

웰로 정책 매칭

코드스테이츠 마지막 2번째 팀프로젝트

팀원 : 오승원(정책정보태깅), 황찬희(유저정보태깅)

웰로 오리지널 콘텐츠

웰로팀이 정성스럽게 쓴 정책 콘텐츠

2022년 1인 소상공인 고용보험료 지원



월 납입 고용보험료의 30% 지원
경기도 소재 1인 소상공인 대상

#경기도 #소상공인 #고용보험료

경영자 코치와 함께하는 스타기업 육성프로그램



선정 기업당 약 1,100만 원 지원
스타기업 발굴 및 육성

#스타트업 #육성프로그램 #경영자

인도

울산 일
1인당 1

#울산 #일

20대 창업청년이라면?

평균 수혜금액 : 222,000만원



내 정책 찾아보기

웰로가 상담해드립니다.

웰로가 선정한 최고의 정책을 신청하실 수
있도록 전담매니저가 도와드립니다

중앙행정기관 고용노동부

국민취업지원제도 II유형

#구직 #취업 #청년 #중장년 #저소득층

상시지원

정책 상세보기

채팅상담

가장 많이 찾아본 정책

어떤 정책을 가장 많이 찾아봤을까요?

주택금융기관 한국주택금융공사

목돈 안드는 전세II

상시지원

중앙행정기관 국토교통부

청년우대형 청약통장

상시지원

```
[3] import pandas as pd
import datetime as dt

import math
```

```
[4] df=pd.read_csv('/content/유저데이터 샘플리스트 서플(AIB전달용) - user_sample_encode.csv')
df1=pd.read_csv('/content/정책데이터 태그데이터리스트(AIB전달용) - 정책정보데이터(POLICY).CSV')
# df2=pd.read_csv('/content/정책정보_2021-12-28.csv', error_bad_lines = False)
```

df.head()



df.iloc[1000]

```
num 21975
gender F
gender_code A02-02
birth 19850610
international_age 36
Age_category 30
home1 대전/대덕구
home1_area_code A01-06-05
home2 대전/대덕구
home2_area_code A01-06-05
study 대학(원) 졸업
study_code A03-04
job 기타
job_code A04-00
house_owner 세대원
marriage 기혼
marriage_code A06-03
have_kids 있다
have_kids_code A07-00
have_kids_code_detail_1 A07-00-03
have_kids_code_detail_2 A07-00-No_code
have_kids_code_detail_3 A07-00-No_code
have_kids_code_detail_4 A07-00-No_code
number_kids 1명
kid_status 초등학생
family_status 해당없음
family_status_code A12-02
job_status 해당없음
income 중위소득 40~60% 사이
illness no
mil_award no
mil_award_code A12-00
support 의료 지원, 교육지원(만8~19세), 성인교육지원
Name: 1000, dtype: object
```

상위구분	지역(시도)	전국	00	A01-00
Prefix	A01	서울특별시	01	A01-01
하위코드존재여! Y		부산광역시	02	A01-02
		대구광역시	03	A01-03
		인천광역시	04	A01-04
		광주광역시	05	A01-05
		대전광역시	06	A01-06
		울산광역시	07	A01-07
		세종특별자치시	08	A01-08
		경기도	09	A01-09
		강원도	10	A01-10
		충청북도	11	A01-11
		충청남도	12	A01-12
		전라북도	13	A01-13
		전라남도	14	A01-14
		경상북도	15	A01-15
		경상남도	16	A01-16
		제주특별자치도	17	A01-17

		경상남도	16	A01-16
		제주특별자치도	17	A01-17
상위구분	성별	무관	00	A02-00
Prefix	A02	남성	01	A02-01
하위코드존재여! N		여성	02	A02-02
상위구분	학력	무관	00	A03-00
Prefix	A03	고등학교졸업	01	A03-01
하위코드존재여! N		대학(원)재학	02	A03-02
		대학(원)휴학	03	A03-03
		대학(원)졸업	04	A03-04
상위구분	직장	무관	00	A04-00
Prefix	A04	대학/대학원생	01	A04-01
하위코드존재여! Y		구직자/실업자	02	A04-02
		근로자/직장인	03	A04-03
		예비창업자	04	A04-04
		소상공인	05	A04-05
		중소기업	06	A04-06
		프리랜서	07	A04-07

