

산재신청 올인원 서비스, 산재메이트(San-Jae Mate) 아이디어 제안서

푸양트리오(김희영, 박서윤, 유지원)

목차

1. 아이디어 구상 및 제안 배경
2. 고용노동 데이터 활용 방안(활용성)
3. 아이디어 상세 설명(실용성)
4. 아이디어 독창성(창의성)
5. 아이디어 기대효과(효과성)
6. 참고문헌

1. 아이디어 구상 및 제안 배경

산재보험은 4대보험 중 하나로 산업재해보상보험으로 지칭되며, 근로자가 업무 중에 발생한 부상, 질병, 장애 또는 사망 등으로부터 보호받기 위해 마련된 사회보험 제도입니다. 산재보험은 업무상 재해를 입은 개인이 근로복지공단에 신청하여 사업주의 동의 여부와 상관없이 혜택을 받을 수 있는 권리이며, 근로자의 과실 여부와 관계없이 근로자가 재해에 따른 보상을 받을 수 있는 무과실책임주의를 원칙으로 합니다. 그러나 김수현 외(2020)에 따르면 기존 산재보험의 처리 방식은 절차가 복잡하여 시간이 오래 걸린다는 행정적 부분과 재활 서비스에 대한 정보 부재로 인해 제대로 된 치료와 재활을 받지 못한다는 서비스 접근성 취약에 대한 문제점이 공통적으로 나타났습니다.

첫째, 신청 단계에서는 산재 은폐와 미인식에 따른 산재보험 이용률이 매우 낮았으며, 노동자 당사자만 신청이 가능하다는 점에서 치료 중에 본인이 직접 신청해야 한다는 불편이 존재한다는 것이었습니다. 특히 심사청구부터 산재 승인까지의 처리 기간이 길고 과정이 복잡하다는 의견과 담당자의 실수나 미숙함으로 행정처리에 문제가 생긴다는 것이었습니다. 나아가 산재 승인 이후에도 치료 이외에 보상과 관련된 행정처리 진행 과정을 신경 써야 한다는 부담감도 확인할 수 있었습니다.

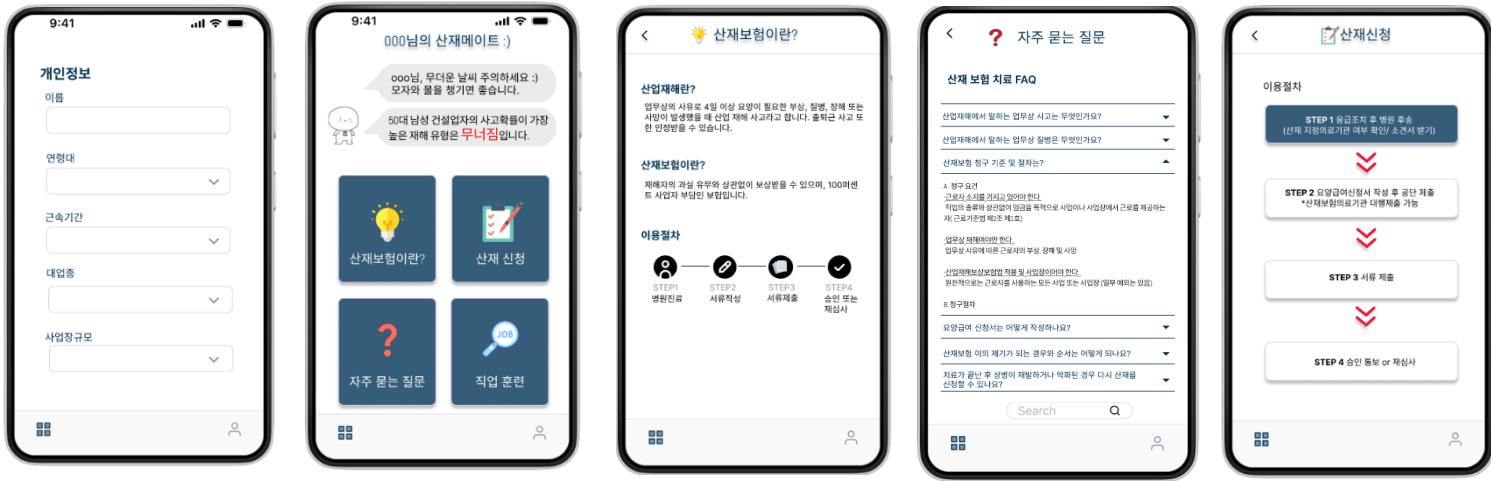
둘째, 치료 단계에서는 정보 제약으로 인해 치료 및 재활 서비스 이용의 한계가 발생하거나, 근로복지공단 소속 병원 및 의료재활 서비스에 대한 정보가 원활히 제공되지 않아 필요한 서비스를 제때 이용하지 못한다는 것을 확인할 수 있었습니다.

따라서 저희는 위와 같은 한계점을 보완하기 위하여 산재 초기부터 요양까지의 각 단계별로 필요한 서비스에 대한 안내와 근로자의 상황에 맞춘 맞춤형 정보를 제공하는 플랫폼 서비스를 고안하게 되었습니다. 이를 통해 복잡한 절차와 긴 처리 기간을 줄여 신속하고 효율적인 산재 처리를 앱을 통해 간편하게 할 수 있도록 하며, 산재 초기부터 요양 및 재활 과정까지 필요한 모든 정보를 제공하여 제대로 된 치료와 재활을 받을 수 있도록 지원합니다.

