
KNOW기반 직업 추천 알고리즘 경진대회

2조 구현서 김태환 임지인

목차

문제 정의 및 목표

분석 개요

분석 내용

분석 결과 및 개선사항

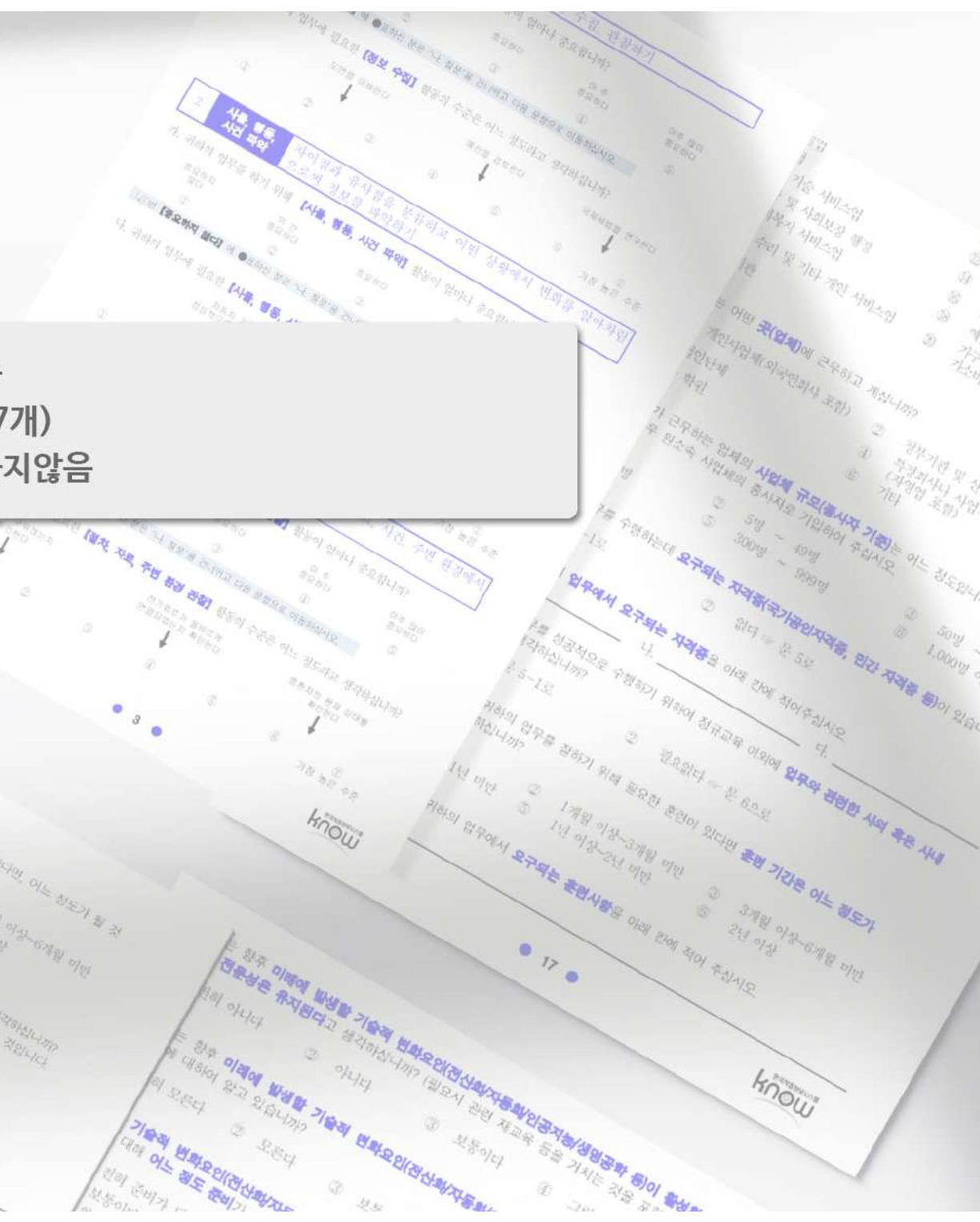
문제 정의

한국고용정보원이 청소년과 성인의 **진로 및 경력설계**,
진로상담, 구인, 구직 등에 도움을 주기 위해서
2001년부터 개발, 운영하고 있는 **조사 데이터**를 기반으로
직업 추천 모델을 만들고 직업과 연관성 높은 직무능력을 탐색 발굴하고자 한다.

