로 정부가 의대생 증원이라는 정책이 타당한지 검증하고자 기획했습니다. 현재 대한민국의 의료 상황과 미래 인구 및 고령화 추세를 분석하고, 다른 나라의 의료제도 및 의사 수 등을 비교하여 이 정책이 실제로 적합한지 평가하고자 합니다.

목표 및 설계

1. 귀무가설 및 대립 가설 수립

귀무가설(H0) : 의대생을 증원해야 한다는 주장은 잘못되었다.

대립가설(H1) : 의대생 증원을 통해 의료 시스템의 질과 효율성을 개선할 수 있다.

2. 연구 문제와 목표

- 대한민국 의사 숫자는 적당한가?

OECD 주요나라들과 의사의 숫자를 비교해본다. (인구 1천명 대비 의사수)

- 대한민국 의사들의 근무환경은 만족스러운가?

대한민국 의사들의 근무시간을 확인해본다.

- 대한민국 의사들의 급여는 적당한가?

의대생 증원으로 인해 의사의 수가 많아진다면 경제적으로 어려운 상황이 다가올 수 있다.

3. 데이터 수집

- 다양한 경로를 통해 데이터를 수집 (온라인 뉴스, 개인 홈페이지, 통계청, OECD 홈페이지 등)

ㄱ. 동아시아 고령화의 대표적 나라이 일본과 비교



ㄴ. OECD 주요 국가들과 봉직의 연봉을 비교

한국 의사 연봉		
국가	봉직의 연봉	1인당 GDP 대비
한국	246,333,222	5.06
네덜란드	245,713,392	3.35
독일	240,454,422	3.43
아일랜드	216,094,464	1.78
영국	198,625,482	3.86
벨기에	180,824,220	2.7
이스라엘	176,747,400	3.51
칠레	171,576,612	5.12
헝가리	151,673,040	3.74
스페인	149,294,682	2.88
노르웨이	148,839,714	1.17
스웨덴	142,930,242	1.86
이탈리아	139,593,384	2.41
프랑스	134,960,634	2.53
멕시코	67,445,172	3.12
폴란드	60,136,290	1.46

2023년 7월 28일 달러 환율 기준

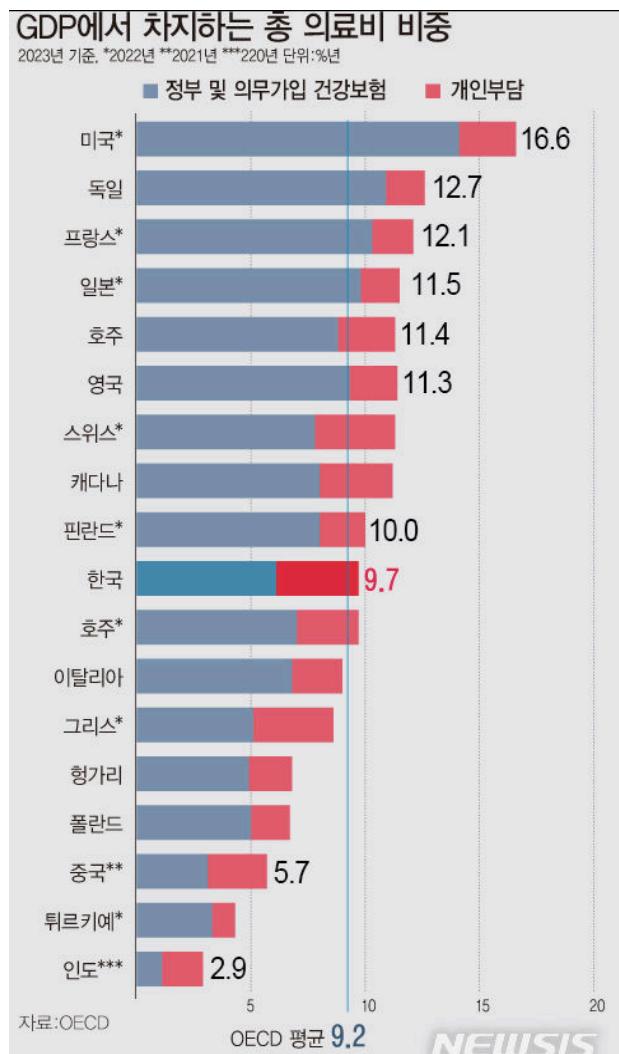
c. 의료비 중 정부·건강보험 비율

	의료비 중 정부·건강보험 비율 단위: %				
	전체 서비스	입원	외래진료	치과	약값
체코	86	97	91	42	56
스웨덴	86	99	90	43	55
독일	85	97	90	67	82
일본	85	92	85	80	72
프랑스	85	96	85	31	83
아이슬란드	84	99	83	41	40
영국	83	95	91	34	66
OECD32	76	90	79	32	56
캐나다	73	92	83	15	38
호주	72	63	87	14	51
스위스	68	83	67	6	59
한국	62	68	57	36	49
그리스	62	66	65	-	51
브라질	41	47	58	33	9

NEWSIS

[서울=뉴스데일리] 전체 의료비 중 정부와 건강보험 커버 비율을 보면 우리나라의 전체 서비스 기준으로 62%로, 뒤에서 세 번째다. 우리보다 뒤쳐진 나라는 그리스와 브라질 뿐이다. 의료서비스 전반에 걸쳐 정부와 건강보험, 즉 공공부문이 커버해 주는 비율이 낮다. (그래픽= 전진우 기자) 618tue@newsis.com

☞ GDP에서 차지하는 총 의료비 비중

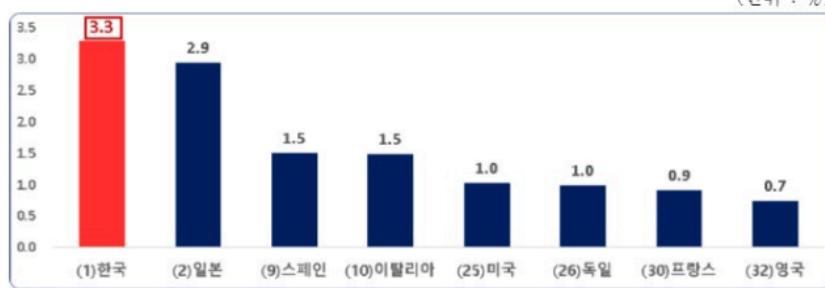


▣. 통계청에서 추정하는 대한민국 고령화 예상 추이

대한민국의 노인 비율 (1960년 ~ 2060년)	
1960년	3.3%
1970년	3.4%
1980년	3.8%
1990년	4.8%
2000년	6.8%
2010년	10.7%
2020년	15.0%
2030년	24.1%
2040년	32.8%
2050년	39.0%
2060년	43.1%
2067년	46.2%

▣. OECD 주요국 연평균 고령화비율 증가율 비교

< [1970년 ~ 2018년] OECD 주요국 연평균 고령화비율 증가율 비교 >
(단위 : %)

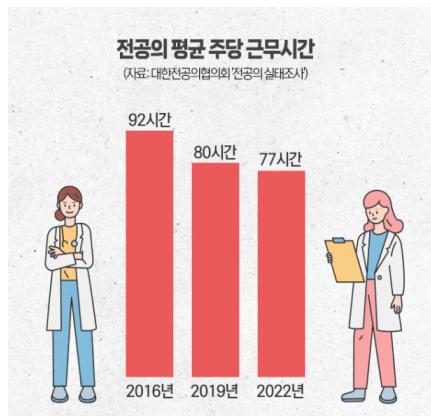


* 자료 : OECD

ㅅ. 대한민국 의사와 OECD 의사당 진찰 건수 비교 (2019년 보건복지부 자료)

우리나라의 의료공백 문제는 OECD 국가와 비교했을 때도
심각한 수준. 한국 의사 1인당 진찰 건수는 6989건으로
OECD 국가 중 1위였으며,
OECD 평균인 2130건보다 3.3배나 많았습니다.

