<머신러닝, Competition 자세한 기술>

목 차

팀원: 김서령, 황건하, 김예향

첫번째 전략 -> 상위권 소스코드를 각자 합치기

먼저, 상위권 소스코드를 각자 분배하여 합친 뒤, 모델링 작업을 하였습니다. 왜냐하면 나중에 앙상블을 진행할 때 있어서 데이터 및 feature의 다양성과 모델의 다양성을 중요시 생각했기 때문입니다. 그래서 각자 모델 selection을 통해서 모델링 작업을 했습니다.

다음과 같이 파일 정리 후 모델링 작업을 하였습니다.

EX)

서령 – 배지환, 김현조 팀 파일 합치기

➔ 중요했던 피처: '숙련도', '근무경력_y', '근무경력_m', '출신대학 * 대학성적', '신입경력', '근무형태+직종', '대학+전공'

건하 – 김채원, 우호경 팀 파일 합치기.

➔ 중요했던 피처: '근무경력 x 대학성적', '근무지역_소분류', '대학_경력_standard deviation', '대학_경력_variance', '대학_경력_mean','대학_경력_sum'

예향 – 김종윤, 이준영 팀 파일 합치기

➔ 중요했던 피처: '마지막근무형태', '대학전공 & 세분화', '근무지역경험횟수', ''

그 후 각자 CatBoost 모델을 활용하여 단일 모델 최고 성능을 만들기 위해 노력하였습니다.

두번째 전략, Feature 생성

단일 모델의 성능을 더 높이기 위해 저희만의 독특한 feature 생성이 필수적이었습니다. 다음은 feature를 만들기 위한 저희만의 방법을 기입했습니다.

1) ID

-> '근무경력','출신대학','대학전공','어학시험','자격증','대학성적'이 같으면 같은 사람이 중복된 것으로 판단하고, 하나의 'ID'로 통합했습니다. 예시는 다음과 같습니다.

예시

```
#ID 생성: 각 컬럼의 값들을 더해서 고유한 사람을 파악
X_train['ID'] = \
X_train['근무경력'].astype(str) + '_' + X_train['출신대학'].astype(str) + '_' +\
X_train['대학전공'].astype(str) + '_' + X_train['어학시험'].astype(str) + '_' +\
X_train['자격증'].astype(str) + '_' + X_train['대학성적'].astype(str)
```

➔ Feature 생성을 한 이유:

: 근무경력을 기준으로 이상치를 찾아보는 도중, 근무경력이 60년 이상인 사람들이 상당히 있었습니다. 그래서 자세하게 분석해보니, '근무경력, 출신대학, 대학전공, 어학시험, 자격증, 대학성적' 등이 똑같은 사람들이 많이 있다는 것을 발견했습니다.

처음에는 이 중복값들을 제거를 한 뒤, 모델링 작업을 실행했지만, 성능이 좋아지지 않았습니다. 그래서이 중복값들을 제거하는 것이 아닌 피처로 만들면 좋을 것 같다는 생각을 했습니다. '근무경력','출신대학','대학전공','어학시험','자격증','대학성적'이 같으면 같은 사람이 중복된 것으로 판단하고, 하나의 'ID'로 통합한 뒤 피처 생성을 했습니다.

```
In [7]: train_df = pd.concat([X_train, y_train], axis=1).reset_index(drop=True)

In [8]: train_df.shape
Out[8]: (16570, 11)

In [9]: train_df = train_df.drop_duplicates(['근무경력','출신대학','대학전공','어학시험','자격증','대학성적']).reset_index(drop=True)

In [10]: train_df.shape
Out[10]: (14605, 11)
```

중복값 발견

➔ 처음에는 '직종, 세부직종, 직무태그, 근무경력, 근무형태, 출신대학, 대학전공, 어학시험, 자격증, 대학성적' 등 10가지 피처를 기준으로 같은 사람이 있으면 중복된 것으로 판단했습니다. 하지만 다 넣었을 때보다 '근무경력, 출신대학, 대학전공, 어학시험, 자격증, 대학성적'만을 넣고 여러번 돌렸울 때 제일 좋은 성능이 나왔기 때문에 6개 피처를 기준으로 'ID' 피처를 생성하였습니다.