문제 정의

데이터 셋 특징

- 설문지 특성상 컬럼 많음
- 직업군이 다양함(약 537개)
- 직업군 별 특징이 명확하지않음



목표

첫번째

KNOW(한국직업정보) 설문 데이터셋을
활용한 직업 추천 알고리즘 개발

두번째

직업과 연관이 높은 설문지 문항 분석 및
영향변수 발굴

세번째

공모전 본선 진출(전체 10위)

분석 개요

STEP 1

전처리

STEP 2

모델링

STEP 3

텍스트 분석
(TF-IDF)

전처리

가설 : 밀린 값은 NaN으로 입력되어있음

근거 : 설문지는 결측값이 있으면 ' '로 입력 됨

밀려적힘
이상한 값이 적힘

특정 직업군에
NaN 값 적힘

객관식 보기가
아닌 다른숫자

중복대답

전처리

■ 밀려적힘 : 나이 column에 대한 답이 성별 column에 밀려 적힘

```
meta2018['bq35'] #성별 질문 보기 1 남성 2 여성
```

```
0      1
1      1
2      1
3      1
4      2
..     ..
9067   1
9068   1
9069   2
9070   1
9071   1
```

```
meta2018['bq35'].unique()
```

```
array([1.0, 2.0, 27.0, 40.0, 31.0, 53.0, 39.0, 26.0, 32.0, 52.0, 50.0,
       42.0, 35.0, 46.0, 37.0, 25.0, 44.0, 47.0, 56.0, '1', '2',
       '라이트룸 등의 컴퓨터 프로그램', '42', '33', '45', '38', '30'], dtype=object)
```

```
meta2018[meta2018['bq35'] == 27.0]
```

bq35	...	bq41_3	...	knowcode
27	...	NaN	...	831101
27	...	NaN	...	821101

전처리

■ NaN 값이 있는 특정 직업군 Column의 Row 찾기

- 2017, 2020 : 없음
- 2018, 2019 : 특정 직업군에 Nan값이 몰림

```
meta2018.loc[[406, 1253, 1548, 1628, 1946, 2938, 3064,  
              3505, 4000, 4076, 4254, 4478, 4718, 4884,  
              5292, 5433, 5574, 5784, 5902, 6094, 6465,  
              6512, 6685, 7480, 7815, 7920, 7992, 8171,  
              8205, 8760, 8783, 8987, 8992, 9016]]['knowcode'].value_counts()
```

Knowcode	숫자	직업군
821101	16	금속가공 제어장치 조작원(용광로·용해로·금속가열로)
831101	14	산업 전기공(항공기·선박·철도기관차·전동차 전기공)
562101	1	계기 점검원 및 가스 점검원
415404	1	시각 디자이너
622304	1	화물차·특수차 운전원
131201	1	통신기기·장비 기술자

전처리

■ 객관식이 아닌 숫자 : 설문지의 정보와 비교

- 질문 48번의 값 : 2보다 큰 값이 있음

■ 확실한 지점부터 밀린 값들을 옮기는 작업

- 질문 48번을 결측치 처리

- 이후 질문 부터 모두 한칸 씩 뒤로 민다

- 사용툴 : 구글 스프레드시트

44. **【저렴한 경쟁】** 귀하는 업무를 수행하면서 동료 혹은 다른 사람들과 얼마나 경쟁을 해야 하나요?

경쟁이 없음 ① 경쟁이 약간 있음 ② 경쟁이 있음 ③ 경쟁이 심함 ④ 경쟁이 매우 심함 ⑤

45. **【장비 속도에 보조 맞추기】** 귀하는 업무를 수행하기 위하여 장비 혹은 기계의 속도에 보조를 맞추는 것이 얼마나 중요하나요?

중요하지 않음 ① 조금 중요함 ② 중요함 ③ 매우 중요함 ④ 극도로 중요함 ⑤

46. **【마감시간】** 귀하는 업무를 수행하면서 얼마나 자주 마감시간을 엄격하게 지켜야 하나요?

전혀 없음 ① 1년에 한 번 이상 그러니 매달 하지 않음 ② 1달에 한 번 이상 그러니 매주 하지 않음 ③ 1주일에 한 번 이상 그러니 매일 하지 않음 ④ 매일 ⑤

47. **【규칙적인 근무】** 귀하는 업무는 근무 일정이 규칙적입니까?

