



# [KNOW기반 직업 추천 알고리즘 경진대회] 데이터 분석 및 사용한 알고리즘 소개

AI학과 2143933 김지선

# 목차

A table of contents

---

1

대회 소개

2

데이터 분석

3

Decision Tree 알고리즘으로 분석

4

User-based CF 알고리즘으로 분석

5

느낀점 및 앞으로의 방향성





# 1

# 1. 대회 소개

대회 소개 및 방법

# 1 대회 소개 - 대회 일정 및 규칙

- 플랫폼:  DAICON
- 주관:  한국고용정보원  
Korea Employment Information Service
- 평가지표: F1-score
- 대회 일정: 2021.12.06~2022.01.28(18:00)
- 링크: <https://daicon.io/competitions/official/235865/overview/description>

# 1 KNOW기반 직업 추천 알고리즘 경진대회

- KNOW 설문조사  
재직자의 성향과 직무, 직무 만족도를 통해서 비슷한 성향을 가진 다른 구직자에게 적합한 직업을 추천해주는 대회
- KNOW 직업 추천 대회의 목적  
재직자가 가진 직업( $w$ )에서 에 필요한 능력에 대한 정보로 비슷한 능력을 가진 구직자에게 그 직업( $w$ ) 추천

## 2. 데이터 분석

파일 분석 및 데이터 구조



2

## 2 학습 데이터의 내용

재직자의 설문응답을 기록한 학습 데이터에 있는 내용

- 업무 중요도
- 업무 수준
- 직무 조사
- 필요 능력
- 직무 만족도
- 재직자의 직업

+ 나눈 기준은 대회측 변수정보에서 라벨링 해준 것에 조금 더 분류한 것

## 2 용어 정리

---

### 1. 재직자

- 설문조사에 답한 사람
- train 데이터의 한 행
- **직업 아는** 상태

### 2. 구직자

- 직업을 추천 받는 사람
- Test 데이터의 한 행
- 재직자와 **같은 설문**에 응답했으나 **직업을 모름**



## 2 용어 정리

---

### 3. 직업

- 약 500개의 직업 리스트로 이중 1개를 구직자에게 추천

### 4. 능력 (직접 정의)

- 중요하다 생각하는 업무 -> 업무 중요도
- 본인이 해당 업무에 높은 수준인 업무 -> 업무 역량
- 중요한 업무이고 && 그에 대한 역량이 높은 업무

### 5. 업무

- 업무 중요도와 업무 역량에서 공통적으로 물어보는 업무

## 2 업무와 능력 예시

6

사물, 서비스,  
사람의 질 판단

사물, 사람의 가치, 중요성, 질을 평가하기

가. 귀하의 업무를 하기 위해 **【사물, 서비스, 사람의 질 판단】** 활동이 얼마나 중요합니까?

중요하지  
않다

①

약 간  
중요하다

②

중요하다

③

아 주  
중요하다

④

아주 많이  
중요하다

⑤

↳ ①번 **【중요하지 않다】** 에 ●표하신 분은 "나. 질문"을 건너뛰고 다음 문항으로 이동하십시오.

나. 귀하의 업무에 필요한 **【사물, 서비스, 사람의 질 판단】** 활동의 수준은 어느 정도라고 생각하십니까?