2. 고용 노동 데이터 활용 방안(활용성)

제공기관명	데이터명	활용 방식
근로복지공단	산재보험 지정의료기관 현황정보	<ul style="list-style-type: none">- 산재 지정의료기관의 주소를 위도와 경도로 추출하여 지정의료기관 위치를 지도 시각화에 활용함.- 해당 데이터를 생성형 AI를 활용한 병원 리뷰 요약을 제시하는데 활용함으로써 산재 근로자들이 원하는 의료기관을 더욱 찾기 쉽게 함.
근로복지공단	산재보험 직업재활기관 현황정보	<ul style="list-style-type: none">- 산재 직업재활기관의 주소를 위도와 경도로 추출하여 지정의료기관 위치를 지도 시각화에 활용함.
고용노동부	사고재해자 개별자료	<ul style="list-style-type: none">- 성별, 연령대, 근속기간, 대업종 등의 데이터를 사용하여 발생 가능한 사고 재해를 예측하는데 활용함

3. 아이디어 상세 설명(실용성)

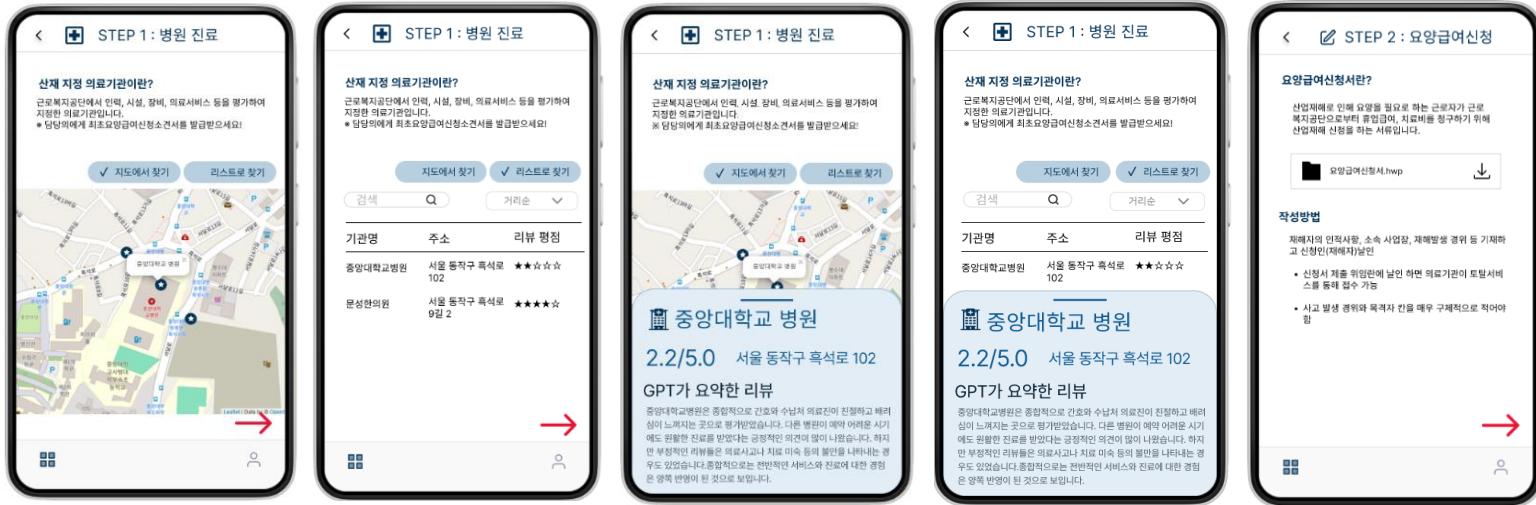


① 산재 예방을 위한 정보제공

- 매일 그날의 날씨를 바탕으로 주의할 점을 알려드립니다.
- 입력한 개인정보(연령대, 근속 기관, 대업종 등)를 바탕으로 발생확률이 높은 재해 유형을 예측하여 현재 사용자의 상황에서 발생확률이 가장 높은 재해 유형을 알려드립니다.

② 산재보험에 대한 정보제공

- ‘산재보험이란?’ 탭에서 산업재해, 산재보험, 이용 절차를 설명함으로써 산재보험에 대한 개념들을 이해할 수 있도록 합니다.
- ‘자주 묻는 질문’ 탭을 통해 산재 처리 과정에서 발생하는 궁금증을 해소할 수 있습니다.