종래한 근무시간의 일정에 따라 근무함 ① 날짜에 따라, 일이 있을 때 따라 근무함 ② 1년 중 특정시기에만 근무함 ③

48. **【재택근무】** 현재 상황에서, 귀하는 업무는 재택근무가 가능하나요?

불가능하다 ① 가능하다 ②

49. **【주말 및 공휴일 근무】** 귀하는 업무를 수행하면서 얼마나 자주 주말 및 공휴일에 출근하십니까?

전혀 없음 ① 1년에 한 번 이상 그러니 매달 하지 않음 ② 1개월에 한 번 이상 그러니 매주 하지 않음 ③ 1개월에 1/2 이상 그러니 매주 하지 않음 ④ 매우 ⑤

50. **【4차산업 도구, 기술 사용】** 귀하는 업무 중 얼마나 다음과 같은 도구나 기술을 사용하십니까?

구분	사용하지 않음	가끔 사용함	자주 사용함
인공지능(프로그램)	①	②	③
클라우드시스템	①	②	③
빅데이터분석	①	②	③
사물인터넷	①	②	③
자율주행차	①	②	③
가상(증강)현실	①	②	③
3D프린터	①	②	③
드론	①	②	③

전처리

중복 대답 : 중복 체크 문항 전처리 (2018년)

중복 체크 문항의 경우, **답변 마다 컬럼이 생성**되기 때문 (예시 : 답변 3개 > 컬럼 3개)

```
def to_other_col(col1,col2,df,number):
    part = df[df[col1] == number]
    wronglist = list(part.index)
    df[col1][wronglist] = 0
    df[col2][wronglist] = number
```

```
to_other_col('bq221','bq222',meta2018, 2)
to_other_col('bq221','bq223',meta2018, 3)
```

```
to_other_col('bq231','bq232',meta2018, 2)
to_other_col('bq231','bq233',meta2018, 3)
to_other_col('bq231','bq234',meta2018, 4)
to_other_col('bq231','bq235',meta2018, 5)
to_other_col('bq232','bq233',meta2018, 3)
to_other_col('bq232','bq234',meta2018, 4)
```

```
to_other_col('bq241','bq242',meta2018, 2)
to_other_col('bq241','bq244',meta2018, 4)
to_other_col('bq241','bq245',meta2018, 5)
to_other_col('bq242','bq243',meta2018, 3)
```

22. [계약 유형] 귀하는 사업주/고객에게서 받는 **보수에 대해 어떤 형태의 세금을 납부**하고 계십니까?

• 2가지 이상 해당할 경우, 모두 말씀해주시기 바랍니다.

- ① 근로소득세 ② 사업소득세 ③ 잘 모름

23. [계약 유형] 귀하의 일자리에서 귀하가 맺은 **계약**은 다음 중 어디에 해당됩니까?

• 2가지 이상 해당할 경우, 모두 말씀해주시기 바랍니다

- ① 근로계약 ② 위임(위탁, 위촉) 계약 ③ 도급계약
④ 잘 모름 ⑤ 해당없음

24. [계약 기간] 귀하가 일자리에서 맺은 **계약기간**은 어떻게 정해져 있습니까?

• 2가지 이상 해당할 경우, 모두 말씀해주시기 바랍니다.

- ① 기간이 정해져 있지 않음 ② 연간(1~3년)단위로 ③ 개월 단위로
④ 물량/과업/프로젝트단위로 ⑤ 잘 모름/해당없음

25. [전속성1] 귀하는 현재 **몇 개의 업체와 근로/위임/도급 계약**이 되어 있습니까?

- ① 1개 ☐ 문 25-1로 ② 2개 ☐ 문 26으로 ③ 3개 이상 ☐ 문 26으로

25-1. [전속성2] 귀하는 **현재 일하는 사업체 외에 다른 사업체와 근로/위임/도급 계약**이 가능합니까?

- ① 예 ② 아니오

26. [사회보험가입여부] 귀하가 **가입되어 있는 사회보험 적용(가입)여부와 비용을 어떻게 부담**하고 있는지 선택해주시기 바랍니다.

(단, 국민연금과 건강보험의 경우 지역가입으로 되어 있는 경우는 미적용으로 체크해주시기 바랍니다.)

