<데이터 전처리 (결측치 제거)>

1. 결측치 제거

결측치를 확인해본 결과, 직무태그 2515개, 근무형태 9909개, 어학시험 11582개, 대학성적 1970개로 총 4가지 컬럼에서 결측치가 있는 것으로 파악되었습니다.

1-1) 직무태그

직무태그는 피쳐의 cardinality가 11815개로 매우 높았습니다.

전반적으로 샘플 각각의 값들을 살펴보았을 때, 이는 원본 데이터 수집 과정에서 스스로 작성할 수 있게 공란으로 비워둔 부분이라고 판단하였습니다.

작성할 수 있는 부분에 아무것도 적지 않은 사람들이 결측치로 처리되었다고 생각하여 결측치를 '없음'이라는 값으로 채웠습니다.

1-2) 근무형태

근무형태 피쳐는 샘플 별 각각의 value가 담고있는 내용들이 다음과 같이 ','(comma)로 연결되어 있었고, 담고있는 내용들이 많았습니다.

'정규직, 해외취업, 파견직'

쉽게 결측값을 무언가로 치부하여 처리하기 어려워, 근무형태 피쳐가 결측값을 가지고 있는 샘플들의 salary 평균을 관찰해보기로 결정했습니다.

전체 Salary 의 평균이 2814인 것에 비하여, 결측값들의 Salary 평균은 2439로 평균에 비해 현저히 낮은 것을 확인하였습니다.

그 결과에 따라, 결측값들을 '계약직'으로 대치하였습니다.

1-3) 어학시험

어학시험의 피쳐의 경우 각각의 샘플들이 응시한 것으로 보이는 시험의 이름이 들어있었습니다.

해당 칸이 비어있다는 것은, 마땅히 작성할 어학시험이 없었다는 것으로 판단, '미응시'로 결측값들을 대치하였습니다.

1-4) 대학성적

대학성적 피처의 점수 값은 '의미를 가지고 있는 수' 이며 그 수의 크기에 따라 의미의 차이가 있기 때문에 실수형 변수로 치부하여 결측값 대치를 하고자 하였습니다.

처음에는 이들을 샘플들의 대학성적 평균으로 대치하여 했으나, 각각의 점수들이 어느정도 범주화되어있는 정수값을 가지고 있다는 것을 확인하였습니다.

결측값을 가지는 샘플들을 다른 샘플들과 비슷하게 위치시키는 편이 더 좋을 것 같아 큰 차이가 없다면 중앙값을 활용하기로 하였습니다.

대학성적의 평균은 71.7 , 중앙값은 70 으로 큰 차이가 없다고 판단되어 결측값들을 샘플들의 중앙값으로 대치하였습니다.

* 실수형 변수들의 이상치는 pipeline에서 IQR을 활용한 전처리기로 처리하였습니다.

<피쳐 엔지니어링 (데이터 클린징 + 피쳐 생성) >

1. 근무경력

근무경력의 데이터는 X년 Y개월의 형태의 value를 가지고 있습니다.

이를 수치형 변수로 처리하기 위해 근무경력을 개월수로 통일하여 새로운 근무경력 피쳐를 생성하였습니다. (X*12 + Y 개월)

원본 데이터 또한 '근무경력_원본'이라는 이름의 피쳐로 새로 저장하여 두었습니다.

2. 근무형태

근무형태의 경우, 담고있는 내용들이 많아, 크게 정규직과 계약직으로 이분화하였습니다.

모든 정보를 버릴수는 없었기에, 이분화하기 전에 Salary와 관련이 있다고 판단된 해외취업여부에 관한 피처를 하나 생성해주었고, 이후 정규직과 계약직으로 나누었습니다.

이때, 인턴이 포함된 데이터들의 타겟의 평균이 눈에 띄게 낮은 결과를 보여, 인턴이 포함된 샘플들을 먼저 계약직으로 분류해주었습니다.