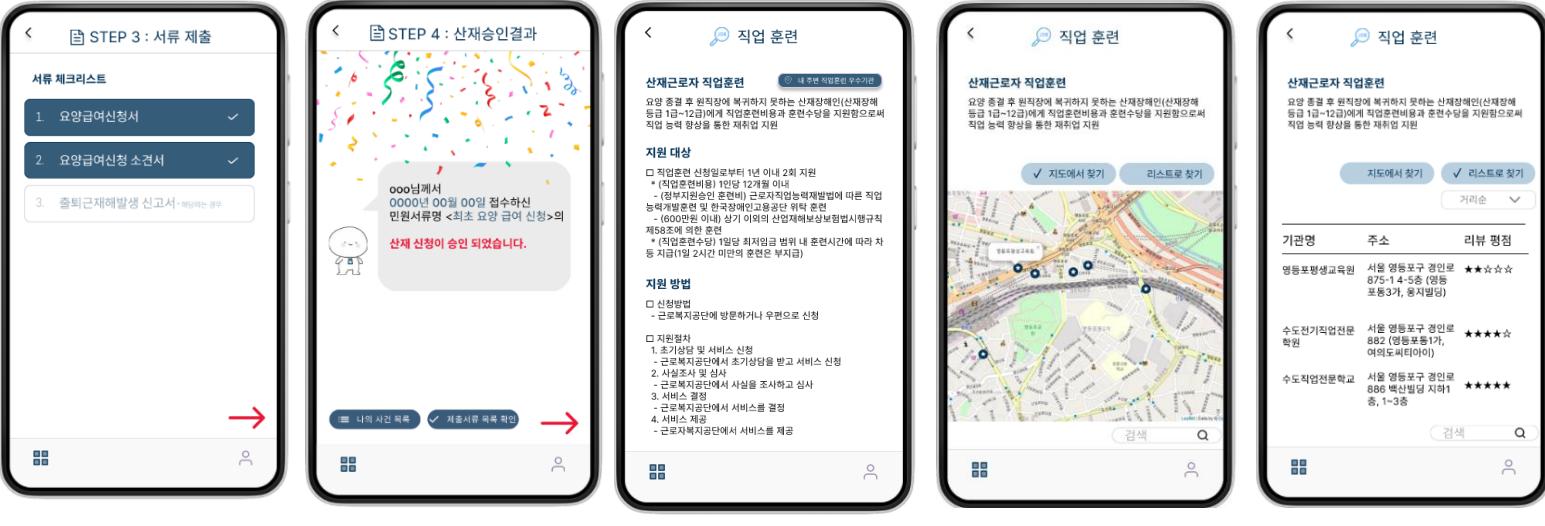


③ 산재신청에 실질적 도움 제공

- 산재 신청 과정을 4단계에 각 단계별로 가이드라인을 제공합니다.
- STEP1: 병원 진료 단계에서는 산재 근로자가 리스트나 지도 형식으로 근처의 산재 지정 의료 기관을 쉽게 찾을 수 있도록 합니다. 또한, 어떤 의료기관을 선택할지에 대한 도움을 주고자 실시간으로 최신 리뷰를 요약하여 제공함으로써 생생하면서도 쉽게 파악하기 용이하도록 합니

다.

- STEP2: 필수 서류인 요양급여신청서를 작성할 수 있도록 어플 내에서 파일을 다운로드할 수 있으며, 작성 방법을 참고하여 작성할 수 있습니다.
- STEP3: 필요한 서류들 중에 빠진 것이 없는지 확인할 수 있습니다.
- STEP4: 산재 승인 결과를 확인할 수 있으며, 산재가 승인된 경우 이후 직업훈련 및 재활기관에 대한 정보를 확인할 수 있는 페이지로 연결합니다. 산재 승인이 반려된 경우, 제출 서류 목록을 확인하여 산재를 재신청할 수 있도록 합니다.



④ 직업훈련에 대한 정보제공

재활과 연관된 직업 훈련 기관 및 프로그램에 대한 정보를 제공하여 근로자들이 필요한 시기에 적절한 서비스를 받을 수 있도록 합니다.

4. 아이디어 독창성(창의성)

저희가 고안한 아이디어에 차별성을 두기 위해서 산재 관련 기관 데이터를 지도에 시각화 하여 서비스를 제공하며, 생성형 AI를 통한 산재 기관 리뷰 요약 제시하여 빠른 기관 선택에 도움을 제공하고, 발생확률이 높은 재해유형을 예측하는 알고리즘을 도입하여 산재 예방에 도움을 주도록 하였습니다.

① 지도 시각화

Geocode

In [1]: `import pandas as pd
import numpy as np
df=pd.read_csv('워크복지공단_산재보험_지정의료기관_현황정보.csv')`

Out[1]:

