

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

푸앙트리오(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 한계점들을 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 요약물 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)

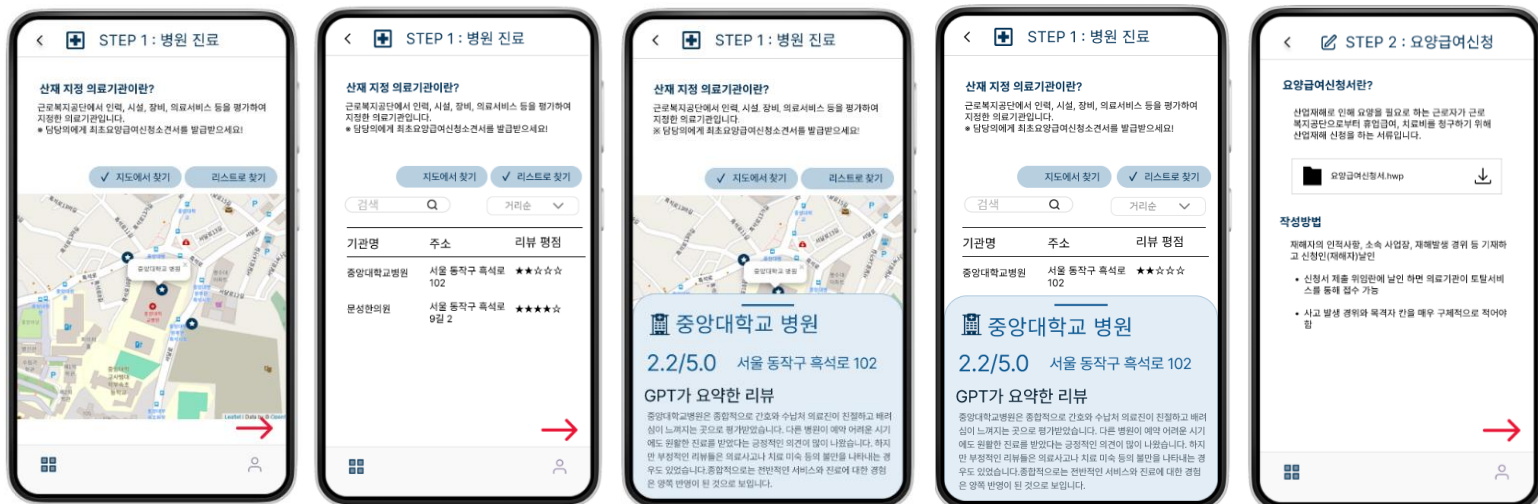


① 산재 예방을 위한 정보제공

- 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.

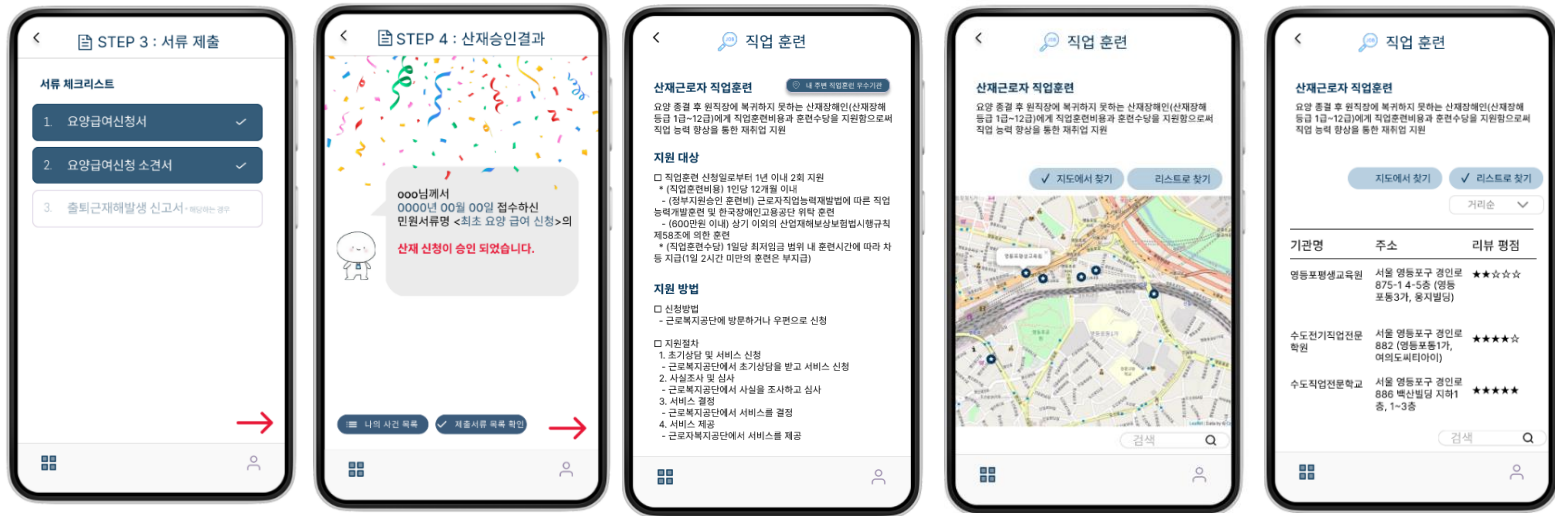


③ 산재신청에 실질적 도움 제공

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

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.



