블라인드 기업리뷰 분석을 통한

'네카라쿠배'

기업문화 파헤치기

[박영준, 김재현, 임보라]

목차

IT 최강 라인업

개발자라면 반드시 알아야 할 IT 기업 5

NAVER

kakao

LINE

coupang

배달의민족

'네카라쿠배' 란?

네이버, 카카오, 라인플러스, 쿠팡,
배달의민족의 첫글자를 따서 만든 신조어

MZ세대가 가장 일하고 싶어하는 회사

블라인드 기업리뷰 장단점 분석

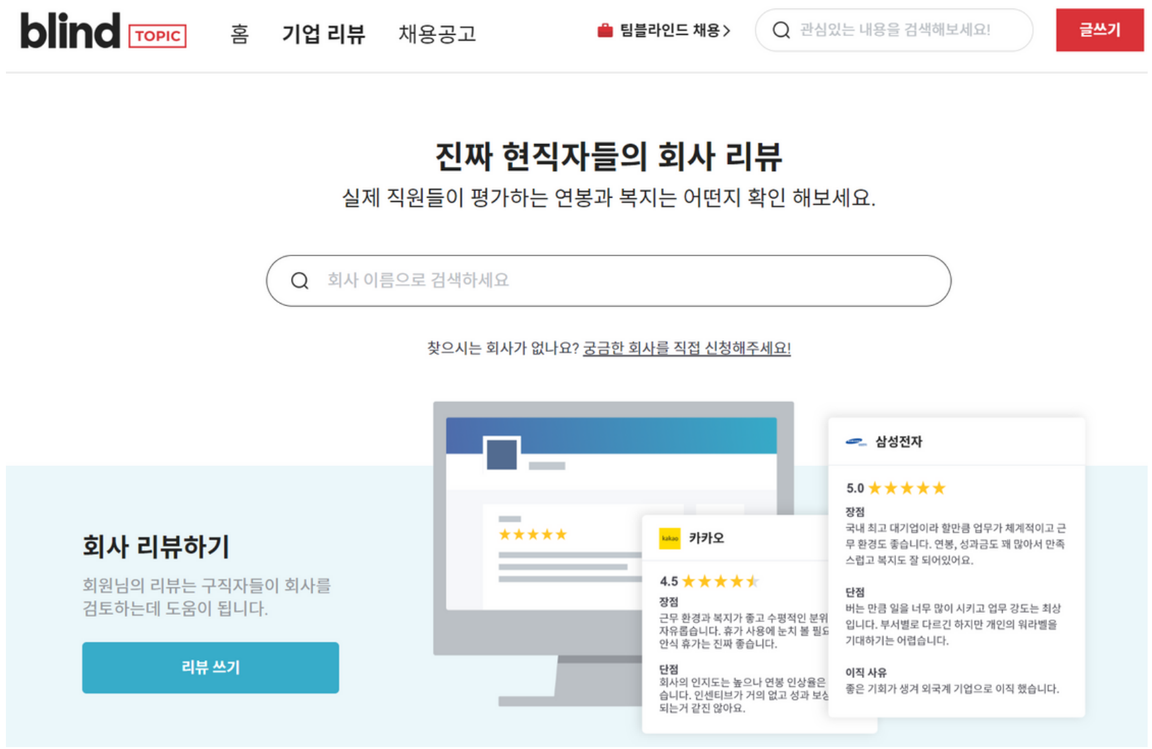
개발자는 어떤 기업에서 일하는 것을 선호할까?

한국 IT업계 선도 기업들의 현직자들의 찐 리뷰분석을
통해 어떤 개발자 문화를 갖추고 있는지 들여다보자!

데이터 설명

01 출처: 블라인드

https://www.teamblind.com/kr/company



인기

“동료들은 좋다 그러나..”

현직원 · P***** · IT 기획·매니지먼트 전문가 - 2022.11.04

장점

재택근무 좋고 수당빵빵하다
재택중에도 열심히 협업해주는 동료들이 든든하다
사내 복지 좋고 자유도가 확보되어 좋음

단점

까라면 까는 문화
일정은 스펙검토 없이 위에서 찍어내린대로 고
여기 아니면 니가 갈데가 어딨냐는 식의 리더들 가스라이팅
라인 줄세우기 문화 최악

“단점이 거의없는회사”

현직원 · h***** · IT 디자이너 - 2022.11.07

장점

커리어향상 좋음
같이일하는 동료들 똑똑함
부바부이긴하지만 우리팀은 야근 많이하는건 지양하는편
연봉 많이올려서 이직함
경영진이 소통을 잘하는편인듯

단점

사소한 물품은 꽁짜로 주면 좋겠음 (ex. 물티슈, 휴지 이런것들...)
법카 사용 범위 넓혀주면 좋겠음

데이터 설명

02 데이터 규모: 총 19,596개의 장단점 리뷰

네이버: 5,226개

카카오: 3,554개

라인플러스: 2,126개

쿠팡: 6,866개

우아한형제들: 1,824개

03 데이터 피쳐 / 타겟

데이터 피쳐: 장단점 리뷰

데이터 타겟: 장점 1, 단점 0

index	company	review	label
0	NAVER	재택근무 좋고 수당빵빵하다재택중에도 열심히 협업해주는 동료들이 든든하다사내 복지 좋고 자유도가 확보되어 좋음	1
1	NAVER	재택근무라는 점과 자율성이 크고 개인 역량을 많이 봐준다는 느낌	1
2	NAVER	- 네임밸류 - 주식보상 등 현금성 복지체계(많진않지만) - 유연한 근무제도	1
3	NAVER	커리어향상 좋음같이일하는 동료들 똑똑함부바부이긴하지만 우리팀은 야근 많이하는건 지양하는편연봉 많이올려서 이직함경영진이 소통을 잘하는편인듯	1
4	NAVER	그래도 수평적인 편.역량을 펼칠 기회도 인프라도 (본인이 잘하면) 있는 편.	1
5	NAVER	남는시간에 뭘하든 신경쓰지 않는다.재택근무 가능 하다.	1
6	NAVER	친절히 알려주는 고수가 많고 세미나가 많고 정보가 많은 회사	1
7	NAVER	나름 자유로운 분위기 / 저년차 주니어에게는 성장하기 좋음	1
8	NAVER	자유 분방했고 우리 팀원분들 분위기도 정말 좋았음 꼰대 문화는 진짜 없었던거 같고 수평적이고 완벽한 자율출퇴근제였음	1
9	NAVER	커리어적인 측면과 업무과제들이 개인적인 성장에도 도움이됨	1
10	NAVER	자율출퇴근개발자 우선 문화(팀바팀)성장을 염두에 둔 좋은 동료들	1

분석과정

1단계 [수집] | 2단계 [전처리] | 3단계 [모델학습] | 4단계 [성능개선] | 결과

(1) url가져오기: requests.get(url) + 원하는 정보 가져오기: beautifulSoup

1차 시도

response [200]을 확인했으나, 태그 정보를 가져올 수 없음

2차 시도

But, 우리가 필요한 장단점은 가져올 수 없음

회사 리뷰를 작성하고 전체 리뷰를 확인하세요!</p>
회사 리뷰를 작성하고 전체 리뷰를 확인하세요!</p>
회사 리뷰를 작성하고 전체 리뷰를 확인하세요!</p>