2) 단과대

-> '대학전공'을 단과대별로 '인문대', '사과대', '경영경제대', '교육대', '예체능대', '공대', '보건계열', '자연과학', '법대', '소프트웨어', '농축산대', '건축대', '신학대', '결측' 등 15개로 분리하여 피처를 생성하였습니다. 복수전공으로 판단되는 데이터는 복수전공을 기준으로 분류하였고, 부전공인 데이터는 본전공으로 대체하였습니다. 밑에는 데이터 전처리를 하였을 때, 저희 팀만의 분류 기준입니다.

주목할 점

헷갈리는 부분이 있다면 직접 대학교 사이트에 들어가서 전공과 단과대를 확인해서 분류하였습니다.

예시

ex) 국어/전산학 -> 전산학으로 대체

ex) 국어, 전산학 -> 전산학으로 대체

부전공으로 판단되는 데이터는 본전공을 기준으로 분류

ex) 국어/전산학(부전공) -> 국어로 대체

➔ 이 피처 생성을 위해 상당히 많은 시간을 기울였으며, 엑셀로 수기 작업을 하며 분류하였습니다. 파일 확인은 '단과대.csv'를 통해 확인할 수 있습니다.

3) 소프트웨어+공대

➔ '단과대'피처에서 '소프트웨어'와 '공대'에 해당하는 데이터를 '공대'로 통합하고, 나머지는 '공대X'로 처리함으로써 '소프트웨어+공대' 피처를 생성하였습니다.

4) 세부직종_연봉

➔ 고용노동통계의 '직종,경력년수,성별 임금 및 근로조건' 통계와 '한국표준직업분류 개정7차'를 참고하여 평균 월급여액이 300만원 미만인 세부직종은 '기타'로 처리하였습니다. (엑셀 수기 작업)

5) NaN_count

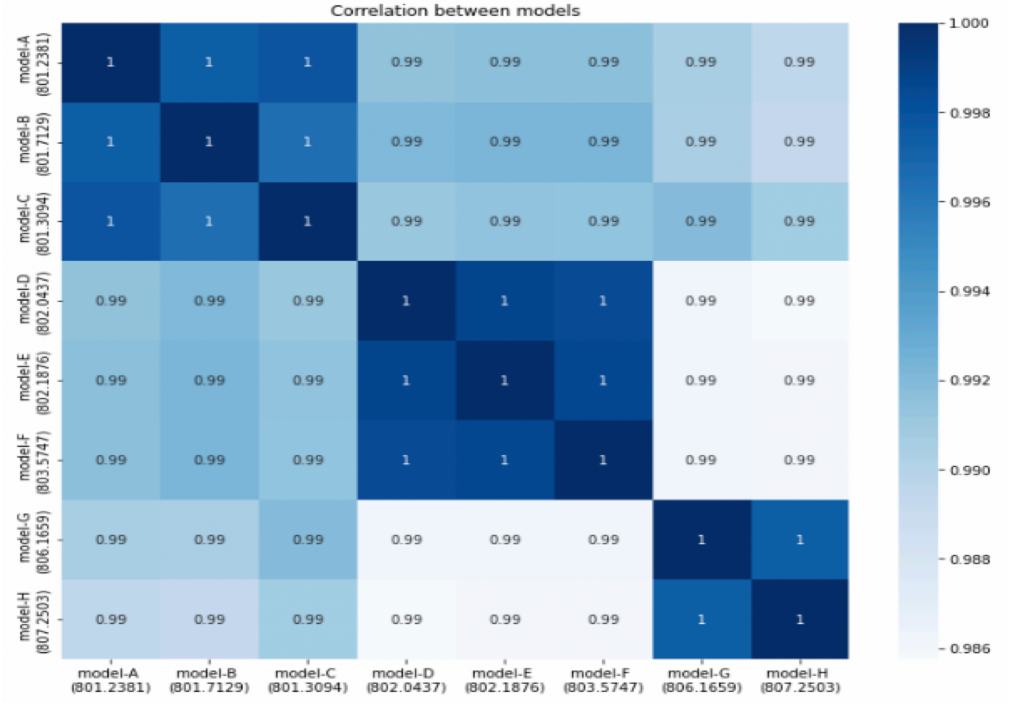
-> 각 행별로 결측치 개수를 카운트하여 피처를 생성하였습니다.

