

2 Progress

- 파트 2,4 줄이기
- 학습시와 예측시의 모델 구조 슬라이드 추가
- 학습시 예시 십자가 슬라이드 추가
- 학습시 계산 예제랑 예제 데이터프레임 꼭 넣기

0 들어가기 앞서서 대회 진행 관련 안내

- KNOW기반 직업 추천 대회로 진행

KNOW기반 직업 추천 알고리즘 경진대회

고용정보원 | 정형데이터 | 추천

₩ 상금 : 총 1,000만원

🕒 2021.12.06 ~ 2022.01.28 18:00

[+ Google Calendar](#)

👤 873명 📅 D-4

잡케어 추천 알고리즘 경진대회


고용정보원 | 정형데이터 | 추천

₩ 상금 : 총 1,000만원

🕒 2021.12.06 ~ 2022.01.28 18:00

[+ Google Calendar](#)

👤 1,264명 📅 D-4



[KNOW기반 직업 추천 알고리즘 경진대회] 데이터 분석 및 추천 알고리즘 구현

AI학과 2143933 김지선

목차

A table of contents

1 대회 소개

2 데이터 분석

3 핵심 알고리즘

4 데이터 전처리

5 알고리즘 구현



1

1. 대회 소개

대회 소개 및 방법

1 대회 소개 - 대회 일정 및 규칙

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

1 KNOW기반 직업 추천 알고리즘 경진대회

- KNOW 설문조사 - 청소년과 성인의 진로 및 구인, 구직 등에 도움을 주기 위해서 운영하고 있는 조사
- 학습 데이터: 재직자의 직업(knwocode)과 **특정 업무의 중요도, 역량, 만족도, 직무정보**
- 테스트 데이터: **업무의 중요도, 역량, 만족도, 직무정보**로 직업(knowcode)을 예측

2. 데이터 분석

파일 분석 및 데이터 구조



2

2 데이터 분석(메타 데이터)

- 년도 별 설문지와 변수정보, knowcode 숫자 별 직업

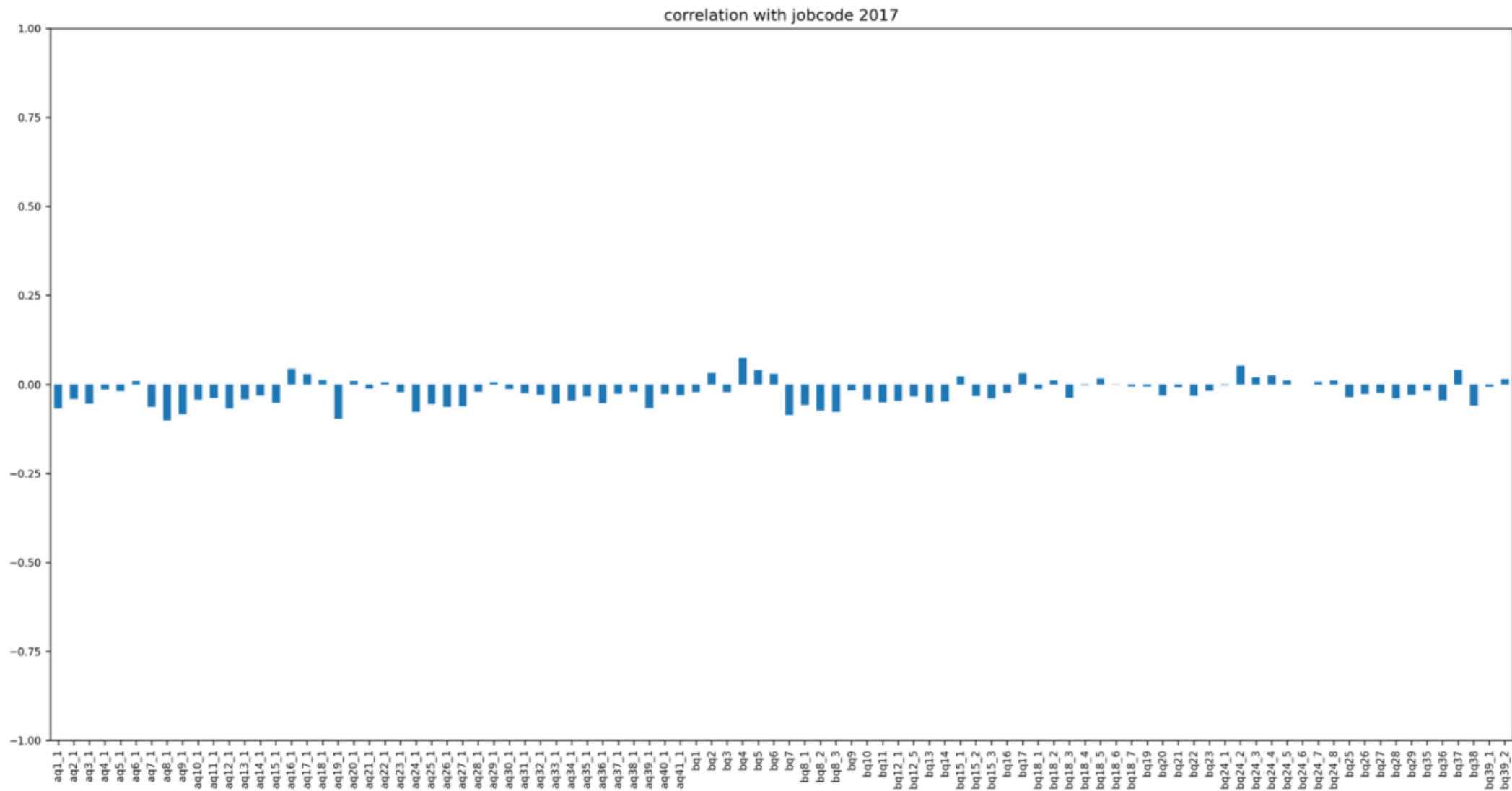
인덱스	파일명	내용
1	2017_KNOW__재직자조사_설문지.pdf	설문지
2	2017_변수값.pdf	변수 값의 뜻
3	2017년_변수정보.pdf	칼럼 뜻
4	2018년_KNOW__재직자조사_설문지.pdf	설문지
5	2018_변수값.pdf	변수 값의 뜻
6	2018년_변수정보.pdf	칼럼 뜻
7	2019년_KNOW__재직자조사_설문지.pdf	설문지
8	2019_변수값.pdf	변수 값의 뜻
9	2019년_변수정보.pdf	칼럼 뜻
10	2020년_KNOW__재직자조사_설문지.pdf	설문지
11	2020_변수값.pdf	변수 값의 뜻
12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