해결책

requests header 중,
1순위) user-agent, referer 정보 입력

⇒ 리뷰의 제목 태그 가져오기 성공

해결책

requests header 중,
2순위) cookies 정보 입력

⇒ 장단점 리뷰 태그 가져오기 성공

분석과정

(2) url가져오기: requests.get(url) + 원하는 정보 가져오기: 정규표현식 이용

3차 시도

장점과 단점이 나눠져 있으나,
태그는 동일 → 장단점 분리 필요

```
<p><strong class="abt">장점</strong><span>재택근무 좋고 수당빵빵하다<br/>재택중에도 열심히 협업해주는 동료들이 든든하다<br/>사내 복지 좋고 자유도가 확보되어 좋음</span></p>
<p><strong class="abt">단점</strong><span>까라면 까는 문화<br/>일정은 스펙검토 없이 위에서 찍어내린대로 고<br/>여기 아니면 니가 갈데가 어딨냐는 식의 리더들 가스라이팅<br/>라인 줄세우기 문화 최악<br/><br/></span></p>
```

해결책

정규표현식 이용
장점 = resp.text에서 '장점'으로 시작하고 '</p>'로 끝나는 그 사이의 모든 문자 가져오기
단점 = resp.text에서 '단점'으로 시작하고 '</p>'로 끝나는 그 사이의 모든 문자 가져오기

⇒ 장단점 분리 성공

분석과정

1단계 [수집] | 2단계 [전처리] | 3단계 [모델학습] | 4단계 [성능개선] | 결과

2단계: 전처리

01

라벨링 해서 데이터프레임 만들기

나눠진 장단점 리뷰를 "장점은 1, 단점은 0"으로 라벨링 후 분석이 용이하도록 데이터 프레임으로 만들기

02

토큰화 + 품타 태깅

5개의 한글 형태소 분석기 중 Okt 객체를 활용해서 형태소 토큰화 및 품사 태깅

03

단어 빈도수 카운트

불용어 사전 만들기 전 토큰화된 단어를 Counter 함수를 활용해 빈도수 확인

분석과정

1단계 [수집] | 2단계 [전처리] | 3단계 [모델학습] | 4단계 [성능개선] | 결과

2단계: 전처리

04

단어 빈도수를 바탕으로 워드클라우드 생성

네이버 장점과 단점 비교

<장점>

<단점>

분석과정

1단계 [수집] | 2단계 [전처리] | 3단계 [모델학습] | 4단계 [성능개선] | 결과

2단계: 전처리

05 N-gram 사용

네이버 장점 기준 unigram - bigram - trigram 비교

분석과정

2단계: 전처리

06

단어 빈도수를 바탕으로 불용어 사전 만들기

빈도수 상위의 단어들 중 조사나 접속사 뿐만 아니라 명사, 동사, 형용사와 같은 단어들 중에서도 불용어로 제거하고 싶은 단어들을 수집하여 **불용어 사전**에 저장

07

노이즈 & 불용어 제거

단어의 글자수가 1개 이하이거나 불용어 사전에 해당된다면 제거

08

명사, 동사, 형용사 추출

품사 중 명사, 동사, 형용사에 해당되는 단어들만 추출

09

TF-IDF 방법으로 벡터화

단어들마다 중요한 정도를 가중치로 부여

분석과정

1단계 [수집] | 2단계 [전처리] | 3단계 [모델학습] | 4단계 [성능개선] | 결과

3단계: 모델 학습

1차 시도 [데이터 비율 7:3]

- Train set: 네이버 리뷰 5,226개
- Test set: 라인플러스리뷰 2,126개
- 불용어 x , bigram 사용
- 로지스틱 c = 1 기본값
- 랜덤포레스트 나무100개 , 깊이 30단

결과

구분	Train score	Test score
로지스틱 리그레션	0.946	0.910
랜덤 포레스트	0.940	0.869

2차 시도 [데이터 비율 8:2]

- Train set: 네이버, 카카오, 쿠팡 총 15,646개
- Test set: 라인플러스, 우아한형제들 총 3,950개
- 불용어 x , bigram 사용
- 로지스틱 c = 1 기본값
- 랜덤포레스트 나무100개 , 깊이 30단

결과

구분	Train score	Test score
로지스틱 리그레션	0.927	0.902
랜덤 포레스트	0.915	0.854

분석과정

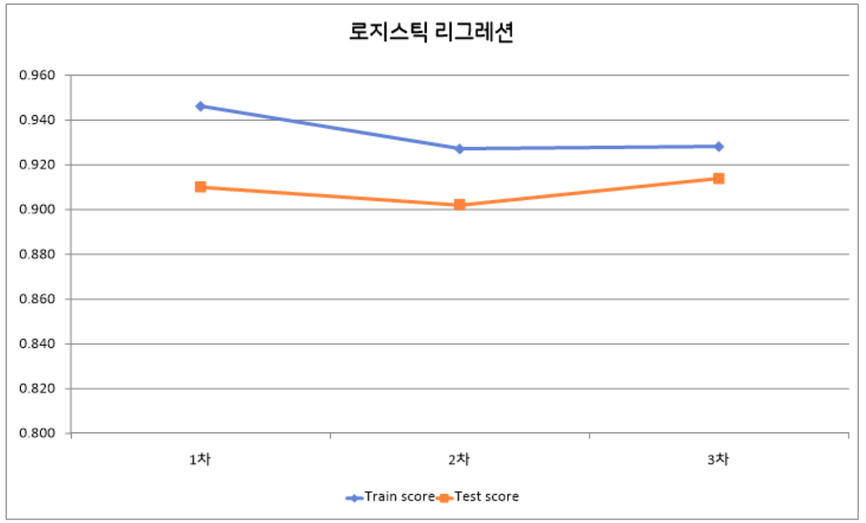
3단계: 모델 학습

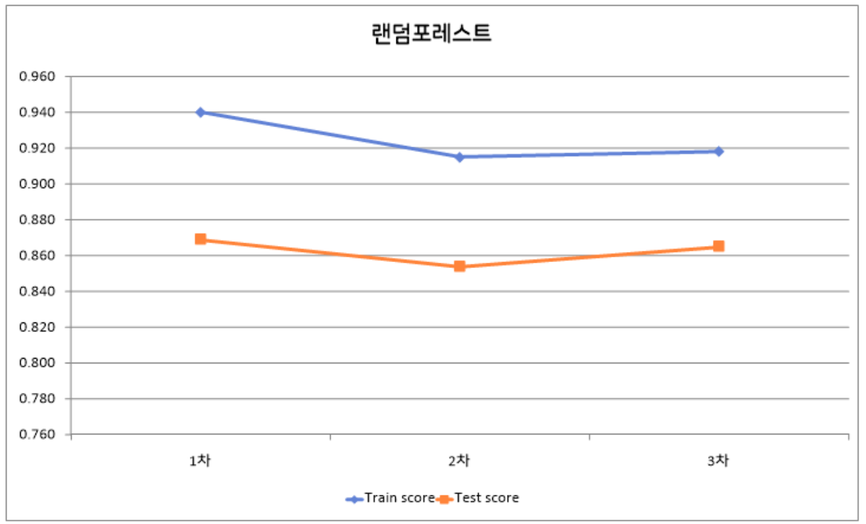
3차 시도 [데이터 비율 8:2]

- train_test_split 라이브러리 활용
- target의 비율을 유지하면서 8:2의 비율로 train, test 데이터를 나눠서 시도
- 불용어 x , bigram 사용
- 로지스틱 c = 1 기본값
- 랜덤포레스트 나무100개 , 깊이 30단

결과

구분	Train score	Test score
로지스틱 리그레션	0.928	0.914
랜덤 포레스트	0.918	0.865





분석과정

1단계 [수집] | 2단계 [전처리] | 3단계 [모델학습] | 4단계 [성능개선] | 결과

3단계: 모델 학습

4차 시도 [불용어 변화에 따른 성능 비교]