연도	의료기관번호	의료기관명	주소	Latitude	Longitude	전화번호
1	2022	1000001 연세대학교의과대학강남세브란스병원	서울 강남구 연세로 211 (도곡동)	37.492549	127.046528	02-2019-2163
2	2022	1000001 연세대학교의과대학강남세브란스병원	서울 강남구 연세로 211 (도곡동)	37.492549	127.046528	02-2019-2163
3	2022	1000001 연세대학교의과대학강남세브란스병원	서울 강남구 연세로 211 (도곡동)	37.492549	127.046528	02-2019-2163
4	2022	1000002 대전부설의학의원	대전 서구 남부순환로 53 (대동)	37.493660	127.036600	042-0225-0122
5	2022	1000016 가톨릭영성의원	6108 서울 강남구 흥이로 601-3 (신사동, 퍼스트빌딩)	37.508818	127.037772	02-594-7575
6	2022	1000025 단양영성의원	37.5221	127.034434	02-3444-7277	
7	2022	1000047 경희대학교 의과대학	0170 서울 강남구 도산대로 303 (신사동)	37.514256	127.031135	02-2222-4853
8	2022	1000050 경희대학교 의과대학	0170 서울 강남구 도산대로 303 (신사동)	37.514256	127.031135	02-2222-4853
9	2022	1000056 강원도보건환경	3250 서울 강남구 강남로 160 (강남동, 한양대학교부속병원)	37.536124	127.130442	02-2224-4856
10	2022	1000081 국립경찰병원	5715 서울 강남구 송파로 123 (가락동)	37.490240	127.123482	02-3430-1169
11	2022	1000086 경기도대학교병원	5000 서울특별시 강남구 테헤란로 120-1 (마장동, 경국대학교병원)	37.540496	127.079454	02-2030-7109
12	2022	1000072 만국보건복지부	5200 서울 강남구 테헤란로 120-1 (마장동, 경국대학교병원)	37.540496	127.079454	02-2225-1206
13	2022	1000077 노동부보건복지부	5200 서울 강남구 테헤란로 120-1 (마장동)	37.533084	127.084542	02-2225-1206
14	2022	1000077 노동부보건복지부	4914 서울 강남구 강남대로 32 (송파동)	37.502938	127.078605	02-894-2111
15	2022	1000026 연세대학교의과대학	5054 서울 강남구 고려로 30 (마사동)	37.554530	127.134072	02-441-2111
16	2022	1000036 이대목동병원	5071 서울 강남구 구반로 516 (방화동)	37.549055	127.150615	02-481-5404
17	2022	1000037 박남보건복지부	5033 서울 강남구 테헤란로 267-1 (마장동)	37.549789	127.078734	02-484-5754
18	2022	1000038 경기대학교병원	5017 서울 강남구 테헤란로 120-1 (마장동)	37.550432	127.079454	02-2225-1206
19	2022	1000043 박진의료재단	5408 서울 강남구 고려대로 1210 (방화동)	37.530403	127.089502	02-481-7700
20	2022	1000059 경기경찰병원	4714 서울 강남구 테헤란로 120-1 (마장동)	37.558616	127.033250	02-2211-3875
21	2022	1000079 금수정의료재단	4728 서울특별시 강남구 흥로 147 (금수정2기) 2층	37.557116	127.019479	02-2291-7674
22	2022	1000081 송파구보건환경	5069 서울 강남구 오감로 10 (마곡동)	37.497155	127.034466	02-400-2286
23	2022	1000085 청사진보건환경	5003 서울 강남구 흥로 2 (마곡동)	37.553045	127.080539	02-405-1177

Geocode

In [1]: `import pandas as pd
import numpy as np
df=pd.read_csv('워크복지공단_산재보험_지정의료기관_현황정보.csv')`

Out[1]:

연도	의료기관번호	의료기관명	주소	Latitude	Longitude	전화번호
0	2022	1000001 연세대학교의과대학강남세브란스병원	서울 강남구 연주로 211 (도곡동)	37.492549	127.046528	02-2019-2163
1	2022	1000050 의료법인 계임병원	6664 서울 서초구 서초대로 116 (방배동)	37.487829	126.995077	02-596-5557
2	2022	1000072 예후암병원	6737 서울 서초구 낙부순환로 350길 33 (망재동)	37.483457	127.036467	02-575-0122-
3	2022	1000116 가톨릭평화외교의원	6109 서울 강남구 연주로 601-3 (신사동, 퍼스트빌딩)	37.508618	127.052211	02-554-7575
4	2022	1000125 안양영성의원	37.5221	127.034434	02-3444-7277	
5	2022	100047 경희대학교 의과대학	37.514256	127.031135	02-2222-4853	
6	2022	100050 경희대학교 의과대학	37.514256	127.031135	02-2222-4853	
7	2022	100056 강원도보건환경	37.536124	127.130442	02-2224-4856	
8	2022	100059 강원도보건환경	37.536124	127.130442	02-2224-4856	
9	2022	100068 경기경찰병원	37.490240	127.123482	02-3430-1169	
10	2022	100068 경기도대학교병원	37.540496	127.079454	02-2030-7109	
11	2022	100072 만국보건복지부	37.540496	127.079454	02-2225-1206	
12	2022	100077 노동부보건복지부	37.533084	127.084542	02-2225-1206	
13	2022	100077 노동부보건복지부	37.502938	127.078605	02-894-2111	
14	2022	100077 노동부보건복지부	37.554530	127.134072	02-441-2111	
15	2022	100026 연세대학교의과대학	37.549055	127.150615	02-481-5404	
16	2022	100036 이대목동병원	37.549789	127.078734	02-484-5754	
17	2022	100037 박남보건복지부	37.550432	127.079454	02-2225-1206	
18	2022	100038 경기대학교병원	37.497155	127.033250	02-2211-3875	
19	2022	100043 박진의료재단	37.530403	127.089502	02-481-7700	
20	2022	100059 경기경찰병원	37.558616	127.033250	02-2211-3875	
21	2022	100079 금수정의료재단	37.557116	127.019479	02-2291-7674	
22	2022	100081 송파구보건환경	37.497155	127.034466	02-400-2286	
23	2022	100085 청사진보건환경	37.553045	127.080539	02-405-1177	

6059 rows x 8 columns

- ‘산재보험 지정의료기관 현황 정보’의 ‘주소’ 데이터를 구글의 확장 프로그램인 geopy를 통해 위도와 경도로 추출한 후, folium API를 활용하여 구글 지도 API에 지정의료기관들을 시각화할 수 있도록 코드를 작성하였습니다.