2 데이터 분석 - 파일 분석(학습 및 테스트 데이터)

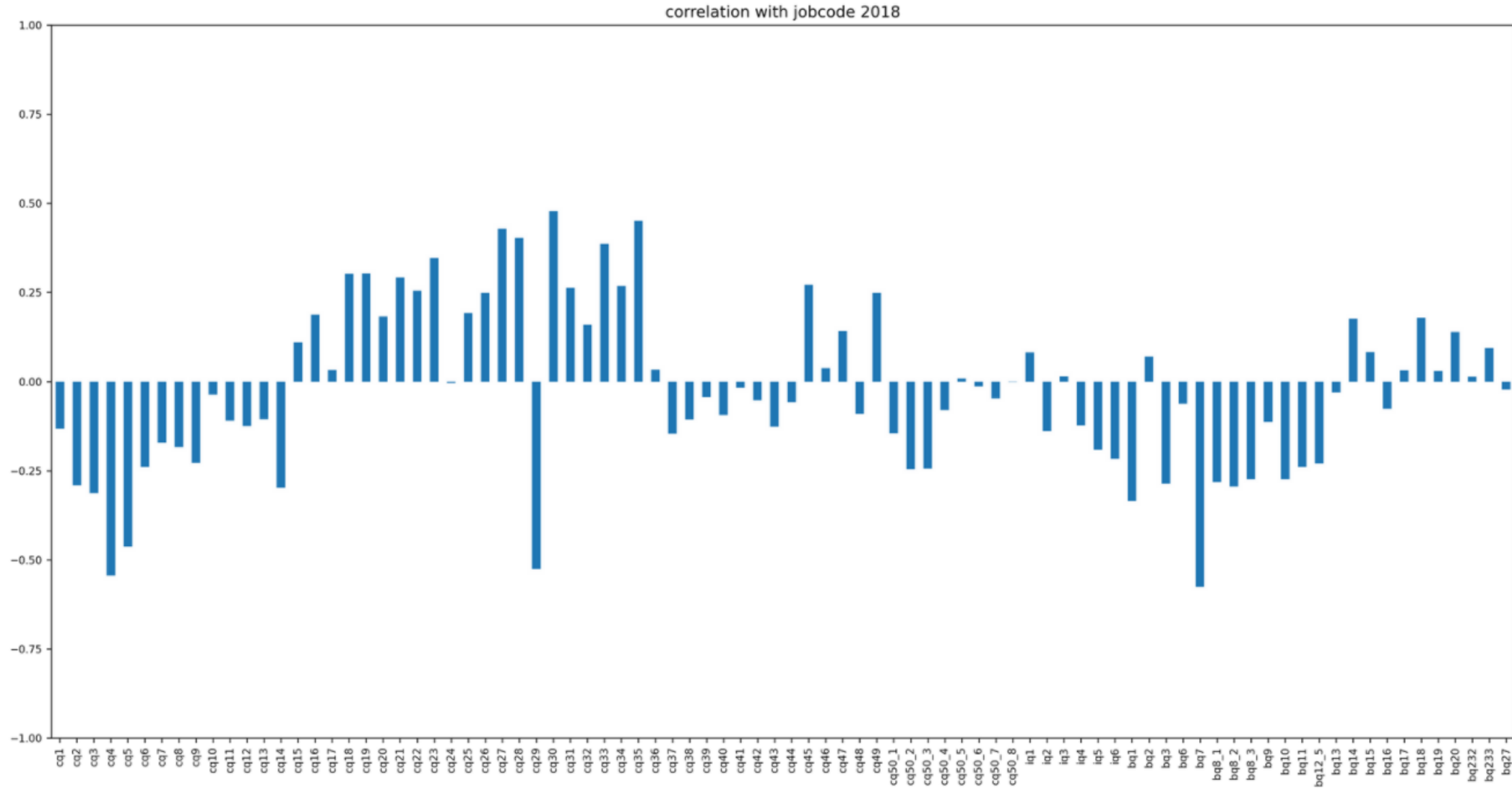
- 4개년의 학습 및 테스트 데이터와 메타데이터 제공 받음

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)