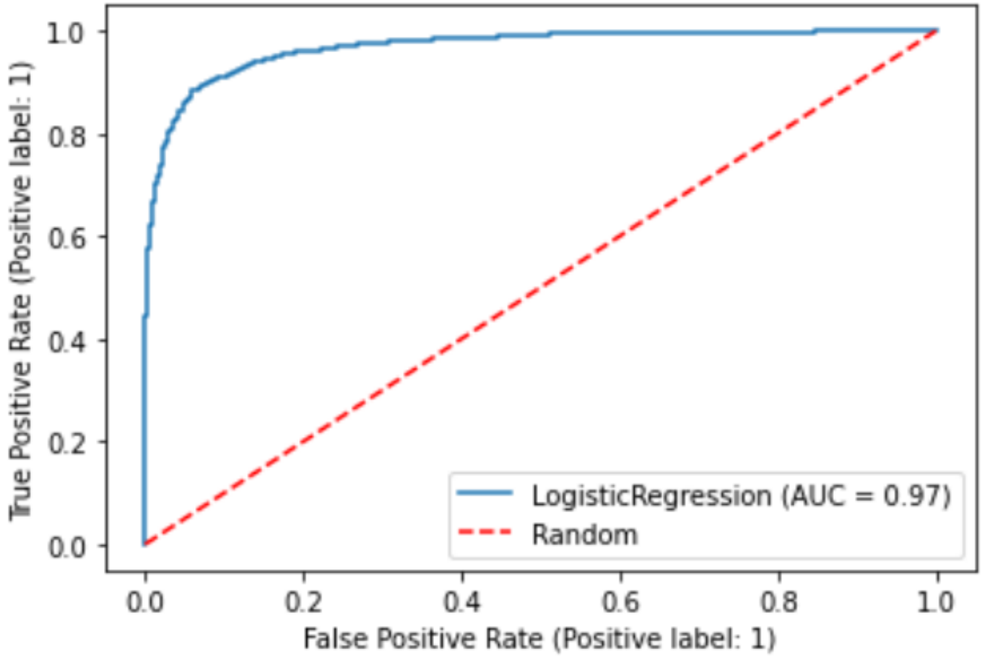
구분	불용어					
	불용어 x		500개		1000개	
	train	test	train	test	train	test
LogisticRegression C = 1 (기본)	0.921	0.911	0.865	0.837	0.855	0.820
LogisticRegression C = 10 (약화)	0.940	0.905	0.888	0.842	0.875	0.820
LogisticRegression C = 0.1 (강화)	0.893	0.886	0.831	0.816	0.818	0.800
RandomForest: 나무100, 깊이 30	0.926	0.867	0.810	0.765	0.802	0.761
RandomForest: 나무100, 깊이 40	0.946	0.872	0.827	0.772	0.817	0.765
RandomForest: 나무100, 깊이 50	0.961	0.876	0.840	0.777	0.829	0.769

분석과정

3단계: 모델 학습

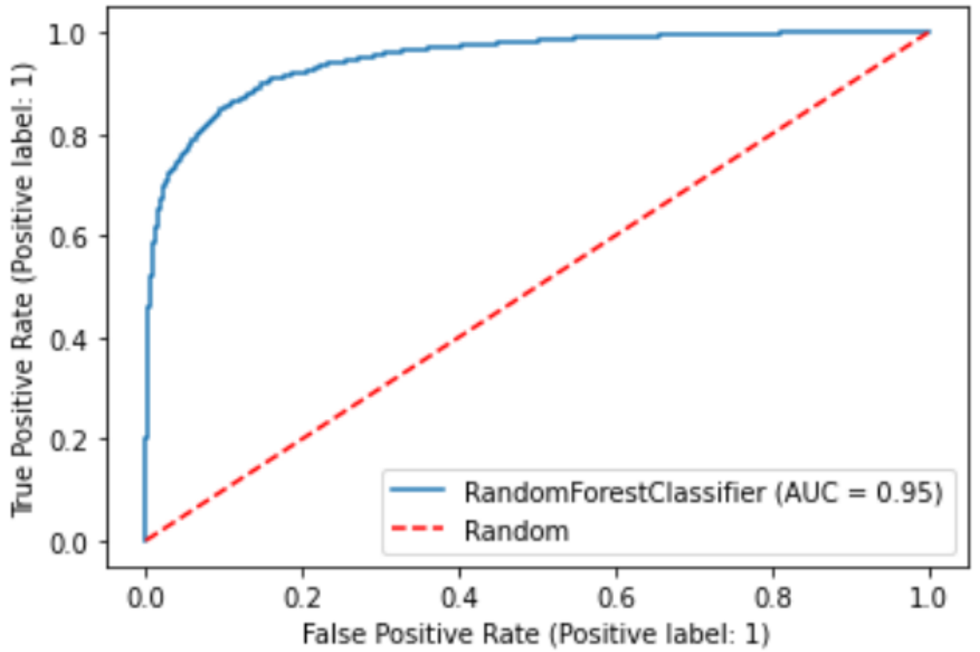
LogisticRegression C = 1 (규제 기본)

정확도	정밀도	재현율	F1 스코어	ROC_AUC
0.911	0.925	0.894	0.909	0.968



RandomForest: 나무100, 깊이 50

정확도	정밀도	재현율	F1 스코어	ROC_AUC
0.876	0.893	0.854	0.873	0.949



분석과정

1단계 [수집] | 2단계 [전처리] | 3단계 [모델학습] | 4단계 [성능개선] | 결과

3단계: 모델 학습

5차 시도 [N-gram 변화에 따른 성능 비교]

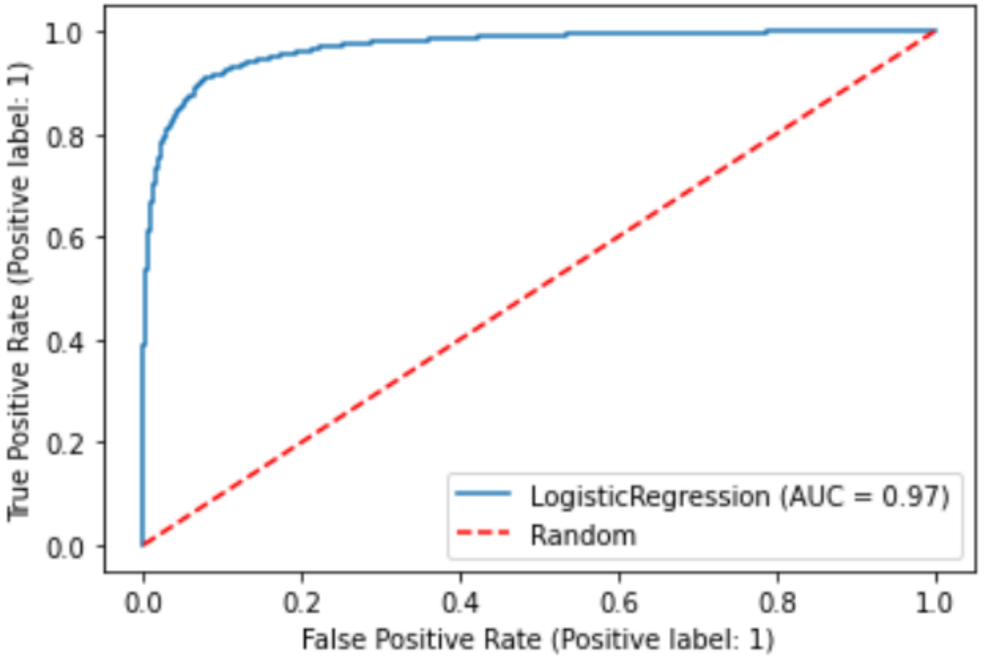
구분	N-gram					
	unigram		bigram		trigram	
	train	test	train	test	train	test
LogisticRegression C = 1 (기본)	0.918	0.907	0.928	0.914	0.923	0.910
LogisticRegression C = 10 (약화)	0.937	0.903	0.947	0.914	0.942	0.907
LogisticRegression C = 0.1 (강화)	0.890	0.881	0.898	0.893	0.894	0.889
RandomForest: 나무100, 깊이 30	0.925	0.865	0.918	0.865	0.931	0.871
RandomForest: 나무100, 깊이 40	0.943	0.868	0.935	0.871	0.947	0.877
RandomForest: 나무100, 깊이 50	0.959	0.873	0.949	0.875	0.961	0.880