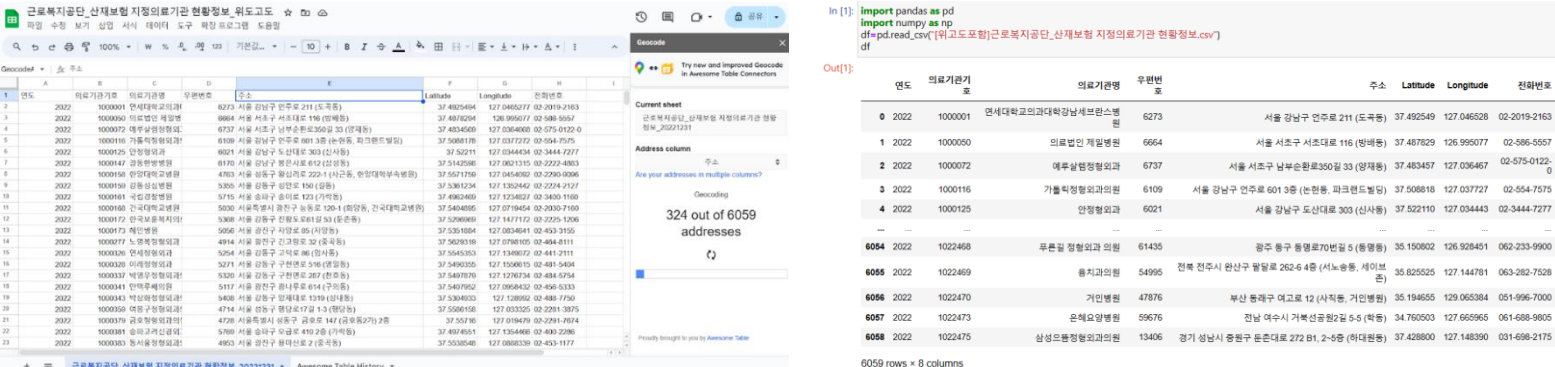
④ 직업훈련에 대한 정보제공

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



- ‘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 재할 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	의료기관명	우편번호	주소	전화번호
0 2022	연세대학교의과대학강남세브란스병원	6273	서울 강남구 언주로 211 (도곡동)	02-2019-2163
1 2022	의료법인 제일병원	6664	서울 서초구 서초대로 116 (방배동)	02-586-5557
2 2022	예루살렘정형외과	6737	서울 서초구 남부순환로350길 33 (양재동)	02-575-0122-0
3 2022	가톨릭정형외과의원	6109	서울 강남구 언주로 601 3층 (논현동, 파크랜드빌딩)	02-554-7575
4 2022	안정형외과	6021	서울 강남구 도산대로 303 (신사동)	02-3444-7277
5 2022	광동한방병원	6170	서울 강남구 봉은사로 612 (상성동)	02-2222-4883
6 2022	한양대학교병원	4763	서울 성동구 왕십리로 222-1 (사근동, 한양대학부속병원)	02-2290-9096
7 2022	강동성심병원	5355	서울 강동구 성안로 150 (길동)	02-2224-2127
8 2022	국립경향병원	5715	서울 송파구 송이로 123 (가락동)	02-3400-1160
9 2022	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 숨길 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 전달
driver.find_element(By.XPATH, r'//*[@id="search.keyword.submit"]').send_keys(Keys.ENTER) # 돋보기 클릭

time.sleep(5)

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

registered = 1 # 카카오맵에 의료기관 미등록
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 정보
try:
    name = room.select('.head_item > .tit_name > .link_name')[0].text
except AttributeError:
    name = searchloc
    registered = 0 # 카카오맵에 등록되지 않은 의료기관임
    review_exist = 0

name

'중앙대학교병원'

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

addr

'서울 동작구 흑석로 102'

# 의료기관 평점
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)): #후기 미제공 or 후기 악의적
        review_exist = 0
        score = "N"
    else:
        score = "X" #카카오맵에 미등록된 의료기관

score

'2.2'
```

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

```
# 리뷰 수집 및 ChatGPT API를 활용한 요약
def extract_review():
    html = driver.page_source
    soup = BeautifulSoup(html, 'html.parser')

    # 후기 목록 찾기
    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 # 리뷰
            if len(comment) != 0:
                rev.append(comment)