사회보험 항목	적용 여부			비용 부담 (적용받는 경우만)		
	적용	미적용	잘 모름	사업자부담	반반부담	본인만 부담
국민연금(직장)	①	②	③	①	②	③

전처리

텍스트 컬럼 지우기

결측치 처리 ' '으로 입력된 값을 np.nan으로 바꿔줌

결측치 0으로 바꾸기

float 형태의 컬럼을 int 로 바꿔줌

#텍스트 컬럼 지우기

```
meta2017 = meta2017.drop(['idx', 'bq4_1a', 'bq4_1b', 'bq4_1c', 'bq5_2', 'bq19_1', 'bq30', 'bq31', 'bq32', 'bq33', 'bq34', 'bq38_1'], axis = 1)
meta2018 = meta2018.drop(['idx', 'bq4_1a', 'bq4_1b', 'bq4_1c', 'bq5_2', 'bq28_1', 'bq29', 'bq30', 'bq31', 'bq32', 'bq33', 'bq37_1', 'bq40'], axis = 1)
meta2019 = meta2019.drop(['idx', 'bq4_1a', 'bq4_1b', 'bq4_1c', 'bq5_2', 'bq18_10', 'bq20_1', 'bq22', 'bq23', 'bq24', 'bq27_1'], axis = 1)
meta2020 = meta2020.drop(['idx', 'bq4_1a', 'bq4_1b', 'bq4_1c', 'bq5_2', 'bq18_10', 'bq18_10', 'bq20_1', 'bq26_1'], axis = 1)
metas = [meta2017, meta2018, meta2019, meta2020]
```

#결측치 처리 ' '으로 입력된걸 진짜 결측인 np.nan으로 바꿔줌

year = 2017

metas = [meta2017, meta2018, meta2019, meta2020]

for meta in metas:

print(year)

for col in list(meta.columns):

for i in range(len(meta)):

if meta[col].iloc[i] == ' ':

meta[col].iloc[i] = np.nan

year += 1

전처리

One Hot Encoding

- 개인의 주관적인 답변이 들어간 값은 제외
- 답이 2개인 것은 컬럼 1개만 남김
- 카테고리로 분류할 수 있는 질문(예: 중졸 이하, 고졸, 대졸, 대학원 이상)은 답변 갯수만큼 컬럼 생성

기준

cat: category 별로 학벌수준

bi: 1이 Yes이고 2가 No인 질문(예: 성별)-> Yes 대한 부분만 0,1로 바꿔서 하나의 컬럼으로 남긴다

bi2: 보기가 3개 이상이지만 binary 형태로 남기고 싶은 질문(예: 1. 없다 2. 적당히 있다 3. 항상 있다) -> 2,3을 있다고 묶어주고 1과 3을 떨군다

bi3: 중복질문 형태라 이전 전처리에서 이미 0과 다른값으로 남겨진 컬럼(예: 0,1 또는 0,2 또는 0,3) -> 0,1의 형태로 바꿔줌

분석 내용

모델링

배경

RandomForest
ExtraTreesClassifier

부스팅

XGBoost
CatBoost

머신러닝

SVM
KNN

모델링

f1 score

ExtraTreesClassifier

0.5673

RandomForest

0.5487

XGBoost

0.3599

CatBoost

0.3714

SVM

0.4025

KNN

0.0452

모델링

01 BayesianOptimization

- 최적의 파라미터 찾기
 - 1) "최적의 값"을 찾아갈 수 있음
 - 2) 상대적으로 시간이 덜 걸림

```
BayesianOptimization(f = et_bo, pbounds = et_parameter_bounds, random_state = 0)
```

Grid Search

시간이 오래걸림

Random Search

하이퍼 파라미터의 범위가 너무 넓으면
일반화된 결과가 나오지 않음
(할 때 마다 결과가 달라짐)

모델링

02 VotingClassifier



```
et = ExtraTreesClassifier(random_state = 30, max_depth = 30, n_estimators = 200)
rf = RandomForestClassifier(random_state = 30, max_depth = 30, n_estimators = 200)
VC = VotingClassifier(estimators=[('rf', rf), ('et', et)], voting = 'soft')
VC.fit(train_x, train_y)
pred_y = VC.predict(test_x)
```