3. 근무지역

근무지역은 보통 생각하는 근무지역과는 달리 3가지 값이 ',(comma)로 구분되어 있었습니다.

이는 희망 근무지역 등 특별한 의미를 가진 데이터일수도 있다고 판단하여 하나로 통일하지 않고, 각각을 근무지역1,2,3으로 나누어 3개의 피쳐를 새로 생성해주었습니다.

4. 대학전공

대학전공 피쳐는, 비슷한 것을 가르치는 학과라고 판단됨에도 불구하고 대학별로 학과 이름이 다른 전처리하기 어려운 데이터였습니다.

일단 각 샘플의 뒷글자 중 '과', '학과'를 제거하였고, 데이터 샘플을 하나하나 찾아보는 과정을 통해 비슷한 학과를 하나의 학과로 처리하였습니다.(Ex: 중문, 중어, Chinese와 같은 값들을 '중국'이라는 하나의 값으로 통일해주었습니다.)

또한 오타 등의 대학전공의 의미를 가지고 있지 않다고 판단되는 부분에 대하여 '기타' 항목을 만들어 처리함으로써 전체 카디날리티를 줄일 수 있었습니다.

5. 어학시험

어학시험의 경우 ' '(space) 값을 가지는 데이터들이 있었습니다.

' '값을 가지는 데이터들을 결측지 즉 미응시로 대체하려 했으나 그 값들의 Salary 평균의 매우 높아 '불확실'이라는 값으로 따로 분류해주었습니다.

또한, 기타시험과 기타가 나누어져 있어, 기타로 통일해주었습니다.

<피처 추가>

1. 직무태그 글자수 생성

직무태그는 위에서 말했듯 , 자신이 작성할 수 있는 공란으로 나와있는 직무태그 글자수와 연봉의 그래프를 그려본 결과, 글자수에 비례하여 평균도 높아짐을 확인했습니다. 따라서 직무태그글자수라는 피처를 새로 추가해주었습니다. 또한 글자수를 범위로 나눠 직무태그글자수범위별 이라는 피처도 새로 생성해주었습니다.

2. 근무지역 소분류, 대분류 추가

근무지역 피쳐를 사용하여 서울, 경기, 인천, 부산 등으로 세분화하여 나눈 '근무지역_소분류' 피쳐와 수도권,비수도권으로 이분화한 '근무지역_대분류 피쳐', 2개를 생성해주었습니다.

3. 근무경력 유무

근무경력을 일전에 실수화하였는데, 그것을 기반으로 근무경력이 없는 경우, 2년 미만인 경우 그리고 근무경력이 2년 이상인 경우의 유무로 나눠주었습니다.

4. 인서울여부

서울에 있는 대학교를 분류하여 새로운 피처를 생성해주었습니다.

5. 직무태그_직무수

직무태그 피쳐에 들어있는 데이터들은, ','(comma) 로 나뉘어져 있는 경우가 많았습니다.

이에, str.split(',') 를 사용하여 각각의 데이터를 쪼개고, 그 수 별로 나누어 numeric feature를 생성할 수 있었습니다.

6. 직종_직무수

직종을 str.split 을 사용하여 나누고, 분할된 개수를 실수형 피쳐로 만들었습니다.

7. 세부직종_세부직종수

세부직종을 str.split을 사용하여 나누고, 분할된 개수를 실수형 피쳐로 만들었습니다.

8. 근무경력x대학성적

피처의 중요도를 뽑아본 결과, 근무경력과 대학성적의 중요도가 가장 높게 나와 두 개의 피처를 곱해준 근무경력*대학성적 피처를 생성해주었습니다.

9. 출신대학수치 x 대학성적

상위권 대학부터 하위권 대학까지 1~5까지 점수를 할당하여 나타낸 출신대학수치 피처를 하나

생성해주었으며, 대학성적과 곱해준 피처도 생성해주었습니다. (피쳐 중요도 표를 참고하였습니다.)