ㅇ. 대한민국 전공의 평균 주당 근무시간



4. 수집한 데이터들을 취합·분석

ㄱ. 의사 숫자에 대한 고찰

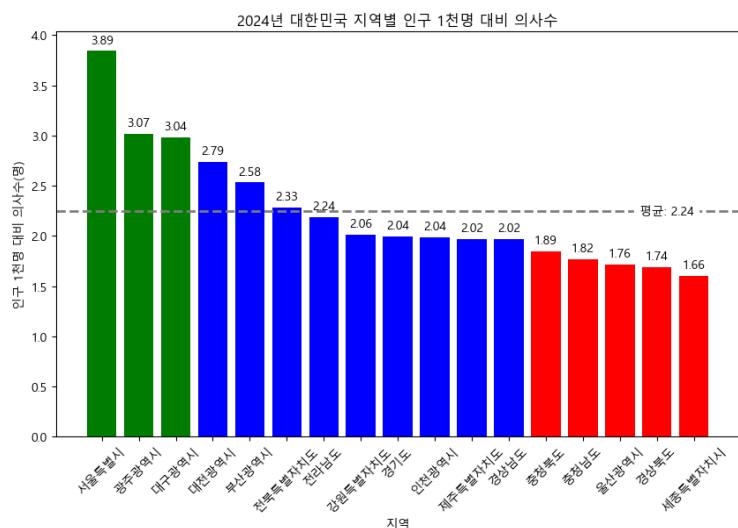
대한민국의 2021년 기준 1천명당 의사 수는 3.07명으로, 고령화가 가장 높은 4개국(이탈리아, 스페인, 한국, 일본)의 평균인 4.78명에 비해 상당히 낮습니다. 이는 절대적 의사 수가 부족함을 나타내며, 대한민국 전공의들의 높은 근무시간과 의사 1인당 진찰건수 등을 통해 재확인할 수 있습니다. 또한, OECD 국가들 중에서 한국 봉직의의 연봉은 1인당 GDP 대비 5.06배로, 칠레에 이어 두 번째로 높은 수준입니다. 그럼에도 불구하고, 한국의 고령화 증가율은 3.3%로 OECD 국가들 중 가장 높으며, 앞서 언급한 4개국의 평균 증가율인 2.3%를 훨씬 상회합니다. 이를 종합할 때, 의대생 증원은 OECD 국가들과 비교해 볼 때 필수적으로 보입니다.

ㄴ. 정부의 의료비 부담에 대한 고찰

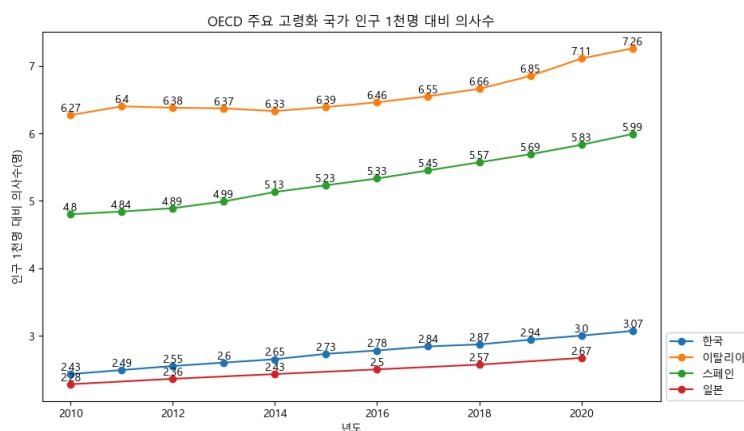
대한민국의 경우 의료비 중 정부·건강보험 비율이 62%로, OECD 32개국 평균 76%에 비해 낮은 편이다. 또한 입원, 외래진료, 약값 등 다양한 분야에서도 그 비율이 낮다. 이는 개인 혹은 가계가 의료비를 직접 부담하는 비중이 높음을 뜻하므로, 보장성이 부족하거나 의료비로 인한 경제적 어려움을 초래할 수 있는 문제점이 있음을 나타낸다. 다만 GDP에서 차지하는 총 의료비 비중은 9.7%로 OECD 평균 9.2%와 크게 차이는 나지 않지만 앞으로 급속한 고령화 사회가 다가온다면 이 수치는 점점 증가할 것이며, 의료 시스템의 부담도 커질 전망으로 보입니다.

5. 데이터 시각화

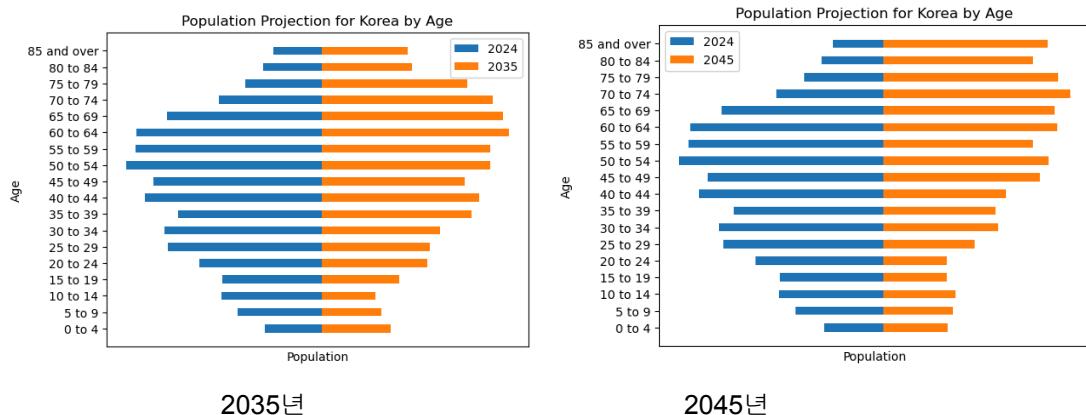
ㄱ. 2024년 대한민국 지역별 인구 1천명 대비 의사수



ㄴ. OECD 주요 고령화 국가 인구 1천명 대비 의사수



ㄷ. 대한민국 인구 그래프 현재와 미래



5. 앱 개발 및 배포

ㄱ. Anaconda Prompt를 이용하여 Streamlit 서버를 구축

```
(base) C:\Users\sr48g>conda env list
# conda environments:
#
base                  *  C:\Users\sr48g\anaconda3
myST                 C:\Users\sr48g\anaconda3\envs\myST
myST3                C:\Users\sr48g\anaconda3\envs\myST3

(base) C:\Users\sr48g>conda activate myST
(myST) C:\Users\sr48g>streamlit run "C:\Users\sr48g\OneDrive\바탕 화면\2nd_project\2nd_project.py"
You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: :8501
```

A screenshot of the Anaconda Prompt window titled 'Anaconda Prompt - streamlit'. It shows the command 'conda env list' being run, listing environments like 'base', 'myST', and 'myST3'. Then, 'conda activate myST' is run, followed by 'streamlit run "C:\Users\sr48g\OneDrive\바탕 화면\2nd_project\2nd_project.py"'. The prompt ends with 'You can now view your Streamlit app in your browser.' and provides local and network URLs for port 8501.