- 시각화된 지정의료기관을 HTML 파일로 저장하였습니다. 사용자들은 어플리케이션을 통해 활용된 지도 API를 통해 인근에 위치한 산재 지정 의료기관들을 손쉽게 확인하여 재활 병원의 접근성을 용이하게 할 수 있습니다. ‘산재보험 직업재활기관 현황정보’ 데이터도 같은 방법으로 지도 시각화를 시행하였습니다.

② 산재 지정의료기관의 최신 리뷰 요약

```

import time
import pandas as pd
import warnings
warnings.filterwarnings('ignore')

from selenium import webdriver # 동적크롤링
from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.common.keys import Keys # Enter
from selenium.webdriver.common.by import By
from bs4 import BeautifulSoup
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.chrome.options import Options
from selenium.common.exceptions import NoSuchElementException
    
```

연도	의료기관번호	의료기관명	우편번호	주소	전화번호
0	2022	1000001	연세대학교의과대학강남세브란스병원	6273	서울 강남구 연주로 211 (도곡동) 02-2019-2163
1	2022	1000050	의료법인 제일병원	6664	서울 서초구 서초대로 116 (방배동) 02-586-5557
2	2022	1000072	예루살렘정형외과	6737	서울 서초구 낭부순환로350길 33 (양재동) 02-575-0122-0
3	2022	1000116	가톨릭청평외과의원	6109	서울 강남구 연주로 601 3층 (논현동, 파크랜드빌딩) 02-554-7575
4	2022	1000125	안정형외과	6021	서울 강남구 도산대로 303 (신사동) 02-3444-7277
5	2022	1000147	광동한방병원	6170	서울 강남구 봉은사로 612 (삼성동) 02-2222-4883
6	2022	1000158	한암대학교병원	4763	서울 성동구 황십리로 222-1 (사근동, 한암대학부속병원) 02-2290-9096
7	2022	1000159	강동성심병원	5355	서울 강동구 성안로 150 (길동) 02-2224-2127
8	2022	1000161	국립경찰병원	5715	서울 송파구 송이로 123 (가락동) 02-3400-1160
9	2022	1000168	건국대학교병원	5030	서울특별시 광진구 능동로 120-1 (화양동, 건국대학교병원) 02-2030-7160

- 동적 크롤링을 수행하기 위해 오픈 소스 프레임워크인 Selenium을 활용하였고, 웹 드라이버는 ChromeDriver를 사용하였습니다. 근로복지공단에서 제공한 ‘산재보험 지정의료기관 현황정보’ CSV 파일을 Pandas 라이브러리를 통해 데이터프레임으로 불러와 df 변수에 저장하였습니다.

```
: df['의료기관명'][1304]
```

```
: '중앙대학교 병원'
```

```
: hospital_idx = int(input())
```

```
1304
```

- 데이터 프레임의 1,304번째 데이터의 ‘의료기관명’ 칼럼은 ‘중앙대학교 병원’임을 확인할 수 있습니다. 정보를 열람하고자 하는 의료기관의 인덱스를 유저에게 입력받아 hospital_idx 변수에 저장하도록 하였습니다.

```
# 카카오 맵으로 이동
url = "https://map.kakao.com/"
chrome_driver_path = 'C:/Users/kellykim/Downloads/chromedriver-win64/chromedriver.exe'
service = Service(chrome_driver_path)
options = Options()
options.add_argument('--headless') # 브라우저를 숨김 모드로 실행
options.add_argument('--no-sandbox')
options.add_argument('--disable-dev-shm-usage')
driver = webdriver.Chrome(service=service, options=options)
driver.get(url)
```

```
: searchloc = df['의료기관명'][hospital_idx] + ' ' + '.join(list(df['주소'][hospital_idx].split())[:2])

search_area = driver.find_element(By.XPATH, r'//*[@id="search.keyword.query"]') # 카카오맵 검색창
search_area.send_keys(searchloc) # 검색어 설정
driver.find_element(By.XPATH, r'//*[@id="search.keyword.submit"]').send_keys(Keys.ENTER) # 둘보기 클릭

time.sleep(5)

driver.find_element(By.XPATH, r'//*[@id="info.main.options"]/li[2]/a').send_keys(Keys.ENTER) # 장소 템

registered = 1 # 카카오맵에 의료기관 등록됨
review_exist = 1 # 후기 개수 0개 or 후기 미제공
```