10. 세부직종 blank 없애기

세부직종 값들의 빈칸을 없애주었으며, 일부 단어들을 보다 큰 범주에 포함되는 형태로 바꾸어 cardinality를 줄여주었습니다. (EX : 게임기획 -> 게임디자인)

11. 세부직종별_필수직무유무

직무태그의 피처에 있는 단어들을 split으로 나누어준 후, 단어별 count를 통해 딕셔내리를 형성, 상위 5개 정도를 필수역량이라는 키워드를 뽑았습니다.

샘플에 세부직종에 필수역량 단어가 들어가 있으면 1로 반환해주는 binary 피처를 생성하였습니다.

12. 숙련도

근무경력을 0년, 3년 이하, 6년 이하, 6년 이상 기준으로 나누어, 경력 없음, 하, 중, 상으로 나눈 숙련도 피처를 생성해주었습니다.

* 이하에 작성한 피쳐들 중 설명이 없는 피쳐들에 대한 설명입니다.

모델의 피처 중요도에 기반하여 그들을 서로 곱하고 더하여 가지고 있는 의미를 여러 방식으로 합하거나,

범주형 피쳐와 수치형 피쳐를 조합하여 groupby 를 사용한 그룹 별 대푯값과 산포도(min,max,std,var,mean,median)들을 활용하여 피쳐를 생성하였습니다.

13. 직무태그_직무수 + 근무경력

14. 세부직종별 근무경력, 직무태그_직무수

15. 출신대학별 근무경력, 대학성적

16. 근무형태별 근무경력, 대학성적, 출신대학수치

17. 대학전공별 근무경력, 대학성적, 출신대학수치

18. 직무태그별 근무경력, 대학성적, 출신대학수치

19. 출신대학 + 전공

20. 근무형태 + 직종

21. 세부직종 + 출신대학

22. 세부직종 + 대학전공

23. 출신대학 + 대학전공

24. 직무태그1

1순위 직무태그 혹은 희망 직무태그일 것이라 판단하였고, 이에 따라 중요피처일 것이라 생각되어 맨 앞에 있는 태그만 새로 추출해주었습니다.

25. 직종, 및 세부직종별 대학성적, 근무경력 평균, 중앙값, 합, 표준편차, 최솟값, 최대

26. 어학시험별 대학성적, 근무경력 평균, 중앙값, 합, 표준편차, 최소, 최대

27. 서울근무

근무지역에 서울이 있는 경우, YES로 넣어주어 binary 피처를 생성했습니다.

28. 피처의 제곱

29. 근무형태

근무형태에 포함된 계약직, 해외취업, 정규직이 포함된 값들을 따로 뽑아, 그 순서를 달리하여 피처로 생성해주었습니다.

30. 근무형태_단순화 및 베테랑 정규직

근무경력이 56이상인 경우, 베테랑 정규직으로 넣어주었습니다.

31. 피처끼리의 합

32. 피처끼리의 곱

33. 마지막 근무형태, 근무지역, 직무태그, 출신대학별 근무경력, 대학성적

34. 근무경력_세분화

경력을 범위로 나누어 세분화해준 피처를 생성해주었습니다.

35. 해외근무지역

근무지역에 해외가 포함되어있는 경우, 이를 따로 나눈 binary 피처를 만들어주었습니다.

36. new_근무지역

광주, 전남, 전북과 같은 지역들을 호남으로 묶어주어 새롭게 범주화 하였습니다.

37. 근무지역_경험횟수

38. 근무지역 관련 피처 생성

39. 수도권 대학교

40. 연고지

대학지역과 근무지역이 같은 피처를 생성해주었습니다.

