

AI 활용 서비스 구현 프로젝트

2팀 손한이 박민욱 양수경 윤정서 전현식 정인교

경북산업직업전문학교

SEMI-PROJECT

— 독자

1

개요

2

팀 구성 및 역할

3

프로젝트 일정

4

실행 결과

5

자체 평가

개요

1

신규 사용자의 직업을 추천하기 위한 AI 설계가 필요

2

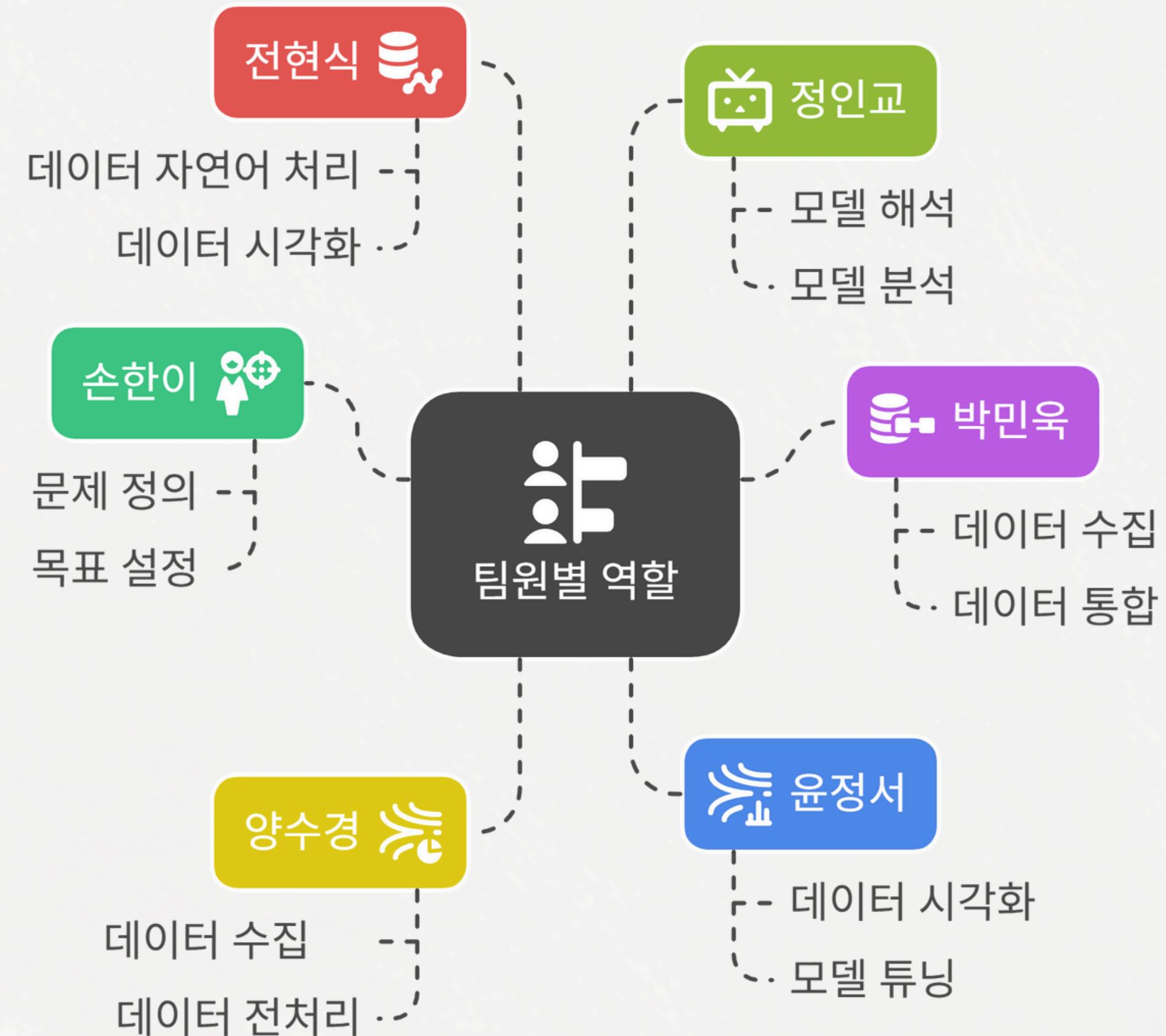
상세한 설계를 위해 자연어 처리 개념을 공부하고 데이터 수집 및 보완

3

사용자 응답에 NCS 직업 정보 데이터를 매칭해보고 개선

팀원별 역할 및 책임

팀 구성원 역할



프로젝트 일정

핵심 시나리오 설정
데이터 수집

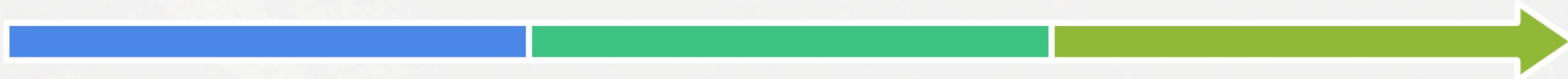
07.01-07.02

데이터 전처리
모델 학습
개선 방향 논의

07.03-07.08

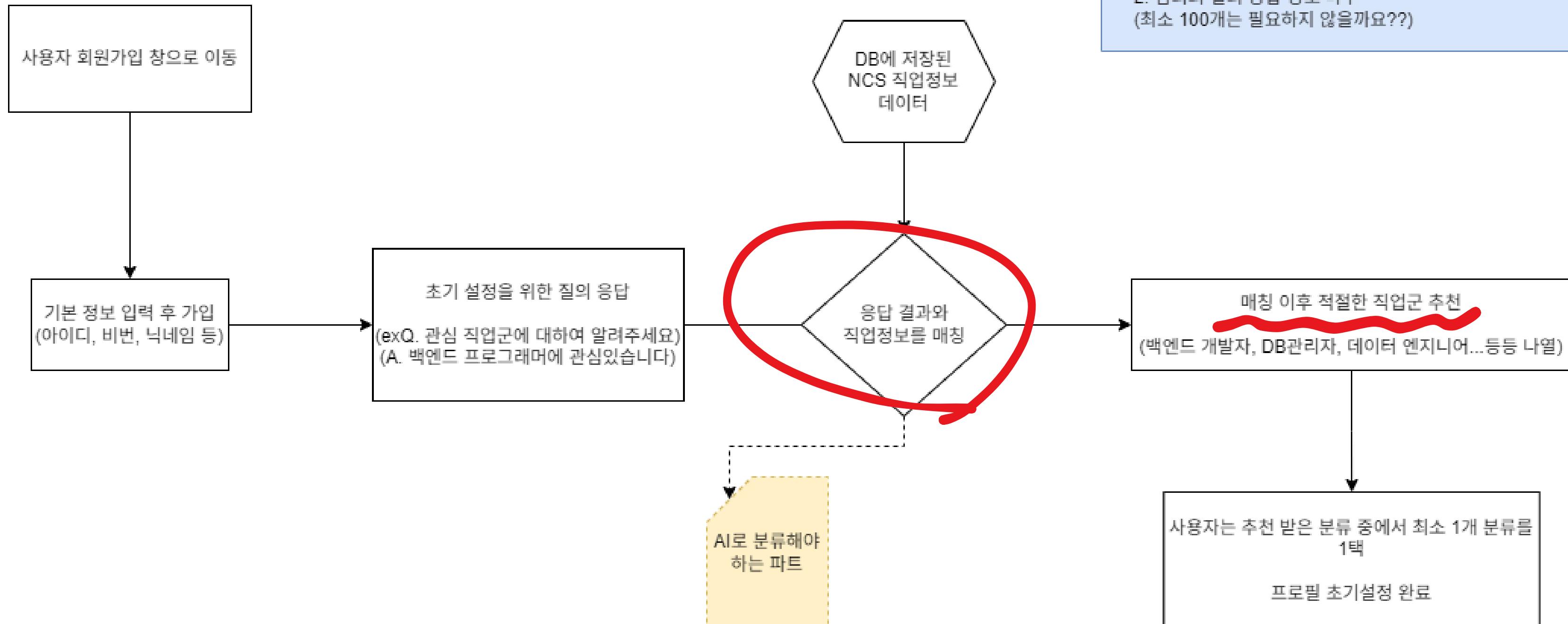
데이터 시각화
프레젠테이션

07.09-07.11



실행 결과

신규 가입자 시나리오



실행 결과

- 데이터 수집 및 전처리

	대분류(24)	중분류(81)	소분류(279)	세분류(1,114)	코드	직업명