6) NotNull_count -

> 각 행별로 결측치가 아닌 값의 개수를 카운트하여 피처를 생성하였습니다.

세번째 전략, 최고 성능이 나온 모델의 상관관계. 확인

각자 최고 성능이 나온 모델인 CatBoost를 기준으로 앙상블을 시도 하기 전에 각 데이터셋의 최고 성능 TOP3을 뽑아서 상관관계를 확인해봤습니다.



본래의 목적은 상관관계가 낮은 모델을 만드는 것이였으나, 위의 사진과 같이 상관관계가 매우 높게 나왔기 때문에, 저희는 하나의 모델로 합치기로 했습니다.

네번째 전략, 단일 모델 성능 최대화 전략 수립

앙상블을 시도하기 전에, 단일 모델의 성능을 최대화하고자 하는 것이 저희 팀의 목표였습니다. 최고 성능 모델이였던 예향 모델을 기준으로 하여, 건하&서령이 했던 모델 중 중요 feature를 뽑은 뒤 합쳐서 다시 모델링을 했습니다. 그 후 중요 feature를 하나씩 넣으면서 최고 성능의 조합을 확인하고자 하였습니다.

<서령의 피처 중요도 확인 시도>

'출신대학*대학성적'
CatBoost CV mean = 827.94 with std = 372.34
'신입경력'
CatBoost CV mean = 829.12 with std = 374.28

'근무형태+직종'
CatBoost CV mean = 828.03 with std = 374.32
대학+전공
CatBoost CV mean = 827.86 with std = 375.61
숙련도
CatBoost CV mean = 829.17 with std = 371.76
직종+세부직종
CatBoost CV mean = 828.53 with std = 373.75
소프트웨어대+공대
CatBoost CV mean = 829.84 with std = 370.15

<건하의 피처 중요도 확인 시도>

CatBoost CV mean = 834.17 with std = 361.95 + '어학시험언어'

CatBoost CV mean = 834.14 with std = 360.71 어학시험언어 빼고 어학시험(지환팀)꺼로 수정

CatBoost CV mean = 833.63 with std = 362.84 ID 빼고

CatBoost CV mean = 833.83 with std = 363.22
근무지역_a, 근무지역_b, 근무지역_c, 추가 + 근무지역은 제외