41. 명문대 졸업 유무

42. 근무지역_개별_경력평균

43. 근무지역1별 대학성적, 근무경력

44. 숙련도별 대학성적, 근무경력

45. 숙련도별 대학성적, 근무경력, 출신대학수치

46. 00별 출신대학수치

출신대학, 근무지역, 근무지역1, 숙련도 별 출신대학 수치의 mean, median, var, std, min, max를 groupby한 피처를 생성했습니다.

47. 출신대학별 근무경력mean과 근무경력의 곱, 합, 뺄셈

48. 오타만 있는 피처 생성

근무태그에 오타만 작성되어 있는 피처를 새로 생성해주었습니다.

<모델링>

1. CatBoost:

포문을 통해 numeric_features와 categorical_features를 나누고 binary_features 에 있는 값들을 categorical_features에서 빼, 세가지 종류의 피처로 나누었고 교수님이 제시해준 파이프라인대로 진행했습니다. numeric_features 중 카디날리티가 2인 이진분류 피처들을 스케일링 할 시 성능 저하를 우려해 categorical_features로 변경해 진행했습니다. preprocessing 이후로 KMeans로 군집화를 5, 10 ,15개의 군집으로 모두 시도해보았습니다.

LGBM 모델에서 learning_rate를 줄이고 iterations를 늘리면, 성능이 올라간다는 내용을 https://lightgbm.readthedocs.io/en/latest/Parameters-Tuning.html#for-better-accuracy 에서 확인하고 아이디어를 착안해 learning_rate를 0.05까지 줄이고, iterations를 5000까지 올려 성능 향상을 도모하였습니다.

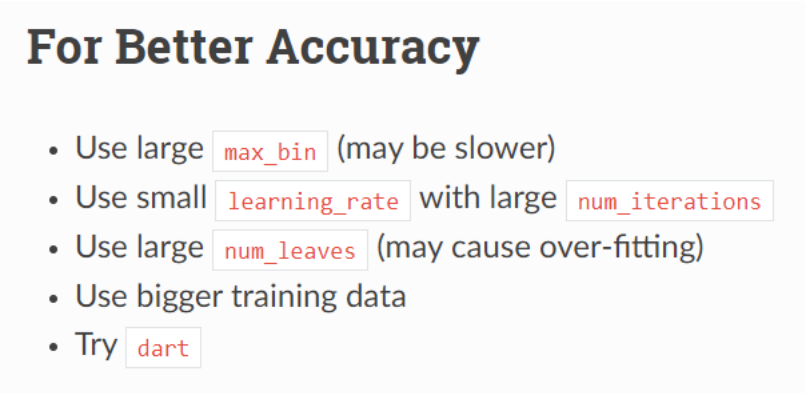
1. CatBoost:

포문을 통해 numeric_features와 categorical_features를 나누고, binary_features 에 있는 값들을 categorical_features에서 뺀 후, 세가지 종류의 피처로 나누었고 교수님이 제시해준 파이프라인대로 진행했습니다. numeric_features 중 카디날리티가 2인 이진분류 피처들을 스케일링 할 시 성능 저하를 우려해 categorical_features로 변경해 진행했습니다.

preprocessing 이후로 KMeans로 군집화를 5, 10 ,15개의 군집으로 모두 시도해보았습니다.

LGBM 모델에서 learning_rate 를 줄이고 iterations 를 늘리면 성능이 올라간다는 내용을

https://lightgbm.readthedocs.io/en/latest/Parameters-Tuning.html#for-better-accuracy



에서 확인하고 아이디어를 착안해 learning_rate를 0.05까지 줄이고 iterations를 5000까지 올려 성능 향상을 도모하였습니다.