훼손된 나무를 없앨  
것인지 결정한다



①

②

③

④

화재로 인한 재산  
피해를 측정한다



⑤

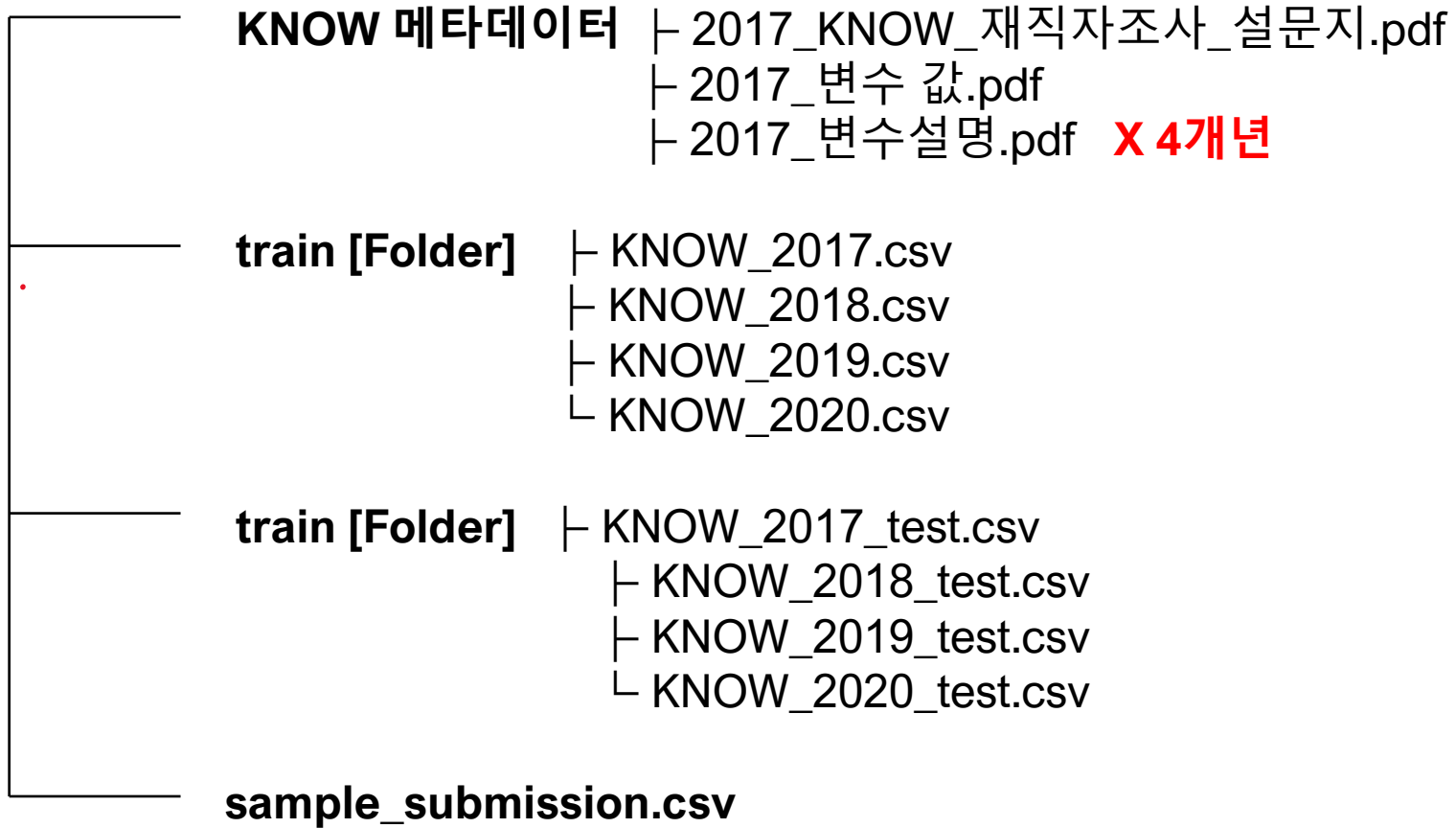
최근 발견된 고대 예술  
작품의 가치를 결정한다



⑥

⑦  
가장 높은 수준

## 2 전체 데이터 디렉터리 구조



## 2 메타 데이터 분석

- 년도 별 설문지와 칼럼 설명 직업 코드 별 직업

인덱스	파일명	내용
1	2017_KNOW__재직자조사_설문지.pdf	설문지
2	2017_변수값.pdf	변수 값의 뜻
3	2017년_변수정보.pdf	칼럼 뜻
4	2018년_KNOW__재직자조사_설문지.pdf	설문지
5	2018_변수값.pdf	변수 값의 뜻
6	2018년_변수정보.pdf	칼럼 뜻
7	2019년_KNOW__재직자조사_설문지.pdf	설문지
8	2019_변수값.pdf	변수 값의 뜻
9	2019년_변수정보.pdf	칼럼 뜻
10	2020년_KNOW__재직자조사_설문지.pdf	설문지
11	2020_변수값.pdf	변수 값의 뜻
12	2020년_변수정보.pdf	칼럼 뜻

## 2 나머지 파일 분석

- 4개년의 학습 및 테스트 데이터와 메타데이터 제공 받음

인덱스	용도	파일명	크기
1	X	sample_submission.csv	(35231, 2)
2	TRAIN	KNOW_2017.csv	(9486, 156)
3	TEST	KNOW_2017_test.csv	(9486, 155)
4	TRAIN	KNOW_2018.csv	(9072, 141)
5	TEST	KNOW_2018_test.csv	(9069, 140)
6	TRAIN	KNOW_2019.csv	(8555, 153)
7	TEST	KNOW_2019_test.csv	(8554, 152)
8	TRAIN	KNOW_2020.csv	(8122, 185)
9	TEST	KNW_2020_test.csv	(8122, 184)

## 2 특이점을 갖는 칼럼 우선 분석

- 업무 중요도와 업무 역량 질문의 반복 구조가 절반 이상을 차지

### 1 정보 수집 모든 관련 자료에서 정보를 수집, 관찰하기

가. 귀하의 업무를 하기 위해 **【정보 수집】** 활동이 얼마나 **중요합니까?**

중요하지 않다      약 간 중요하다      중요하다      아 주 중요하다      아주 많이 중요하다  
①                      ②                      ③                      ④                      ⑤

L①번 **【중요하지 않다】**에 ●표하신 분은 "나. 질문"을 건너뛰고 다음 문항으로 이동하십시오.

나. 귀하의 업무에 필요한 **【정보 수집】** 활동의 수준은 어느 정도라고 생각하십니까?

도면을 이해한다      예산을 검토한다      국제세법을 연구한다  
↓                      ↓                      ↓  
①                      ②                      ③                      ④                      ⑤                      ⑥                      ⑦  
가장 높은 수준

### 5 제품, 사건, 정보의 수치 추정 크기, 거리, 양을 추정하거나, 업무 활동을 하기 위하여 시간, 비용, 자원, 자재를 결정하기

가. 귀하의 업무를 하기 위해 **【제품, 사건, 정보의 수치 추정】** 활동이 얼마나 **중요합니까?**

중요하지 않다      약 간 중요하다      중요하다      아 주 중요하다      아주 많이 중요하다  
①                      ②                      ③                      ④                      ⑤

L①번 **【중요하지 않다】**에 ●표하신 분은 "나. 질문"을 건너뛰고 다음 문항으로 이동하십시오.

나. 귀하의 업무에 필요한 **【제품, 사건, 정보의 수치 추정】** 활동의 수준은 어느 정도라고 생각하십니까?

이삿짐 운송 상자에 넣을  
가정용 가구의 크기를  
가능한다      대형 재난이 발생했을 때  
도시를 벗어나는데  
필요한 시간을 추정한다      전세계 바다아래의  
천연 자원 매장량을  
추정한다  
↓                      ↓                      ↓  
①                      ②                      ③                      ④                      ⑤                      ⑥                      ⑦  
가장 높은 수준

## 2 업무 중요도와 업무 역량에 초점을 맞춘 이유

- 업무 중요도 질문에 1번 중요하지 않다 선택시 다음 업무 역량 질문 건너뛸  
+건너뛸 == **중요하지 않은 업무에 대해서는 업무 수준은 물어보지 않음**
- 반복 구조로 묻는데에 **추출할 수 있는 의미가 있다고 추론**

## 2 특이점을 갖는 칼럼 우선 분석(실제 데이터)

- 특이점은 실제 데이터에서도 아래 구조로 이루어져 있었다.