상위구분	직장	무관	00	A04-00
Prefix	A04	대학/대학원생	01	A04-01
하위코드존재여! Y		구직자/실업자	02	A04-02
		근로자/직장인	03	A04-03
		예비창업자	04	A04-04
		소상공인	05	A04-05
		중소기업	06	A04-06
		프리랜서	07	A04-07
상위구분	가구원	무관	00	A05-00
Prefix	A05	무주택 세대주	01	A05-01
하위코드존재여! N		무주택 세대원	02	A05-02
		주택 세대주	03	A05-03
		주택 세대원	04	A05-04
상위구분	결혼	무관	00	A06-00
Prefix	A06	미혼	01	A06-01
하위코드존재여! N		기혼	02	A06-02
		이혼	03	A06-03
		재혼	04	A06-04

상위구분	자녀	없음	00	A07-00
Prefix	A07	있음	01	A07-01
하위코드존재여! Y				
상위구분	소관기관유형	중앙행정기관	00	A08-00
Prefix	A08	공공기관 등	01	A08-01
하위코드존재여! N		교육청	02	A08-02
		광역시도	03	A08-03
		시군구	04	A08-04
		민간기관	05	A08-05
상위구분	지원유형	현금	00	A09-00
Prefix	A09	현물	01	A09-01
하위코드존재여! Y		이용권	02	A09-02
		서비스	03	A09-03
		의료지원	04	A09-04
		상담/법률지원	05	A09-05
		문화/여가지원	06	A09-06
		민원	07	A09-07
		봉사/기부	08	A09-08

전국	00	A01-00			
서울특별시	01	A01-01	전체	00	A01-01-00
			종로구	01	A01-01-01
			중구	02	A01-01-02
			용산구	03	A01-01-03
			성동구	04	A01-01-04
			광진구	05	A01-01-05
			동대문구	06	A01-01-06
			종로구	07	A01-01-07
			성북구	08	A01-01-08
			강북구	09	A01-01-09
			도봉구	10	A01-01-10
			노원구	11	A01-01-11
			은평구	12	A01-01-12
			서대문구	13	A01-01-13
			마포구	14	A01-01-14
			양천구	15	A01-01-15
			강서구	16	A01-01-16
			구로구	17	A01-01-17
			금천구	18	A01-01-18
			영등포구	19	A01-01-19
			동작구	20	A01-01-20

무관	00	A04-00			
대학/대학원생	01	A04-01	대학생	00	A04-01-00
			대학원생	01	A04-01-01
구직자/실업자	02	A04-02	건설/재글직	00	A04-02-00
			경영/사무/금융/보험직	01	A04-02-01
			교육/법률/사회복지/경찰/소방/군인직/공무원직	02	A04-02-02
			농림어업직	03	A04-02-03
			미용/여행/숙박/음식/경비/돌봄/청소직	04	A04-02-04
			보건/의료직	05	A04-02-05
			설비/정비/생산직	06	A04-02-06
			연구/공학기술직	07	A04-02-07
			영업/판매/운전/운송직	08	A04-02-08
			예술/디자인/방송/스포츠직	09	A04-02-09
근로자/직장인	03	A04-03	건설/재글직	00	A04-03-00
			경영/사무/금융/보험직	01	A04-03-01
			교육/법률/사회복지/경찰/소방/군인직/공무원직	02	A04-03-02
			농림어업직	03	A04-03-03
			미용/여행/숙박/음식/경비/돌봄/청소직	04	A04-03-04
			보건/의료직	05	A04-03-05
			서비/정비/생산직	06	A04-03-06

만나이 계산기

```
import pandas as pd
from datetime import datetime, date

# Creating a list of date of birth
dob = {'DOB': ['19860513', '20181212', '20060423']}
```

```
# Creating dataframe
df_ = pd.DataFrame(data = dob)
```

```
# This function converts given date to age
def age(born):
    born = datetime.strptime(born, "%Y%m%d").date()
    today = date.today()
    return today.year - born.year - ((today.month,
                                        today.day) < (born.month,
                                                        born.day))

new_age = df['birth'].apply(str).apply(age)

display(new_age)
```

```
[27] df.insert(3, 'international_age', new_age)
```

```
[28] df['international_age']
```