- 크롬 드라이버를 활용하여 카카오맵(“<https://map.kakao.com/>”)으로 이동합니다.
 - 카카오맵의 검색창에 의료기관명과 주소의 앞부분을 입력하도록 하였습니다.
 - 카카오맵에는 a. 카카오맵에 등록되지 않은 의료기관인 경우, b. 등록되어 있지만 후기를 제공하지 않는 의료기관인 경우, c. 등록되어 있으면서 후기를 제공하지만 아직 후기가 없는 경우로 나뉩니다. 이 경우들을 구분하기 위하여 registered, review_exist 변수를 두었습니다.

```
# 의료기관 등록
try:
    name = room.select('.head_item > .tit_name > .link_name')[0].text
except AttributeError:
    name = searchloc
registered = 0 # 카카오맵에 등록되지 않은 의료기관일 경우
review_exist = 0

name

# 품질기준 별점
if registered:
    addr = room.select('.addr > p')[0].text
else:
    addr = df['주소'][hospital_idx]

addr

# 서울 종로구 흑석로 102

# 카카오기준 별점
if registered:
    score = room.select('.rating > .score > em')[0].text
    review_cnt = room.select('.rating > a > em')[0].text
    if (score == "" or not int(review_cnt)): #평가가 0개인 경우 or 평가 아직 없음
        review_exist = 0
        score = "X"
    else:
        score = "X" #카카오맵에 등록됨으로 의료기관
else:
    score = "X"

score
```

- 의료기관의 이름, 주소, 평점이 각각 추출됩니다. 유저가 앞서 선택한 의료기관이 ‘중앙대학교 병원’이었으므로, 의료기관 이름으로는 ‘중앙대학교병원’, 주소로는 ‘서울 동작구 흑석로 102’, 평점으로는 2.2점이 추출되었습니다. 또한, 해당 과정에서 리뷰 존재 여부가 review_exist 변수에 저장됩니다.

```
# 웹툰 수집 및 ChatGPT API로 풍자문 요약
def extract_review():
    html = driver.page_source
    soup = BeautifulSoup(html, 'html.parser')

    # 초기 설정 초기화
    review_lists = soup.select('.list_evaluation > li')

    count = 0
    rev = []

    if len(review_lists) != 0:
        for review in review_lists:
            comment = review.select('.txt_comment > span')[0].text # B
            if len(comment) != 0:
                rev.append(comment)

    # 드라이버 종료 및 유통 정보
    driver.close()
    driver.switch_to.window(driver.window_handles[0])
    time.sleep(2)

    return rev

if review_exist:
    # 상세페이지로 이동
    driver.find_element(By.XPATH, r'//*[@id="info.search.place.list"]/li['+str(i)+']/div[5]/div[4]/a[1]').send_keys(Keys.ENTER)
    driver.switch_to.window(driver.window_handles[-1])
    time.sleep(2)
    # 웹툰 정보가 최대 5번 표시되는 경우
    for i in range(5):
        try:
            more_rvs = driver.find_element(By.CLASS_NAME, 'txt_more') # 초기 표시된 B
            more_rvs.click()
            time.sleep(1)
        except NoSuchElementException:
            pass

    rev = extract_review() # 초기 수집 및 B
else:
    rev = "N/A"
```

- 의료기관의 리뷰가 존재하면 카카오맵 페이지의 ‘상세보기’를 클릭하고 리뷰를 추출합니다. 여기서 ‘리뷰 더보기’를 최대 5번까지 클릭하도록 하여 너무 적은 수의 최신 리뷰를 활용하지 않도록 하였습니다. 추출된 리뷰는 위와 같았습니다.

```

from openai import OpenAI

# open ai에서 발급받은 api key를 등록합니다.
OPENAI_YOUR_KEY = '#your open ai api key'
# 사용 모델을 설정합니다. chat GPT는 gpt-3.5-turbo를 사용합니다.
MODEL = "gpt-3.5-turbo"
USER_INPUT_MSG = "다음의 하나 이상의 병원 리뷰를 요약해서 종합한 하나의 리뷰를 출력해줘. 리뷰 요약을 할 때 주제는 특정인이 아닌 제3자인 것처럼 해줘. 서술어 스타일은 '


# OpenAI API 키 설정
client = OpenAI(api_key=OPENAI_YOUR_KEY) # 사용자의 API 키로 대체해야 함

if review_exist:
    response = client.chat.completions.create(
        model = MODEL,
        messages = [ {"role" : "user", "content" : USER_INPUT_MSG} ]
    )
    review_summary = response.choices[0].message.content
else:
    review_summary = "X"

print(review_summary)

```

증강대학교병원은 간호생님들과 수납의사까지 모두 친절하여 원활한 진료를 받을 수 있었고, 다른 병원에서는 어려운 예약도 쉽게 받을 수 있어서 감사했습니다. 의료사고로 인해 고통받은 적이 있으나, 전체적으로 의사와 간호사들은 친절하고 진료를 잘 받을 수 있었다는 평이 많았습니다. 하지만 일부 의료사고 및 불친절한 직원들이 걱정되어 다소 신중한 복문을 권고하는 의견도 보입니다.

- 추출한 리뷰를 담은 리스트를 ChatGPT API를 활용하여 요약하도록 하였습니다. 그 결과는 review_summary에 저장되게 됩니다. 만약 리뷰가 없는 경우라면 리뷰 요약의 결과 대신 ‘X’가 출력되도록 하였습니다.