ㄴ. Streamlit을 이용하여 간단한 웹 앱을 배포

공공데이터를 이용한 AI 챗봇
과정

조장 : 김한솔

조원 : 이혁빈, 활성현

역할분담

김한솔 : 데이터수집·정리·시각화, 앱 개발 및 배포, 사전 및 중간 발표

이혁빈 : 데이터수집·정리·시각화

활성현 : 데이터 수집·정리, 최종 발표

2 번째 프로젝트 😊

대한민국 의사증원, 몇 명이 적절한가

데이터 분석 데이터 시각화 프로젝트 결과

대한민국은 OECD 38개국들과 비교하여 1천명당 의사수가 2021년 기준 3.07명으로 고령화 수치가 가장 높은 4개국(이탈리아, 스페인, 한국, 일본) 평균인 4.78명에 상당히 못 미쳤습니다

시장 경제 체제에서 수요와 공급이 가격을 결정하듯이 OECD 국가들 중 농직의 연봉은 1인당 GDP 대비 5.06배로 질레 5.12배 다음 2번째로 높았다

그럼에도 불구하고 고령화 증가 비율은 3.3%로 OECD 국가들 중 가장 높으며 앞선 4개국의 고령화 증가율 평균은 2.3%에 불과한것으로 나타났다

이에 첫 번째로 내릴 수 있는 결론은 의대생 증원은 OECD 국가들의 상황과 비교할 때, 필수적으로 보인다.

This screenshot shows a Streamlit application interface. On the left, there's a sidebar with developer information and roles. The main area has a title '2 번째 프로젝트 😊' and a subtitle '대한민국 의사증원, 몇 명이 적절한가'. Below the title are three colored buttons: yellow for '데이터 분석', red for '데이터 시각화', and green for '프로젝트 결과'. The main content area contains text comparing South Korea's physician density to other OECD countries, mentioning high aging rates and low physician ratios. It also notes the high wage-to-GDP ratio in the market economy.

성과

< 기술적 후기 >

1. 데이터 추출 및 가공

OECD 공식 홈페이지에서 다운받은 csv 파일에서 국가별 의사 수와 인구 데이터를 **Pandas** 라이브러리를 활용해 추출하여 가공하였습니다. 각 국가의 의사수를 해당 국가의 인구 수로 나누어 1000명당 의사수 비율을 계산했습니다. 이러한 데이터를 정리하고 가공하는 과정에서 데이터의 일관성을 유지하며, 분석의 기초가 되는 지표들을 도출 해냈습니다.

2. 데이터 시각화

제공한 데이터를 보다 직관적으로 이해할 수 있도록 데이터 시각화 작업을 진행했습니다. 이를 위해 **Matplotlib** 라이브러리를 활용하여 각 국가의 의사 비율을 그래프 형태로 표현했습니다. 이 시각화 작업을 통해 국가간의 의사 비율 차이를 한 눈에 파악할 수 있었으며, 이를 바탕으로 대한민국의 의사 수가 다른 OECD 국가들과 비교했을 때의 모습을 명확히 알 수 있었습니다.

3. 앱 개발 및 배포

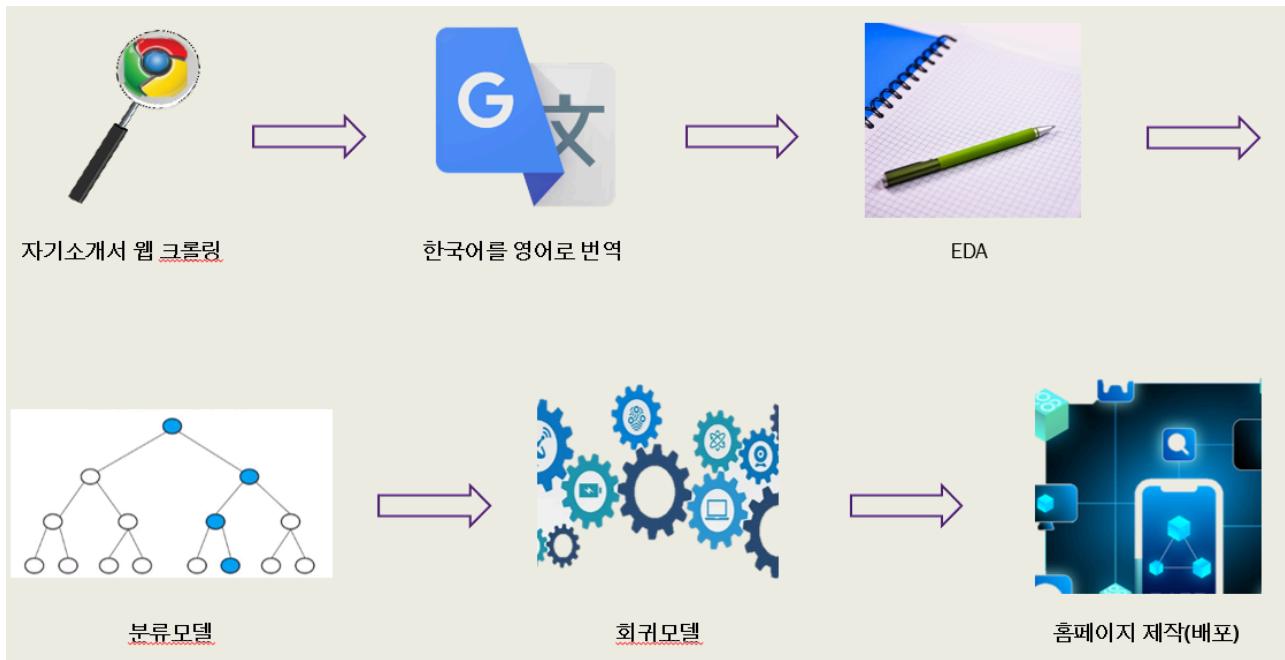
마지막으로, 프로젝트 결과를 사용자들에게 쉽게 공유하고자 **Streamlit**을 사용하여 웹 애플리케이션을 배포하였습니다. 이 앱을 통해 사용자들이 직접 분석 결과를 탐색할 수 있으며, 시각화된 데이터를 바탕으로 다양한 인사이트를 얻을 수 있습니다.

<의사소통 후기>

프로젝트 진행 중 팀원 1명이 나가면서 프로젝트 마무리에 대한 불안감이 생겼습니다. 이를 극복하기 위해 함께 식사하는 등 의사소통의 기회를 더 가지고 팀 분위기를 좋게 유지하려고 노력하였습니다. 이러한 노력 덕분에 팀의 결속력을 강화할 수 있었고, 프로젝트를 무사히 마무리 할 수 있었습니다.

또한, 팀장으로서의 책임감을 가지고 주말에도 시간을 내어 앱 개발과 배포를 마무리 했습니다. 비록 완성도는 다소 낮았지만, 기획부터 데이터 추출과 가공, 시각화, 웹 애플리케이션 개발까지 전 과정을 수행한 점에 의미를 두고 있습니다. 이를 통해 데이터 분석과 앱 개발이 실제 의사결정에 어떻게 기여할 수 있는지 깊게 이해할 수 있었으며, 앞으로의 프로젝트에 큰 도움이 될 것이라 생각합니다.

