잡케어 추천 알고리즘 경진대회

어린이 탐정단

CONTENTS

잡케어 추천 알고리즘 경진대회 _ 어린이 탐정단

Chapter 01

Feature 생성

지평_제출용 feature

1. **person_rn, contents_rn 활용 :** person_rn 값과 contents_rn 값을 활용하기 위해 두 값을 곱한 피처와 더한 피처를 각각 만들었다.
2. **contents_open_dt 관련**
   - 컨텐츠를 열람한 시간대가 관련이 있을 것이라 생각하여 contents_open_dt 중 시간에 해당하는 부분만 뽑아내어 target encoding 해주었다.
3. **매칭**
   - D 코드의 대-중-소-세 / H 코드의 대-중 / L 코드의 대-중-소-세 코드를 각 코드에 맞게 merging
   - 사람의 선호특성과 컨텐츠의 특성을 비교하여 일치하면 1, 다르면 0을 추출하는 전처리를 진행하였고, person_prefer_f, person_prefer_g 이 두개의 피처는 모두 같은 값을 가지고 있고, id, person_rn, contents_rn, contents_open_dt 피처는 학습에 필요 없을 것으로 생각되어 모두 drop했다.
4. **일치여부에 따른 score 생성 :** D, H코드 1, 2, 3 순위 별 일치스코어 생성(D코드 : 대-중-소-세 / H코드 : 대-중)
5. **L 코드 관련 :** L 코드는 컨텐츠에만 존재하는 속성이므로 특징을 좀 더 두드러지게 하기위해 대-중-소-세 코드 값을 더한 피처, 곱한 피처 생성했다.
6. **D, H 코드 각 순위별 일치 스코어 활용 :** D, H 코드의 각 순위별 일치 스코어를 각 순위별로 곱한 피처, 더한 피처 생성했다.
7. **D, H 코드별 취향 확인 :** 각 코드별 순위별로 차이가 얼마나 나는지 확인하기 위해 1, 2, 3 순위 중 두가지 조합의 diff 값 생성했다.
8. **D, H 코드별 순위별 일치스코어 합**
9. **순서형 변수였던 e코드 관련해서 person의 e코드 값과 contents의 e 코드 값의 차를 구함**
10. **기본적인 피처 엔지니어링 :** D, H, E, L 코드를 이용해 만든 기본 피처들을 각각 곱하고 더하는 피처 생성했다.

만석_제출용 feature　　지평_제출용 feature ＋ α

1. KMeansFeaturizer를 이용한 이용자와 컨텐츠의 연관성

이용자 군집 생성

-> 이용자의 특성을 가지는 피쳐 중 유니크 값을 10개 이하의 피쳐를 대상으로 이용자 군집 생성했다.

(군집 개수는 초반에 임의로 피쳐 수 * 10 개 로 했고, 성능이 가장 좋았다.)

컨텐츠 군집 생성

-> 컨텐츠의 특성을 가지는 피쳐 중 유니크 값을 10개 이하의 피쳐를 대상으로 이용자 군집 생성했다.

(군집 개수는 초반에 임의로 피쳐 수 * 10 개 로 했고, 성능이 가장 좋았음)

이용자 컨텐츠의 조합

-> 군집과 군집의 연관성을 피쳐로 표현하기 위해서 “이용자군집_컨텐츠군집” 으로 문자열 결합을 해준 피쳐를 생성했다.

2. 제공 데이터 이용 피쳐

- 시간데이터에서 월(month), 일(day), 시간(time) 생성했다.
- True/False 로 이루어진 bool 형식의 데이터 d_?_match 의 True 개수, h_?_match의 True 개수

3. 제공데이터 컨텐츠, 이용자 선호도 피쳐 병합을 이용한 특징 도출

컨텐츠의 특성과 이용자 선호도의 특성을 갖는 피쳐 생성

-> 컨텐츠와 이용자 선호도 특징을 나타내는 피쳐들을 각각의 곱을 구한다.

(데이터가 1에서부터 시작하여 등차수열로 단순히 라벨인코딩 데이터이기 때문에 피쳐와 피쳐의 차이를 두기 위해서 제곱, 세제곱을 해주었다.
컨텐츠의 경우는 많은 피쳐가 있어 제곱하여 곱할 경우 너무 큰 숫자가 나오기 때문에 제곱 하지 않았다.)