_1로 끝남	업무 중요도
_2로 끝남	업무 역량

	aq1_1	aq1_2	aq2_1	aq2_2	aq3_1	aq3_2	aq4_1	aq4_2	aq5_1	aq5_2	...	aq37_1	aq37_2	aq38_1	aq38_2	aq39_1	aq39_2	aq40_1	aq40_2	aq41_1
0	3	3	3	3	3	3	4	4	3	4	...	2	2	2	2	5	2	2	2	
1	4	5	4	5	3	4	3	4	3	4	...	3	4	3	4	2	2	1	0	
2	3	4	3	4	3	4	5	6	4	5	...	3	4	3	4	1	0	1	0	
3	3	3	3	3	3	5	4	5	4	6	...	4	4	4	4	4	4	4	2	
4	4	5	3	4	3	4	4	5	3	4	...	2	2	3	4	2	2	1	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
9481	3	5	2	4	3	3	2	2	2	3	...	3	4	3	5	2	3	3	3	
9482	5	5	5	5	5	5	3	4	4	5	...	4	4	4	5	2	1	1	0	
9483	3	3	4	6	3	3	4	5	4	5	...	2	2	1	0	1	0	1	0	
9484	3	5	3	5	4	5	3	4	3	5	...	4	5	4	5	4	4	1	0	
9485	3	4	3	4	3	4	3	4	3	4	...	3	4	3	4	2	3	3	4	



## 2 업무중요도 파일

<Importance2017>: 1~5사이 선택 응답

Importance2017

	aq1_1	aq2_1	aq3_1	aq4_1	aq5_1	aq6_1	aq7_1	aq8_1	aq9_1	aq10_1	...	aq32_1	aq33_1	aq34_1	aq35_1	aq36_1	aq37_1
<b>0</b>	3	3	3	4	3	3	2	2	2	3	...	1	2	2	3	3	2
<b>1</b>	4	4	3	3	3	1	1	1	1	2	...	1	3	3	1	3	3
<b>2</b>	3	3	3	5	4	1	1	3	3	3	...	1	3	3	3	3	3
<b>3</b>	3	3	3	4	4	3	3	4	5	4	...	4	5	5	4	4	4
<b>4</b>	4	3	3	4	3	1	1	1	1	3	...	1	2	2	1	2	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
<b>9481</b>	3	2	3	2	2	2	3	2	2	3	...	2	2	3	3	2	3
<b>9482</b>	5	5	5	3	4	5	4	5	4	4	...	3	4	4	4	4	4
<b>9483</b>	3	4	3	4	4	3	3	1	2	3	...	1	2	2	2	3	2
<b>9484</b>	3	3	4	3	3	4	4	4	4	4	...	2	4	4	4	4	4
<b>9485</b>	3	3	3	3	3	3	3	3	4	3	...	3	3	3	3	3	3