    # 다시 검색 버튼을 클릭
    driver.close()
    driver.switch_to.window(driver.window_handles[0])
    time.sleep(2)

    return rev

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

    rev = extract_review() # 리뷰 추출 함수 실행
else:
    rev = "X"
```

```
rev

[
    '1. 일단입니다. 대학 병원 후기를 적게 읽을 줄 몰랐는데 중앙대병원 전체적으로 간호선생님들과 수간처 의사행까지 다 너무 친절함. 간차 팀가 배려심도 기본적으로 가지고 계신게는 매개서기분종게 친절하게 전로 잘 받았어요. 시기가 시기인지라 다른 병원은 예약도 너무 어려웠는데중앙대 병원 관계한 모든 분들을 감사드립니다.(호중기알라프가 내과, 영상의학과, 진단영상)',
    '시립관서서울 의료원과 기사왕고 왔습니다.',
    '아들사고로 머리 치유안되는 몸고성인데 타원에서 영장난관치료비 지출이며 영장난 후유증을 안긴 곳.사소하게는 다른 병에서 언어장애 할까 피부에 하는 항장제 테스트도 마프게 유제 박듯 한술(간호사들 다들 아파라다다... 잘못해서 그런 것도 모르겠음...) 신장 검드라 오듯동안 술에 저혈.아무튼 개인적으로 굿고 작은 의료사고가 있던 병원. 실력도 의술-다소 드문 병원 하나에 대한 지식자체가 이해 없었음(청약은 본인들 후지를 가지고 그런 중상의 병원은 없애드니 환자들 미상한 사람 취급해내뵤. 나중 의료학과 사건만 뒤져봐도 나오던걸). 당연히 후에 타원에서 진단 받고 이 병원은 다시 안감.',
    '평약중앙대과오송출교수님 항상감사드립니다. 저희친정대마님 암병문에불안하고 항상우물머하셔는데혹는말글로 인사해주시고 손잡아주시고 따뜻한말씀한마디에 저희대마님 힘든투병생활동안 기운충만으셨습니다만려하시고 말남알까지과 골과 불마가섯지만 입장실에서 자신감마지막을 적갈하섯지만 오송출교수님게말사민사드려하신다대제게 다감들사키섯습니다 감사한마음충만할것을 알아보니 갑장면법에저속원다하여 선행도드러지듯하고 바쁜진료시간에 인사드려는것은 다른입원우분들의수용관시간을복는것같아. 이곳에 글올립니다오송출교수님고말백중영내과 귀영고해복신(외오영내과나소녀치암살살대해주섯습니다)진료간호사님(진료소통을위해 교수님의의사를 전달해주시고마음분인데. 성함을 잊었습니다.)과 입원실장주 간호사님들 (불마가시기전6월의입원시간동안 많이해해주섯습니다)마라속대여감사드립니다! 앞으로도 일몰지니대병원을올때마다 감사한가억이날것같습니다감사했습니다.',
    'HIV 약명검사 안됨. 관체적으로 정신이 하나도없고 체게가 60대나 보겠소 수준으로 열악함. 대학 시절부터 수준까지 정말 한숨만 나온다. 밑에 수도룩한 의과사고 후기를들 보면 믿을 없었단다라고 할아기도 만만하 수준. 대체서 대학사실대 서장한 중증외사 인지 이해가 안감. 국공체용 으로 하려도 감지적지안감.',
    '이비인후과', 무슨 검사시간이 13시 20분부터 일하는데 적어도 먼저 중 사람들중 중이며 적어할 수 있게 하던가사람들 다 새치가 하려고 기다리는데 일면서 그러네중이 한 강쓰는게 어려운데가 먼저 온 사람은 바로되는 방식이네요.12시부터 점심도 안 먹도 일에서 풀면서 기다리는게 뽀든서1을 접수만저 해어하는건 왜 13시 20분 다 되어서 알려줘서후송위로 말리는지 사람 호구만드시네.',
    '열대 가지마세요 재발 그냥 가만 안됨',
    '자... 입원 xㄴㄴ',
    '여기가 양한장로 시용한 의사 감싸기, 나물과라 혁의 힘 및 거짓말, 피해자부에게 천만원 합의 재시, 남편에 자살시도하게 만든 중양대병원인가요?',
    '병원에서 젊은 교수들로 운영되는것들처럼으로 여기고 임용을 하는지는모르겠으나 실력있고 경험많은 멘토가 없어서인지실력이 현저히 떨어지는 사람들 교수로채용해서 환자를 보게 한다는게 관재에겐 너무도무모한시스템이 아닐까 사실 무섭게 느껴진다.',
    '의류원 직장을 잃도 못하고 내가지 않음.',
    '선생님들도 친절하고 진료 잘 받고 왔습니다.',
    '예약외관 전체 안됨. 사전 예약을 미리하고 와도 상할때마다 1시간이상 대기해야하고 얼마나 더 기다려야하는지 예상해지지만 안해도 얻을.병원 운영 개선 시급.',
    '중증외과전선터 방문했는데 두번 불려보니 불친절하게 차종내과 진료보면서 한숨만 계속 내주고 뒤하는데도 인지 그리고 인턴이라그런가 피검사하고 핏가려는데 왜게 머물든지 관경한다. 여기 있는 약을 먹었다.',
    '화장실을 사용하러 왔더니 외부인 출입금지?...병원 앞에 왔다.',
    '의사선생님들 친절하시고 자비한 사람해주어요.',
    '생사를 다루는 순간이 아니란 절대 가지 양을 병장. 간호사들의 환자 뒷담화는 중대 병원의 문화이다.',
    '안과 김지택 교수님 항상 감사했습니다. 덕분에 눈 건강 유지하면서 살고 있습니다.',
    '여기 댓글의 별 한개 (그리고도 선택의 여지가 없어 아플 수 없어 한 개) 존 분을 리뷰 사실입니다. 제가 중대 병원 초기(생진 지 얼마 안 앓을 때)에 이미, 응급실에서 말도 안되는 오진을 경험한 후 절대 안 가는 병원이었어요. 심한 날개 지었고, 저녁 시간에 머물 수 있어 사고가 나서 병문명하러 간 여권하던중요. 다 최악인 건, 안 해도 되는 입사들 술 데 있어 두드리는 거예요. 의사가 내한 오진 때문에! 결국 그 검사하는 환자들... 절대 가지 마세요 중대 병원. 그후에 최악은 응급실.',
    '외과에 방문했는데 이렇게 불친절한 의사는 처음이네요 아파서 가니까 왜 왔냐고 그래서 어쩌라는거냐고 하는데 이게 의사 입에서 나올 해간가요? 다시는 방문하고 싶지 않습니다.',
    '다 너무 불친절하고 개방문서사 없었음에 비해 입원하는데 다 애설쓰고 불친절하고... 불마가서서 혈관에 바늘 꽂다가 살수하는 것에 맞붙을 다시 붓고 붓고 불친절하였고 뒤고 호중고 급하게 두는 시간 열려나서 올림.',
    '자인 응급실 향로 및 방문. 차종장같은 공사소음과 직원을 여수선함에 실망. 이비인후과로 옮겼는데 시설이나 시스템 낙후되보에 실망. 그러나 의사를분 친절하게 설명해가며 오한서진 꼼꼼히 치료해주고 걱정 덜어줌. 간호사나 원무직원분들까지 인간적으로 너무 친절..']
```


- 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 모형을 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 주체는 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)
```