0	01. 사업관리	01. 사업관리	01. 프로젝트관리	01. 공적개발원조사사업관리	210.0	정부·공공행정 전문가
1	01. 사업관리	01. 사업관리	01. 프로젝트관리	01. 공적개발원조사사업관리	254.0	국가·지방행정 사무원
2	01. 사업관리	01. 사업관리	01. 프로젝트관리	01. 공적개발원조사사업관리	255.0	공공행정 사무원
3	01. 사업관리	01. 사업관리	01. 프로젝트관리	02. 프로젝트관리	261.0	기획·마케팅 사무원
4	01. 사업관리	01. 사업관리	01. 프로젝트관리	03. 산학협력관리	263.0	총무 사무원 및 대학 행정조교
...
1755	24. 농림어업	04. 수산	04. 어촌개발	01. 어촌체험상품개발	261.0	기획·마케팅 사무원
1756	24. 농림어업	04. 수산	04. 어촌개발	01. 어촌체험상품개발	5211.0	여행상품 개발자
1757	24. 농림어업	04. 수산	04. 어촌개발	02. 어촌체험시설운영	263.0	총무 사무원 및 대학 행정조교
1758	24. 농림어업	04. 수산	04. 어촌개발	03. 어업환경개선	1553.0	환경공학 기술자 및 연구원
1759	:	NaN	NaN	NaN	NaN	NaN

