제 1회 국민대학교

AI빅데이터 분석 경진대회

결과 발표 보고서 | Team 쉼표네

팀장 : 홍승표 | 팀원 : 김태현, 김현우, 남다예 , 서다슬

Contents

01

안녕하세요,
저희는 쉼표네 입니다.

'쉼표네' 는 동료 팀 '쵸리네' 에서 영감을 받은 팀장의 **독단적 결정**으로 생성된 이름으로써,

가족과도 같은 협동심으로 대회에 임하겠다는 각오가 담겨있습니다.

팀장 : 홍승표 | 팀원 : 김태현, 김현우, 남다예 , 서다슬

저희는 모든 사항을 합의해서 진행하기 보다는 각자 고민해보고

최고 Output을 낸 Idea를 공유해가며 Develop하는 방식을 선택하였습니다.

02

대회 소개

이번 대회에서는 Total HR Service를 제공하는 (주)스카우트의 후원을 받아 유연한 노동시장으로의 변화 흐름에 맞추어,
구직자 입장에서는 자신의 이력과 경력에 맞춤화된 채용 공고를 추천받을 수 있고 구인기업 입장에서는
공고에 적합한 인재를 미리 선별하는 도구로 활용할 수 있도록 **채용공고 추천 알고리즘 개발**을 제안합니다.

이력서 등 구직자 관련 데이터와 **채용 공고 관련 데이터**, 그리고 **지원 히스토리 데이터**를 활용하여
구직자에게 맞춤화된 채용 공고를 자동으로 추천할 수 있는 알고리즘을 개발함에 따라
지원자는 적성에 맞는 채용 공고에 지원하여 직무 만족도를 높이고
구인기업은 직종에 맞는 핵심 인재를 선발할 수 있으리라 기대할 수 있습니다.

02

대회 소개

대회규칙 설명

사용 데이터 셋

- Apply_train : 지원 히스토리 30%
  (Resume_seq, Recruitment_seq)
- Resume 관련 데이터
  (Resume, Edu, Cert, Lang)
  * Key : Resume_seq
- Recruitment 관련 데이터
  (Recruitment, Company)
  * Key : Recruitment_seq

평가 지표

- Recall 5
- 참가자들은 각 구직자들에게
  가장 적합한 공고 5개를 추천한다
- 추천한 공고 5개가 나머지 히스토리
  70%에 포함되는지를 평가한다
- 각 구직자 별로 정답률을 구한 뒤,
  정답률의 평균값을 평가 지표로 한다

제한 사항

- 모든 참가자에 대해 5개 추천 필요
- 이미 지원한 공고는 제외하고 추천

03

타임라인

Negative Sampling을 통한 Boosting 모델을 이용한 방식과

Cosine Similarity를 이용한 방식을 모두 시도해보았습니다.

02

<table>
<tr><th rowspan="2">일정</th><th colspan="2">모델</th><th rowspan="2">결과</th><th rowspan="2">순위</th></tr>
<tr><th>Boosting</th><th>Cosine similarty</th></tr>
<tr><td>10/25</td><td colspan="2">팀 결성</td><td></td><td></td></tr>
<tr><td>10/26</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>10/27</td><td>Negative Boosting 활용한
LightGBM 모델 설계</td><td>Base Line Code 분석</td><td>-</td><td>-</td></tr>
<tr><td>10/28</td><td colspan="2" rowspan="4">EDA (마감 전까지 계속 진행)</td><td rowspan="4">-</td><td rowspan="4">-</td></tr>
<tr><td>10/29</td></tr>
<tr><td>10/30</td></tr>
<tr><td>10/31</td></tr>
<tr><td>11/1</td><td>Negative Boosting 코드 공유</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>11/2</td><td>EDA + Negative Boosting 적용</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>11/3</td><td rowspan="3">submission 까지 가능한 모델 개발 (V1)</td><td>-</td><td rowspan="3">train auc: 0.827
test auc: 0.692
public : 0.00715</td><td rowspan="3">226위</td></tr>
<tr><td>11/4</td><td>-</td></tr>
<tr><td>11/5</td><td>-</td></tr>
</table>

02

일정	모델		결과	순위
	Boosting	Cosine similarty		
11/6	EDA 결과 공유 및 최종 합의		-	-
11/7	모델 수정 필요 (train/test 분리 시 오류)	feature 반영한 유사도 모델 공유받음	-	-
11/8	**수정 모델 공유 (V2)**	-	train auc: 0.751 test auc: 0.718 public : ??	? 위
11/9	팀 간 EDA 아이디어 공유	-	-	-
11/10	*** XGBoost 모델 전환 결정 (One-hot encoding 가능하도록 전처리) * Apply 유사도를 사용하여 로직 개선 * RandomSearchCV 적용 (V3)**	Apply_train을 특정 조건에 따라 필터링 하는 아이디어 공유받음	train auc: 0.774 test auc: 0.713 public : ??	? 위