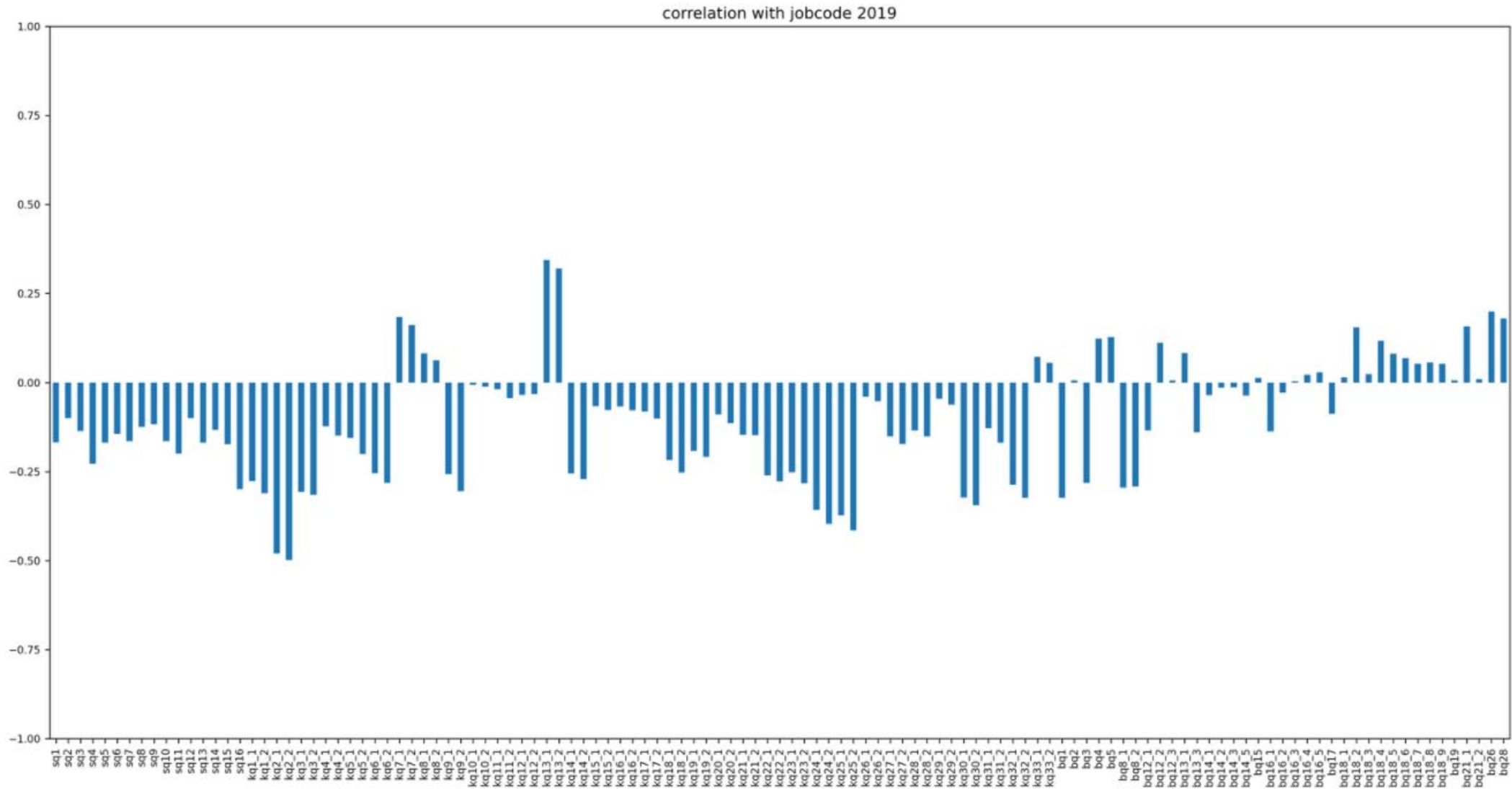
2 4개년 학습 데이터 상관 분석 - 2017년



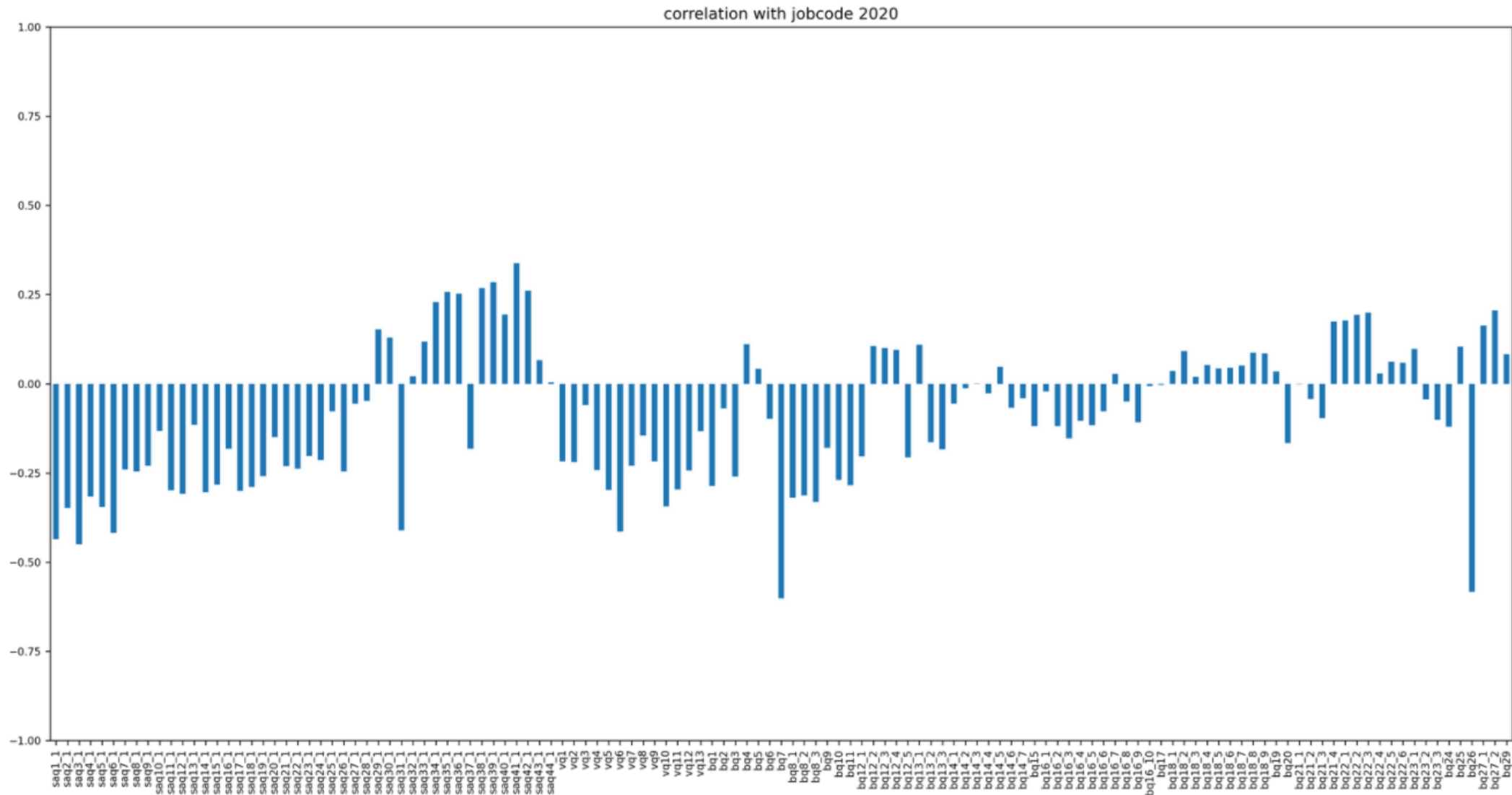
2 4개년 학습 데이터 상관 분석 - 2018년



2 4개년 학습 데이터 상관 분석 - 2019년



2 4개년 학습 데이터 상관 분석 - 2020년



2 데이터 분석: KNOW_2017.csv 칼럼 분석

2017년의 데이터의 shape은 (9486, 156)으로 156개의 칼럼으로 구성되어 있고 크게 아래 5가지 내용을 담고 있다.

- **업무 중요도**
- **업무 수준**
- 직무 조사
- 필요 능력
- **직무 만족도**
- 재직자의 직업(knowcode)

2 특이점을 갖는 칼럼 우선 분석

- 특이점1: 업무 중요도와 업무 역량 질문의 반복 구조 !!
aq1_1 ~ aq41_2(82 / 156)

1 정보 수집 모든 관련 자료에서 정보를 수집, 관찰하기

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

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 **중요합니다**?

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 KNOW_2017.csv 특이점을 갖는 칼럼 우선 분석(실제 데이터)

- 특이점은 실제 데이터에서도 아래 구조로 이루어져 있었다.

_1	업무중요도
_2	업무역량

	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 KNOW_2017.csv 특이점을 갖는 칼럼 우선 분석

- 특이점 2.1번 응답시 다음 업무역량 질문 건너 뛴다.