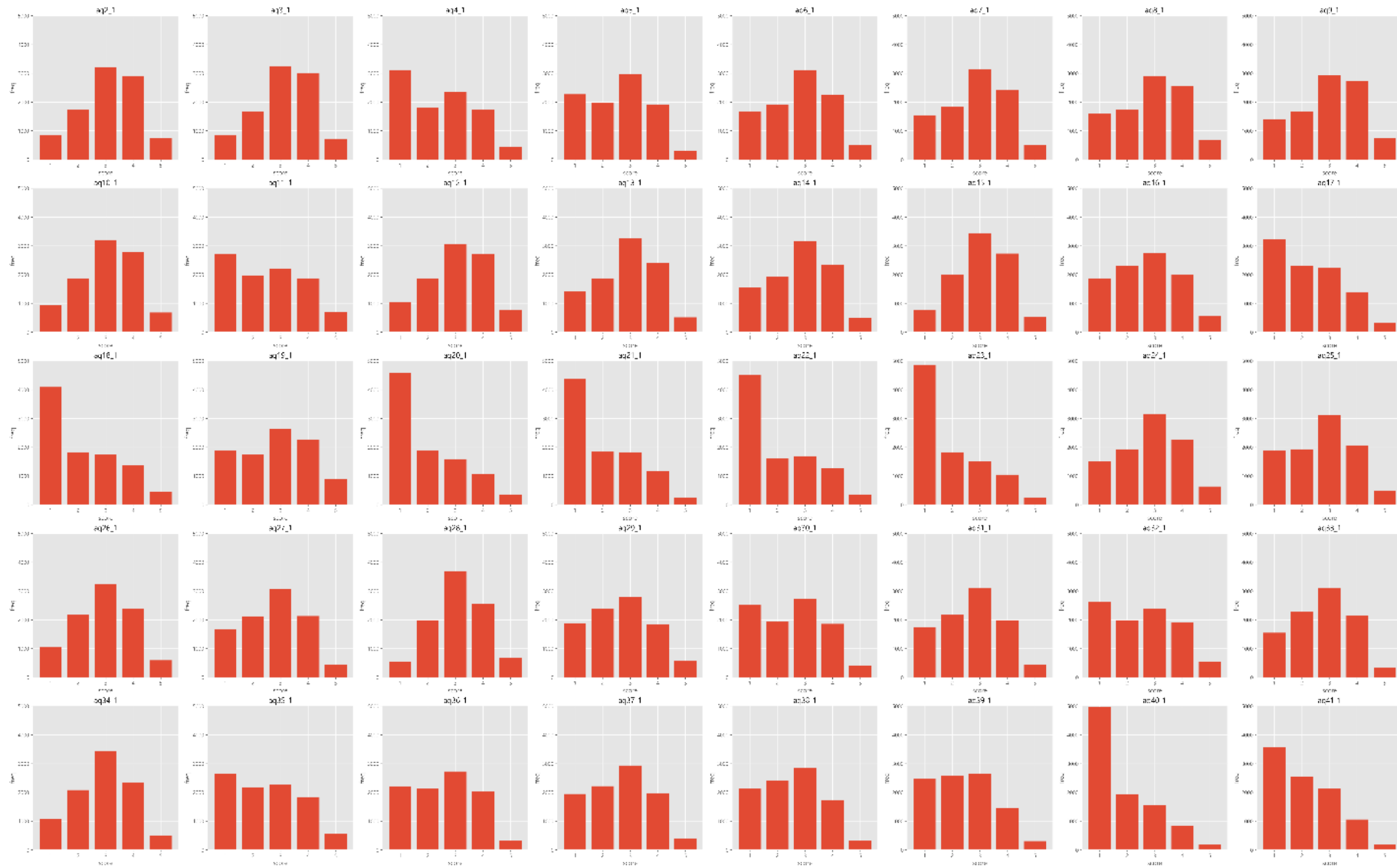
## 2 업무 역량 파일

<level2017>: 1~7 사이 선택 응답

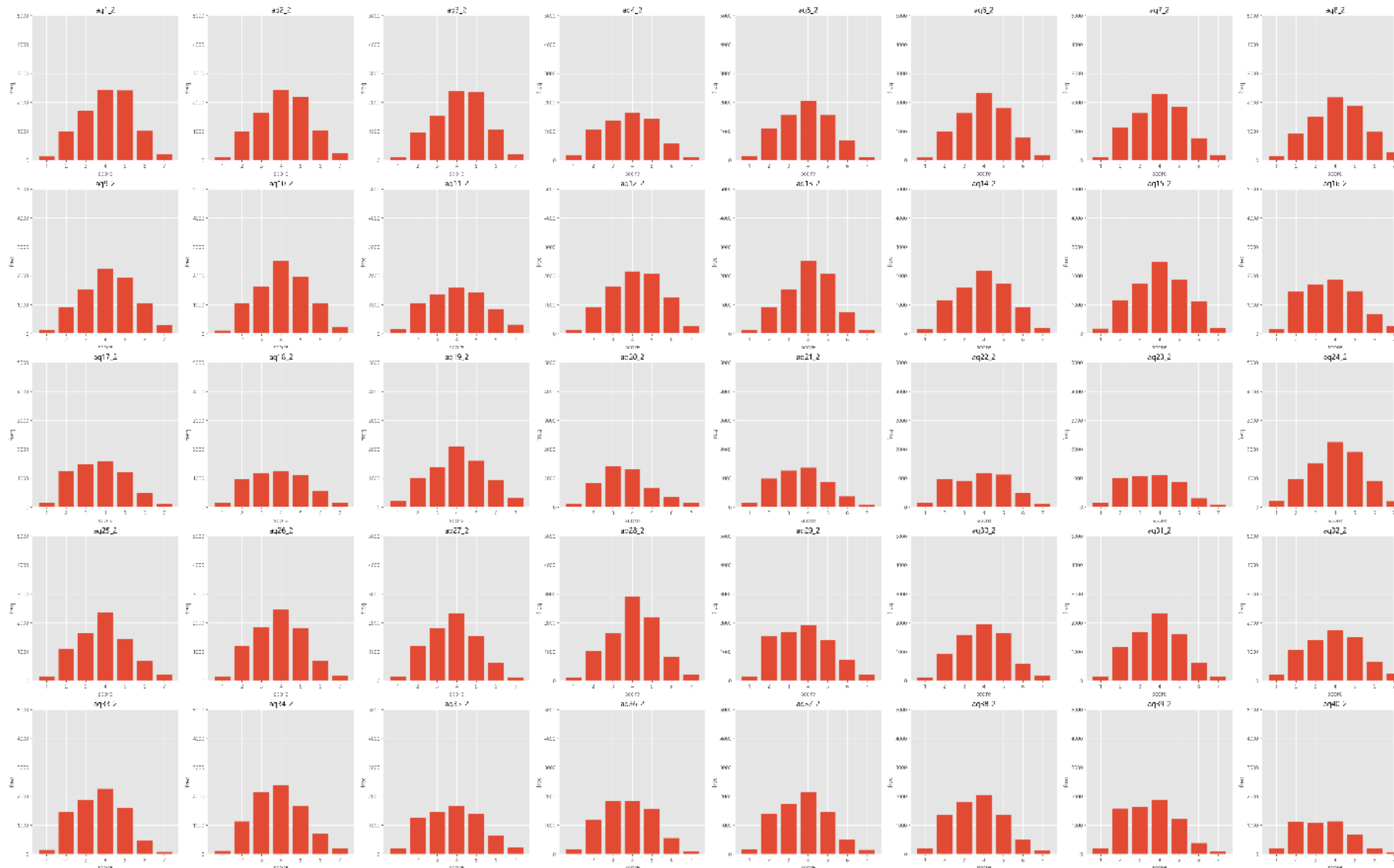
level2017

	aq1_2	aq2_2	aq3_2	aq4_2	aq5_2	aq6_2	aq7_2	aq8_2	aq9_2	aq10_2	...	aq31_2	aq32_2	aq33_2	aq34_2	aq35_2	aq36_2	aq37_2	aq3
<b>0</b>	3	3	3	4	4	3	2	2	2	3	...	3	0	2	5	4	4	2	
<b>1</b>	5	5	4	4	4	0	0	0	0	3	...	4	0	4	4	0	4	4	
<b>2</b>	4	4	4	6	5	0	0	4	4	4	...	4	0	4	4	4	4	4	
<b>3</b>	3	3	5	5	6	5	4	5	5	5	...	4	3	5	4	3	4	4	
<b>4</b>	5	4	4	5	4	0	0	0	0	4	...	0	0	2	3	0	3	2	
<b>...</b>	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
<b>9481</b>	5	4	3	2	3	3	3	3	3	4	...	3	3	3	4	4	3	4	
<b>9482</b>	5	5	5	4	5	5	4	5	5	5	...	4	4	5	5	3	4	4	
<b>9483</b>	3	6	3	5	5	4	4	0	2	3	...	4	0	2	2	2	4	2	
<b>9484</b>	5	5	5	4	5	6	5	5	6	6	...	5	2	4	5	4	5	5	
<b>9485</b>	4	4	4	4	4	5	3	5	4	3	...	4	4	4	4	4	4	4	

## 2 업무중요도 파일 응답 분포



# 2 업무역량 파일 응답 분포



## 2 응답분포를 기반으로 평균 정의

- 응답의 분포를 기준으로 업무 중요도()와 업무 역량의 기준을 잡을 수 있다.