모델링

Rule 기반 텍스트 맞추기

1:1 매칭되는 특정 직업군에서 **자격증**을 가진 **데이터 매칭률 70%** 살리기

```
meta2017[meta2017['bq4_1a'] == '안경사면허증']['knowcode'].value_counts()
```

knowcode	값
307301	5

매칭률
100%

```
meta2017[meta2017['bq4_1a'] == '박사학위']['knowcode'].value_counts()
```

knowcode	값
211101	15
110101	2
110203	1
121103	1
414702	1

매칭률
75% 이상

TF-IDF

여러 개의 자격증을 가지고 있어도
TF-IDF로 **knowcode**를 매칭시킴

STEP 1

유사도 찾는데 방해될 키워드 제거하기

Before

1071	안경사자격증
1253	안경사 면허증
1912	안경사
2055	국가공인안경사
2469	안경사면허증
2886	안경사면허증
4070	안경사 자격증
4168	안경사면허증
4312	안경사면허증
5515	안경사

After

1071	안경사
1253	안경사
1912	안경사
2055	안경사
2469	안경사
2886	안경사
4070	안경사
4168	안경사
4312	안경사
5515	안경사

STEP 2

자격증을 기준으로 유사도를 조사해 KNOWCODE 찾기

TEST 데이터의 자격증 'AAA'가
TRAIN 데이터의 자격증 'AAA'를 포함한 Row 중에서
특정 Knowcode가 전체의 70% 이상을 차지하는 '111111'을 가져옴

TEST 데이터
자격증 'AAA'

TRAIN 데이터
자격증 'AAA', 'AAA BBB', 'CCC AAA DDD' ...

Knocode 111111 : 전체 72%

Knocode 123456 : 전체 15%

Knocode 234555 : 전체 13%

TF-IDF

```
metatest2017[['license']].iloc[341]
```

license 안경사

```
x2017 = tf_idf_license_filter_print(meta2017,
                                     metatest2017[341:342],
                                     'license',
                                     sim_score=0.7,
                                     percent=0.7)
```

	license	knowcode	sim_score	#sim_score = 유사도
1071	안경사	307301	1.0	
1253	안경사	307301	1.0	
1912	안경사	307301	1.0	
2055	안경사	307301	1.0	
	...			
6696	안경사	307301	1.0	
6742	안경사	307301	1.0	
8000	안경사	307301	1.0	
8537	안경사	307301	1.0	

knowcode : 307301, count : 16

매칭률
100%

```
metatest2017[['license']].iloc[1521]
```

license 기계조립

```
x2017 = tf_idf_license_filter_print(meta2017,
                                     metatest2017[1521:1522],
                                     'license',
                                     sim_score=0.7,
                                     percent=0.7)
```

	license	knowcode	sim_score
973	기계조립	817201	1.0
4809	기계조립	816103	1.0
6129	기계조립	817101	1.0
6258	기계조립	214301	1.0

.....

knocode : 817201, count : 4
 knocode : 214301, count : 3
 knocode : 817101, count : 2
 knocode : 816103, count : 2

매칭률
70% 이하

TF-IDF

여러 개의 자격증을 가지고 있어도
TF-IDF로 knowcode를 매칭시킴

STEP 3

자격증을 다른 기준 유사도를 조사해 KNOWCODE 찾기

TEST 데이터의 자격증 'AAA'가
TRAIN 데이터의 자격증 'AAA'를 포함한 Row 중에서
특정 Knowcode가 전체의 70% 이상을 차지하는 '11111'을 가져옴

TEST 데이터
자격증 'AAA'