프로젝트	자소서 적합 예측 프로젝트
개발기간	2024.06.04 ~ 2024.06.24 (총 15일, 120시간)
참여인원	4명
담당업무	<p>프로젝트 제안(100%)</p> <p>프로젝트 기획(100%)</p> <p>데이터 수집(25%) : 자소서 크롤링</p> <p>데이터 분석(33%) : 자소서 워드 클라우드 생성</p> <p>분류 모델 개발(50%) :</p> <ul style="list-style-type: none"> NLTK와 KoNLPy를 이용하여 한국어 자소서들의 분류 모델을 구현(100%) NLTK를 이용하여 영어 자소서들의 분류 모델을 구현(25%) RandomForest, SVM, Ensemble 사용 GridSearch, RandomSearch 사용 <p>회귀 모델 개발(50%) :</p> <ul style="list-style-type: none"> NLTK와 KoNLPy를 이용하여 한국어 자소서들의 회귀 모델을 구현(100%) 앱 개발 및 배포(10%) : Google Cloud를 이용하여 구현 시도
개발환경	GitHub, Jupyter Notebook, Visual Studio Code, Google Colab
사용도구	Optuna, KoNLPy, Doc2Vec, Scikit-learn(RandomForest GridSearch), NLTK, Selenium, FastAPI
사용기술	Python, Machine Learning



개요

1. 목표 및 문제 정의

- 자기소개서 평가

자기소개서 검토가 필요한 사람들에게 자기소개서의 완성도를 평가하는 기능을 제공합니다. 이 프로젝트는 특히 졸업예비생, 취업준비생, 그리고 이직을 준비하는 이들에게 큰 도움이 될 것입니다.

2. 주요 기능 및 특징

- 데이터 크롤링

Selenium을 활용하여 합격자들의 자기소개서를 크롤링하고 네이버 파파고 번역기를 이용한 후, Doc2Vec 알고리즘으로 벡터화하여 분석을 진행하였습니다.

- 분류모델

최신모델(SVM, Light GBM, XGBoost)들을 사용하여 모델을 개발하였으며 Optuna를 이용해 최적의 하이퍼파라미터 튜닝을 진행하였습니다.

- 웹 배포

누구나 쉽게 접근할 수 있도록 웹사이트로 배포되어, 사용자는 별도의 설치없이 onrender 사이트를 통해 서비스에 접속하여 본인의 자기소개서를 평가받을 수 있습니다.

- 자기소개서 평가

사용자가 본인의 자기소개서를 입력하면, 모델이 이를 분석하여 평가를 내립니다.

3. 대상 사용자

- 졸업예비생, 취업준비생, 이직

진로를 준비하는 다양한 사용자들이 자기소개서 검토를 위해 사용할 수 있습니다.

4. 기술 스택

- 다양한 Python 라이브러리 사용

Optuna, KoNLPy, Doc2Vec, Scikit-learn, NLTK 등 다양한 Python 라이브러리를 사용하여 데이터 처리, 분석 및 모델링을 수행하였습니다.

- 머신러닝

문서 임베딩 및 하이퍼파라미터 튜닝 최적화를 통해 머신러닝의 모델 성능을 최적화하였습니다.

기획 의도(동기)

많은 사람들이 졸업 이후 한 직장에서 오래 근무하며 단 한 번의 취업준비만 하는 시대는 지나갔습니다. 이제는 지속적으로 더 좋은 직장을 찾아 이직을 준비하며 그만큼 자기소개서의 완성도에 신경을 더 많이 쓰는 사람들이 늘어나고 있습니다. 그들 중 어떤 자기소개서가 좋은 것인지 명확한 기준을 모르는 사람들이 대부분이며, 본인의 자기소개서를 전문가에게 보여주기 부끄러워하는 사람들도 있습니다. 이 모든 걱정과 어려움 속에 있는 사람들에게 자기소개서를 웹사이트에 입력하기만 하면 평가해주는 기능을 제공하고 싶었습니다.

목표 및 설계

1. 데이터 수집(크롤링)

Selenium을 이용하여 잡코리아 사이트에서 전문가의 평점(1점~5점)이 있는 합격자 자소서 중 IT(개발, 데이터)분야만을 선택해서 가지고 온다(323개). 크롤링 후 파일은 csv 형식으로 저장을 한 후 정규화를 거쳐 Doc2Vec을 이용하여 자소서들을 벡터화합니다.

2. 분류 모델 기준 설정

크롤링을 한 자소서들은 각각 1점부터 5점까지의 점수가 있으며, 최초로 1점은 불합격, 5점은 합격으로 분류를 시작하였습니다. 그렇지만 1점과 5점의 자소서들의 숫자가 많으므로 1점과 2점을 불합격으로, 4점과 5점을 합격으로 분류한 후 최종 모델 생성을 진행하였습니다.

3. 분류 모델 생성(한국어)

NLTK 및 KoNLPy Tokenizer를 이용한 후 RandomForest, SVM, Ensemble 기법 등을 통해 모델 생성 후 성능을 확인해보았습니다. 하이퍼파라미터튜닝은 RandomSearch와 GridSearch를 이용하였습니다.

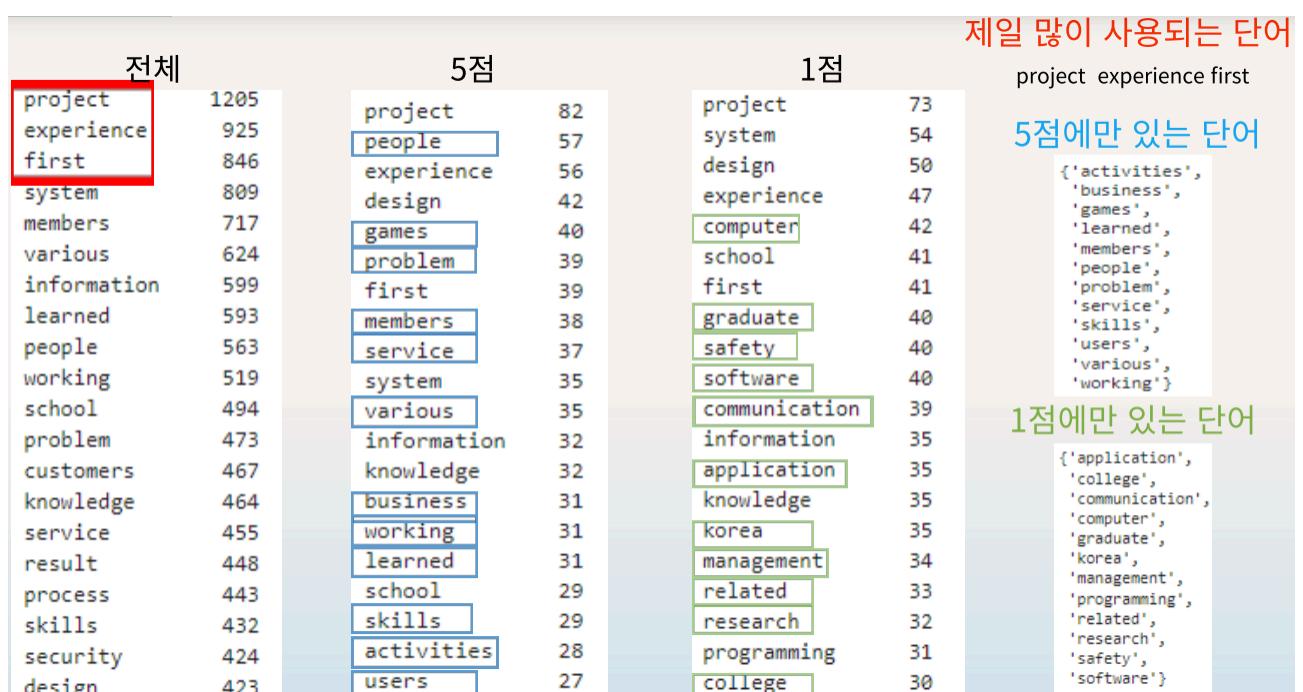
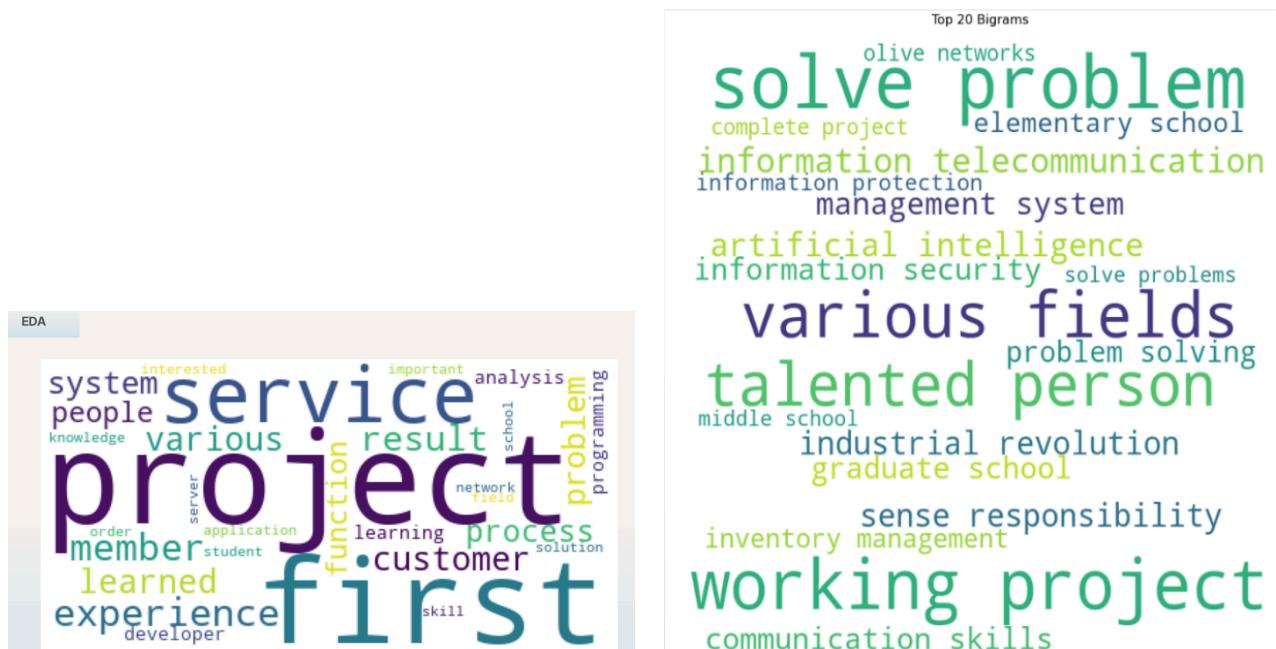