[1760 rows x 6 columns]

- 수집 데이터 출처 - 공공 데이터 포털
(data.go.kr) 등

- NaN에 해당하는 컬럼제거
- 데이터 분석에 중요하지 않은 컬럼 제거



	대분류(24)	중분류(81)	소분류(279)	세분류(1,114)	코드	직업명
0	사업관리	사업관리	프로젝트관리	공적개발원조사사업관리	정부·공공행정 전문가	
1	사업관리	사업관리	프로젝트관리	공적개발원조사사업관리	국가·지방행정 사무원	
2	사업관리	사업관리	프로젝트관리	공적개발원조사사업관리	공공행정 사무원	
3	사업관리	사업관리	프로젝트관리	프로젝트관리	기획·마케팅 사무원	
4	사업관리	사업관리	프로젝트관리	산학협력관리	총무 사무원 및 대학 행정조교	
...
1754	농림어업	수산	수산자원관리	염생산	농림어업 단순 종사원	
1755	농림어업	수산	어촌개발	어촌체험상품개발	기획·마케팅 사무원	
1756	농림어업	수산	어촌개발	어촌체험상품개발	여행상품 개발자	
1757	농림어업	수산	어촌개발	어촌체험시설운영	총무 사무원 및 대학 행정조교	
1758	농림어업	수산	어촌개발	어업환경개선	환경공학 기술자 및 연구원	

[1759 rows x 5 columns]

— 실행 결과 - 모델 학습

```
1 print("Sentence-BERT 모델 로딩 중... 잠시만 기다려 주세요.")
2 # 사용 가능한 유효한 한국어 모델 이들을 사용합니다.
3 embedding_model = SentenceTransformer('jhgan/ko-sroberta-multitask')
4 print("모델 로딩 완료!")
5
6 # NCS 직업군 설명 임베딩 (한 번만 수행)
7 job_descriptions = (df['대분류(24)'] + ' ' + df['중분류(81)'] + ' ' +
8                      df['소분류(279)'] + ' ' + df['세분류(1,114)'] + ' ' + df['직업명']).tolist()
9
10 job_embeddings = embedding_model.encode(job_descriptions, convert_to_tensor=True)
```

- Sentence-BERT 한국어 사전 학습 모델인 'jhgan/ko-sroberta-multitask'를 로딩
- 데이터프레임(df)의 컬럼의 내용을 합쳐 각 직업에 대한 설명 텍스트 리스트를 생성
- 로딩된 Sentence-BERT 모델을 사용하여 임베딩(수치 벡터로 변환)
- 임베딩 결과는 직업 간의 의미적 유사도를 계산하는 데 사용

실행 결과

- 사용자의 정보를 입력
받아 추천직업 정보를
출력해주는 함수

```
1 def recommend_jobs():
2     print("n== NCS 직업군 추천 시스템 ===")
3     skills = input("보유한 지식, 기술, 경험 등을 자유롭게 입력하세요: ").strip()
4
5     if not skills: # 기술/경험 입력은 필수
6         print("당신이 목표로 하고있는 직업 추천이 가능합니다")
7         return
8
9
10    # 입력 정보를 하나의 텍스트로 결합 (임베딩 모델 입력으로 사용)
11    # 이름, 나이, 성별 정보를 포함하면 모델이 불필요한 정보를 학습할 수 있으므로, 여기서는 기술 및 경험에 집중합니다.
12    user_profile_text = f"{skills}"
13    print(f"n입력된 기술 및 경험: {user_profile_text}")
14    print("추천 직업군을 분석 중입니다...")
15
16    # 사용자 프로필 임베딩
17    user_embedding = embedding_model.encode(user_profile_text, convert_to_tensor=True)
18
```

```
20
21    # NCS 직업군 임베딩과의 코사인 유사도 계산
22    cosine_scores = util.cos_sim(user_embedding, job_embeddings)[0]
23
24    # 결과 DataFrame 생성 및 정렬
25    recommendation_df = pd.DataFrame({
26        'job_name': df['직업명'],
27        'description': df['세분류(1,114)'],
28        'similarity_score': cosine_scores.cpu().numpy()
29    })
30    recommendation_df = recommendation_df.sort_values(by='similarity_score', ascending=False).reset_index(drop=True)
31
32    # 추천 결과 출력
33    print("n== 당신을 위한 추천 NCS 직업군 Top 5 ===")
34    for idx, row in recommendation_df.head(5).iterrows():
35        print(f"n{idx+1}. 직업명: {row['job_name']}")
36        print(f"    유사도: {row['similarity_score']:.4f}")
37        print(f"    직업분류: {row['description'][:150]}") # 내용이 길 경우 일부만 출력
38
39    # 사용자에게 다음 단계를 제안
40    print("n추천 결과가 만족스러우신가요? 추가로 궁금한 점이 있거나 다른 시도를 해보고 싶으시다면 알려주세요.")
41
42    return recommendation_df # Return the dataframe
```

```
1 # 메인 실행 부분: 추천 함수 호출 후 시각화 함수 호출
2 if __name__ == "__main__":
3     recommended_df = recommend_jobs() # 직업 추천 함수 호출 및 결과 저장
4     visualize_recommendations_horizontal(recommended_df) # 새로운 시각화 함수 호출
...
***
```