분석과정

3단계: 모델 학습

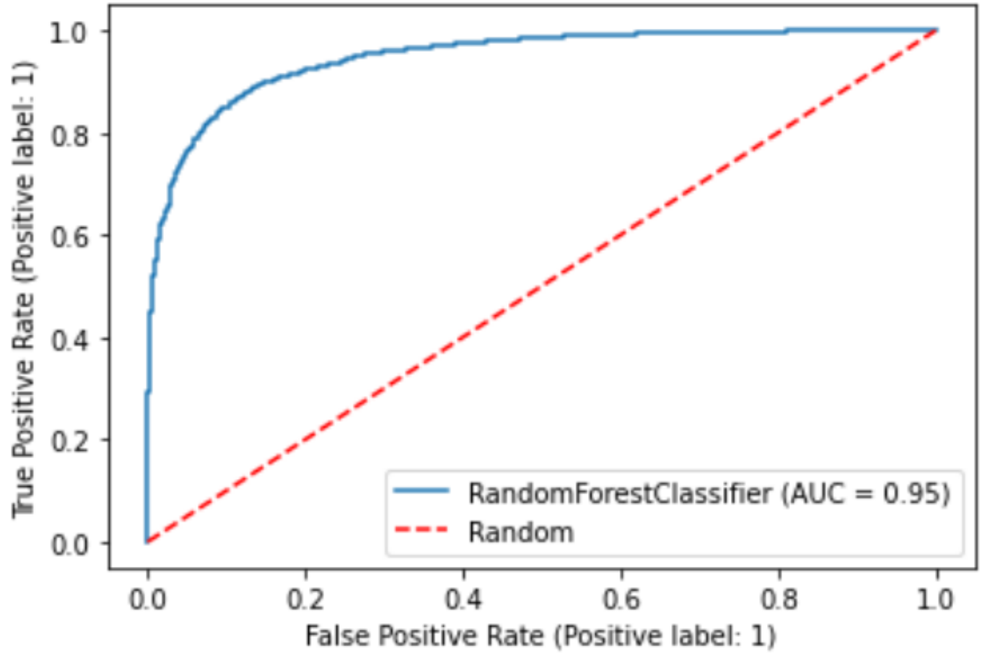
LogisticRegression C = 10 (규제 약화)

정확도	정밀도	재현율	F1 스코어	ROC_AUC
0.914	0.922	0.904	0.913	0.969



RandomForest: 나무100, 깊이 50

정확도	정밀도	재현율	F1 스코어	ROC_AUC
0.875	0.896	0.849	0.872	0.949



분석과정

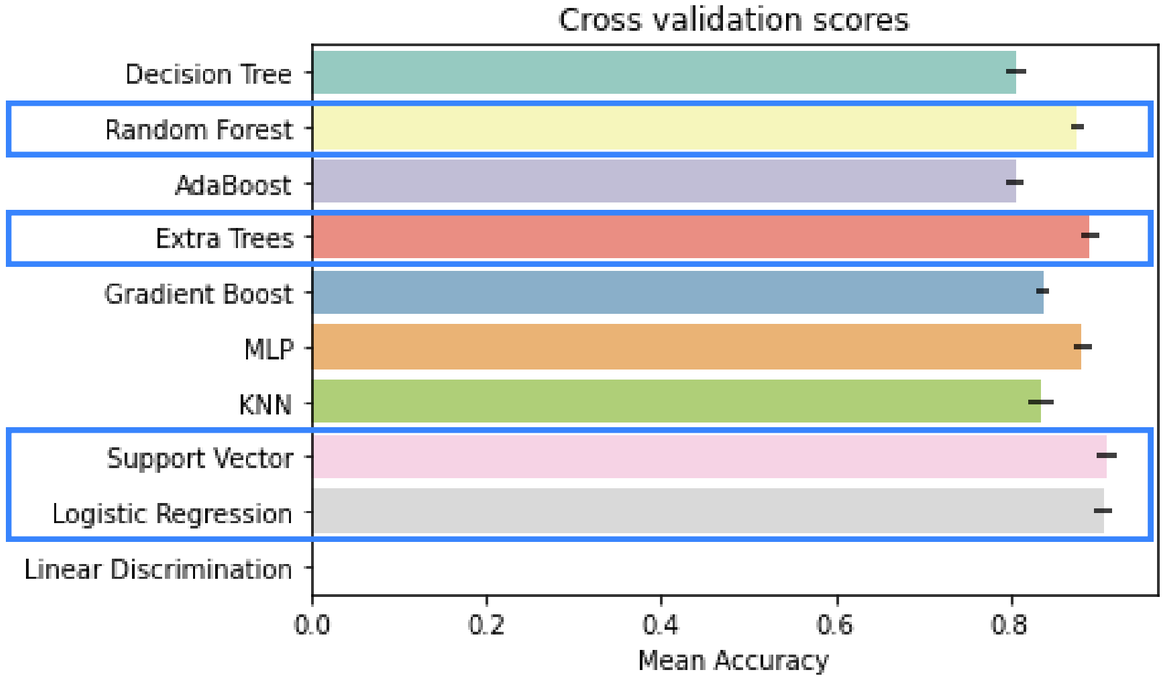
1단계 [수집] | 2단계 [전처리] | 3단계 [모델학습] | 4단계 [성능개선] | 결과