### <업무 중요도>

-Importance2017

Importance2017.mean()

aq2_1	3.091503
aq3_1	3.100464
aq4_1	2.423888
aq5_1	2.571157
aq6_1	2.784735
aq7_1	2.837128
aq8_1	2.882248
aq9_1	2.969007
aq10_1	3.035210
aq11_1	2.553553
aq12_1	3.023825
aq13_1	2.855787
aq14_1	2.809509
aq15_1	3.020346
aq16_1	2.683428
aq17_1	2.284735
aq18_1	2.175206
aq19_1	2.835547
aq20_1	2.008644
aq21_1	2.044697
aq22_1	2.078115

### <업무 역량>

-level2017

level2017.mean()

aq1_2	3.847987
aq2_2	3.742463
aq3_2	3.776407
aq4_2	2.565992
aq5_2	2.929159
aq6_2	3.280308
aq7_2	3.312988
aq8_2	3.418090
aq9_2	3.533418
aq10_2	3.663504
aq11_2	2.847987
aq12_2	3.720852
aq13_2	3.404913
aq14_2	3.322897
aq15_2	3.674678
aq16_2	3.048493
aq17_2	2.428105
aq18_2	2.167405
aq19_2	3.239300
aq20_2	1.879401
aq21_2	1.948134

## 2 데이터 분석-만족도 칼럼

- 만족도와 불만족도 나눠서 데이터프레임 생성후 위와 같은 방식으로 평균 처리

인덱스	만족/불만족	칼럼명	칼럼 정의	데이터 예시	범위
1	만족	bq8_1	사회적 평판_사회적 기여/타인의 인정 받음	4	객관식 (1~5)
2	만족 +	bq8_2	사회적 평판_자녀의 동일 직업 선택시 지지	3	객관식 (1~5)
3	만족	bq8_3	사회적 평판_자녀에게 동일 직업 권유	3	객관식 (1~5)
4	만족	bq12_1	직무만족_급여	4	객관식 (1~5), 해당없음 (9)
5	만족	bq12_2	직무만족_승진	3	객관식 (1~5), 해당없음 (9)
6	만족	bq12_3	직무만족_상사	4	객관식 (1~5), 해당없음 (9)
7	만족	bq12_4	직무만족_동료	4	객관식 (1~5), 해당없음 (9)
8	만족	bq12_5	직무만족_전반적	4	객관식 (1~5), 해당없음 (9)
9	만족	bq13	직무만족_10년전 대비 위상	4	객관식 (1~5)
10	만족	bq14	직무만족_10년후 위상	3	객관식 (1~5)
11	만족	bq15_1	직무몰입_전직 희망	2	객관식 (1~5)
12	만족	bq15_2	직무몰입_평생직업 인식	4	객관식 (1~5)
13	만족	bq15_3	직무몰입_전직 인식	3	객관식 (1~5)
1	불만족	bq18_1	직업 스트레스_업무량 과다	4	객관식 (1~5)
2	불만족	bq18_2	직업 스트레스_기일 업무 반복적 업무	4	객관식 (1~5)
3	불만족	bq18_3	직업 스트레스_고객과 접촉	2	객관식 (1~5)
4	불만족	bq18_4	직업 스트레스_감정 관리 및 조절	3	객관식 (1~5)
5	불만족	bq18_5	직업 스트레스_직업 유지 걱정	2	객관식 (1~5)
6	불만족	bq18_6	직업 스트레스_직업으로 인한 우울감	3	객관식 (1~5)
7	불만족	bq18_7	직업 스트레스_전체적인 직업 스트레스	3	



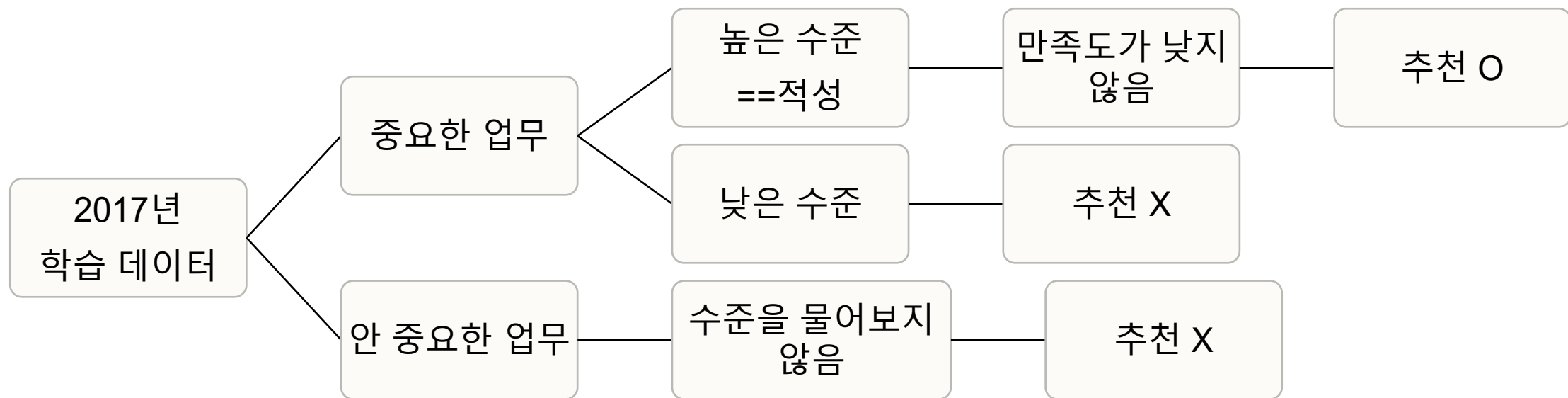
# 3

## 결정트리 알고리즘

Knowcode와 칼럼이 대응하는 데이터  
프레임 생성

### 3 핵심 알고리즘, 직업코드 별 칼럼 추출하는 알고리즘

- 추천 - 능력 데이터프레임에 직업코드별 특성 데이터로 넣는 것
- 만족도가 극단적으로 낮지 않음 != 만족도에 1번 or 불만족도에 5번 (매우 불만족)