02

일정	모델		결과	순위
	Boosting	Cosine similarty		
11/11	스코어 개선되지 않아 Boosting 모델 개발 중단	Apply_train 필터링 작업 시작 (V4)	-	-
11/12	-	필터링 조건 탐색 및 적용 (조건 1~5 생성, 가중치 서치)	recall5 : 0.1221 public : 0.16674	Public
11/13	-	최종 제출	recall5 : 0.1221 public : 0.16674 private : 0.17169	Private 30위

04

진행 기록

V1 ~ V2 : LightGBM

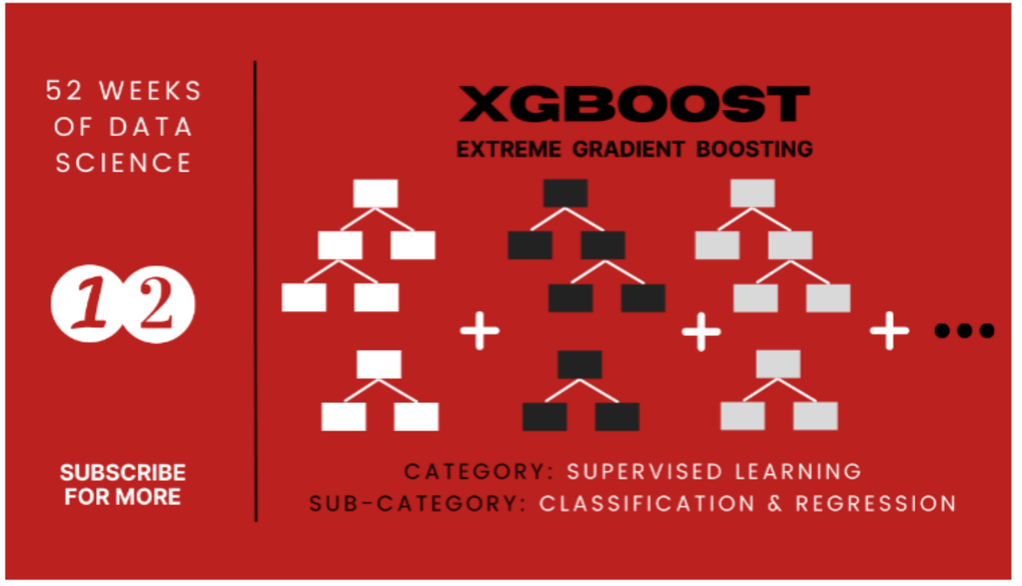
V3 : XG Boost

V4 : Apply_train을 Sampling 한 뒤, Cosine Similarity 적용하여 추천

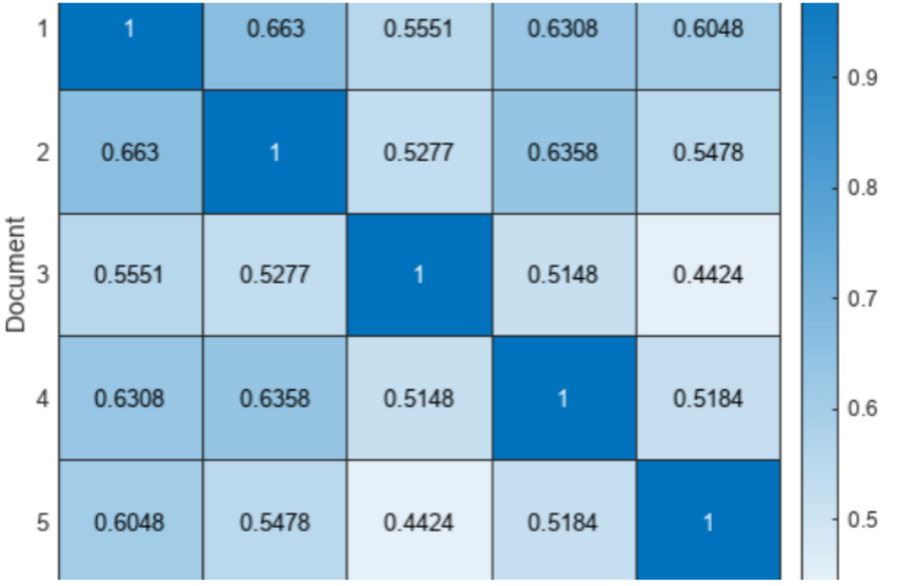
04

진행기록

부스팅 vs 유사도



경진대회는 Boosting이 일반적으로 성과가 좋다 !
우리가 배운 내용을 써먹어보면서 우리 것으로 만들자 !
=> 우리가 해본 Boosting은 이진 분류인데..??