지역 태깅(home1, home2)

```
[50] region=df['home1'].str.split('/').str[0]
     sub_region=df['home1'].str.split('/').str[1]
```

```
# 도시(상위코드)는 좀 더 세분해서 딕셔너리 형태로 만들
cities = {'서울': 'A01-01', '부산': 'A01-02', '대구': 'A01-03', '인천': 'A01-04', '광주': 'A01-05',
          '대전': 'A01-06', '울산': 'A01-07', '세종': 'A01-08', '경기': 'A01-09', '경기도': 'A01-09',
          '강원': 'A01-10', '강원도': 'A01-10', '충청북도': 'A01-11', '충북': 'A01-11', '충청남도': 'A01-12',
          '충남': 'A01-12', '전라북도': 'A01-13', '전북': 'A01-13', '전라남도': 'A01-14', '전남': 'A01-14',
          '경상북도': 'A01-15', '경북': 'A01-15', '경상남도': 'A01-16', '경남': 'A01-16', '제주': 'A01-17'}
```

```
[52] #도시 하위코드
df4 = pd.read_excel('/content/정책데이터 태그셋리스트(A18전달용).xlsx', sheet_name=6)
df4=df4.iloc[3:247,4:7]
col4=['상세지역', '하위코드', '전체코드']
df4.columns=col4
df4['전체코드_세부']=df4['전체코드'].str[-2:]
#세부코드 dict형태로 생성
df4.index = df4['상세지역']
sub_cities=df4['전체코드_세부'].to_dict()
```

```
[53] def code_generate(text, city_code, home):

    region=df[home].str.split('/').str[0]
    sub_region=df[home].str.split('/').str[1]

    lst=[]
    for i in range(0, len(text)):
        if text[i] in city_code.keys():
            code=city_code[text[i]]
            lst.append(code)
        else:
            lst.append('No_code')
    return lst
```

```
[54] A1=code_generate(region, cities, 'home1')
     A2=code_generate(sub_region, sub_cities, 'home1')
```

▼ common_tag_func

```
✓  def code_common(common_code,home):  
  
    text=df[home]  
  
    lst=[]  
    for i in range(0,len(text)):  
        if text[i] in common_code.keys():  
            code=common_code[text[i]]  
            lst.append(code)  
        else:  
            lst.append('No_code')  
    return lst  
  
    data={'common_code':lst}  
    code_frame=pd.DataFrame(data)
```


mb_6(marrige) 태깅

```
[62] marriage={'무관': 'A06-00', '미혼': 'A06-01', '이혼': 'A06-02', '기혼': 'A06-03', '재혼': 'A06-04'}
```

```
• M=code_common(marriage,'marriage')  
df.insert(13,'marriage_code',M)
```

mb_3(study)태깅

```
[64] study={'무관': 'A03-00', '고등학교 졸업': 'A03-01', '대학(원) 재학': 'A03-02', '대학(원) 휴학': 'A03-03', '대학(원) 졸업': 'A03-04'}
```

```
[65] S=code_common(study,'study')
```

```
#10번째 칸에 study의 코드 열 생성  
df.insert(10,'study_code',S)
```

mb_7(have_kids)태깅, 자녀상태의 상위코드

```
[66] have_kids={'있다': 'A07-00', '없다': 'A07-01'}
```

```
[67] HK=code_common(have_kids,'have_kids')
```

```
#10번째 칸에 study의 코드 열 생성  
df.insert(16,'have_kids_code',HK)
```

mb_sex(gender)태깅

```
[73] gender={'무관': 'A02-00', 'M': 'A02-01', 'F': 'A02-02'}
```

```
• G=code_common(gender,'gender')  
df.insert(2,'gender_code',G)
```

mb_4(job)태깅

```
[75] job={'기타': 'A04-00', '대학(원)생': 'A04-01', '구직자': 'A04-02', '근로자': 'A04-03', '예비창업자': 'A04-04', '기업인/소상공인': 'A04-05', '프리랜서': 'A04-07'}
```

```
[76] J=code_common(job,'job')  
df.insert(13,'job_code',J)
```

mb_14(mil_award)상위코드 태깅

```
[77] #독립유공자는 상위코드가 존재하지 않아 부득이하게 해당없음 코드의 A12-00코드 부여  
mil_award={'5, 18 민주유공자': 'A12-01', '독립유공자': 'A12-00', '해당없음': 'A12-00', 'no': 'A12-00', '고려제 후유(의)증': 'A12-01-01', '국가유공자': 'A12-07', '참전유공자': 'A09-04',  
           '재대군인': 'A12-08', '보훈보상대상자': 'A12-03-00', '고려제 후유(의)증': 'A12-01-00', '특수임무유공자': 'A12-01'}
```

```
[78] H0=code_common(mil_award,'mil_award')
```

```
#30번째 칸에 mil_award의 코드 열 생성  
df.insert(20,'mil_award_code',H0)
```

```
user_tag.head(10)
```