==== NCS 직업군 추천 시스템 ====
보유한 지식, 기술, 경험 등을 자유롭게 입력하세요: [REDACTED]

실행 결과

- 입력한 정보에 맞게 직업을 추천하며 사용자가 쉽게 이해할수 있게 시각화(막대그래프)

1 # 메인 실행 부분: 추천 함수 호출 후 시각화 함수 호출

```
2 if __name__ == "__main__":
3     recommended_df = recommend_jobs() # 직업 추천 함수 호출 및 결과 저장
4     visualize_recommendations_horizontal(recommended_df) # 새로운 시각화 함수 호출
```


== NCS 직업군 추천 시스템 ==

보유한 지식, 기술, 경험 등을 자유롭게 입력하세요: 저는 웹개발자가 되기위해 풀스택 웹개발자 교육과정을 수료하며 5개의 팀프로젝트를 수행한 경험이 있습니다.

입력된 기술 및 경험: 저는 웹개발자가 되기위해 풀스택 웹개발자 교육과정을 수료하며 5개의 팀프로젝트를 수행한 경험이 있습니다.

추천 직업군을 분석 중입니다...

== 당신을 위한 추천 NCS 직업군 Top 5 ==

1. 직업명: 웹 개발자

유사도: 0.5544

직업분류: PaaS엔지니어링

2. 직업명: 웹 개발자

유사도: 0.5303

직업분류: 응용SW엔지니어링

3. 직업명: 웹 개발자

유사도: 0.5274

직업분류: 미러닝콘텐츠개발

4. 직업명: 웹 개발자

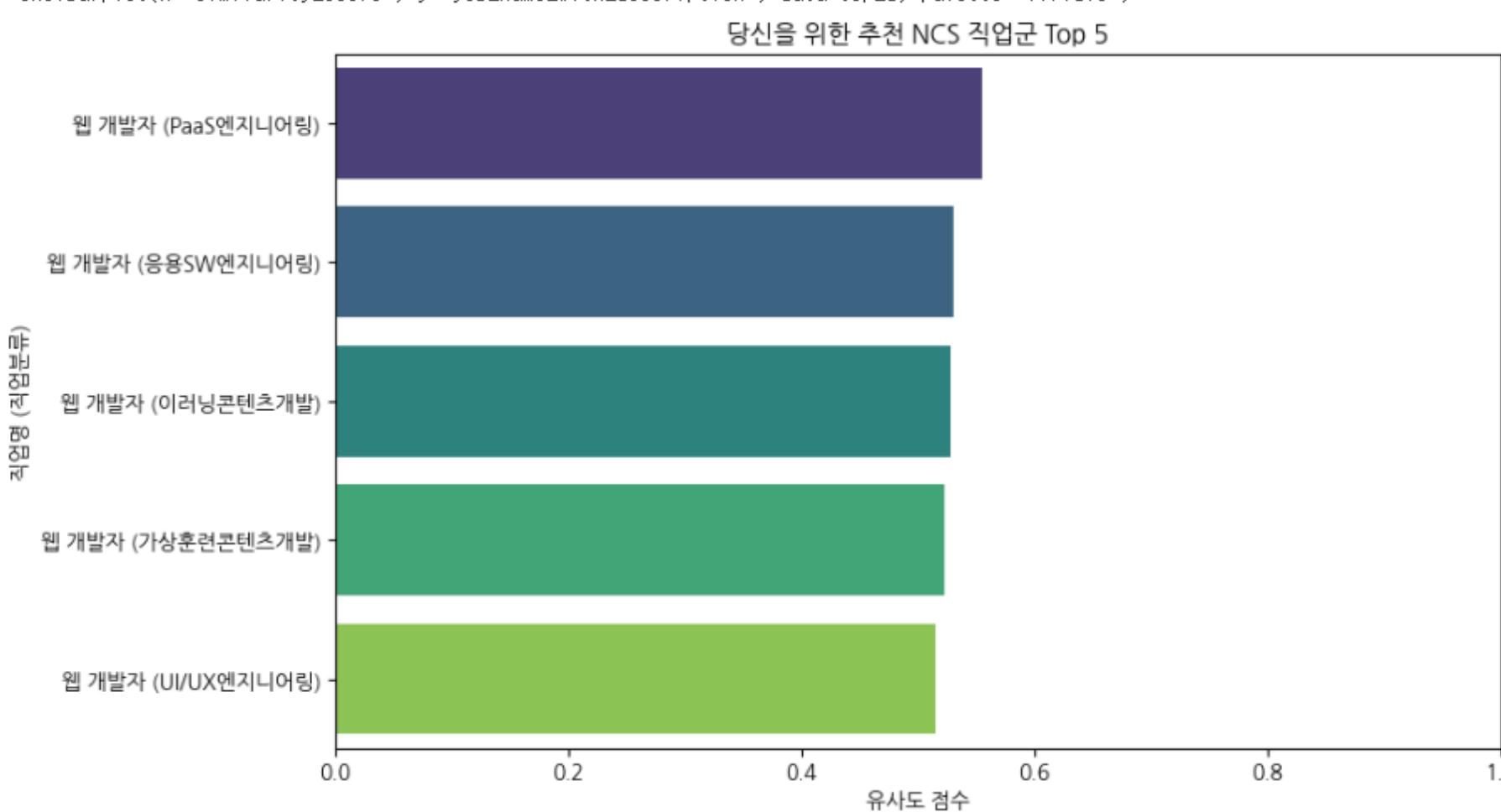
유사도: 0.5229

직업분류: 가상훈련콘텐츠개발

5. 직업명: 웹 개발자

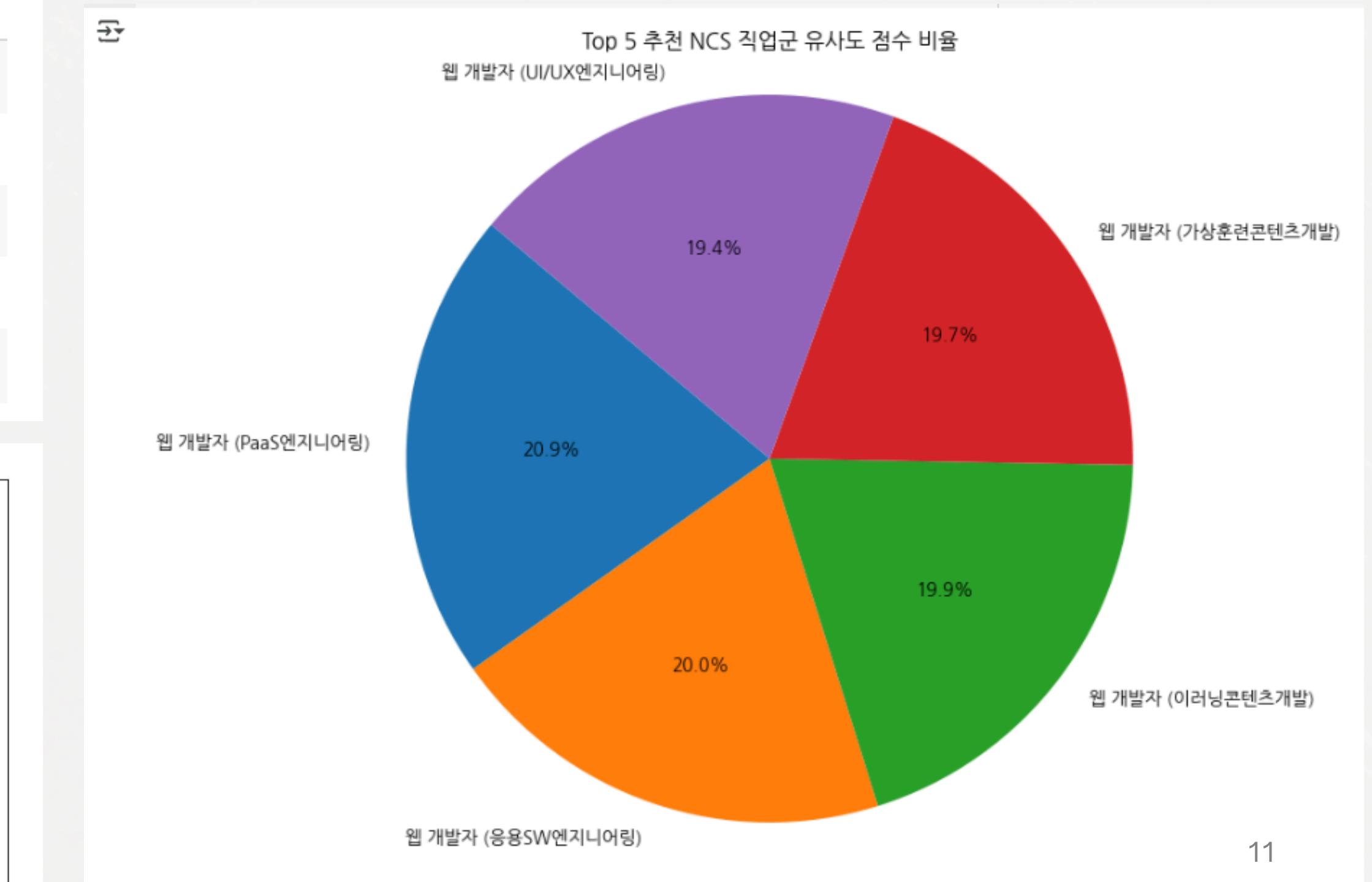
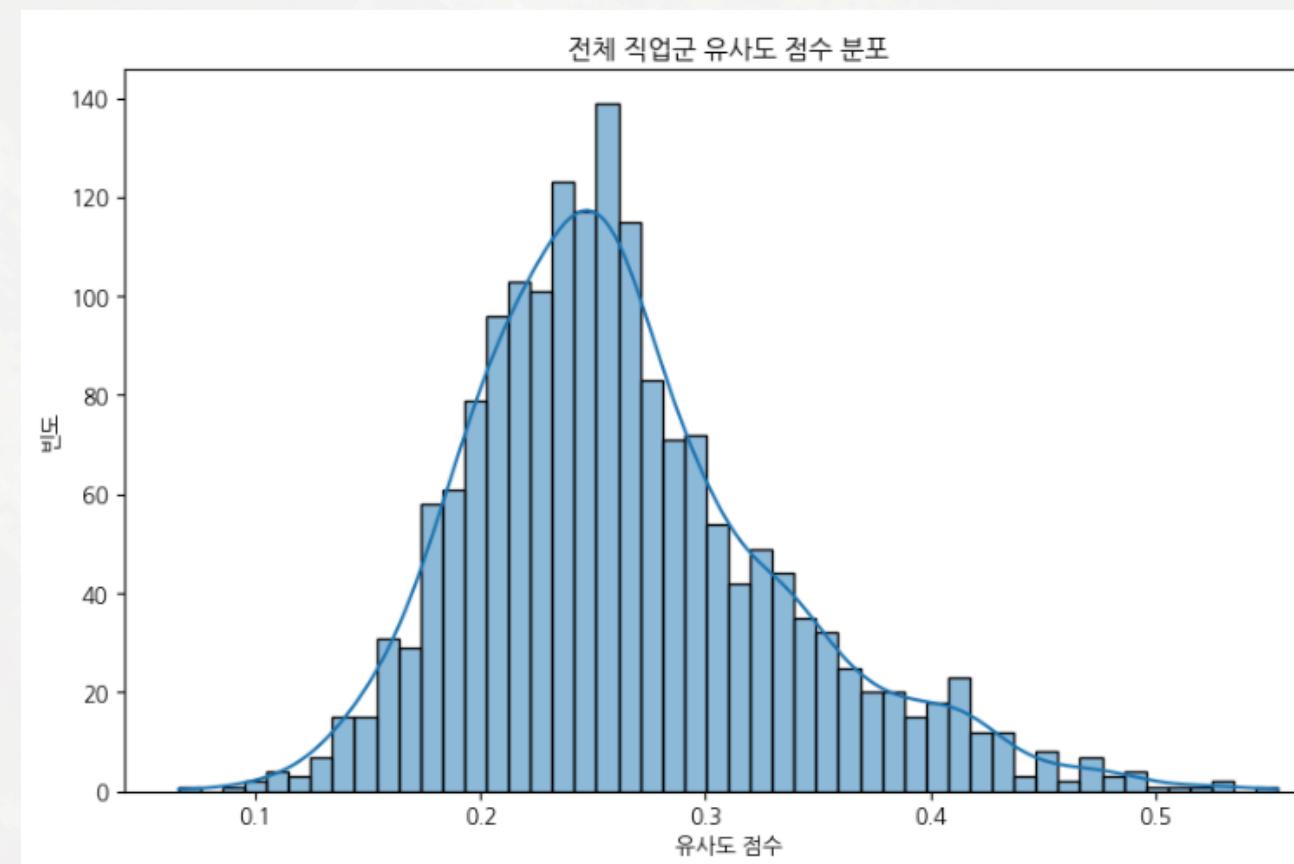
유사도: 0.5149