성현_제출용 feature

1. person_rn, contents_rn 활용 : person_rn 과 contents_rn을 곱한 피처, 더한 피처를 만들어 사람과 컨텐츠 사이의 관계를 알고자 하였다.

2. 컨텐츠 열람 시기 : 컨텐츠를 열람한 시간대가 관련이 있을 것이라 생각하여 contents_open_dt 중 시간에 해당하는 부분만 뽑아내어 target encoding 해주었다.

3. 매칭

- D 코드의 대-중-소-세 / H 코드의 대-중 / L 코드의 대-중-소-세 코드를 각 코드에 맞게 merging
- 사람의 선호특성과 컨텐츠의 특성을 비교하여 일치하면 1, 다르면 0을 추출하는 전처리를 진행하였고, person_prefer_f, person_prefer_g 이 두개의 피처는 모두 같은 값을 가지고 있고, id, person_rn, contents_rn, contents_open_dt 피처는 학습에 필요 없을 것으로 생각되어 모두 drop했다.

4. 선호컨텐츠 속성의 동일 여부 판단

- 컨텐츠속성과 선호 컨텐츠속성에서 겹치는 것이 C, D, E, H 속성이기 때문에 관련된 피쳐들의 동일 여부를 판단하여 bool값으로 추출하는 피쳐를 생성하였다.

5. 의미를 가지지 않는 피쳐 & 그대로 사용 시 leakage일 위험이 있는 피쳐 삭제

- F, G 속성피쳐는 모두 동일한 값을 가지기 때문에 drop하였고, 컨텐츠와 사람의 고유번호 피쳐는 leakage의 위험이 있을 것으로 판단하여 drop하였다.

6. 동일여부 피쳐들의 결합

- D, H속성에 종류가 1, 2, 30이 있는데 이를 선호하는 우선순위로 생각하여 각각 대, 중, 소, 세분류의 동일여부를 판단한 피쳐들의 합을 이용한 피쳐를 만들어주었다. L코드 또한 동일여부를 판단한 피쳐의 합을 이용하여 생성하였다.

7. 동일여부 피쳐들의 결합2 : 6번에서 만든 피쳐들 중 1에 해당하는 D, H를 곱하고 더하는 방식으로 피쳐를 만들어주었다. 이를 2와 3도 적용하였다.

8. 동일여부 피쳐들의 결합3

- 같은 속성에서 우선순위 1, 2, 3의 차이를 보기 위해 각각을 빼주는 방식의 피쳐를 만들어주었다. 또한 같은 속성의 우선순위 1, 2, 3의 합도 만들어주었다.

성현_제출용 feature

9. 동일여부 피쳐들의 결합4 : 위의 방식들을 혼용하여 D와 E, H와 E, L과 E의 조합들도 만들어주었다.

10. J 속성 : J 속성이 1인 행은 J_1이 1~5, 2인 행은 6~10이라는 규칙을 보고 두 피쳐를 곱하여 하나의 J를 만들어주면 좋겠다 생각하여 생성하였다.

11. A 속성 : 10번과 같은 방식으로 하려했으나 A의 경우 곱했을 때 겹치는 경우의 수가 있기 때문에 이를 방지하기 위해 문자열로 변환한 후 결합해주었다.

12. D 속성

- 사람이 선호하는 컨텐츠의 속성과 선택한 컨텐츠의 속성의 관계를 보면 좋은 예측을 할 수 있을 것이라 생각하여 우선순위(1,2,3)별 선호컨텐츠 D속성 대분류 값을 행으로, 컨텐츠 D속성 대분류 값을 열로, value값을 target의 평균으로 하여 pivot_table을 만들어주었다. 그렇게 만든 3개의 pivot_table을 각각의 선호컨텐츠 D속성 대분류 값과 컨텐츠 D속성 대분류 값으로 매핑하여 피쳐를 생성해주었다. 그 다음 이 3개의 피쳐를 제곱하여 가중치를 크게 해 주었다.