중앙대학교병원은 간호선생님들과 수납의사까지 모두 친절하여 원활한 진료를 받을 수 있었고, 다른 병원에서는 어려운 예약도 쉽게 받을 수 있어서 감사했습니다. 의료사고로 인해 고통받은 적이 있는 분들도 있으나, 전체적으로 의사와 간호사들은 친절하고 진료를 잘 받을 수 있었다는 평가 많았습니다. 하지만 일부 의료사고 및 불친절한 직원들이 걱정되어 다소 신중한 방문을 권고하는 의견도 보였습니다.

- 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()
```

순	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)
```

```

### 결정 트리
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('결정 트리 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)

```

로지스틱 회귀 정확도: 0.8513528643403621
결정 트리 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)

```

[96] 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
0.9935856318152662

- 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'])

```

0.9999377993406731
{'min_impurity_decrease': 0.0001}
[0.9999378 0.99980096 0.99980096 0.99955216 0.99955216]

```

[65] 제안된 코드에 라이선스가 적용될 수 있습니다. | art1117/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.


```
[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
```

- ```

테스트할 샘플 데이터 추출
sample_data = encoded_feature.iloc[5:].to_frame().T
sample_data

연도 성별 성별_여 연령대_10대 연령대_20대 연령대_30대 연령대_40대 연령대_50대 연령대_60대 근속기간_10년이상 ... 재해유형_사입전도교동사고 재해유형_사입전도교동사고 재해유형_산소결핍 재해유형_이상도물제조속 재해유형_콜드배임질량 재해유형_청_제척행사 재해유형_청_목적행위 재해유형_청_목적파열 재해유형_화재 재해유형_화력물류수출전속

5 2022 True False False False False False True False False ... False False False False False False False False
1 rows x 63 columns

예측
y_pred = clf.predict(sample_data)

```

- ```
0.점 > y_pred[0]
8

0.점 > [131] target_encoded[5]
8

0.점 > df_target.iloc[y_pred[0],0]
'무너짐'
```

- ## 5. 아이디어 기대효과(효과성)

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 장애 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>