4. 한국어 → 영어 번역

한국어를 이용한 분절이 어렵다고 판단하고 네이버 파파고 번역기를 사용해 번역을 진행하였습니다. Selenium을 이용하여 csv파일로 된 한국어 자소서에서 내용을 가지고 온 후 번역된 결과값을 새로운 csv 파일로 저장하였습니다. 네이버 파파고 번역기는 한 번에 최대 3천자까지 들어갈 수 있으므로, 길이 제한을 두어 글이 3천자가 넘는 경우 변수로 x1, x2로 설정하여 각각을 번역기에 돌린 후 두 값을 translated_x1과 translated_x2에 저장한 후 합친 다음 결과를 받도록 하였습니다.

5. EDA

번역기를 돌린 번역한 자소서들에서 빈도가 높은 단어들을 n-gram을 통해 확인 후 워드클라우드를 만들어 시각화 했습니다. 또한, 모든 자소서, 1점 자소서, 5점 자소서 분류간에 특이점이 있는지 확인하였습니다.



5. 분류 모델 생성(영어)

NLTK의 토큰화를 활용하여 평점을 기준으로 1,2점은 불합격, 4,5점은 합격으로 나눈 후 분류 모델을 개발했습니다. 총 5가지(Logistic Regression, SVM, RandomForest, Light GBM, XGBoost) 모델을 사용하여 결과를 비교해보았으며, 최적의 성능을 얻기 위해 Optuna를 이용하여 각 모델의 하이퍼파라미터 튜닝을 진행하였습니다.

분류 모델				
Random Forest				
Logistic Regressor				
precision	recall	f1-score	support	
0	0.85	0.71	0.77	24
1	0.74	0.67	0.80	23
accuracy			0.79	47
macro avg	0.80	0.79	0.79	47
weighted avg	0.80	0.79	0.79	47
SVM				
precision	recall	f1-score	support	
0	0.78	0.88	0.82	24
1	0.85	0.74	0.79	23
accuracy			0.81	47
macro avg	0.81	0.81	0.81	47
weighted avg	0.81	0.81	0.81	47
Light GBM				
precision	recall	f1-score	support	
0	0.73	0.67	0.70	24
1	0.68	0.74	0.71	23
accuracy			0.70	47
macro avg	0.70	0.70	0.70	47
weighted avg	0.70	0.70	0.70	47
XG boost				
precision	recall	f1-score	support	
0	0.72	0.75	0.73	24
1	0.73	0.70	0.71	23
accuracy			0.72	47
macro avg	0.72	0.72	0.72	47
weighted avg	0.72	0.72	0.72	47

6. 회귀 모델 생성

한국어와 영어 모두 평점 1,2,4,5점을 기준으로 회귀모델을 만들어 보았으나 평가지표가 낮게 나와, 시도를 한 것에 의의를 두었습니다.

7. 홈페이지 제작 및 배포

FastAPI를 통해 프로그램을 만들었으며, 다양한 개발자들에게 쉽게 코드를 공유하고자 GitHub주소를 QR코드를 제작하였습니다(기한 만료). 홈페이지 주소를 통해 들어간 후 자소서를 입력하고 시간을 기다리면 결과를 확인할 수 있습니다. (홈페이지 주소 : <https://essay-classifier.onrender.com/>)



성과

< 기술적 후기 >

1. 데이터 수집 및 전처리

프로젝트 시작 단계에서 **Selenium**을 활용하여 크롤링 및 번역 과정을 수행하였으며, 이후 **EDA** 과정에서 정규화 및 워드클라우드 작업과 **Doc2Vec** 모델로 벡터화하는 과정을 경험했습니다. 특히나 **NLTK**와 **KoNLPy**를 사용하여, 한국어 자연어 처리 기술에 대해 깊은 관심을 갖게 되었습니다.

2. 모델링과 하이퍼파라미터 튜닝

다양한 머신러닝 모델을 사용해 자소서 평가 모델을 생성하고, **RandomSearch**, **GridSearch**, **Optuna** 등을 통해 하이퍼파라미터 튜닝을 경험했습니다.