③ 발생확률 높은 재해유형 예측

▶ import pandas as pd

```

df = pd.read_csv("./content/drive/MyDrive/고용노동부_공모전/고용노동부_사고재해자 개별자료_20221231.csv", encoding='euc-kr')
df.head()

```

	연도	성별	연령대	근속기간	대업종	사업장규모	재해자구분	재해유형
0	2022	여	30대	6개월 미만	제조업	50인~99인	사고사망자	끼임
1	2022	남	50대	6개월 미만	건설업	5인~9인	사고사망자	무너짐
2	2022	남	30대	6개월 미만	제조업	10인~19인	사고사망자	폭발파열
3	2022	남	60대 이상	6개월 미만	운수·창고·통신업	30인~49인	사고사망자	사업장외교통사고
4	2022	남	50대	1년~2년	운수·창고·통신업	10인~19인	사고사망자	사업장외교통사고

- 고용노동부에서 제공한 ‘사고재해자 개별자료’ csv 파일을 Pandas 라이브러리를 통해 데이터프레임으로 불러와 df 변수에 저장하였습니다.
- 재해유형을 알 수 없는 행들은 제거한 뒤, 타겟값인 ‘재해유형’ 컬럼만 라벨인코딩 하고 피쳐값들은 원핫인코딩 해줍니다. 이후 로지스틱 회귀, 결정트리, XgBoost 모델의 정확도를 확인해보았습니다. 각 모델의 정확도는 다음과 같았습니다.

```

[] # 원핫인코딩
encoded_feature = pd.get_dummies(df)
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(encoded_feature,target_encoded, test_size = 0.25, stratify=target_encoded)

### 로지스틱 회귀
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
lr.fit(X_train, y_train)

from sklearn.metrics import accuracy_score
y_pred = lr.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print('로지스틱회귀 정확도:', accuracy)

```

```

#### 결정트리
from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier(max_depth=19,min_impurity_decrease=0.0001)
clf.fit(X_train, y_train)
pred = clf.predict(X_test)
accuracy = accuracy_score(y_test, pred)
print('결정트리 정확도:', accuracy)

#### XgBoost
import xgboost as xgb
xgb_classifier = xgb.XGBClassifier(random_state=42)
xgb_classifier.fit(X_train, y_train)

y_pred = xgb_classifier.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print('XgBoost 정확도:', accuracy)

로지스틱회귀 정확도: 0.8513528643403621
결정트리 정확도: 0.9997760776264228
XgBoost 정확도: 1.0

```

- 로지스틱 회귀 정확도: 0.85, 결정 트리 정확도: 0.99, XgBoost 정확도: 1.0
- 정확도가 1.0인 것은 비정상적인 값으로 판단하여 제외하고 정확도가 가장 높았던 결정 트리를 채택하였고, 과적합 여부를 확인하기 위해 훈련 세트에서 검증 세트를 분리하여 훈련 세트와 검증 세트의 점수를 비교해보았습니다.

```

[94] from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(encoded_feature,target_encoded, test_size = 0.2, stratify=target_encoded)

sub_input, val_input, sub_target, val_target = train_test_split(X_train, y_train, test_size=0.2, random_state=42)
print(sub_input.shape, val_input.shape)

(68595, 63) (17149, 63)

[95] from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier(max_depth=15)
clf.fit(sub_input, sub_target)

print(clf.score(sub_input, sub_target))
print(clf.score(val_input, val_target))

0.9944893942707195
0.9995856318152662

```

- 68,595개의 훈련 세트와 17,149개의 검증 세트를 사용해 비교한 결과, 훈련 세트의 점수가 더 높았지만 과적합으로 보기 어려울 정도로 미세한 차이였습니다.

```

✓ [64] 제안된 코드에 라이선스가 적용될 수 있습니다.| 
# 하이퍼파라미터 튜닝
from sklearn.model_selection import GridSearchCV
params = {'min_impurity_decrease': [0.0001, 0.0002, 0.0003, 0.0004, 0.0005]}
gs = GridSearchCV(DecisionTreeClassifier(random_state=42), params, n_jobs=-1)
gs.fit(X_train, y_train)

dt = gs.best_estimator_
print(dt.score(X_train, y_train))
print(gs.best_params_)
print(gs.cv_results_['mean_test_score'])

☒ 0.999377993405731
{'min_impurity_decrease': 0.0001}
[0.999378 0.99980096 0.99980096 0.99955216 0.99955216]

✓ [65] 제안된 코드에 라이선스가 적용될 수 있습니다.| art1111/making-tars
best_index = np.argmax(gs.cv_results_['mean_test_score'])
print(gs.cv_results_['params'][best_index])

☒ {'min_impurity_decrease': 0.0001}

✓ [66] params = {'min_impurity_decrease':np.arange(0.0001, 0.001, 0.0001),
               'max_depth': range(5,20,1),
               'min_samples_split': range(2,100,10)}

```

- 그리드서치(GridSearch)를 사용해 최적의 파라미터 값을 찾았습니다.

```
❶ 제안된 코드에 라이선스가 적용될 수 있습니다.|  
gs = GridSearchCV(DecisionTreeClassifier(random_state=42), params, n_jobs=-1)  
❷ gs.fit(X_train, y_train)
```

```
✓ [69] print(gs.best_params_)  
→ {'max_depth': 19, 'min_impurity_decrease': 0.0001, 'min_samples_split': 2}
```

```
✓ [70] np.max(gs.cv_results_['mean_test_score'])  
→ 0.9998009578901537
```