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

①

약 간
중요하다

②

중요하다

③

아 주
중요하다

④

아주 많이
중요하다

⑤

나. ①번 **【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
가늠한다



①

②

③

④

⑤

⑥

대형 재난이 발생했을 때
도시를 벗어나는데
필요한 시간을 추정한다



전 세계 바다아래의
천연 자원 매장량을
추정한다



⑦
가장 높은 수준

2 데이터 분석: 나머지 칼럼 분석

분류	칼럼명	직업 정보로 활용 가능	질문
직업 정보	bq1	O	1. [산업 유형] 귀하는 어떤 곳(산업)에 근무하고 계십니까?(21개 중 1선택)
요구 자격증 / 활용도구, 프로그램	bq4, bq4_1a, bq4_1b, bq4_1c, bq31	?	4. [요구자격] 귀하의 업무의 수행하는데 요구되는 자격증(국가 공인자격증, 민간 자격증 등)이 있습니까?
직업(일과) 관련	bq30, bq32, bq33, bq34	?	30. [유사직업명] 현장에서 귀하의 직업을 달리 부르는 명칭이 있다면 있는 대로 적어주시기 바랍니다.
만족도 관련	[직무 만족] - bq8_1, bq8_2, bq8_3, bq12_1, bq15_1, bq15_2, bq15_3 [직업 스트레스] - bq18_1, bq18_2, bq18_3, bq18_4, bq18_5, bq18_6, bq18_7	O	12. [직무만족] 귀하가 현재 하고 있는 일에서 다음과 같은 내용에 어느 정도 만족하십니까?
업무 정보	bq2, bq3, bq39_1, bq39_2, bq40, (bq41_1, bq41_2, bq41_3)	?	39. [고용형태] 귀하의 직장에서 고용형태는 다음 중 어디에 해당하니까?
재직자 정보	bq36, bq37, bq38, bq38_1(전공)	?	37. [연령] 귀하의 나이는 만 몇 세입니까?
의견 포함	bq5, bq5_1, bq5_2, bq6, bq7, bq16, bq17, bq35	X	5. [요구훈련] 귀하의 업무를 성공적으로 수행하기 위하여 정규 교육 이외에 업무와 관련한 사외 혹은 사내 훈련이 필요하다고 생각하십니까?
변화 관련	bq13, bq14, bq9, bq10, bq11, bq19~bq29	X	19. [향후 일자리변화] 향후 10년 후 귀하가 종사하는 직업의 일자리는 어떻게 변화할 것이라고 생각하십니까? 20. [직업세계 변화 관련] 귀하의 현재 직업에서 수행하고 있는 업무가 기술적 변화요인(전산화/자동화/인공지능/생명공학) 때문에 어느정도 변화할 것이라고 생각하십니까?
보류	bq30(유사 직업명)		