3. 웹 애플리케이션 개발 및 배포

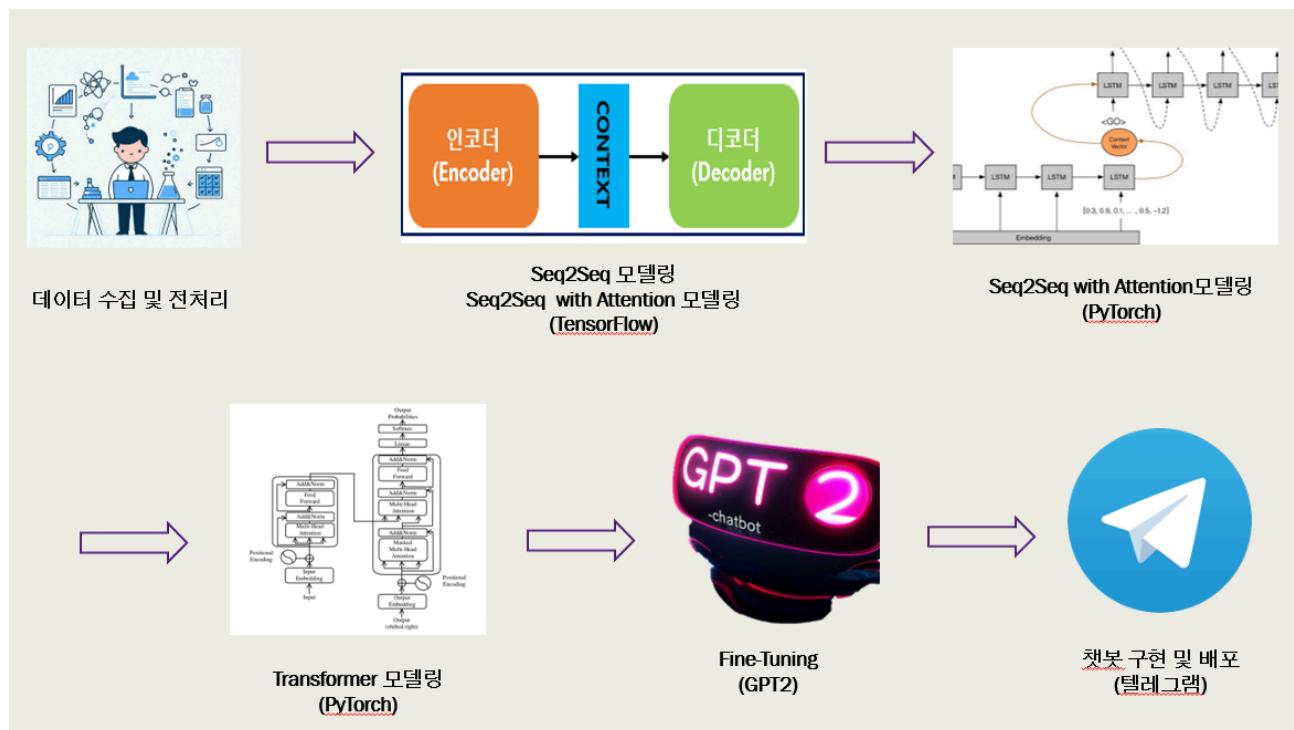
FastAPI를 통해 웹 애플리케이션을 개발하고, **QR코드**를 통해 다양한 사용자와 결과를 공유할 수 있도록 배포했습니다. 웹 애플리케이션 부분에서 사전지식들이 대부분 없었기에 **Google Cloud**, **AWS**, **Kubernetes** 등을 스스로 공부하며 웹 개발 및 배포에 관한 기술적 역량을 강화했습니다.

<의사소통 후기 >

프로젝트에 앞서 참여한 인원들 모두 이직 혹은 취업을 준비하고 있었기에, 스스로에게 도움이 최대한 될 수 있는 결과물을 만들려고 노력을 함께 했습니다. 전문가 혹은 자기소개서 검수를 하는 다른 곳들에서는 비교적 높은 금액을 요구하기에 무료로 최대한 많은 이들에게 도움을 줄 수 있도록 하자는 취지로 팀원들 모두의 기투합하여 진행하였습니다.

이번 프로젝트에서는 머신러닝의 핵심기술을 팀원 모두가 익히고자 하였습니다. 320여개의 자소서를 80개씩 나누어서 크롤링한 후, 분류모델 제작, 회귀모델 제작, 하이퍼파라미터 튜닝 등의 작업을 진행했습니다. 팀원 모두가 각자의 기준으로 먼저 프로젝트를 진행한 후 중간 중간 발생하는 문제들을 공유하고 함께 해결해나가는 방식으로 협업을 강화했습니다. 이를 통해 팀원 간의 적극적인 의사소통이 더 이루어졌으며 성공적으로 프로젝트의 전 과정을 마무리 할 수 있었습니다.

프로젝트	챗봇 프로젝트
개발기간	2024.07.11 ~ 2024.07.31 (총 15일, 120시간)
참여인원	4명(팀장)
담당업무	<p>프로젝트 제안(100%) 프로젝트 기획(100%) 사전발표(100%) 데이터 수집(100%) 데이터 전처리(25%) : 스타벅스, 이디야 커피, 탐앤탐스 등 체인점 삭제 같은 상호명이나 내용이 2개이상인 경우 단일화</p> <p>Seq2Seq 모델 구현(100%) :</p> <ul style="list-style-type: none"> Tensorflow를 이용하여 Seq2Seq, Seq2Seq with Attention 모델 구현 PyTorch를 이용하여 Seq2Seq 모델 구현 <p>PyTorch를 이용하여 Transformer 모델 구현(10%) :</p> <ul style="list-style-type: none"> SentencePiece 분절화를 거친 Transformer 모델 구현 KoBERT 분절화를 거친 Transformer 모델 구현 <p>GPT2를 Fine-Tuning한 모델 구현(100%) :</p> <ul style="list-style-type: none"> PyTorch를 이용하여 GPT2에 Fine-Tuning한 모델 구현 <p>최종 챗봇 구현 및 배포(50%) :</p> <ul style="list-style-type: none"> 텔레그램 챗봇 구현 (GPT2 Fine-Tuning 모델)
개발환경	Google Colab, Visual Studio Code, GitHub, Jupyter Notebook
사용도구	Tensorflow, PyTorch, SentencePiece, KoBERT, GPT2
사용기술	Python, Deep Learning, Seq2Seq, Attention mechanism, Fine-Tuning, Transformer



개요

1. 목표 및 문제 정의

- 서울시 맛집 추천 AI 챗봇 구현

한국 사람들에게 있어 가장 중요한 문제 중 하나인 “무엇을 먹을지”에 대한 문제를 해결하기 위해, 실제 맛집을 추천하고자 맛집 추천 챗봇을 구상하였습니다.

2. 주요 기능 및 특징

- 데이터 수집 및 전처리

AI 허브에서 제공하는 JSON 형식의 서울시 맛집 데이터를 수집하였으며, 스타벅스, 이디야 커피, 탐앤탐스 등 체인점과 영어 및 특수문자가 포함된 상호명은 삭제하여 고유의 맛집만을 사용할 수 있도록 전처리하였습니다. 이후 랜덤으로 2000개의 상호를 추출하여 맛집으로 가정한 후 추천 시스템의 기본 데이터로 활용했습니다.

- **Seq2Seq** 모델링

TensorFlow와 PyTorch를 이용하여 Attention 매커니즘이 추가된 모델을 각각 구현하여 성능을 확인하였습니다.

- **Transformer** 모델링

자연어 모델 제작에 특화된 PyTorch를 이용하여 SentencePiece와 KoBERT를 활용한 모델을 각각 구현하고 성능을 확인하였습니다.

- **GPT-2 Fine-Tuning**

Fine-Tuning 기술을 활용하여 챗봇을 만들고, 직접 구현한 Transformer 모델과 성능을 비교하고자 했습니다.

- 챗봇 배포

Telegram을 활용해 챗봇을 배포하고, 맛집 추천 서비스의 성능을 확인하였습니다.