- 최적의 파라미터는 max_depth가 19일때, min_impurity_decrease가 0.0001입니다. 이 값을 적용했을 때, 점수가 0.998까지 향상되었습니다.

```
❶ # 테스트할 샘플 데이터 추출  
sample_data = encoded_feature.iloc[5,:].to_frame().T  
sample_data  
❷ 연도 성별_남 성별_여 연령대_10대 연령대_20대 연령대_30대 연령대_40대 연령대_50대 연령대_60대 이상 근속기간_10년 이상 ... 재해유형_사업장내교통사고 재해유형_사업장외교통사고 재해유형_경.산소 결핍 재해유형_이상운동불체접촉 재해유형_경단배열재접촉 재해유형_경.체육행사 재해유형_경.폭력행위 재해유형_화재 재해유형_화학물질누출 재해유형_화학물질누출총촉  
5 2022 True False False False False True False False ... False  
1 rows × 63 columns  
❸ [59] # 예측  
y_pred = clf.predict(sample_data)
```

- 앞에서 개발한 모델을 사용해 test 데이터의 5번째 값으로 발생 확률이 가장 높은 재해 유형을 예측해보았습니다. (정답은 ‘무너짐’입니다.)

```
✓ [60] y_pred[0]  
→ 8  
✓ [61] target_encoded[5]  
→ 8  
✓ [62] df_target.iloc[y_pred[0],0]  
→ '무너짐'
```

- 예측 결과, ‘무너짐’으로 제대로 예측했음을 알 수 있습니다.
- 총 107,180개의 행을 가진 데이터를 활용하여 연도, 성별, 연령대, 근속기간, 대업종, 사업장 규모를 바탕으로 개인별 발생 가능성이 높은 재해 유형을 예측해보았습니다. 이를 바탕으로 사용자가 주의해야 할 사항을 미리 인지할 수 있습니다.

5. 아이디어 기대효과(효과성)

김수현 외(2020)에 따르면, 산재 처리에 불편함을 느낀 근로자들은 공통적으로 산재보험 제도의 처리 과정의 복잡성과 치료 및 재활 서비스에 대한 정보 부족으로 인해 서비스 접근성의 한계를 느꼈다고 토로하였습니다. 이에 ‘산재메이트’는 다음과 같은 기대효과를 제공합니다.

첫째, 산재 예방 효과와 진입 장벽을 낮출 수 있습니다. 근로자가 입력한 개인정보를 바탕으로 현재 상황에서 발생할 수 있는 재해 유형을 예측하여 제시함으로써 산재 예방의 효과를 높이고, 산재 서비스 이용에 대한 진입 장벽을 낮추는 데 기여합니다.

둘째, 행정 처리 기간을 단축하고 효율성을 증대시킬 수 있습니다. 근로자들이 플랫폼을 통해 산재 서류를 업로드하여 신청하고, 산재 승인 여부를 확인할 수 있도록 하여 지연과 누락 등의 행정 처리 절차 문제점을 보완합니다. 또한, 각 단계별 필요 서류와 서비스에 대한 정보를 제공함으로써 행정 처리 기간을 단축하고 근로자들의 부담을 줄일 수 있습니다.

셋째, 산재 관련 기관에 대한 접근성을 향상시키고 정보를 제공합니다. 근로복지공단에서 제공한 산재보험 지정 의료기관, 산재 노동자 직업훈련 우수 기관, 산재보험 직업재활 기관 현황 정보를 지도로 제시하여 산재 관련 기관에 대한 접근성을 용이하게 합니다. 특히, 산재보험 지정 의료기관의 AI 리뷰 요약을 제공하여 빠른 병원 선택에 도움을 주며, 병원 및 재활 서비스에 대한 정보 부족으로 인한 장해 발생을 최소화할 수 있습니다.

넷째, 산재보험 제도에 대한 이해도를 증대시키고 사회적 비용을 절감할 수 있습니다. 근로자와 사업주가 산재보험 제도를 명확하게 이해하고 활용할 수 있도록 ‘자주 물어보는 질문’ 탭을 통해 산재 제도에 관한 정보를 빠르게 확인할 수 있게 합니다. 또한, 사업장에서 산재 승인을 거부할 경우에도 근로자가 산재 승인을 신청할 수 있다는 등의 산재보험에 대한 오해와 잘못된 인식을 바로잡아, 고용 안정화와 고용 및 재취업에 대한 사회적 비용 절감에 기여할 수 있습니다.

이와 같이, ‘산재메이트’는 산재 예방에서부터 신청, 승인, 치료 및 재활에 이르는 모든 과정에서 효율적이고 신속한 처리를 지원하며, 근로자와 사업주 모두에게 실질적인 도움을 제공할 것입니다.

6. 참고문헌

- 김수현, 이정은 & 이자호.(2020) 산재근로자의 직업복귀 과정에 관한 연구: 원직장 복귀를 중심으로, *장애와 고용*, 30(1), 121-158
- 법제처. (2024.06.15). 산업재해보상보험 개관. 찾기쉬운 생활법령정보. <https://easylaw.go.kr/CSP/CnpClsMain.laf?csmSeq=570&ccfNo=1&cciNo=1&cnpClsNo=1>
- 한방산재케어. 산재보험치료 FAQ. 한방산재케어. <https://www.sanjaecare.com/faq>