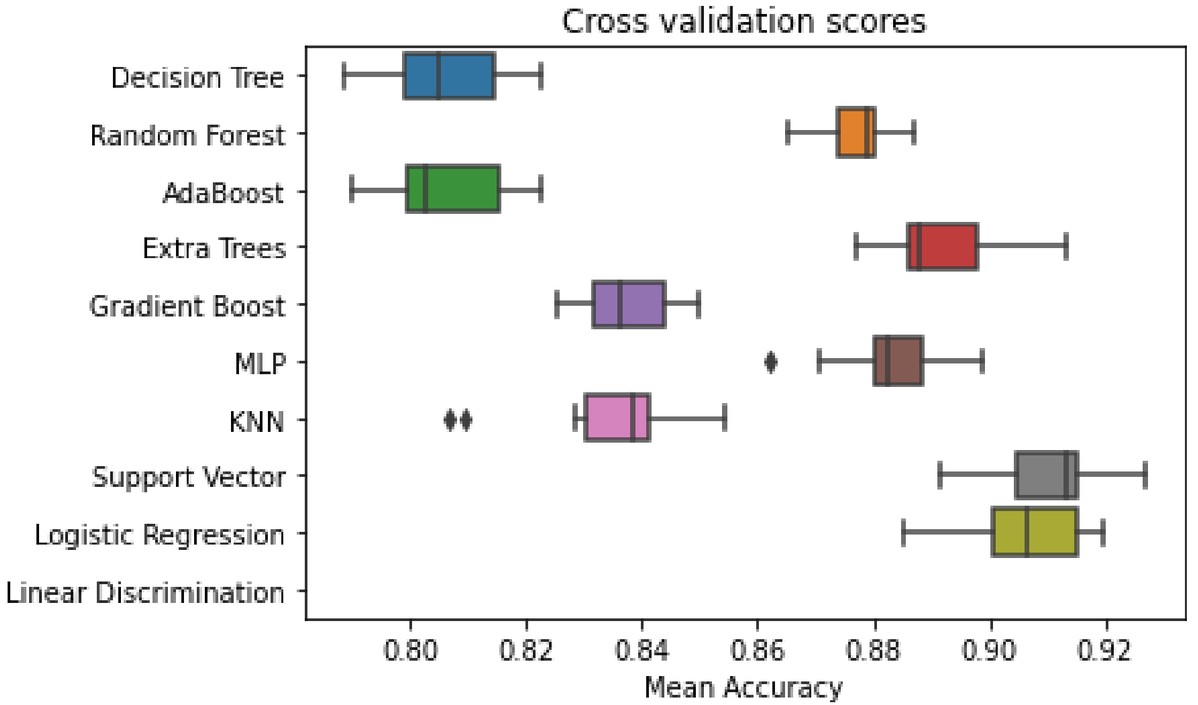
4단계: 성능 개선 - 교차검증

	Decision Tree	Random Forest	AdaBoost	Extra Trees	Gradient Boost	MLP	KNN	Support Vector	Logistic Regression
0	0.800383	0.878827	0.791454	0.889668	0.830357	0.870536	0.836097	0.894133	0.896046
1	0.794005	0.867985	0.799107	0.882653	0.825255	0.879464	0.809311	0.902423	0.899235
2	0.812500	0.873724	0.801658	0.886480	0.828444	0.882653	0.839286	0.912628	0.905612
3	0.815051	0.887117	0.817602	0.897959	0.847577	0.896046	0.850765	0.924745	0.919643
4	0.806122	0.880740	0.809311	0.913265	0.841837	0.898597	0.838010	0.915179	0.917092
5	0.798469	0.865434	0.801658	0.876913	0.836735	0.862245	0.806760	0.891582	0.885204
6	0.788768	0.878749	0.790045	0.885769	0.835992	0.890236	0.828334	0.910657	0.904914
7	0.822591	0.887045	0.818124	0.897256	0.850032	0.882578	0.840459	0.926611	0.918315
8	0.821953	0.873644	0.822591	0.886407	0.835354	0.882578	0.854499	0.913848	0.907466
9	0.804084	0.879387	0.804084	0.898532	0.844927	0.883216	0.841736	0.914486	0.909381
mean	0.806393	0.877265	0.805563	0.891490	0.837651	0.882815	0.834526	0.910629	0.906291
std	0.010886	0.006824	0.010588	0.009881	0.007891	0.010318	0.014940	0.010986	0.010258

분석과정

4단계: 성능 개선 - 교차검증





분석과정

1단계 [수집] | 2단계 [전처리] | 3단계 [모델학습] | 4단계 [성능개선] | 결과

4단계: 성능 개선 - 하이퍼 파라미터 튜닝(최적의 파라미터 조합, 최고점수 출력)

로지스틱 리그레션

[파람 그리드 설정]

penalty	l1, l2
C	0, 4, 10

결과

[최적의 파라미터 조합]

- C=2.7825594022071245

[최고 점수(accuracy)]

0.9091618994438873

분석과정

1단계 [수집] | 2단계 [전처리] | 3단계 [모델학습] | 4단계 [성능개선] | 결과

4단계: 성능 개선 - 하이퍼 파라미터 튜닝(최적의 파라미터 조합, 최고점수 출력)

엑스트라 트리

[파람 그리드 설정]

max_depth	None
min_samples_split	2, 3, 10
min_samples_leaf	1, 3, 10
bootstrap	False
n_estimators	100,300
criterion	gini

결과

[최적의 파라미터 조합]

- min_samples_split = 10

[최고 점수(accuracy)]

0.8918737708867848

분석과정

1단계 [수집] | 2단계 [전처리] | 3단계 [모델학습] | 4단계 [성능개선] | 결과

4단계: 성능 개선 - 하이퍼 파라미터 튜닝(최적의 파라미터 조합, 최고점수 출력)

랜덤포레스트 1

[파람 그리드 설정]

max_depth	10,20,30
min_samples_split	2, 3, 10
min_samples_leaf	1, 3, 10
bootstrap	False
n_estimators	100,300,500
criterion	gini

결과

[최적의 파라미터 조합]

- max_depth=30
- bootstrap=False
- n_estimators=300

[최고 점수(accuracy)]

0.8626571799747339

분석과정

4단계: 성능 개선 - 하이퍼 파라미터 튜닝(최적의 파라미터 조합, 최고점수 출력)

랜덤포레스트 2

[파람 그리드 설정]

max_depth	None
min_samples_split	2, 3, 10
min_samples_leaf	1, 3, 10
bootstrap	False
n_estimators	100,300
criterion	gini

결과

[최적의 파라미터 조합]

- bootstrap=False
- min_samples_split=3
- n_estimators=300

[최고 점수(accuracy)]

0.881220330346561

분석과정

분석결과

하이퍼 파라미터 튜닝 적용

모델명	최적의 파라미터 조합	최고 점수
LogisticRegression	C=2.7825594022071245	0.909
ExtraTree	min_samples_split = 10	0.892
RandomForest 1	max_depth=30 bootstrap=False n_estimators=300	0.863
RandomForest 2	bootstrap=False min_samples_split=3 n_estimators=300	0.881

수작업

모델명	최적의 파라미터 조합	최고 점수
LogisticRegression	C=10	0.914
RandomForest	max_depth=50 bootstrap=False n_estimators=100	0.875