3. 대상 사용자

- 맛집을 원하는 누구나

서울시에 거주하거나 서울로 여행을 와 맛집을 찾는 모든 사람

4. 기술 스택

- 개발환경

GitHub, Jupyter Notebook, Visual Studio Code, Google Colab

- 프로그래밍 언어 및 라이브러리

Python, PyTorch, Tensorflow, SentencePiece, KoBERT, GPT-2

- 모델링 및 알고리즘

Deep Learning, Seq2Seq, Attention Mechanism, Transformer, Fine-Tuning

기획 의도(동기)

한국 사람들에게 있어 가장 중요한 문제 중 하나가 먹는 것입니다. 흔히 식도락 여행이라고 어디를 갈 때에도 한국인들은 어떤 음식을 맛있게 먹을 수 있는지, 금강산도 식후경이라는 속담처럼 한국인들은 밥을, 밥심을 중요하게 여깁니다. 오늘 어떤 음식을 먹어야지를 항상 생각하는 우리들에게 진짜 맛집을 소개해주는 챗봇이 있으면 좋을 것 같아 프로젝트를 기획하게 되었습니다.

목표 및 설계

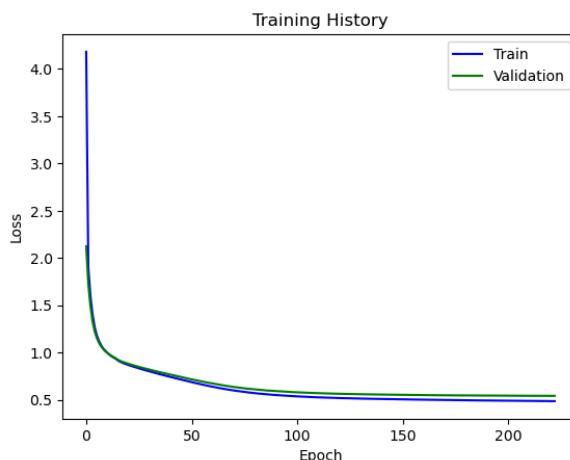
1) 데이터 수집 및 전처리

- AI Hub에서 데이터를 가지고와서 전처리를 진행 :
 - ㄱ. 스타벅스, 이디야 커피, 탐앤탐스 등 체인점과 영어 및 특수문자가 포함된 상호명은 삭제
 - ㄴ. 단순 정보를 문장형으로 만들기 위해 조사를 추가
 - ㄷ. 같은 상호명이나 내용이 2개이상인 경우 단일화
 - ㄹ. 상호명과 주소 정보를 결합하여 랜덤 질문 작성

2) 모델링

1. Seq2Seq 모델 구현

- Tensorflow를 이용하여 Seq2Seq, Seq2Seq with Attention 모델 구현
 - ㄱ. 폴더 내에 있는 JSON 파일을 읽어와서 리스트에 넣은 후 데이터프레임으로 변환
 - ㄴ. 정규화 후 Seq2Seq 모델링 후 평가 실시



Input sentence : 이 곳에 인접한 시설이 있나요?

Correct sentence : 아차산역 3번 출구

Machine translated : 흑석역 4번 출구

Input sentence : 이 곳의 주요 메뉴는 무엇이 있나요?

Correct sentence : 딸기잼, 카스테라

Machine translated : 아메리카노, 카페라떼

Input sentence : 이 곳에 인접한 시설이 있나요?

Correct sentence : 지하철 뚝섬역 8번 출구

Machine translated : 흑석역 4번 출구

Input sentence : 영업시간은 몇시부터 몇시까지로 기재되어 있나요?

Correct sentence : 10:00/11:00~22:00

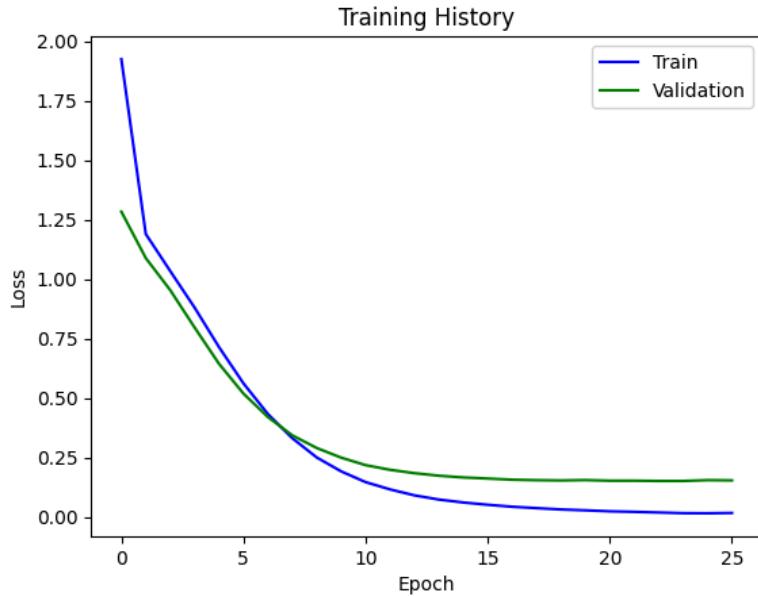
Machine translated : 알수없음

Input sentence : 휴무일은 어떻게 되나요?

Correct sentence : 매주 월요일

Machine translated : 없음

ㄷ. Seq2Seq with Attention 모델링 후 평가 실시 (Adam optimizer 사용)



Input sentence : 향기고을의 대표적인 메뉴를 알 수 있나요?
 Correct sentence : 향기고을의 메뉴에는 청국장, 제육떡볶이2인이 있습니다
 Machine translated : 향기고을의 메뉴에는 청국장, 제육떡볶이2인이 있습니다

Input sentence : 선식당의 주소는 어떻게 되나요?
 Correct sentence : 선식당의 주소는 서울 영등포구 선유로 33 문래대림아파트101동 1층상가입니다
 Machine translated : 선식당의 주소는 서울 영등포구 선유로 33 문래대림아파트101동 1층상가입니다

Input sentence : 놀부부대찌개 불광역점에 인접한 시설을 알 수 있나요?
 Correct sentence : 놀부부대찌개 불광역점에 인접한 시설에는 불광역 8번출구, 불광역 9번출구, 강너머남촌 솟불닭갈비, 디저트39 불광점가 있습니다
 Machine translated : 놀부부대찌개 불광역점에 인접한 시설에는 불광역 8번출구, 불광역 9번출구, 강너머남촌 솟불닭갈비, 디저트39 불광점가 있습니다

Input sentence : 오니기리와이규동 군자역점의 영업시간은 어떻게 되나요?
 Correct sentence : 오니기리와이규동 군자역점의 영업시간은 09:30/10:00-21:30입니다
 Machine translated : 오니기리와이규동 군자역점의 영업시간은 09:30/10:00-21:30입니다

Input sentence : 크로네베이커리의 주소를 알 수 있나요?
 Correct sentence : 크로네베이커리의 주소는 서울 동대문구 경희대로 17입니다
 Machine translated : 크로네베이커리의 주소는 서울 동대문구 경희대로 17입니다

- PyTorch를 이용하여 음절단위 분절화 후 Seq2Seq with Attention 모델링 후 평가 실시 (Adam optimizer 사용)

질문: 연희 에스프레소 바의 주요 메뉴는 무엇이 있나요?
 실제 답변: 연희 에스프레소 바의 메뉴에는 연희 에스프레소, 스트라파짜토이 있습니다
 예측 답변: 가락 잠실 플래그점의 메뉴에는 알기 그린거, 하나베이 있습니다

질문: 카츠젠 잠실나루점의 연락처가 어떻게 되나요?
 실제 답변: 카츠젠 잠실나루점의 연락처는 02-415-9929입니다
 예측 답변: 커피 신령점의 연락처는 0507-1315-8292입니다