2. LGBM:

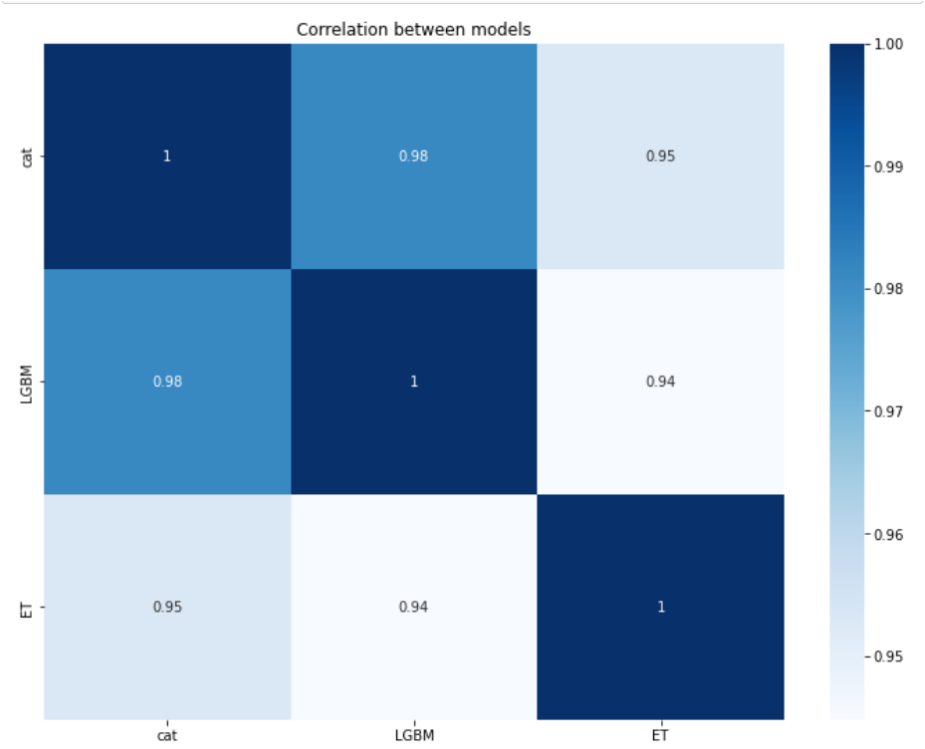
교수님이 주신 LGBM_v5를 거의 따르되, CatBoost에서 성능이 올랐었던 이유로 cv를 15로 올렸고, 사이트에서 확인한대로 learning_rate를 0.01까지 줄이고, iterations를 10000까지 올려 성능 향상을 꾀했습니다. LGBM이 Catboost와 비슷한 부스팅 계열의 모델이라는 생각에 착안해 Catboost단일 최고성능이 나온 데이터에서 피처를 조금 추가하고 전처리도 약간식 다르게 진행한 데이터로 LGBM 모델을 돌려 상관관계를 줄여 앙상블 시 보다 향상된 성능이 나오게 해보았습니다.

3. ExtraTree:

Catboost와 LGBM이 둘다 비슷한 부스팅 모델임을 떠올리고 앙상블시 다른 유형의 회귀 모델을

사용하면 효과가 더 좋을 것이라는 생각이 들어 tree 기반 모델중 가장 좋다고 생각되는 ExtraTree를 사용했습니다. 모델링 방법은 교수님의 LGBM_v5을 참고해 필요한 파라미터들만 변경하고 사용했습니다.

<앙상블>



예전 앙상블 과제할 때의 자료를 활용해 상관계수를 뽑아내고 그래프를 그려 catboost model, LGBM model, Extra model 세 개를 가중평균을 하였습니다. 여러 가지 경우의 수로 돌렸지만 lgbm 모델이 catboost model모델과 상관관계가 예상한대로 매우 높았으므로 Extra model에 어느정도 높은 가중치를 주게 하였습니다. 그중 가장 성능이 좋았던 것은 (cat:0.6, lgbm:0.15, extratree:0.25)와 (cat:0.75, extratree:0.25)로 가중치를 설정했을 때와 (cat:0.75, extratree:0.25)로 가중치를 설정했을때가 가장 성능이 높아 두개를 제출했습니다.