2 데이터 분석-만족도 칼럼

- 만족도와 불만족도 나눠서 데이터프레임 생성

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	

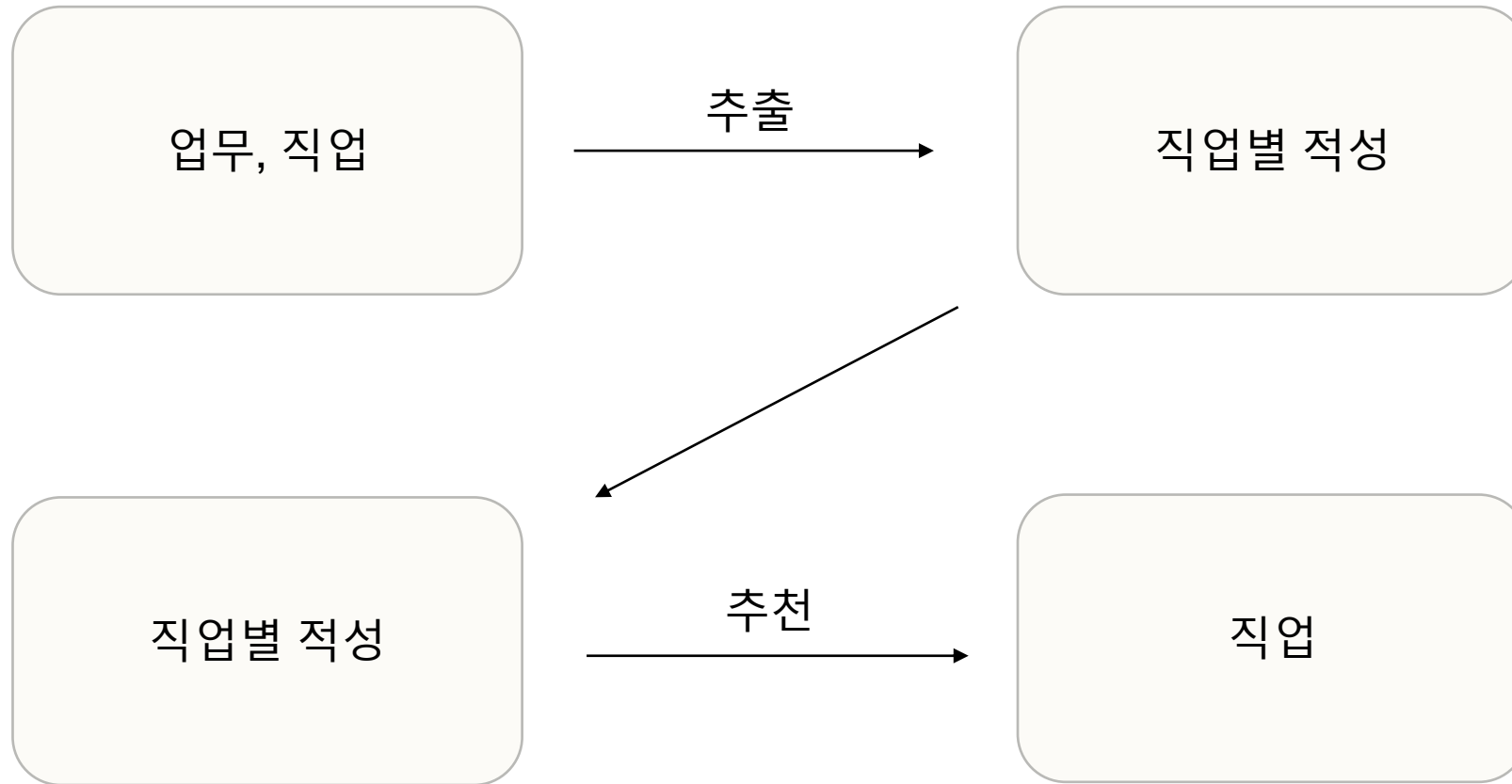


3

핵심 알고리즘

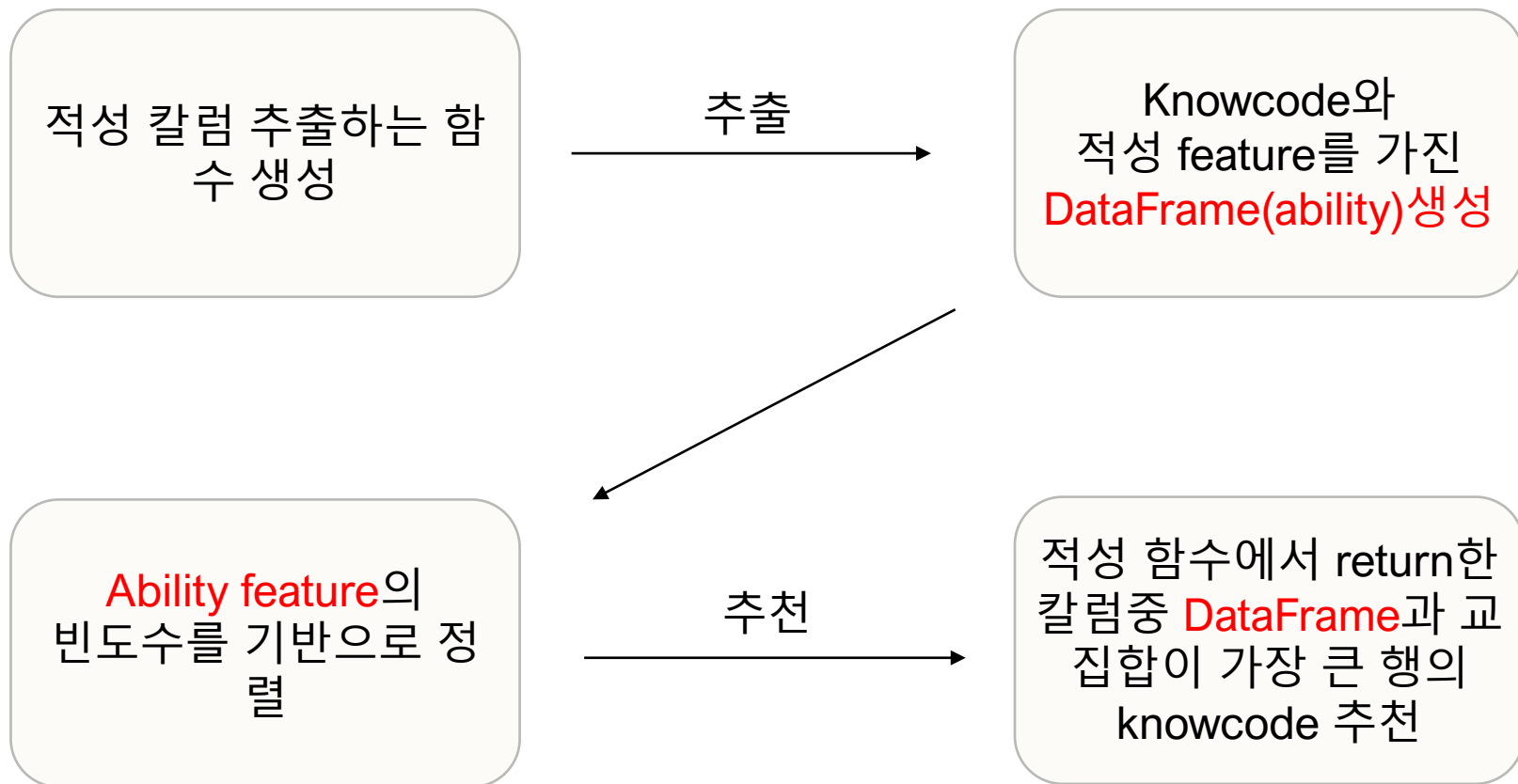
Knowcode와 칼럼이 대응하는 데이터
프레임 생성

3 전체 알고리즘



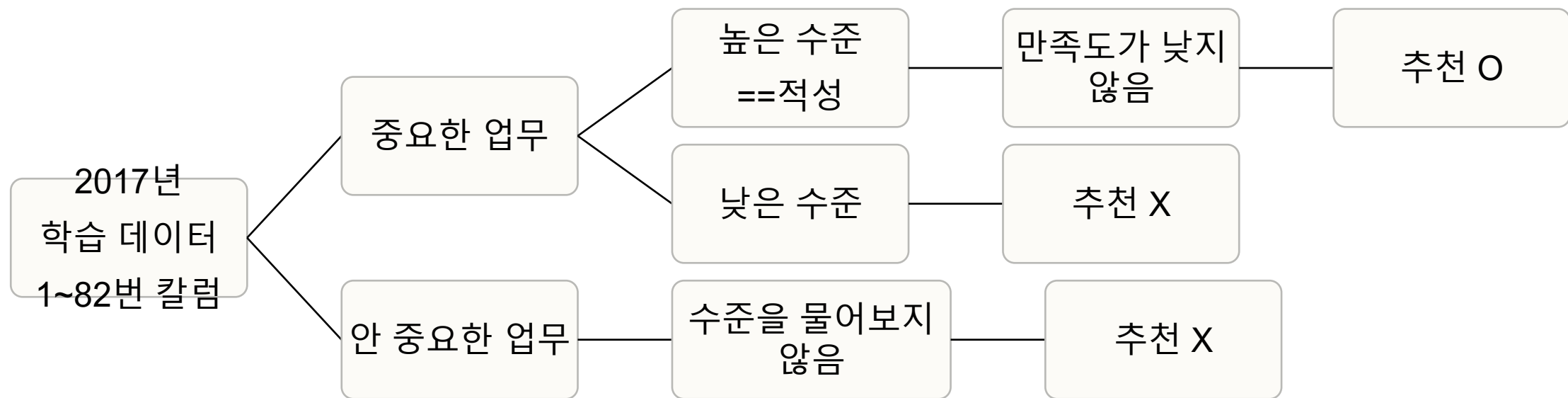
3 전체 알고리즘

- 칼럼이 Knowcode와 1~41번까지 칼럼 첫 번째 질문으로 구성된 데이터프레임 생성



3 핵심 알고리즘, 데이터 프레임에 적성 넣는 알고리즘

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



4. 데이터 전처리

결측치 처리 및 응답 분포에 따른 평균 잡기



4

4 업무 중요도와 업무 역량 칼럼(1~82) 결측치 분석

- 결측치 확인 결과 업무역량 질문에 1번을 선택한 사람들의 수와 다음 업무 수준 질문의 결측치가 동일

idx : 0, 0, 0	
aq1_1 : 0, 0, 0	
aq1_2 : 585, 598, 1183	<code>train2017_csv[(train2017_csv['aq1_2'].isnull()) & (train2017_csv['aq1_1'] == 1)].shape</code>
aq2_1 : 0, 0, 0	(585, 156)
aq2_2 : 861, 862, 1723	
aq3_1 : 0, 0, 0	<code>train2017_csv[(train2017_csv['aq3_2'].isnull()) & (train2017_csv['aq3_1'] == 1)].shape</code>
aq3_2 : 843, 785, 1628	(843, 156)
aq4_1 : 0, 0, 0	
aq4_2 : 3118, 3099, 6217	
aq5_1 : 0, 0, 0	<code>train2017_csv[(train2017_csv['aq5_2'].isnull()) & (train2017_csv['aq5_1'] == 1)].shape</code>
aq5_2 : 2282, 2170, 4452	(2282, 156)
aq6_1 : 0, 0, 0	
aq6_2 : 1676, 1612, 3288	
aq7_1 : 0, 0, 0	<code>train2017_csv[(train2017_csv['aq7_2'].isnull()) & (train2017_csv['aq7_1'] == 1)].shape</code>
aq7_2 : 1537, 1470, 3007	(1537, 156)
aq8_1 : 0, 0, 0	
aq8_2 : 1606, 1546, 3152	
aq9_1 : 0, 0, 0	
aq9_2 : 1396, 1352, 2748	

22

4 업무 중요도와 업무 역량 칼럼(1~82) 결측치 처리

- 정상 결측치: 결측치는 0으로 할당해서 처리
- 자연 결측치: 평균값으로 처리

<Before>		<After>
aq11_2 : 4, 6, 10		aq11_2 : 0, 0, 0
aq12_1 : 0, 0, 0		aq12_1 : 0, 0, 0
aq12_2 : 0, 0, 0		aq12_2 : 0, 0, 0
aq13_1 : 0, 0, 0		aq13_1 : 0, 0, 0
aq13_2 : 0, 0, 0		aq13_2 : 0, 0, 0
aq14_1 : 0, 0, 0		aq14_1 : 0, 0, 0
aq14_2 : 3, 9, 12		aq14_2 : 0, 0, 0