13. Person 과 contents의 관계

- 공통데이터 1번에서도 얘기했듯 person과 contents의 관계가 중요할 것이라 판단되어 관계를 그룹화하여 피쳐를 만들고자 하였다.
  허나 여기서 발생하는 문제는 person의 unique값과 contents의 unique값이 train셋과 test셋에 차이가 난다는 점 때문에 test셋에 매핑하는 것이 어려웠다.
  그래서 최대한 test셋에 있는 것이 train에 포함되어 있는 피쳐를 생성하고자 노력하여 그룹화를 진행해보았다. 최적의 조합이지만 NaN값이 나올 수 있기 때문에
  4가지 조합을 통해 NaN값을 대체해주었다. 또한 조합들을 이용하여 target_encoding을 하려 했으나 과적합이 의심되어 size값만 사용하였다.
- 첫번째 조합은 D1의 대분류와 H1의 대분류 속성의 조합이었다. 이를 이용하여 size피쳐를 만들어주었다.
- 두번째는 D1의 대분류와 E속성의 조합으로 위의 과정을 동일하게 수행한 뒤 NaN값을 대체해주었다.
- 세번째는 H1의 대분류와 E속성의 조합으로 NaN값을 대체해주었다.
- 마지막으로 H1의 대분류와 C속성의 조합으로 대체하였다. 나머지 NaN값은 평균값으로 대체해주어 피쳐 생성을 완료하였다.

경서_cluster feature 제출용

"cluster" feature를 만들기 위해 사용

지평_제출용 feature + 민석_제출용 feature

GuassRankScaler

CatboostClassifier
threshold = median

Feature selection

contents_attribute_mul
: 해당 feature 식에 따라 결측치를 채워줌

결측치 처리

PCA

clustering

PCA feature에 Kmeans를 사용하여 군집화 진행
(군집 개수 = 20)

"cluster" feature 생성

딥러닝 기반의 피쳐셀렉션 방법인 shap 사용

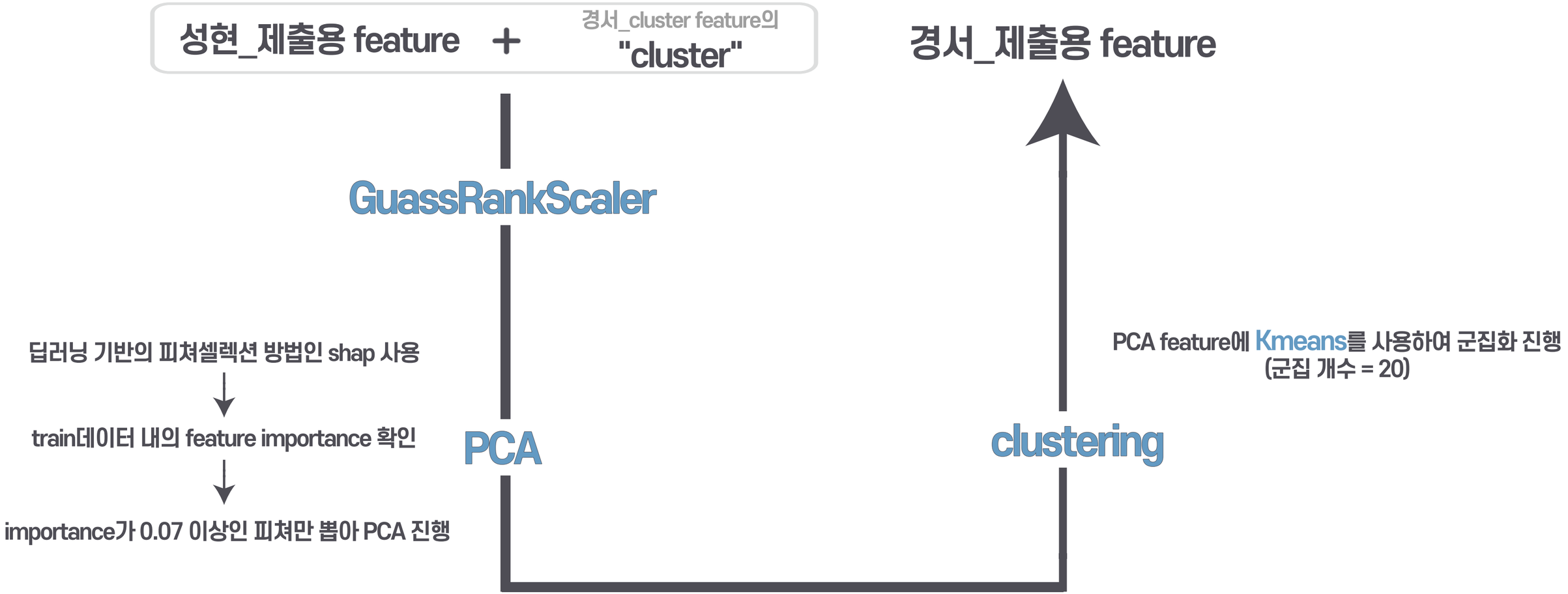
↓

train데이터 내의 feature importance 확인

↓

importance가 0.07 이상인 피쳐만 뽑아 PCA 진행

경서_제출용 feature



Chapter 02

Scaling & Feature selection

Scaling

지평_제출용 feature, 민석_제출용 feature, 성현_제출용 feature에 **GaussRankScler**를 사용했다.