CatBoost CV mean = 833.37 with std = 362.39 근무지역 추가해서

CatBoost CV mean = 833.67 with std = 359.53
'해외근무지역', 'ID'만 추가해서 + 어학시험도 바꿔줌

CatBoost CV mean = 834.00 with std = 361.06 호경팅 it, 법 등등 추가

CatBoost CV mean = 833.04 with std = 361.21 건하팀 단과대 추가

CatBoost CV mean = 832.46 with std = 362.22 단과대 빼고 다시

<예향의 피처 중요도 확인 시도>

최고성능 검증 스코어 829.37

근무희망형태 제거:
827.85 with std = 369.45

어학시험언어 추가 : 827.44

직종 추가: 827.64 with std = 370.85

nan_count 추가: 828.95 with std = 374.26

notnull_count 추가 :
827.65 with std = 377.91

어학시험언어/nan_count :
828.73 with std = 372.9

어학시험언어/직종:
829.23 with std = 372.66

nan_count/notnull_count :
828.96 with std = 373.88

어학시험언어/notnull_count :
828.21 with std = 375.36

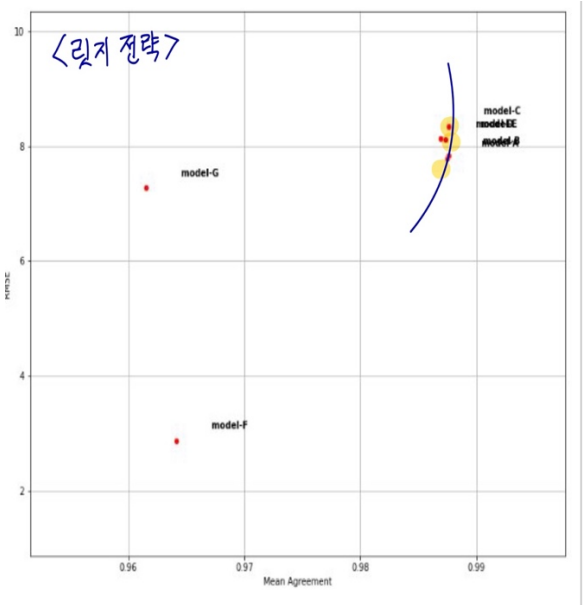
직종/nan_count:
828.85 with std = 371.08

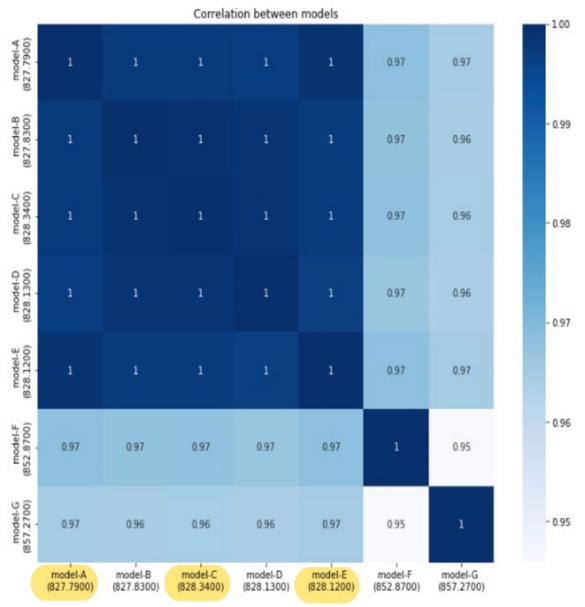
직종/notnull_count:
829.02 with std = 373.16

어학시험언어/notnull_count/직종:
829.36 with std = 371.66
어학시험언어/nan_count/직종:
828.28 with std = 370.70
어학시험언어/nan_count/notnul_count/직종:
828.81 with std = 373.21

위에는 각 데이터셋의 feature 중요도 확인을 위한 시도 결과입니다.

다섯번째 전략, 단일 모델 앙상블 시도 (교수님 코드 이용)





그 후 제일 유의미하지 않았던 feature를 빼고 나서 가장 성능을 높은 모델 7개로 상관관계를 확인하고, 그 중 모델A, 모델C, 모델E를 뽑아서 앙상블을 시도했습니다.

modelA의 피처에는

numeric_features = ['근무경력','대학성적','근무경력^2','NaN_count','NotNull_count','출신대학*대학성적']
categorical_features = ['직종','세부직종','출신대학','대학전공','어학시험','자격증','근무경력_세분화','마지막근무형태','근무지역_a','근무지역_b','근무지역_c','해외근무지역','단과대','대학전공_and_경력세분화','ID','근무경력_y', '근무경력_m','근무희망형태','세부직종_연봉','숙련도','신입경력','소프트웨어대+공대','근무형태+직종']
binary_features = ['직무태그', '근무지역','근무형태']

대학전공은 배지환팀 코드로 처리하였습니다.

modelC의 피처에는

numeric_features = ['근무경력','대학성적','근무경력^2','NaN_count','NotNull_count','출신대학*대학성적']
categorical_features = ['직종','세부직종','출신대학','대학전공','어학시험','자격증','근무경력_세분화','마지막근무형태','근무지역

_a','근무지역_b','근무지역_c','해외근무지역','단과대','대학전공_and_경력세분화','ID','근무경력_y', '근무경력_m', '근무지역_소분류','근무희망형태','어학시험언어','상위어학시험','new_근무지역','근무지역_경험횟수','세부직종_연봉','숙련도','신입경력','소프트웨어대+공대','대학+전공','근무형태+직종']
binary_features = ['직무태그, '근무지역','근무형태']

modelE 피처에는

numeric_features = ['근무경력','대학성적','근무경력^2','NaN_count','NotNull_count','출신대학*대학성적']
categorical_features = ['직종','세부직종','출신대학','대학전공','어학시험','자격증','근무경력_세분화','마지막근무형태','근무지역_a','근무지역_b','근무지역_c','해외근무지역','단과대','대학전공_and_경력세분화','ID','근무경력_y', '근무경력_m','근무희망형태','세부직종_연봉','숙련도','신입경력','소프트웨어대+공대','근무형태+직종','대학+전공']
binary_features = ['직무태그', '근무지역','근무형태']

대학전공은 배지환팀 코드로 처리하였습니다.