대회에서 제공한 Base line 코드는 유사도를 활용했다 !
Base line에서 피처를 추가해서 점수를 올리자 !
=> 피처를 적용할 수록 점수가 떨어진다..

04

진행기록 : Boosting

초기모델 설계(V1)

train auc : 0.827

test auc : 0.692

public score : 0.007

STEP.01

Data Merge -> EDA

- 키 컬럼들을 기준으로 merge
- 결측치 치환, 피처 생성, 스케일링, 인코딩
- 구직자 기준으로 공고 데이터를 groupby -> 구직자 feature

STEP.02

Negative Sampling
-> train/test 데이터셋 생성

- 데이터 셋을 이진분류가 가능하도록 변환 (Target 컬럼을 생성)
- sklearn import하여 train / test 데이터 셋 생성

STEP.03

학습 -> 학습평가

- LightGBM 사용
- roc_auc 지표 사용

STEP.04

예측 -> Output

- 생성된 모델에 모든 구직자 X 모든 공고 조합을 input하여 예측확률을 구함
- 사전 지원 내역을 제외하고 각 구직자 별 지원 확률이 높은 공고 5개를 output

04

진행기록 : Boosting

모델 수정 (V2)

- 모델 학습이 잘 이루어 졌다면 구직자별 예측 점수 상위에 이미 지원한 공고가 랭크 되어있어야 하는데, 전혀 나타나고 있지 않음
- 심지어 Recall 점수가 0.0 (그 많은 사람 중 단 하나의 공고도 맞추지 못함..)
- 피처 부족 현상으로 결론, 추가 EDA 진행

- train / test 양 쪽에 모든 구직자 정보가 있도록 직접 split 되도록 수정

STEP.01
Data Merge -> EDA

STEP.02
Negative Sampling
-> train/test 데이터셋 생성

STEP.03
학습 -> 학습평가

STEP.04
예측 -> Output

train auc : 0.827
test auc : 0.692
public score : 0.007

08

진행기록 : Boosting

V2 Main feature list

희망직무

섬유;봉제;가방;의류의 빈도수가 높음
=> 해당 직무면 1, 아니면 0

군소 범주는 '기타' 직종으로 통합

자격증

'운전', 'MOS', '워드프로세서' 등의
일반적인 자격증 제거

1명만 가지고 있는 특수 자격증 제거

모집직무

모집 직무코드는 총 4자리이며,
앞 2자리가 대분류일 것을 가정
=> 앞 2자리로 인코딩
(code_22, code_23, code_24...)

어학

특정 어학 자격증 보유 여부로 인코딩
=> lang_2, lang_3, lang_4 ...

roc auc 지표는 조금씩 개선되나, 정작 Recall 점수는 0.0으로 그대로인 상황에서 스코어를 계속 확인해보는 것은 무의미하다고 판단

public score : 0.007 -> 0.011로 상승.

주최측에서 준 파일을 그대로 업로드하면 0.159였던 상황 -> 마냥 피처만 생성할 수는 없고 뭔가 변화가 필요한 시점

04

진행기록 : Boosting

모델 수정 (V3)

train auc : 0.85

test auc : 0.76

public score : 0.056

스코어는 8배 상승!! 그러나....

- train/test auc가 모두 0.7 정도인것으로 보아 아직 언더피팅이라고 판단.
  오히려 피처를 줄여서 단순하게 만들면 일단 오버피팅이라도 발생시킬 수 있지 않을까?
- 구직자가 가진 학위가 공고가 요구하는 학위보다 낮으면 추천하면 안되지 않을까?

STEP.01
Data Merge -> EDA

STEP.02
Negative Sampling
-> train/test 데이터셋 생성

- LightGBM의 한계일까?
  데이터를 모두 범주형으로 바꾸고 XGBoost 적용

STEP.03
학습 -> 학습평가

STEP.04
예측 -> Output

- 모든 구직자 X 특정 공고 (구직자 별 예측값이 높은 상위 n개 공고) 조합을 사용하여 정확도 상승, 속도 개선