### 3 직업코드와 능력에 대한 관계를 빈도로 표현하는 Matirx 생성

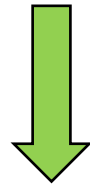
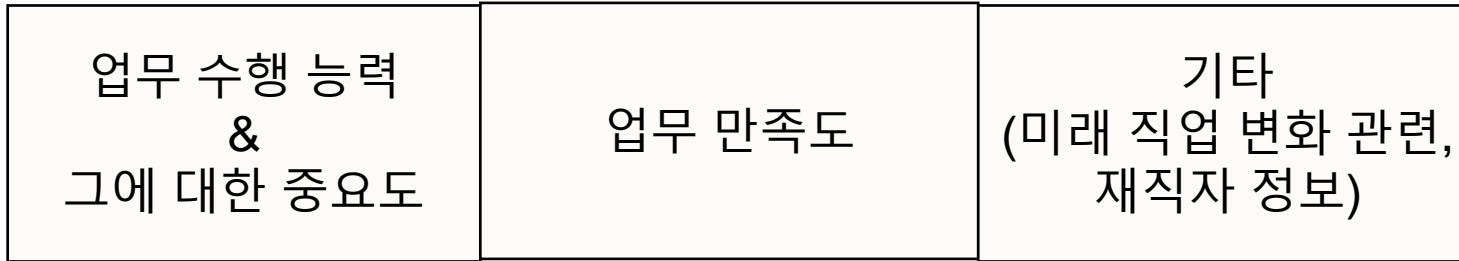
- 직업마다 필요한 능력을 표현하기 위해서임
- 칼럼 구성 – 직업코드(knowcode)와 능력 칼럼으로 이루어짐
- 업무 중요도 칼럼 변수로 새 행렬을 생성

	knowcode	aq1_1	aq2_1	aq3_1	aq4_1	aq5_1	aq6_1	aq7_1	aq8_1	aq9_1	...	aq32_1	aq33_1	aq34_1	aq35_1	aq36_1	aq37_1	aq38_1	aq39_1	a
0	11102	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
1	11201	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
2	12101	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
3	12201	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
4	12301	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
532	902101	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
533	902201	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
534	903101	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
535	904101	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
536	904201	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0

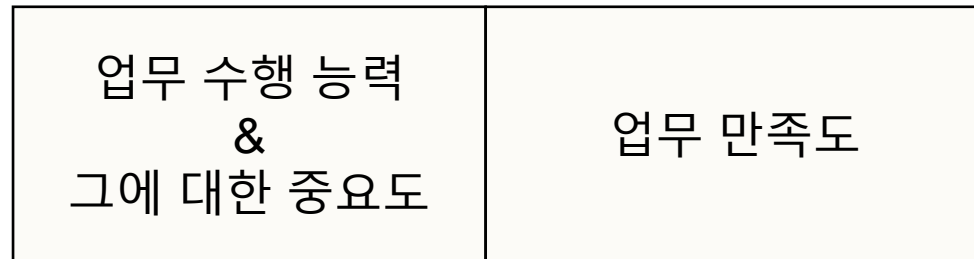
537 rows × 42 columns

### 3 Train Algorithm Step 1

- 이 단계까지는 Train 데이터나 Test 데이터 모두 거침



1차 가공



### 3 Train Algorithm Step2 – 공통과정

업무 수행 능력 & 그에 대한 중요도	업무 만족도
----------------------------	--------



칼럼 개수를 줄임

칼럼을 반복문 돌리며 아래 조건을 만족시키는 행만 남김  
if ( 한 유저의 업무중요도에 대한 개별 응답이 전체 업무 중요도 평균보다 높고 )  
if ( 한 유저의 업무 역량이 " )  
If ( 만족도 1 or 5 점이 아닐 때 )



[학습 시]

생성한 행렬에서 해당 유저의 직업코드  
행에서 위 조건 만족시킨 칼럼 +1



[예측 시]

직업코드 Matrix 기반으로 예측

### 3 공통과정 이후 학습 시 알고리즘

- 생성한 Matirx에 해당 유저의 직업코드 행에서 위 조건 만족시킨 칼럼 +1

ex) 계산 예시

- Train 한 행(2017년 학습 데이터)

idx	aq1_1	aq1_2	aq2_1	aq2_2	aq3_1	aq3_2	aq4_1	aq4_2	aq5_1	...	bq37	bq38	bq38_1	bq39_1	bq39_2	bq40	bq41_1	bq41_2	bq41_3	knowcode	
1	1	4	5	4	5	3	4	3	4	3	...	38	4	건축공학	1	1	1	NaN	NaN	2400	140204

- Matrix 한 행(Matrix)

knowcode	aq1_1	aq2_1	aq3_1	aq4_1	aq5_1	aq6_1	aq7_1	aq8_1	aq9_1	...	aq32_1	aq33_1	aq34_1	aq35_1	aq36_1	aq37_1	aq38_1	aq39_1	a
0	11102	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
1	11201	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
2	12101	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
3	12201	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
4	12301	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
...	...	+1	...	+1	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
532	902101	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
533	902201	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0

### 3 공통과정 이후 예측 시 알고리즘

- Matrix와 교집합(빈도수)가 가장 큰 직업코드 추천

	knowcode	aq1_1	aq3_1	aq41_1	sum
0	11102	7	7	7	21
1	11201	15	15	15	45
2	12101	4	4	4	12
3	12201	31	30	32	93
4	12301	11	12	12	35
...	...	...	...	...	...
532	902101	12	12	12	36
533	902201	3	3	3	9
534	903101	4	4	4	12
535	904101	10	10	11	31
536	904201	3	3	3	9

..	aq37_1	aq37_2	aq38_1	aq38_2	aq39_1	aq39_2	aq40_1	aq40_2	aq41_1	a
..	3	4	3	4	2	2	1	0	1	

_1	aq9_1	...	aq32_1	aq33_1	aq34_1	aq35_1	aq36_1	aq37_1	aq38_1	aq39_1
12	13	...	10	12	13	8	13	9	7	0
3	3	...	4	3	3	3	4	2	3	0
8	6	...	5	7	6	4	7	4	4	0
34	35	...	31	33	31	22	34	23	26	30
12	12	...	12	11	7	14	13	13	13	10
...	...	...	...	...	...	...	...	...	...	...
4	4	...	8	4	4	4	5	4	4	0
2	2	...	1	1	1	3	1	0	3	0
2	2	...	0	1	3	3	2	2	0	0