aq15_1 : 0, 0, 0		aq15_1 : 0, 0, 0
aq15_2 : 0, 0, 0		aq15_2 : 0, 0, 0
aq16_1 : 0, 0, 0		aq16_1 : 0, 0, 0
aq16_2 : 0, 0, 0		aq16_2 : 0, 0, 0
aq17_1 : 0, 0, 0		aq17_1 : 0, 0, 0
aq17_2 : 0, 0, 0		aq17_2 : 0, 0, 0
aq18_1 : 0, 0, 0		aq18_1 : 0, 0, 0
aq18_2 : 0, 0, 0		aq18_2 : 0, 0, 0
aq19_1 : 0, 0, 0		aq19_1 : 0, 0, 0
aq19_2 : 0, 0, 0		aq19_2 : 0, 0, 0
aq20_1 : 0, 0, 0		aq20_1 : 0, 0, 0



4 만족도 결측치 처리도 같은 방식으로 진행

- 공백: ' ' -> np.nan 으로 처리
- 나머지: 평균값 할당

```
positivSatisfact2017.isnull().sum()
```

```
bq8_1      0  
bq8_2      0  
bq8_3      0  
bq12_1     0  
bq12_2     0  
bq12_3     0  
bq12_4     0  
bq12_5     0  
bq13       0  
bq14       0  
bq15_1     0  
bq15_2     0  
bq15_3     0  
dtype: int64
```

```
negativSatisfact2017.isnull().sum()
```

```
bq18_1     0  
bq18_2     0  
bq18_3     0  
bq18_4     0  
bq18_5     0  
bq18_6     0  
bq18_7     0  
dtype: int64
```

4 업무중요도와 업무 역량 파일로 분리

- _1과 _2를 기준으로 파일을 분리

```
importance = []  
level = []  
for col in Response2017_csv.columns[1:-1]:  
    try:  
        if(col.split("_")[1]=="1"):  
            importance.append(col)  
        elif(col.split("_")[1]=="2"):  
            level.append(col)  
    except Exception as e:  
        pass
```

```
Importance2017 = train2017_csv[importance]  
level2017 = train2017_csv[level]
```

4 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
0	3	3	3	4	3	3	2	2	2	3	...	1	2	2	3	3	2
1	4	4	3	3	3	1	1	1	1	2	...	1	3	3	1	3	3
2	3	3	3	5	4	1	1	3	3	3	...	1	3	3	3	3	3
3	3	3	3	4	4	3	3	4	5	4	...	4	5	5	4	4	4
4	4	3	3	4	3	1	1	1	1	3	...	1	2	2	1	2	2
...
9481	3	2	3	2	2	2	3	2	2	3	...	2	2	3	3	2	3
9482	5	5	5	3	4	5	4	5	4	4	...	3	4	4	4	4	4
9483	3	4	3	4	4	3	3	1	2	3	...	1	2	2	2	3	2
9484	3	3	4	3	3	4	4	4	4	4	...	2	4	4	4	4	4
9485	3	3	3	3	3	3	3	3	4	3	...	3	3	3	3	3	3