결과해석

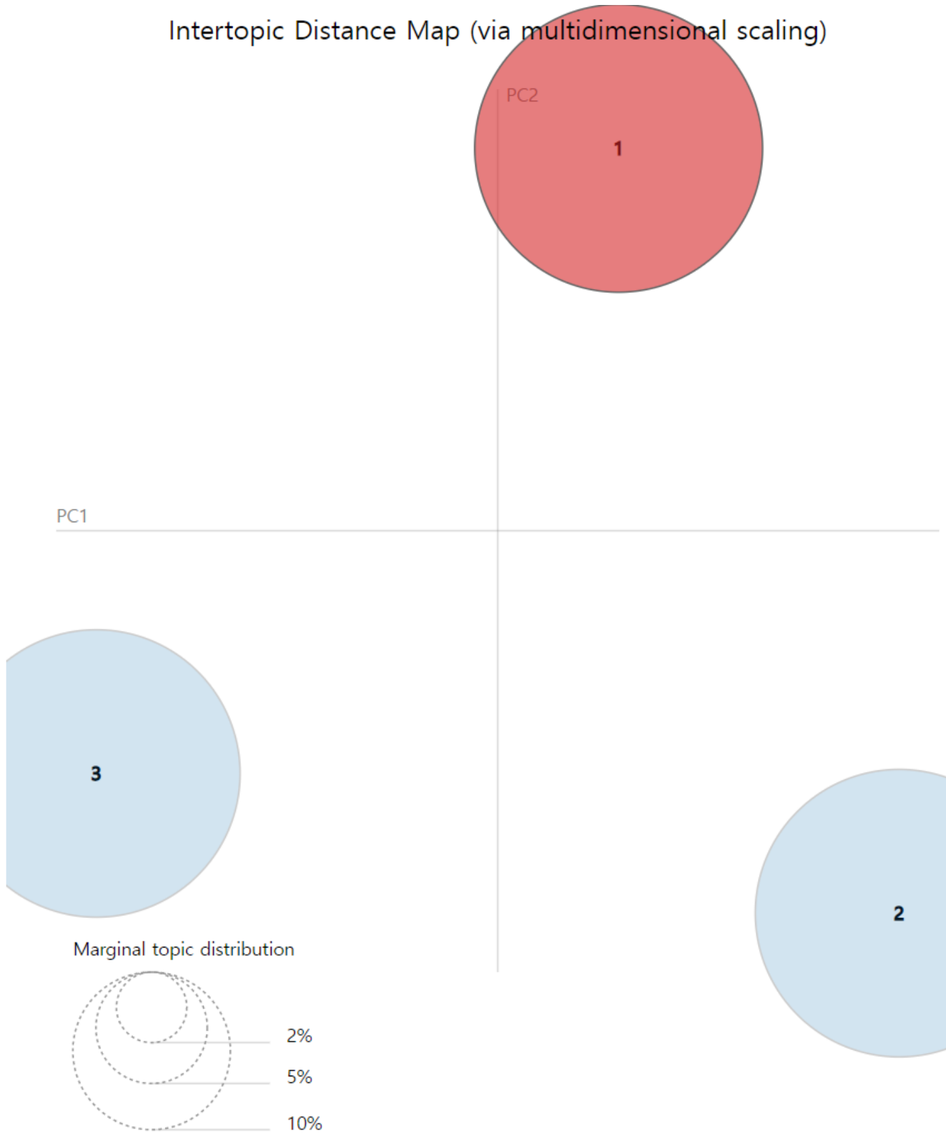
토픽모델링 적용하여 '네카라쿠배' 장점 분석

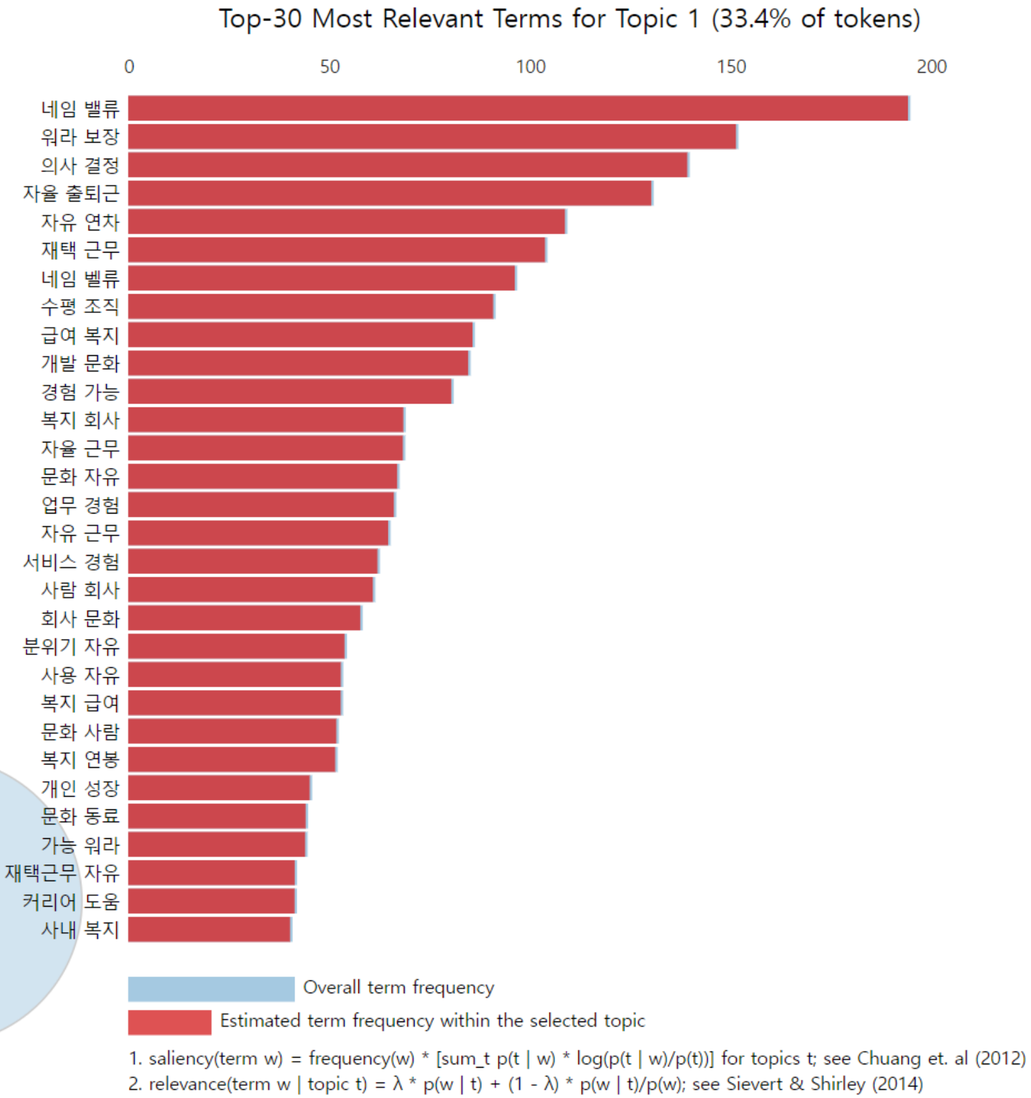
장점

Topic 1	('네임 밸류', 212.65), ('워라 보장', 165.75), ('의사 결정', 152.48), ('자율 출퇴근', 142.64), ('자유 연차', 119.11), ('재택 근무', 113.67), ('네임 벨류', 105.42)
Topic 2	('수평 문화', 319.23), ('자유 분위기', 263.53), ('조직 문화', 220.03), ('근무 시간', 157.61), ('성장 회사', 153.98), ('회사 복지', 87.69), ('유연근 무제', 86.24)
Topic 3	('사내 문화', 236.52), ('근무 환경', 171.22), ('연차 사용', 151.4), ('성장 가능', 134.71), ('기업 문화', 125.44), ('수평 분위기', 112.8), ('야근 수당', 107.35)