- 지평_제출용 feature의 경우, ('content_L_code_sum', 'content_L_code_mul', 'L_E_mul', 'L_E_sum')는 num feature로, 나머지는 cat feature로 설정하여 진행했다.
- 성현_제출용 feature의 경우, D속성의 조합을 봤던 피쳐는 제외하고 진행했다. 나머지 피쳐는 모두 범주형 피쳐로 반영했다.

※ GaussRankScaler를 사용한 이유 : 성능이 StandardScaler나 MinMaxScaler보다 우수했다.

feature selection

scaling을 진행한 민석_제출용 feature, 경서_제출용 feature는 **SelectFromModel**에 model은 **Catboost**, threshold는 **median**을 사용하여 feature selection을 진행했다.

- 경서_제출용 feature의 경우, ('content_L_code_sum', 'person_prefer_mul', 'contents_attribute_mul', 'prefer_attribute_com', 'D_H_1_mul', 'D_H_2_mul', 'D_H_3_mul', 'D_H_1_sum', 'D_H_2_sum', 'D_H_3_sum') 는 num feature로, 나머지는 cat feature로 설정하여 진행했다.

Chapter 03

Modeling

Modeling

지평_제출용, 민석_제출용, 성현_제출용, 경서_제출용 모두 modeling 방법이 동일하다.

- 데이터의 대부분이 숫자의 형식을 띄고 있지만, 범주형 변수의 성격이 있기 때문에 **OOF**방법을 사용한 **5fold Catboost**를 사용했다.
- Catboost의 경우 하이퍼 파라미터를 조정하는 데에 시간이 매우 오래 걸리며 성능 차이도 크게 나지 않는 것으로 알고 있어 하이퍼 파라미터는 따로 조정하지 않았다.
- 앙상블위해 각 파일 별로 predict 값이 아닌 proba 값을 저장했다.

Post process

- eval metric이 'F1-score'인 특성을 사용하고자 이를 구성하는 recall과 precision의 특징을 이용했고, 모델이 예측한 predict_proba에서 0과 1을 예측하는 threshold를 조정했다.
- 우리는 recall을 높이는 방향(threshold를 낮추는 방향)으로 결정했고, 코드는 국정원요원님의 코드를 참고했다.
- 우리의 시도로는 threshold가 0.35일 때 가장 LB성적이 높았다.

Chapter 04

활용방안

활용방안

Predict_proba 값(이하 value로 통일)에 따른 계층적 적용

threshold = 0.35 기준

1. threshold > value

-> 해당 컨텐츠에 지원할 가능성이 적다

-> 해당 컨텐츠의 대분류코드(D, H)와 다른 것, 지원한 사람의 선호 대분류코드(D, H) 위주로 추천

* 해당 컨텐츠의 대분류코드와 사람의 선호 대분류코드가 같은 경우 예외

2. threshold <= value < 0.5

-> 해당 컨텐츠에 지원할 가능성이 애매하다.

-> 해당 컨텐츠의 대분류코드(D, H)와 지원한 사람의 선호 대분류코드(D, H) 위주로 추천

3. 0.5 <= value < 0.61

-> 해당 컨텐츠에 지원할 가능성이 높은 편이다.

-> 해당 컨텐츠의 대분류-중분류코드(D, H) 위주로 추천

4. 0.61 <= value

-> 해당 컨텐츠에 지원할 가능성이 아주 높은 편이다.

-> 해당 컨텐츠의 대분류-중분류-소분류코드(D, H) 위주로 추천

최종 제출 파일 proba 값의 describe

count	46404.000000
mean	0.482127
std	0.172625
min	0.004815
25%	0.369497
50%	0.501047
75%	0.611281
max	0.904829

감사합니다.

어린이 탐정단 이지평 장성현 최민석 윤경서

주관 데이콘

대회 잡케어 추천 알고리즘 경진대회

발표일시 2022.02.10