4 업무 역량 파일

<level2017>: 1~7 사이 선택 응답

level2017

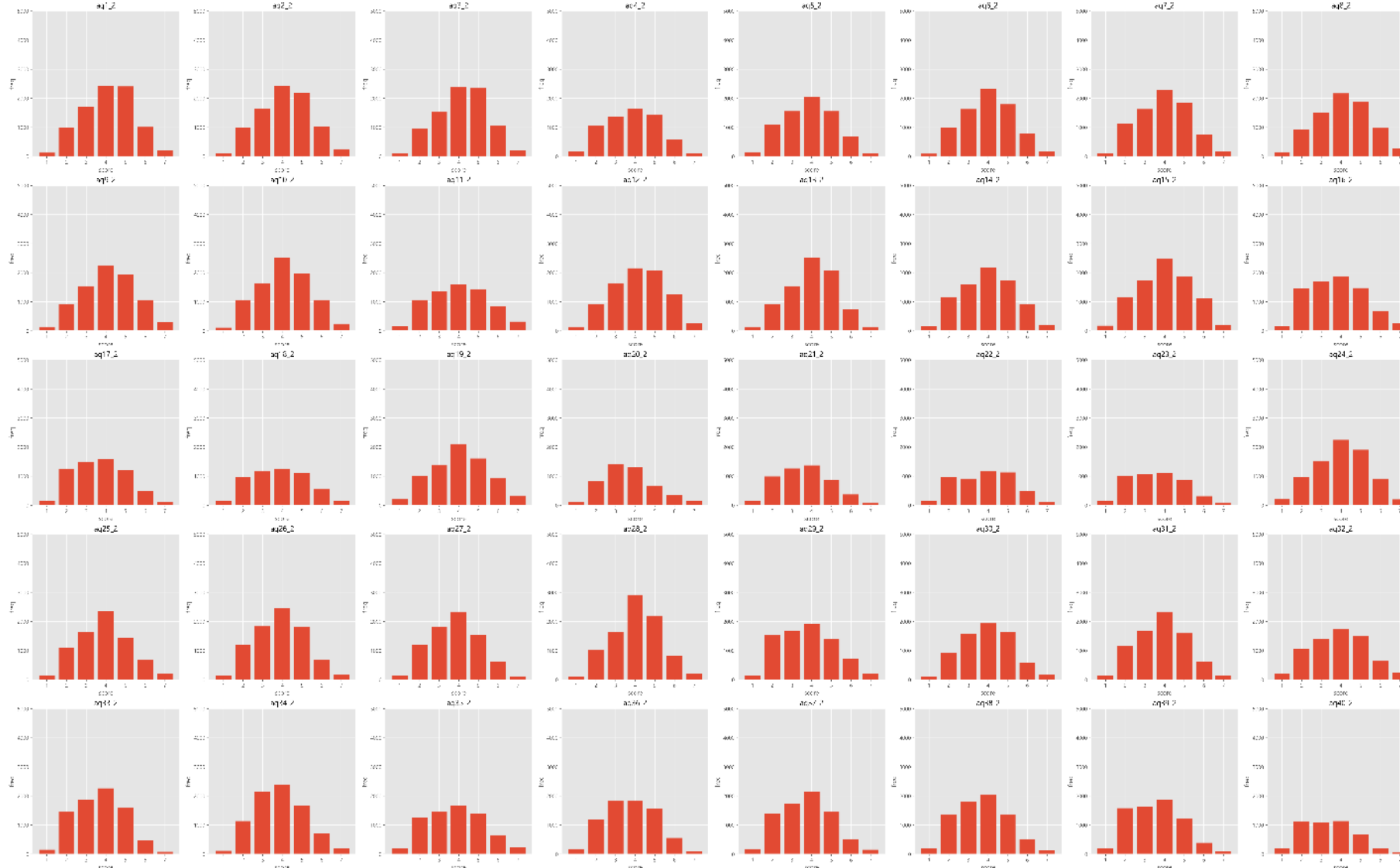
	aq1_2	aq2_2	aq3_2	aq4_2	aq5_2	aq6_2	aq7_2	aq8_2	aq9_2	aq10_2	...	aq31_2	aq32_2	aq33_2	aq34_2	aq35_2	aq36_2	aq37_2	aq3
0	3	3	3	4	4	3	2	2	2	3	...	3	0	2	5	4	4	2	
1	5	5	4	4	4	0	0	0	0	3	...	4	0	4	4	0	4	4	
2	4	4	4	6	5	0	0	4	4	4	...	4	0	4	4	4	4	4	
3	3	3	5	5	6	5	4	5	5	5	...	4	3	5	4	3	4	4	
4	5	4	4	5	4	0	0	0	0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

4 업무중요도 파일 응답 분포



4

업무역량 파일 응답 분포



4 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

5

알고리즘 구현

Knowcode와 칼럼이 대응하는 데이터
프레임 생성

5 빈도수를 나타내는 데이터프레임 생성

- knowcode와 능력 칼럼으로 이루어진 DataFrame 생성
- 이 파일은 knowcode별로 적성인 칼럼의 빈도수를 나타냅니다.
+능력: 2017년도 train데이터 1~82번 칼럼

aptitude2017

	knowcode	aq1_1	aq1_2	aq2_1	aq2_2	aq3_1	aq3_2	aq4_1	aq4_2	aq5_1	...	aq37_1	aq37_2	aq38_1	aq38_2	aq39_1	aq39_2	aq40_1	aq40_2
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
...
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
534	903101	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
536	904201	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0

5 적성 찾는 함수

- 적성 = 중요한 업무 && 잘하는 업무(높은 업무 역량).

```
def get_idx(idx):
    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
    중요X: 비추천
    중요0 역량0 만족도X: 추천
    중요0 역량0 만족도X: 추천
    중요0 역량X : 비추천
    중요0 역량X : 비추천
    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
    중요X: 비추천")
    print("중요X: 비추천")
    return 0
```

업무이고

있으면 |

```
atisfact2017.iloc[idx,:].mean()!=5:
rowcodePer idx,j] +=1
```

5 생성된 데이터 프레임

```
# 만족도 포함 전  
test_apititude2017
```

```
kr # 만족도 조정 후  
# test_apititude2017  
te
```

		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
0		0	11102	12	1	6	0	6	8	13	12	13 ...	10	12	13	8	13	9	7	6
1		1	11201	4	4	3	1	2	3	2	3	3 ...	4	3	3	3	4	2	3	2
2		2	12101	5	2	4	1	5	8	7	8	6 ...	5	7	6	4	7	4	4	8
3		3	12201	18	17	17	4	14	20	26	34	35 ...	31	33	31	22	34	23	26	36
4		4	12301	9	7	11	5	9	8	9	12	12 ...	12	11	7	14	13	13	13	12
...	
532	5	532	902101	6	7	6	10	8	5	4	4	4 ...	8	4	4	4	5	4	4	5
533	5	533	902201	1	1	3	4	5	1	3	2	2 ...	1	1	1	3	1	0	3	2
534	5	534	903101	1	1	1	7	3	0	1	2	2 ...	0	1	3	3	2	2	0	0
535	5	535	904101	4	2	4	8	8	8	6	7	5 ...	7	8	7	4	7	6	7	6
536	5	536	904201	1	0	0	2	2	0	0	0	0 ...	0	2	1	1	2	2	0	0