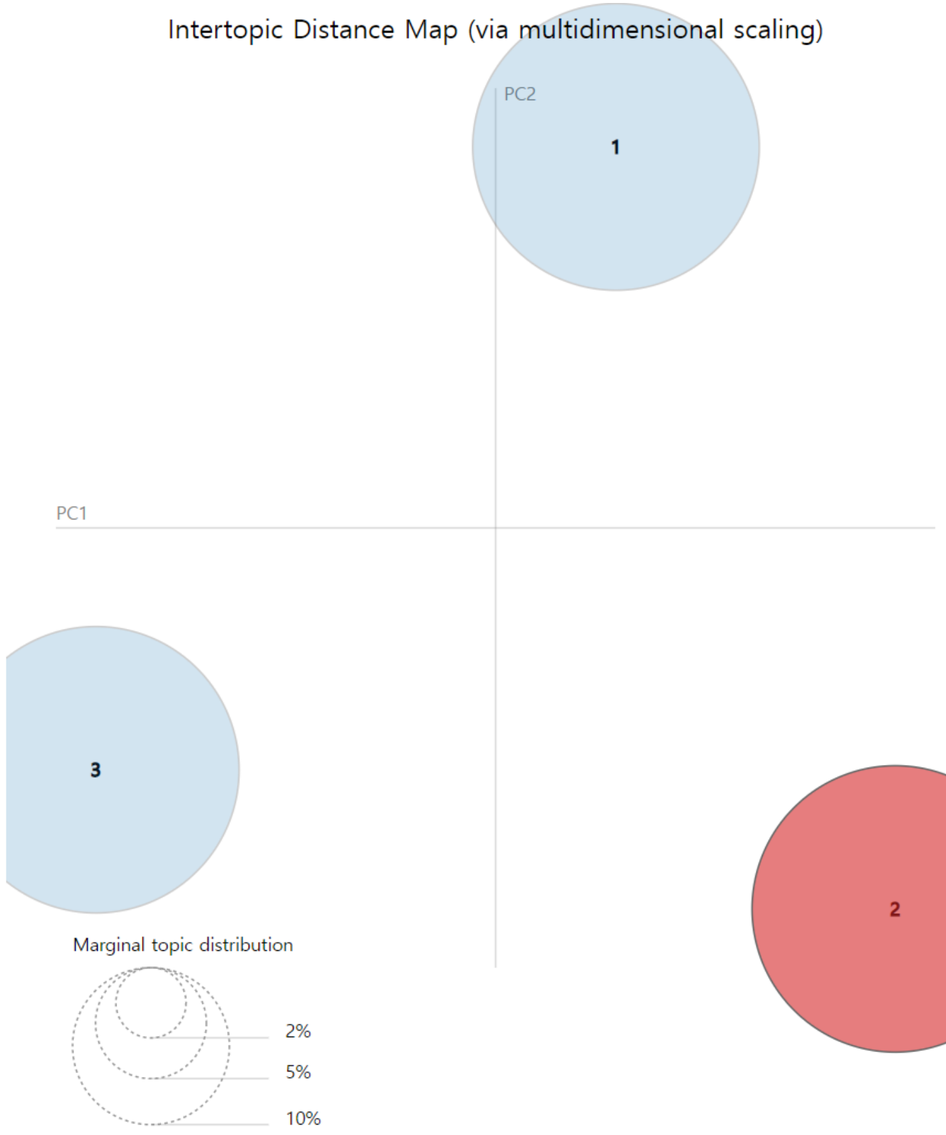
Topic	빈도수
수평 문화	319.23
자유 분위기	263.53
사내 문화	236.52
조직 문화	220.03
네임 밸류	212.65
근무 환경	171.22
워라 보장	165.75
근무 시간	157.61
성장 회사	153.98
의사 결정	152.48
연차 사용	151.4
자율 출퇴근	142.64

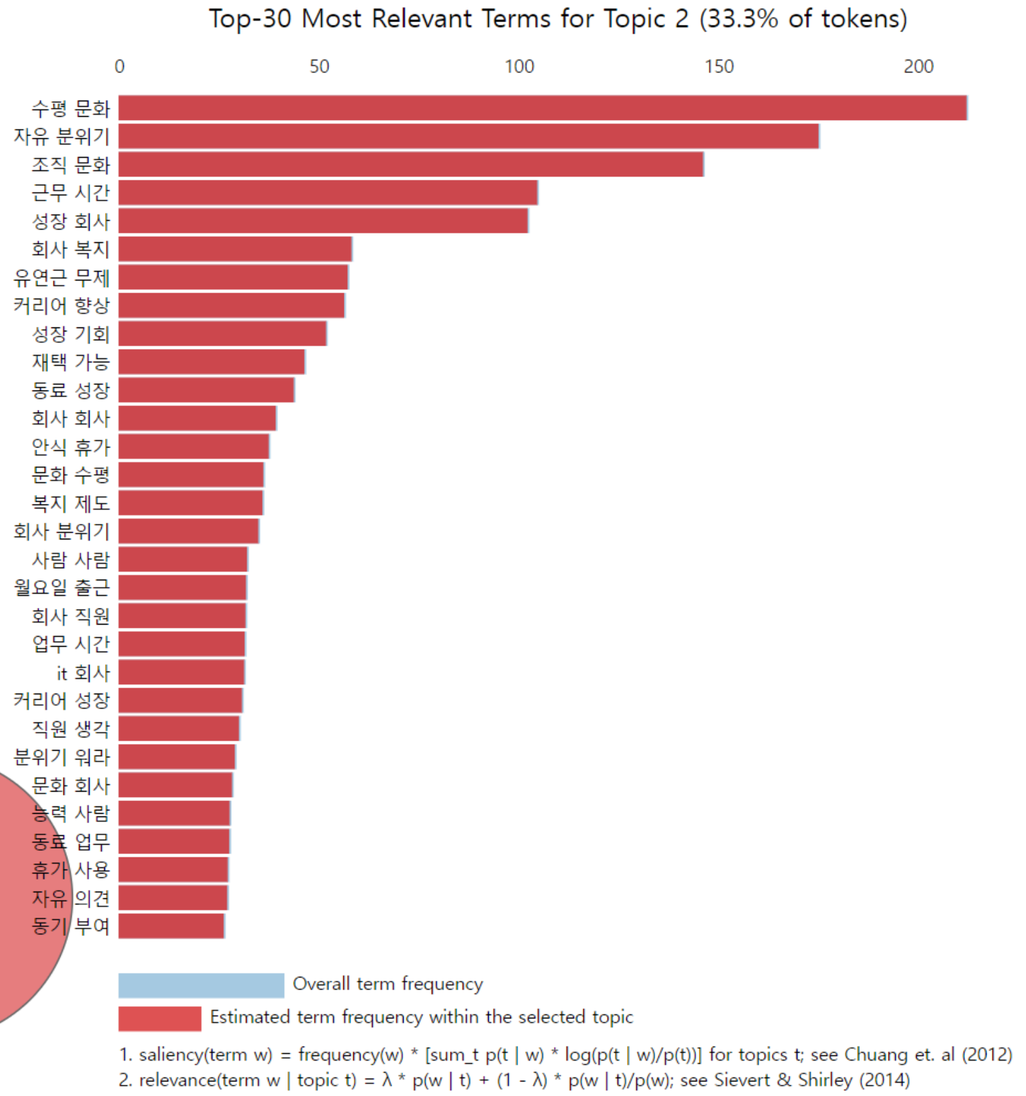
결과해석


Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
1
2
3
Marginal topic distribution
2%
5%
10%