	Age_category	international_age	gender_code	home1_area_code	home2_area_code	study_code	job_code	marriage_code	have_kids_code	have_kids_code_detail_1	have_kids_code_detail_2	have_kids_code_detail_3
0	120	122	A02-01	A01-11-01	A01-11-01	No_code	A04-00	No_code	No_code	No_code-No_code	No_code-No_code	No_code-No_code
1	120	121	A02-01	A01-09-03	A01-09-03	A03-04	A04-03	A06-01	A07-01	A07-01-No_code	A07-01-No_code	A07-01-No_code
2	110	117	A02-01	A01-16-08	A01-16-08	No_code	No_code	No_code	No_code	No_code-No_code-No_code	No_code-No_code	No_code-No_code
3	100	103	A02-01	A01-09-26	A01-09-26	A03-04	A04-03	A06-03	A07-00	A07-00-00	A07-00-No_code	A07-00-No_code
4	100	101	A02-01	A01-06-04	A01-06-04	A03-01	A04-01	A06-01	A07-01	A07-01-No_code	A07-01-No_code	A07-01-No_code
5	100	101	A02-01	A01-17-00	A01-17-00	No_code	A04-00	A06-01	A07-01	A07-01-No_code	A07-01-No_code	A07-01-No_code
6	100	100	A02-02	A01-05-03	A01-05-03	No_code	A04-00	A06-01	A07-01	A07-01-No_code	A07-01-No_code	A07-01-No_code
7	90	99	A02-01	A01-01-14	A01-01-14	A03-01	A04-02	A06-01	A07-01	A07-01-No_code	A07-01-No_code	A07-01-No_code
8	80	82	A02-02	A01-11-01	A01-11-01	A03-02	No_code	A06-01	A07-01	A07-01-No_code	A07-01-No_code	A07-01-No_code
9	80	80	A02-01	A01-09-10	A01-09-10	A03-01	A04-00	A06-01	A07-01	A07-01-No_code	A07-01-No_code	A07-01-No_code

```
policy_tag.head(30)
```

구분	Unnamed: 0	정책 ID	정책서비스 ID	서비스명	태그-지역(시도)	태그-신청절차	태그-성별	태그-직장	태그-가구원	태그-대상연령시작	태그-대상연령끝	태그-자녀
0	0	2	105100000001	근로·지역장려금	A01-00	A10-01	A02-00	A04-00	A05-00	0.0	99.0	A07-01
1	1	3	116010000001	월세 자금보조	A01-00	A10-02	A02-00	A04-00	A05-00	0.0	99.0	A07-00
2	2	4	1160100000027	'힘내라 대한민국'특별운영자금 (산업은행)	A01-00	A10-01	A02-00	A04-00	A05-00	0.0	99.0	A07-00
3	3	5	1192000000001	친환경 에너지 절감 장비 보급	A01-00	A10-02	A02-00	A04-00	A05-00	0.0	99.0	A07-00
4	4	6	1192000000007	행랑사고 극선 심판발령인 신청 지원	A01-00	A10-02	A02-00	A04-00	A05-00	0.0	99.0	A07-00
5	5	7	1192000000008	읍·서버 응선경비 지원	A01-00	A10-02	A02-00	A04-00	A05-00	0.0	99.0	A07-00
6	6	8	1192000000010	합작수산물 관세 감면 추진	A01-00	A10-02	A02-00	A04-00	A05-00	0.0	99.0	A07-00
7	7	9	1192000000012	귀어 창업 및 주택구입 지원(원자)	A01-00	A10-02	A02-00	A04-00	A05-00	0.0	99.0	A07-00
8	8	10	1192000000016	월양어업 경영자금 지원	A01-00	A10-02	A02-00	A04-00	A05-00	0.0	99.0	A07-00
9	9	11	1192000000019	작업장 소멸 할은노동조합 조합원 생계지원금 지급	A01-00	A10-02	A02-00	A04-00	A05-00	0.0	99.0	A07-00
10	10	12	1192000000022	월양어선안전관리조 명칭 변경	A01-00	A10-01	A02-00	A04-00	A05-00	0.0	99.0	A07-00
11	11	13	1192000000023	양식시설 현대화 자금 지원	A01-00	A10-02	A02-00	A04-00	A05-00	0.0	99.0	A07-00
12	12	14	1192000000024	수산장비 구입 지원	A01-00	A10-02	A02-00	A04-00	A05-00	0.0	99.0	A07-00
13	13	15	1192000000025	배합사료 무료구입비용 운영비 용자 지원	A01-00	A10-02	A02-00	A04-00	A05-00	0.0	99.0	A07-00
14	14	16	1192000000027	산지 소 소비자 유통자금 용자 지원	A01-00	A10-02	A02-00	A04-00	A05-00	0.0	99.0	A07-00
15	15	17	1192000000043	재해복구자금	A01-00	A10-02	A02-00	A04-00	A05-00	0.0	99.0	A07-00
16	16	18	1192000000044	수산업영민회생자금	A01-00	A10-02	A02-00	A04-00	A05-00	0.0	99.0	A07-00
17	17	19	1192000000045	어업경영자금 지원	A01-00	A10-02	A02-00	A04-00	A05-00	0.0	99.0	A07-00