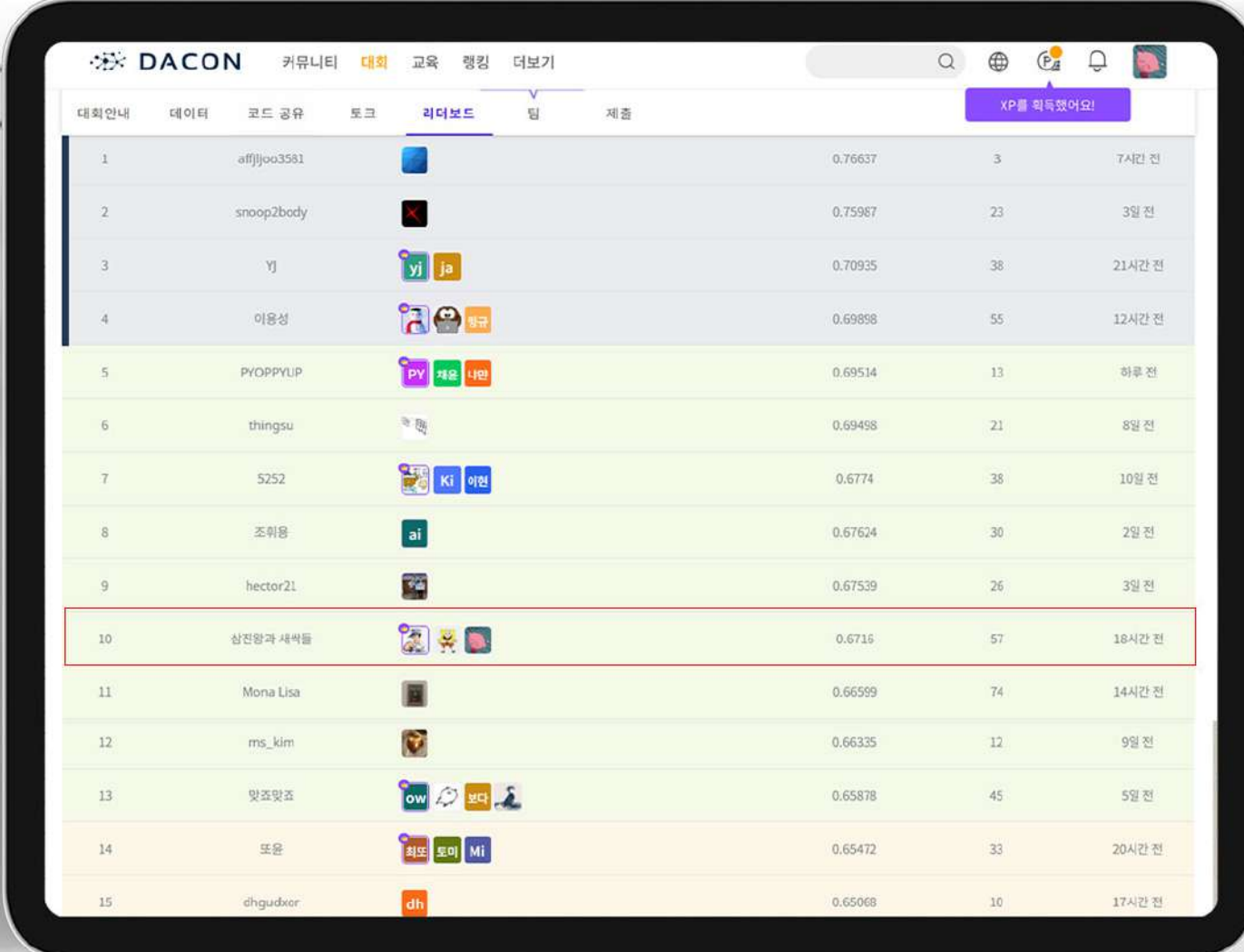
TRAIN 데이터
자격증 'AAA'
자격증 'JJJ KKKK AAA'

유사도 100%
유사도 70% 이하

Knocode 11111 : 전체 72%
Knocode 123456 : 전체 15%
Knocode 234555 : 전체 13%

분석 결과

총 제출 수 : 57회
최종 점수 : 0.6716
등수 : 10등(상위 4%)



DAEON 커뮤니티 대회 교육 랭킹 더보기

대회안내 데이터 코드 공유 토크 리더보드 팀 제출 XP를 획득했어요!

1	affjjo3581		0.76637	3	7시간 전
2	snoop2body		0.75987	23	3일 전
3	yj		0.70935	38	21시간 전
4	이용성		0.69898	55	12시간 전
5	PYOPPYUP		0.69514	13	하루 전
6	thingsu		0.69498	21	8일 전
7	5252		0.6774	38	10일 전
8	조위용		0.67624	30	2일 전
9	hector21		0.67539	26	3일 전
10	삼전왕과 새싹들		0.6716	57	18시간 전
11	Mona Lisa		0.66599	74	14시간 전
12	ms_kim		0.66335	12	9일 전
13	맛조맛조		0.65878	45	5일 전
14	또윤		0.65472	33	20시간 전
15	dhgudxor		0.65068	10	17시간 전

한계점

- | 도메인 지식을 활용한 feature engineering 필요
- | 설문지 특성상 주관이 많이 들어간 row들 판단하기 어려움

개선 사항

- | Text column들을 활용한 딥러닝 모델 추가 활용 필요
- | Tabgan을 활용해 data augmentation 진행 후 모델 학습

감사합니다