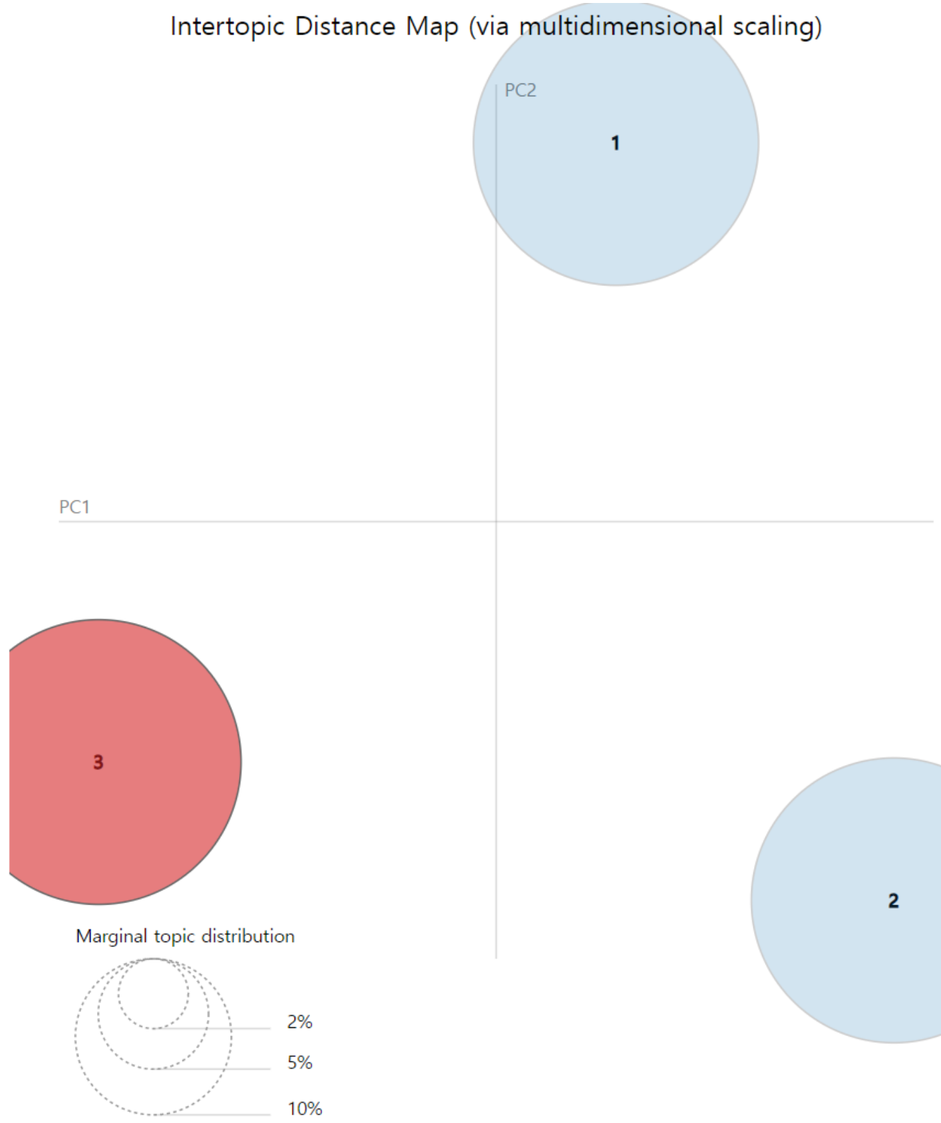

Top-30 Most Relevant Terms for Topic 1 (33.4% of tokens)
0
50
100
150
200
네임 밸류
워라 보장
의사 결정
자율 출퇴근
자유 연차
재택 근무
네임 벨류
수평 조직
급여 복지
개발 문화
경험 가능
복지 회사
자율 근무
문화 자유
업무 경험
자유 근무
서비스 경험
사람 회사
회사 문화
분위기 자유
사용 자유
복지 급여
문화 사람
복지 연봉
개인 성장
문화 동료
가능 워라
재택근무 자유
커리어 도움
사내 복지
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

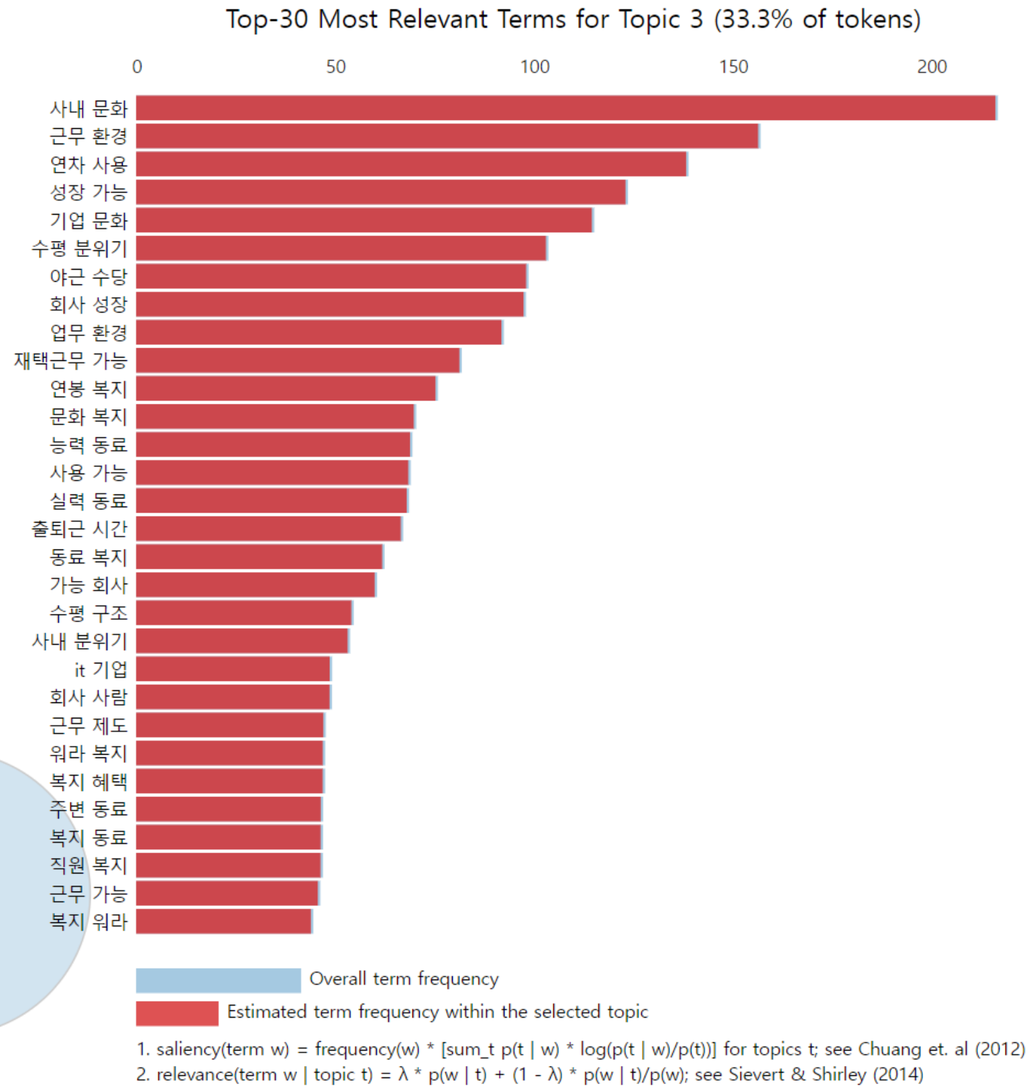
결과해석


Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
1
2
3
Marginal topic distribution
2%
5%
10%


Top-30 Most Relevant Terms for Topic 2 (33.3% of tokens)
0
50
100
150
200
수평 문화
자유 분위기
조직 문화
근무 시간
성장 회사
회사 복지
유연근 무제
커리어 향상
성장 기회
재택 가능
동료 성장
회사 회사
안식 휴가
문화 수평
복지 제도
회사 분위기
사람 사람
월요일 출근
회사 직원
업무 시간
it 회사
커리어 성장
직원 생각
분위기 워라
문화 회사
능력 사람
동료 업무
휴가 사용
자유 의견
동기 부여
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

결과해석


Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
1
2
3
Marginal topic distribution
2%
5%
10%


Top-30 Most Relevant Terms for Topic 3 (33.3% of tokens)
0
50
100
150
200
사내 문화
근무 환경
연차 사용
성장 가능
기업 문화
수평 분위기
야근 수당
회사 성장
업무 환경
재택근무 가능
연봉 복지
문화 복지
능력 동료
사용 가능
실력 동료
출퇴근 시간
동료 복지
가능 회사
수평 구조
사내 분위기
it 기업
회사 사람
근무 제도
워라 복지
복지 혜택
주변 동료
복지 동료
직원 복지
근무 가능
복지 워라
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

결과해석

MZ세대가 일하고 싶어하는 회사 조건

01 수평적인 조직문화 (수평문화, 사내문화, 조직문화)

02 자율적인 분위기

03 대기업이라는 네임밸류

04 근무환경 (많은 트래픽과 다양한 경험)

05 워라밸 보장

감사합니다
Q&A