질문: 생어거스틴 김포공항점의 주소는 어떻게 되나요?
 실제 답변: 생어거스틴 김포공항점의 주소는 서울 강서구 하늘길 38입니다
 예측 답변: 촌서명림 불광역점의 주소는 서울 금천구 시흥대로 253 1층 오스시스 1층입니다

질문: 노룬산분식의 주차시설이 있나요?
 실제 답변: 노룬산분식의 주차시설이 있습니다
 예측 답변: 로우켓는의 주차시설이 있습니다

질문: 짚신매운갈비찜 시흥사거리점의 영업시간은 몇시부터 몇시까지로 기재되어 있나요?
 실제 답변: 짚신매운갈비찜 시흥사거리점의 영업시간은 11:30 - 24:00입니다
 예측 답변: 신세기 영등포구청 시청점의 영업시간은 11:00 - 22:00입니다

2. PyTorch를 이용하여 Transformer 모델 구현

- SentencePiece 분절화와 KoBERT 분절화를 거쳐 모델링 후 평가

Sample 1:	Input: 까르페디엠의 위치를 알 수 있나요? Predicted: 강남구의 맛집으로는 웨커피 신사가 알려져 있습니다 Target: 까르페디엠의 위치는 서울 용산구 한강대로38길 11-8 1층입니다	Sample 6:	Input: 성동구 맛집 추천 부탁해 Predicted: 성동구의 맛집으로는 에를피가 있습니다 Target: 성동구의 맛집으로는 설수피글렛을 추천드립니다
Sample 2:	Input: 노사천마당의 영업시간은 어떻게 되나요? Predicted: 거나의 맛집으로는 미사리밀 및 초계국수 가산하이시티점을 추천드립니다 Target: 노사천마당의 영업시간은 10:00~22:00입니다	Sample 7:	Input: 목탄장 여의도점의 주차시설이 있는지 알 수 있나요? Predicted: 목탄장 여의도점의 메뉴에는 구운 포카치아, 과일 단새우가 있습니다 Target: 목탄장 여의도점의 주차시설은 있습니다
Sample 3:	Input: 스페이스 라사의 휴무일은 어떻게 되나요? Predicted: 수술호를 직화꼬치비빔류의 맛집으로는 말하브레드 금천구청점이 있습니다 Target: 스페이스 라사의 휴무일은 매주 월요일입니다	Sample 8:	Input: 바르다김선생 달산역점에 인접한 시설이 있나요? Predicted: 바르다김선생 달산역점에 인접한 시설에는 당산역 12번 출구, 메가커피, 맘스터치가 있습니다 Target: 바르다김선생 달산역점에 인접한 시설에는 당산역 12번 출구, 메가커피, 맘스터치가 있습니다
Sample 4:	Input: 브롤로에 인접한 시설을 알 수 있나요? Predicted: 강남구의 맛집으로는 웨커피 신사가 알려져 있습니다 Target: 브롤로에 인접한 시설에는 둘레길도식후경, 세븐일레븐이 있습니다	Sample 9:	Input: 을아래의 대표적인 메뉴를 알 수 있나요? Predicted: 을아래의 메뉴에는 매운갈비찜, 굽중갈비찜이 있습니다 Target: 을아래의 메뉴에는 매운갈비찜, 궁중갈비찜이 있습니다
Sample 5:	Input: 감성커피 천호점의 휴무일은 어떻게 되나요? Predicted: 강남구의 맛집으로는 웨커피 신사가 알려져 있습니다 Target: 감성커피 천호점의 휴무일은 없습니다	Sample 10:	Input: 덮세권의 대표적인 메뉴는 무엇이 있나요? Predicted: 덮세권의 메뉴에는 부대동, 규동가 있습니다 Target: 덮세권의 메뉴에는 부대동, 규동가 있습니다 Evaluation complete!

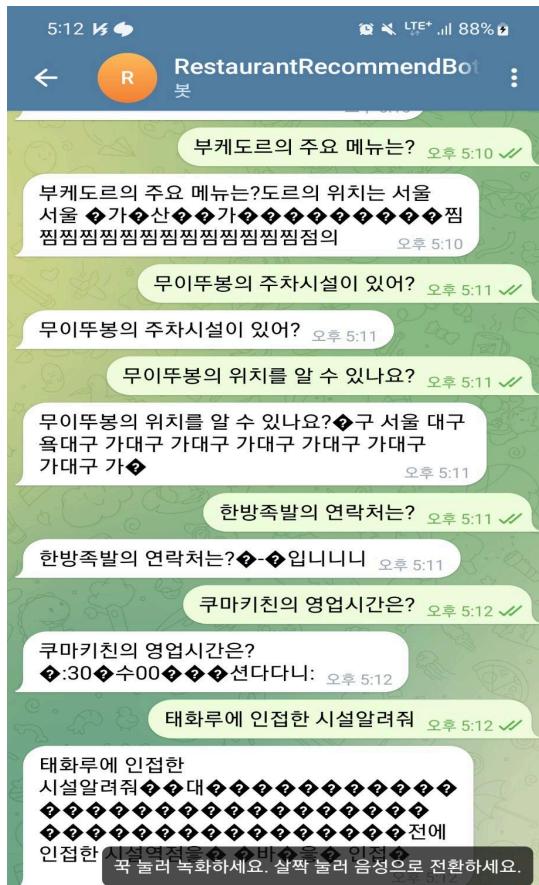
sentencepiece

kobert

- PyTorch를 이용하여 GPT2를 Fine-Tuning한 모델 구현 (Adam optimizer 사용)

3) 챗봇 구현 및 배포

- 텔레그램 챗봇 구현(GPT2 Fine-Tuning 모델)



메시지



- 텔레그램 챗봇 구현(KoBERT 분절화를 거친 Transformer 모델)



성과

<기술적 후기>

1. 다양한 모델 구현

TensorFlow와 PyTorch를 다루어 Seq2Seq, Seq2Seq with Attention, Transformer, 그리고 GPT-2 Fine-Tuning까지 다양한 모델을 성공적으로 구현하고 평가를 해보았습니다.

2. 최적화 및 성능 튜닝

Adam optimizer를 통해 모델의 성능을 극대화하고 성능을 비교해보았습니다.

3. 배포 및 구현

최초 데이터를 수집할 때, AI Hub에서 서울뿐만 아니라 인천, 경기권의 음식점 데이터들도 함께 다운로드 받았으나 프로젝트에 주어진 시간이 부족하여 모두 구현하지는 못했던 점이 아쉬웠습니다. 그렇지만, 서울에 한정하여 텔레그램을 통해 실질적으로 챗봇을 구현하고 배포하여 실용적으로 기술이 사용될 수 있음을 확인했습니다.

<의사소통 후기>

이번 프로젝트에서는 딥러닝의 핵심기술을 팀원들이 익히고자 하였습니다. 이전 머신러닝 프로젝트를 함께 했었기에 기존의 팀워크를 더욱 발휘하여 분업 후 프로젝트를 진행하였습니다. 2명의 팀원이 Seq2Seq 모델링을 하는 동안 다른 2명의 팀원들이 전처리에 시간을 투자하였고, 그 이후 각 팀원이 담당한 역할과 책임을 명확히 하여 Transformer 모델링과 Fine-Tuning 및 앱 배포까지 각자 맡은 부분을 성공적으로 완수하여, 프로젝트 중 팀원간의 협력이 잘 이루어졌습니다. 또한, 프로젝트 진행 중 발생한 기술적 문제들에 대해서 적극적으로 팀원간 의사소통을 통해 함께 해결해내고자 하였으며, 중간중간 피드백을 통해 프로젝트의 방향을 수정하고 개선할 수 있었습니다.