### 3 Train Data 학습 알고리즘 코드

- 적성 = 중요한 업무 && 잘하는 업무(높은 업무 역량) && 만족도가 아주 낮지 않으면

```
for idx in tqdm(range(train2017_csv.shape[0])):
    knowcodePeridx=train2017_csv.iloc[idx,155]
    print(f"\n\nidx: {idx}")
    print(knowcodePeridx)
    for jdx,j in enumerate(Importance2017):
        if Importance2017.iloc[idx,jdx]>1:
            if level2017.iloc[idx,jdx]>level:
                if positivSatisfact2017.iloc[idx,jdx]>0:
                    test_apptitude2017.iloc[idx,jdx]>0:
                        print("중요0 역량0 만족도X: 추천")
                    else:
                        print("중요0 역량0 만족도X: 추천")
                else:
                    print("중요0 역량X: 비추천")
            else:
                print("중요X: 비추천")
```

```
idx: 9485
15201
중요X: 비추천
중요X: 비추천
중요X: 비추천
중요0 역량0 만족도X: 추천
중요0 역량0 만족도X: 추천
중요0 역량0 만족도X: 추천
중요0 역량X : 비추천
중요0 역량0 만족도X: 추천
중요0 역량0 만족도X: 추천
중요X: 비추천
중요X: 비추천
중요X: 비추천
중요0 역량0 만족도X: 추천
중요0 역량0 만족도X: 추천
중요0 역량X : 비추천
```

```
idx: 0
825101
중요X: 비추천
중요X: 비추천
중요X: 비추천
중요0 역량0 만족도X: 추천
중요0 역량0 만족도X: 추천
중요0 역량X : 비추천
중요X: 비추천
중요X: 비추천
중요X: 비추천
중요0 역량0 만족도X: 추천
중요0 역량0 만족도X: 추천
중요0 역량X : 비추천
중요0 역량X : 비추천
중요X: 비추천
```

### 3 생성된 직업코드별 필요한 능력을 나타낸 행렬 생성됨

```
# 만족도 포함 후
test_aptitude2017n
# 만족도 조정 후
test_aptitude2017
```

	knowcode	aq1_1	aq2_1	aq3_1	aq4_1	aq5_1	aq6_1	aq7_1	aq8_1	aq9_1	...	aq32_1	aq33_1	aq34_1	aq35_1	aq36_1	aq37_1	aq38_1	aq39_1	ac
0	11102	12	1	6	0	6	8	13	12	13	...	10	12	13	8	13	9	7	6	
1	11201	4	4	3	1	2	3	2	3	3	...	4	3	3	3	4	2	3	2	
2	12101	5	2	4	1	5	8	7	8	6	...	5	7	6	4	7	4	4	8	
3	12201	18	17	17	4	14	20	26	34	35	...	31	33	31	22	34	23	26	36	
4	12301	9	7	11	5	9	8	9	12	12	...	12	11	7	14	13	13	13	12	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
532	902101	6	7	6	10	8	5	4	4	4	...	8	4	4	4	5	4	4	5	
533	902201	1	1	3	4	5	1	3	2	2	...	1	1	1	3	1	0	3	2	
534	903101	1	1	1	7	3	0	1	2	2	...	0	1	3	3	2	2	0	0	
535	904101	4	2	4	8	8	8	6	7	5	...	7	8	7	4	7	6	7	6	
536	904201	1	0	0	2	2	0	0	0	0	...	0	2	1	1	2	2	0	0	

537 rows × 42 columns

### 3 Test Data 예측 알고리즘 코드

```
for idx in tqdm(range(test2017_csv.shape[0])):
    user=test2017_csv.iloc[idx,0]
    if test2017_csv.iloc[idx,0]==0:
        continue
    user_apititude=[]

    for jdx,j in enumerate(Importance2017):
        if Importance2017.iloc[idx,jdx]>Importance2017_mean[jdx]: # 중요한 업무이고
            if level2017.iloc[idx,jdx]>level2017_mean[jdx]: # 업무 역량이 있으면
                if positivSatisfact2017.iloc[idx,:].mean()!=1 and negativSatisfact2017.iloc[idx,:].mean()!=5:
                    user_apititude.append(Importance2017.columns[jdx])

    if len(user_apititude)==0:
        continue

    user_apititude.append('knowcode')
    max_knowcode=0
    max_freq=0

    calcFrame=test_apititude2017.copy()[user_apititude]
    calcFrame.loc[:,'sum']=calcFrame.sum(axis=1)-calcFrame.loc[:,'knowcode']
    max_freq=calcFrame.iloc[:,-1].max()
    max_row=calcFrame.loc[calcFrame['sum']==max_freq]
    max_knowcode=max_row['knowcode'].tolist()

    test2017_csv.loc[idx,'knowcode']=max_knowcode
```



### 3 결정 트리 알고리즘 스코어

- 터무니 없이 낮은 스코어

```
test2017_csv
```

```
test_aptitude2017.loc[test_aptitude2017['knowcode'] == 212101]
```

wcode	aq1_1	aq2_1	aq3_1	aq4_1	aq5_1	aq6_1	aq7_1	aq8_1	aq9_1	...	aq33_1	aq34_1	aq35_1	aq36_1	aq37_1	aq38_1	aq39_1	aq40_1	aq41_1	sum
212101	66	73	68	55	54	106	91	132	121	...	112	123	154	121	121	99	96	61	54	3567

13 columns

3	4	5	5	6	4	6	3	4	4	...	35	6	화학	1	1	1	4100	NaN	3000	212101.0
4	5	6	4	5	4	5	1	0	1	...	36	4	광고홍보	1	1	1	2800	NaN	2000	212101.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9481	3	4	4	5	4	5	5	6	4	...	52	1	NaN	2	6	NaN	NaN	3400	NaN	212101.0
9482	4	5	4	6	5	6	5	6	5	...	48	5	요업과	1	1	1	7000	NaN	2400	212101.0
9483	3	2	1	0	2	1	3	3	1	...	44	2	인문계	2	6	NaN	NaN	4500	NaN	212101.0
9484	4	5	3	4	3	4	1	0	1	...	44	4	컴퓨터공학	1	1	1	6000	NaN	4000	212101.0
9485	3	4	4	5	4	4	4	5	3	...	42	3	기계	1	1	1	3000	NaN	2000	212101.0

rows × 156 columns

### 3 결정 트리 알고리즘 스코어 낮은 이유

- 원인: 새로 만든 행렬에서 빈도수가 과하게 많은 행이 있으면 그 행으로 모두 채워져서 예측 못함
- 결론: 이론까지는 좋았어도 실제 코드에 적용했을 때 예상치 못한 변수로 인해 사용할 수 없는 알고리즘으로 판명됨
- 대안: 다른 알고리즘을 적용해서 문제를 풀어보자