직업분류: UI/UX엔지니어링



실행 결과 - 시각화(테이블, 히스토그램, 원그래프)

직업명	직업분류	유사도 점수
웹 개발자	PaaS엔지니어링	0.554430
웹 개발자	응용SW엔지니어링	0.530275
웹 개발자	이러닝콘텐츠개발	0.527438
웹 개발자	가상훈련콘텐츠개발	0.522879
웹 개발자	UI/UX엔지니어링	0.514886



— 유사도를 올리는 방법

1

임베딩 모델 변경

2

유사도 측정 방식 변경

3

데이터 전처리 개선

4

프롬프트 엔지니어링(좀 더 명확하게 질문 개선)

실행 결과

- 두 번째 모델 학습

```
1 print("Sentence-BERT 모델 로딩 중... 잠시만 기다려 주세요.")
2 # 사용 가능한 유효한 한국어 모델 이름을 사용합니다.
3 embedding_model = SentenceTransformer('snunlp/KR-SBERT-V40K-klueNLI-augSTS')
4 # embedding_model = SentenceTransformer('jhgan/ko-sroberta-multitask')
5 print("모델 로딩 완료!")
6
7 # NCS 직업군 설명 임베딩 (한 번만 수행)
8 # 세분류 정보를 임베딩 생성
9 df['combined_description'] = df['세분류(1,114)']
10 job_descriptions_combined = df['combined_description'].tolist()
11 job_embeddings = embedding_model.encode(job_descriptions_combined, convert_to_tensor=True)
```

- Sentence-BERT 한국어 사전 학습 모델인 '[snunlp/KR-SBERT-V40K-klueNLI-augSTS](#)'를 로딩
- 로딩된 Sentence-BERT 모델을 사용하여 임베딩(수치 벡터로 변환)
- 임베딩 결과는 세분류로 학습시킨 직업 간의 의미적 유사도를 계산하는 데 사용