✓
8주



```
IL=len(user_tag['international_age'])

list=[]

for j in range(0,IL):
    lst=[]
    a=user_tag['international_age'][j]
    for i in range(0,1000):
        if (a >= policy_tag['태그-대상연령시작'][i] and a <= policy_tag['태그-대상연령끝'][i]):

            lst.append(policy_tag['서비스명'][i])

    list.append(lst)
```

✓
3주

```
[10] # list_data=pd.DataFrame(list, columns='연령 서비스명')
      list_data=pd.DataFrame(list)
```

✓
4

```
[11] list_data.to_csv('/content/sample_data/df__user_code_service')
```

✓
4분



```
list=[]
for i in range(0,len(user_tag)):
    lst=[]
    for j in range(0,1000):
        if (policy_tag['태그-지역(시도)'][j]==user_tag['home1_area_code'][i]):
            lst.append(policy_tag['서비스명'][i])
    list.append(lst)
```

✓
0초

[30] len(list)

20010

✓
4분

[31]

```
list_1=[]
for i in range(0,len(user_tag)):
    lst=[]
    for j in range(0,1000):
        if (policy_tag['태그-직장'][j]==user_tag['job_code'][i]):
            lst.append(policy_tag['서비스명'][i])
    list_1.append(lst)
```

```
✓ 4초 ▶ a=pd.DataFrame(list)
b=pd.DataFrame(list_1)
# c=pd.DataFrame(list_2)
```

```
✓ 4초 [94] a.to_csv('/content/sample_data/df_area_code_service')
b.to_csv('/content/sample_data/df_job_code_service')
# c.to_csv('/content/sample_data/df_kids_code_service')
```

```
✓ 0초 [121] len(a)
```

20010

```
✓ 0초 [122] len(b)
```

20010

```
✓ 0초 [123] len(user_service)
```

628

```
[120] def result(number):
      print('지역에 따른 정책 : ',a[0][:628][number])
      print('직업에 따른 정책 : ',b[0][:628][number])
      print('나이에 따른 정책 : ',user_service[:628][number])
```

결론

-머신러닝 사용하기 위해 충분히 생각(X)

-작업시간의 분배를 생각하지 못하고 모든 태그작업을 시도

- 그리하여 모델을 만들시간이 촉박해짐

- if문으로 가능한 많은 태그를 정리하려 했으나 각컬럼당 20000개가 넘어

거기에 태그 매칭까지 생각하면 하나의 유저컬럼당 몇시간씩 걸려

어마어마한 작업 소모가 일어남 그래서 어쩔수 없이 유저당 1000개의

태그로 마무리 지음

- 유저가 원하는 정보를 넣어 매칭하는 시스템을 만들지 못해 추후 좋은 포트폴리오를 위해
업데이트 예정

감사합니다