## 4. User-based Collaborative Filtering로 분석

코사인 유사도 사용

4

## 4 User-based CF(Collaborative Filtering) 알고리즘 도입

- CF의 User-based로 새로운 user A가 입력으로 주어질 때  
A와 설문 응답이 가장 비슷한 user B의 직업코드(target)를 A에게 추천  
+이때 유사도의 기준 – Cosine Similarity

# 4 CF 사용 방법

- User Based Collaborative Filtering(사용자 기반 협업 필터링) 사용

데이터	User	Item	Value
KNOW 직업추천	User	1. 설문에 대한 질문 칼럼 2. Target	설문에 대한 응답
무비렌즈 데이터	User	영화 제목 리스트	영화별 평점

- 공통점: User가 같다
- 차이점: Item이 KNOW는 2개다 / KNOW는 숫자와 텍스트 응답 모두 O
- 결론: KNOW에서 Cosine 유사도를 기반으로 추천해보자

## 4 코사인 유사도란

- 두 벡터간 각도의 코사인 값을 이용하여 측정된 벡터간의 유사한 정도

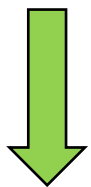
$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$

```
def cos_sim(A, B):  
    return dot(A, B)/(norm(A)*norm(B))
```

## 4 코사인 유사도로 직업을 추천 기본 원리

1. 텍스트를 제외한 한 행을 한 벡터로 하여 두 벡터를 비교하여 유사도 계산 가능

idx	aq1_1	aq1_2	aq2_1	aq2_2	aq3_1	aq3_2	aq4_1	aq4_2	aq5_1	...	bq37	bq38	bq38_1	bq39_1	bq39_2	bq40	bq41_1	bq41_2	bq41_3	know
0	0	3	3	3	3	3	4	4	3	...	52	2	실업	1	1	1	4000		2200	8
1	1	4	5	4	5	3	4	3	4	3	...	38	4	건축공 학	1	1	1		2400	1



두 벡터에 대해 코사인 유사도 계산

```
vec1=vec.iloc[2,:].to_numpy()
vec2=vec.iloc[4,:].to_numpy()

cos_sim(vec1,vec2)
```

0.9683811119902656

## 4 User-based CF로 직업 추천 알고리즘

1. Train 데이터에서 직업 칼럼과 텍스트 제거
2. train 데이터와 test 데이터의 상호 유사도 행렬 생성
3. 상호 유사도 행렬을 기반으로 한 Test의 모든 유저와 Train의 모든 유저의 상호 유사도를 계산
4. 계산한 상호 유사도 행렬을 기반으로 Test User와 가장 유사도가 높은 Train 유저의 직업코드를 추천



## 4 상호 유사도 기록할 행렬 생성

- 로우 인덱스 = Train 데이터의 인덱스
- 칼럼 인덱스 = Test 데이터의 인덱스

```
# 상호 유사도 기록할 행렬 생성 test(칼럼-세로) , train (로우-가로)  
df=pd.DataFrame(index=range(0,9486),columns=range(0,9486))
```

df

+

	0	1	2	3	4	5	6	7	8	9	...	9476	9477	9478	9479	9480	9481	9482	9483	9484	9485
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9481	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9482	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9483	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9484	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9485	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Max Value

## 4 Predict Accuracy – Cosine Similarity (진행중)

```

: from numpy import dot
  from numpy.linalg import norm

def cos_sim(A, B):
    return dot(A, B)/(norm(A)*norm(B))

for i in tqdm(range(df.shape[0])):
    train_user=train2017_csv.iloc[i,:].to_numpy() # train
    for j in range(df.shape[0]):
        test_user=test2017_csv.iloc[j,:].to_numpy()
        df.iloc[i,j]=cos_sim(train_user,test_user)

```

[illegible]



# 5

## 5. 느낀점 및 앞으로의 계획

# 6 이 데이터분석을 통해 얻게 된 점 및 느낀점

안녕하세요 교수님! 저는 AI학과 김지선입니다.

벌써 다시 월요일이 돌아왔는데 교수님께서도 보람찬 한 주 되시길 바라겠습니다.

다름이 아니라 지난번에 코사인 유사도를 알아보라고 알려주셔서 CF까지 보다가 궁금한 점이 생겨 메일을 보냅니다.

제가 데이콘 KNOW 직업 추천 대회 분석에 CF 알고리즘을 써보고 싶어서 찾아보는데 CF가 직업 추천 대회에 사용될 수 헛갈리는 부분이 두 가지가 있어서 여쭙보고 싶습니다.

기본적인 CF에 대한 예제로는 아래 글을 보고 이해했습니다. 아래 글은 무비렌즈 데이터를 CF 알고리즘으로 푼 게시글입니다.

[https://skifree64.github.io/machine\\_learning/2019/11/25/collaborative-filtering.html](https://skifree64.github.io/machine_learning/2019/11/25/collaborative-filtering.html)

저는 CF의 User-based로 새로운 user A가 입력으로 주어질 때 A와 설문 응답이 가장 비슷한 user B의 knowcode(target)

첫 번째는 데이터의 특성이 다른데 같은 알고리즘을 적용해도 되는지 궁금합니다.

위 글에서 item은 영화에 대한 평점으로 특정한 타겟값이 없고 다 feature느낌인데 제가 분석하는 직업추천 데이터는 item

→>>

제목 없음

데이터 처리 방법

1.KNOW에서 한대로 일단 채우는 방식(일괄적 처리)

2.KNOW + fillna 등으로 가능성 높은 값으로 처리 → 함수화 (v)

data:image/png;base64,iVBORw0KGgoAAAANSUUhEUgAAAwUAAAGlCAYAAACr5lq+AAAAAXNSR0IArs4c6QAAAAR





**Thanks!**