537 rows × 42 columns

5 DataFrame을 기반으로 Test data의 knowcode 예측 (1)

DataFrame 기반으로 knowcode 예측하는 코드

```
for idx in range(test2017_len):
    user=test2017_csv.iloc[idx,0]
    if test2017_csv.iloc[idx,0]==0:
        continue
    user_apititude=[]
    print(f"###nidx: {idx}")
    print(user)
    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:
                    # 이것은 그냥 user 당
                    user_apititude.append(Importance2017.columns[jdx])
                    #test_apititude2017.loc[test_apititude2017['knowcode']==knowcodePer idx,j]+=1
                    print("중요0 역량0 만족도X: 추천")
                else:
                    print("중요0 역량0 만족도X")
            else:
                print("중요0 역량X : 비추천")
        else:
            print("중요X: 비추천")
```

5 Aptitude를 기반으로 Test data의 knowcode 예측 (2)

```
# 여기서 교집합 큰 knowcode 예측하는 코드 작성
if len(user_ap_titude)==0:
    continue
print(f"{user} {user_ap_titude}")

user_ap_titude.append('knowcode')
for i in range(test_ap_titude2017_len):
    # 1. 모든 열에 대해서 user_ap_titude리스트 열만 출력
    calcFrame=test_ap_titude2017.copy()[user_ap_titude]
    # 2. 모든 행별 로우의 합계 구함
    calcFrame.loc[:, 'sum']=calcFrame.sum(axis=1)-calcFrame.loc[:, 'knowcode']
    calcFrame
    # 3. 가장 빈도가 큰 knowcode 추천
    # max_knowcode와 일치하는 행의 knowcode를 test2017_csv에 타겟(knowcode) 칼럼 만들어서 값 대입
    max_freq=0
    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
```


5 최종 스코어(작업중)

Train	Test	Train Shape
Train2017	Test2017	(9486, 156)
Train2018	Test2018	(9072, 141)
Train2019	Test2019	(8555, 153)
Train2020	Test2020	(8122, 185)
총계	Sample_submission	(35231, 2)

5. 코사인 유사도를 기반 추천

새로운 알고리즘 도입 및 다른 알고리즘으로 추가 분석

5

5 새로운 알고리즘

- 코사인 유사도를 기반으로 추천 (by sklearn cos sim)
- 랜덤 포레스트를 이용하여 추천

5 코사인 유사도란

5 코사인 유사도로 직업을 추천

- 상호 유사도 행렬 이용 !!

5 코드

5 Predict Accuracy – Cosine Similarity

5 랜덤 포레스트

5 랜덤 포레스트로 직업을 추천하는 코드

5 Predict Accuracy – Random Forest



5

6. 한계점 및 기대방안

Knowcode와칼럼이대응하는데이터

6 한계점 및 기대점

- 한계점1: 기존의 딥러닝이나 머신러닝 알고리즘 혹은 수식을 사용하지 않고 자체적으로 생성한 결정트리로 낸 결론이어서 성능이 떨어질 수도 있음
- 한계점2: 데이터 결측치 전처리 방식이 아쉬움
- 기대점: 텍스트로 된 주관식 응답을 자연어 처리해서 생성한 데이터 프레임의 feature로 활용하면 성능이 더 높아질거라 기대됨

6 느낀점

- 특정 데이터 셋에 대한 과적합을 매우 조심해야함
- 알고리즘 이론을 실습으로 옮겼을 때도 같은 결과가 나올 수 있는지 확인해야함

6 Now Progresss, 현재 진도와 이 데이터분석을 통해 얻게된 점

- Pandas(머신러닝 라이브러리) 익숙해지기(v)
- 데이터 분석 경험(v)
- 머신러닝 분류 문제 구조 파악 완료(v)
- 직접 예측 함수 등을 구현해보니까 다른 머신러닝 코드의 함수를 볼 때 쉽게 이해가 감

6 Next Step: 상반기 계획

- 해당 프로젝트 **기술보고서 작성 예정**
- 머신러닝과 딥러닝 이론에 대한 공부
- Pytorch, keras(딥러닝 라이브러리) 익숙해지기
- 논문을 읽으며 이론에 대해 보충
- Kaggle을 계속 진행하며 데이터 분석에 대한 감 키우기
- 알고리즘 하루에 1문제 풀기
- 선형 대수학, 통계 공부

TF-IDF 행렬의 크기는 20,000의 행을 가지고 47,847의 열을 가지는 행렬입니다. 다시 말해 20,000개의 영화를 표현하기 위해서 총 47,847개의 단어가 사용되었음을 의미합니다. 또는 47,847차원의 문서 벡터가 20,000개가 존재한다고도 표현할 수 있을 겁니다. 이제 20,000개의 문서 벡터에 대해서 상호 간의 코사인 유사도를 구합니다.

```
cosine_sim = cosine_similarity(tfidf_matrix, tfidf_matrix)
print('코사인 유사도 연산 결과 : ',cosine_sim.shape)
```

코사인 유사도 연산 결과 : (20000, 20000)

코사인 유사도 연산 결과로는 20,000행 20,000열의 행렬을 얻습니다. 이는 20,000개의 각 문서 벡터(영화 줄거리 벡터)와 자기 자신을 포함한 20,000개의 문서 벡터 간의 유사도가 기록된 행렬입니다. 모든 20,000개 영화의 상호 유사도가 기록되어져 있습니다. 이제 기존 데이터프레임으로부터 영화의 타이틀을 key, 영화의 인덱스를 value로 하는 딕셔너리 title_to_index를 만들어둡니다.

- 벡터 1개==단어 1개
- 유사도를 구할 때는 n개의 데이터가 있으면 nXn 행렬 구조로 만들어서 상호간의 유사도를 구하고 정렬

4

선택한 영화의 제목을 입력하면 코사인 유사도를 통해 가장 overview가 유사한 10개의 영화를 찾아내는 함수를 만듭니다.

```
def get_recommendations(title, cosine_sim=cosine_sim):
    # 선택한 영화의 타이틀로부터 해당 영화의 인덱스를 받아온다.
    idx = title_to_index[title]

    # 해당 영화와 모든 영화와의 유사도를 가져온다.
    sim_scores = list(enumerate(cosine_sim[idx]))

    # 유사도에 따라 영화들을 정렬한다.
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)

    # 가장 유사한 10개의 영화를 받아온다.
    sim_scores = sim_scores[1:11]

    # 가장 유사한 10개의 영화의 인덱스를 얻는다.
    movie_indices = [idx[0] for idx in sim_scores]

    # 가장 유사한 10개의 영화의 제목을 리턴한다.
    return data['title'].iloc[movie_indices]
```

- 위 코드에서 title만 입력 받는 이유는 title로 전체 뽑기 위한 식별자이자 타겟 !!!!
- https://skifree64.github.io/machine_learning/2019/11/25/collaborative-filtering.html
- 위 사이트는 데이터 분석 정리글로 굿 참고



Thanks!