이렇게 세가지를 기하평균(P = 0.0000000001)으로 앙상블하여 score가 797.71078 값을 도출했습니다.

여섯번째 전략, Blending 앙상블 진행

X_train_bd.csv, y_train_bd.csv, X_test_bd.csv를 활용하여피처를 다음과 같이 설정했습니다.

numeric_features = ['근무경력','대학성적','근무경력^2','NaN_count','NotNull_count','출신대학*대학성적']
categorical_features = ['직종','세부직종','출신대학','대학전공','어학시험','자격증','근무경력_세분화','마지막근무형태','근무지역_a','근무지역_b','근무지역_c','해외근무지역','단과대','대학전공_and_경력세분화','ID','근무경력_y', '근무경력_m', '근무지역_소분류','근무희망형태','어학시험언어','상위어학시험','new_근무지역','근무지역_경험횟수','세부직종_연봉','숙련도','신입경력','소프트웨어대+공대','대학+전공','근무형태+직종']
binary_features = ['직무태그','근무지역','근무형태']

다양한 모델링을 활용하여 앙상블을 진행하였습니다. 모델은 다음과 같습니다.

1) RandomForestRegressor,
2) GradientBoostingRegressor
3) ExtraTreesRegressor
4) LGBMRegressor
5) XGBRegressor
6) CatBoostRegressor
7) NGBRegressor

먼저, 첫번째 전략은 7개 모델을 모두 사용하여 성능별로 가중치를 부여했습니다.

1) CatBoost CV mean = 828.50 with std = 370.50 -> 1등 가중치: 0.5
2) GRB CV mean = 856.83 with std = 371.90 -> 2등 가중치: 0.1
3) LGBM CV mean = 859.22 with std = 385.47 -> 3등 가중치: 0.1
4) XGB CV mean = 873.40 with std = 400.50 -> 4등 가중치: 0.085
5) NGB CV mean = 884.81 with std = 366.05 -> 5등 가중치: 0.075
6) RandomForest CV mean = 889.35 with std = 368.87 -> 6등 가중치: 0.075
7) ETR CV mean = 899.41 with std = 371.74 -> 7등 가중치: 0.065

-> public score: 807.43359

그 결과, Public 성능이 좋지 않아서 다른 전략을 수립하였습니다.

두번째 전략은 첫번째 전략에서 성능이 안좋은 모델.(NGB , RF, ETR 3개 제외)을 제외하고, 앙상블을 시도했습니다.

1) CatBoost CV mean = 828.50 with std = 370.50 -> 1등 가중치: 0.75
2) GRB CV mean = 856.83 with std = 371.90 -> 2등 가중치: 0.1
3) LGBM CV mean = 859.22 with std = 385.47 -> 3등 가중치: 0.1
4) XGB CV mean = 873.40 with std = 400.50 -> 4등 가중치: 0.05
-> publich score : 800.97026

그 결과, Public 성능이 조금 나아진 것을 확인할 수 있었습니다.

일곱번째 전략, 앙상블 진행

다섯번째 전략과 여섯번째 전략의 결과를 활용하여 앙상블을 진행하였습니다. 높은 score가 나온 것끼리 다시 앙상블을 진행하여 성능을 높이고자 하였습니다.

자세한 앙상블 실행 순서와 파일은 2번파일, '김서령-황건하-김예향_소스코드_실행절차'에서 확인하시길 바랍니다.

마지막, 이번 컴피티션을 하면서 느낀점 및 한계점

1. 기존 피처를 삭제해서는 안된다.
2. 결측값 처리는 생각보다 별로 중요하지 않다.
3. 일반화 성능을 높이기 위해 피처를 생성할 때에는 보편적으로 합당할 만한 피처를 생성하는 것이 매우 중요하다.
4. 데이콘 또는 캐글 대회에 참가하여 머신러닝을 더 배워야겠다고 다짐함.
5. 항상 질문에 빠른 답장을 해주시는 조윤호 교수님께 정말 감사합니다!!