실행 결과

- 다른 유사도 측정 방식을

사용한 결과

```
--- NCS 직업군 추천 시스템 ---
이름을 입력하세요: 문정서
나이를 입력하세요: 24
성별을 입력하세요 (남/여): 남
Q1. 당신이 목표로 하고있는 직업군 분야를 작성해주세요.: 경영회계
Q2. 해당 목표를 위해 실행했던 활동(공모전 참여)이나 실적(자격증 보유 여부 등)이 있나요? 펀드투자권유대행인
Q3. 직군을 목표로 하는 이유가 있다면 말씀해주세요. 주식투자에 관심이 많아서

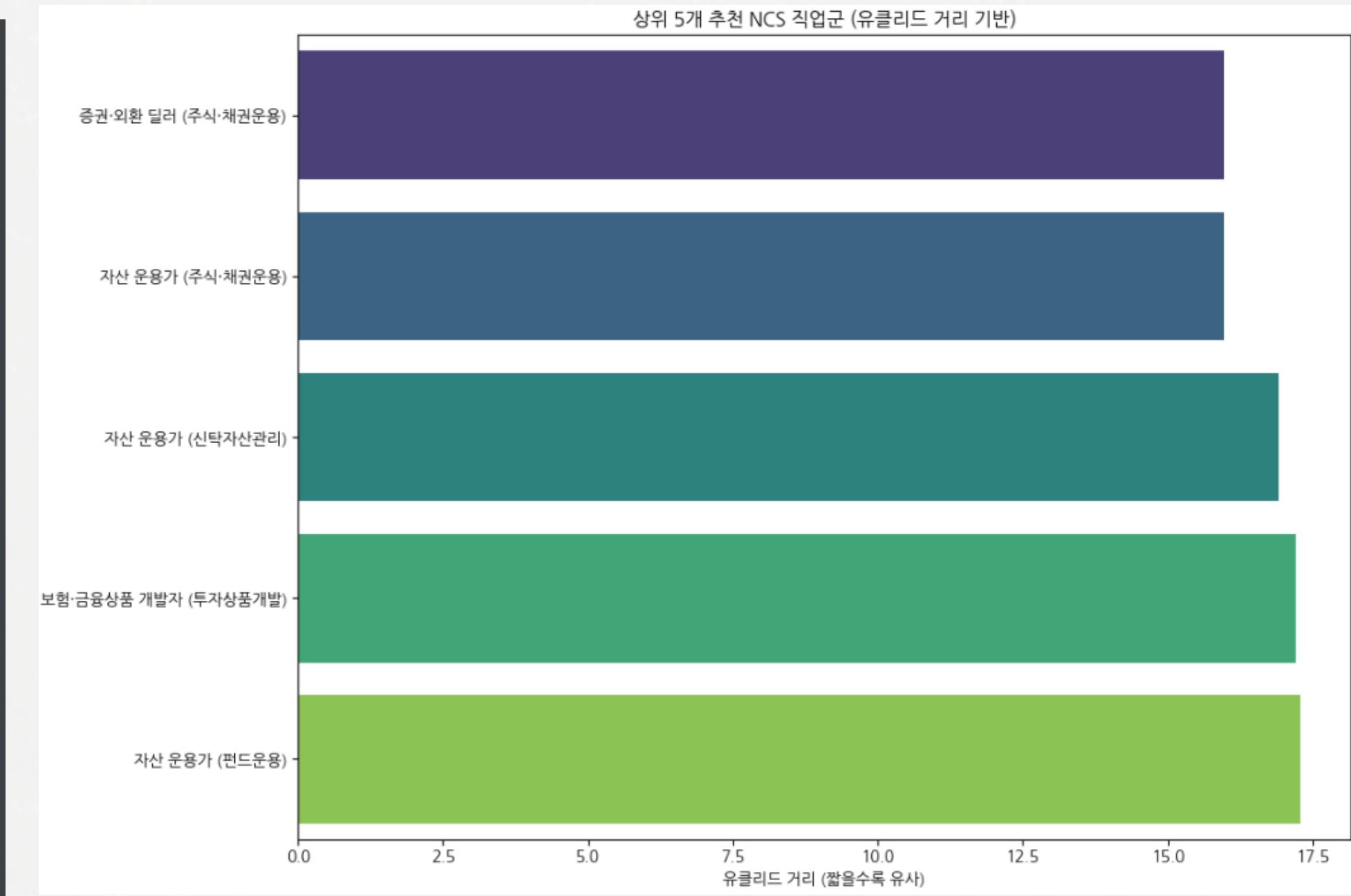
입력된 기술 및 경험: 경영회계 펀드투자권유대행인 주식투자에 관심이 많아서
추천 직업군을 분석 중입니다...

--- 당신을 위한 추천 NCS 직업군 Top 5 ---
1. 직업명: 자산 운용가
   거리: 15.9809
   대분류: 금융
   중분류: 금융
   소분류: 자산운용
   세분류: 주식 · 채권운용

2. 직업명: 증권 · 외환 딜러
   거리: 15.9809
   대분류: 금융 · 보험
   중분류: 금융
   소분류: 자산운용
   세분류: 주식 · 채권운용

3. 직업명: 자산 운용가
   거리: 16.9172
   대분류: 금융 · 보험
   중분류: 금융
   소분류: 자산운용
   세분류: 신탁자산관리
```

```
# NCS 직업군 임베딩과의 유clidean 거리 계산
# 유clidean 거리가 짧을수록 유사도가 높음
# torch.cdist는 (N, D)와 (M, D) 형태의 텐서 간 거리를 계산.
distances = torch.cdist(user_embedding.unsqueeze(0), job_embeddings.unsqueeze(0))[0][0]
```



자체 평가 의견

손한이

자연어 처리와 딥러닝 모델에 관한 지식을 알게 되는 시간이었습니다. 데이터의 학습량과 데이터의 질에 따라 상대적인 유사도가 떨어지더라도 시각 데이터의 정확도가 늘어나는 점을 알게 되었습니다.

박민욱

모델의 정확도를 높이기 위해 단순히 양이 많은 데이터보다, 정제된 양질의 데이터가 훨씬 중요하다는 사실을 알게 되었습니다.

양수경

필요한 정보의 양과 질의 중요함을 느꼈습니다. 다른 분들이 하는 것을 보고 많이 배웠습니다.

프로젝트를 진행하면서, 우리가 원하는 정보를 정확하게 얻기 위해서는 신뢰할 수 있고 적절한 데이터를 기반으로 모델을 학습시키는 것이 중요하다고 느꼈습니다.

윤정서

머신러닝 딥러닝에 대해서 공부하고 불러온 데이터를 학습시켜서 평소에 사용하면 챗봇을 직접 만들수 있다는 점이 신기하고 재미있었습니다.

전현식

막연하게만 느껴졌던 인공지능이라는 주제가 손에 잡히는 기초 지식으로 정리되는 시간이었습니다. 더 깊이 공부해보고 싶은 분야가 생겼습니다.

정인교

QnA

THANK YOU FOR LISTENING