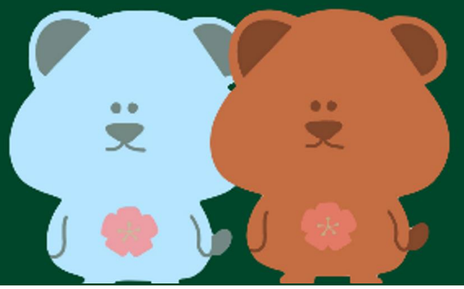
EURON 방학 프로젝트 최종발표회

# 제 2회 코스포 x 데이콘 도서 추천 알고리즘 AI경진대회 분석

입문초급팀_유런도서관

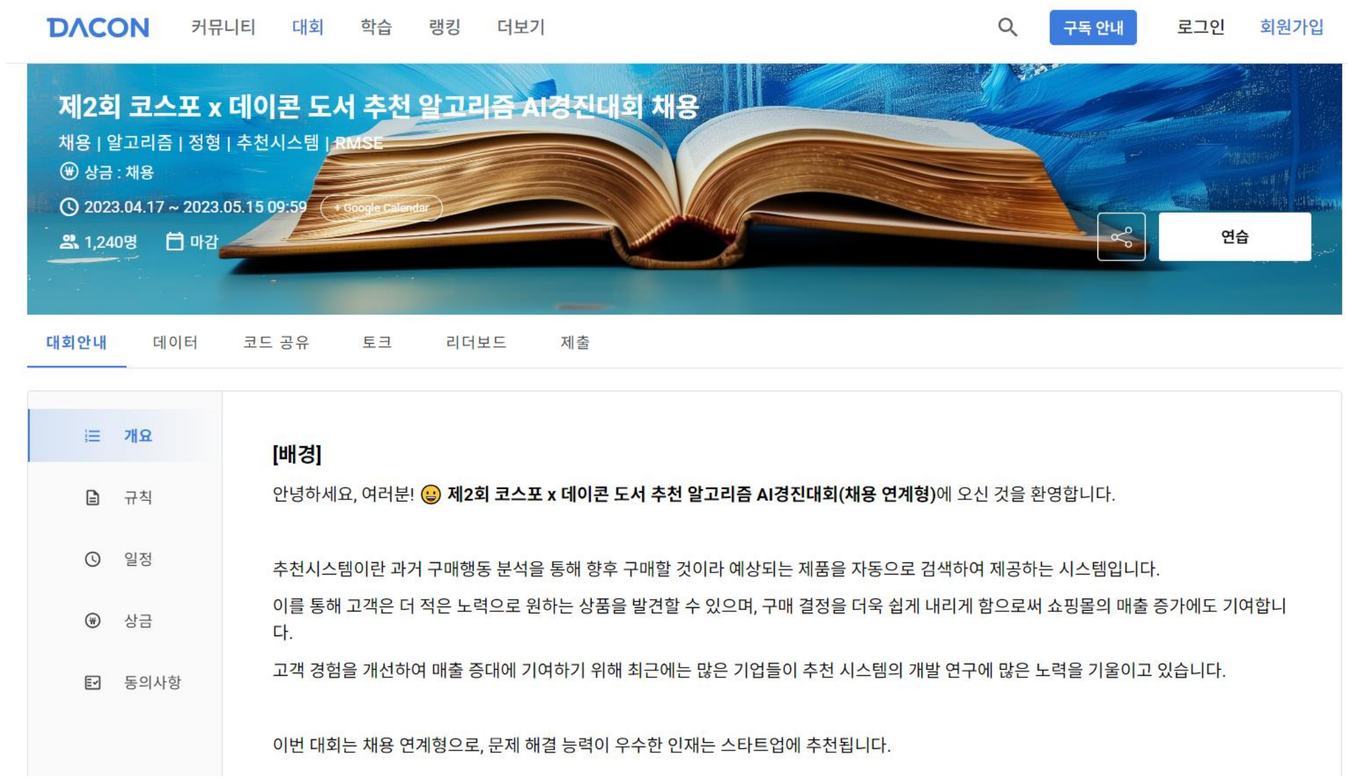
민소연 최유미 허성은



# CONTENTS

# 01 Intro – 대회 소개

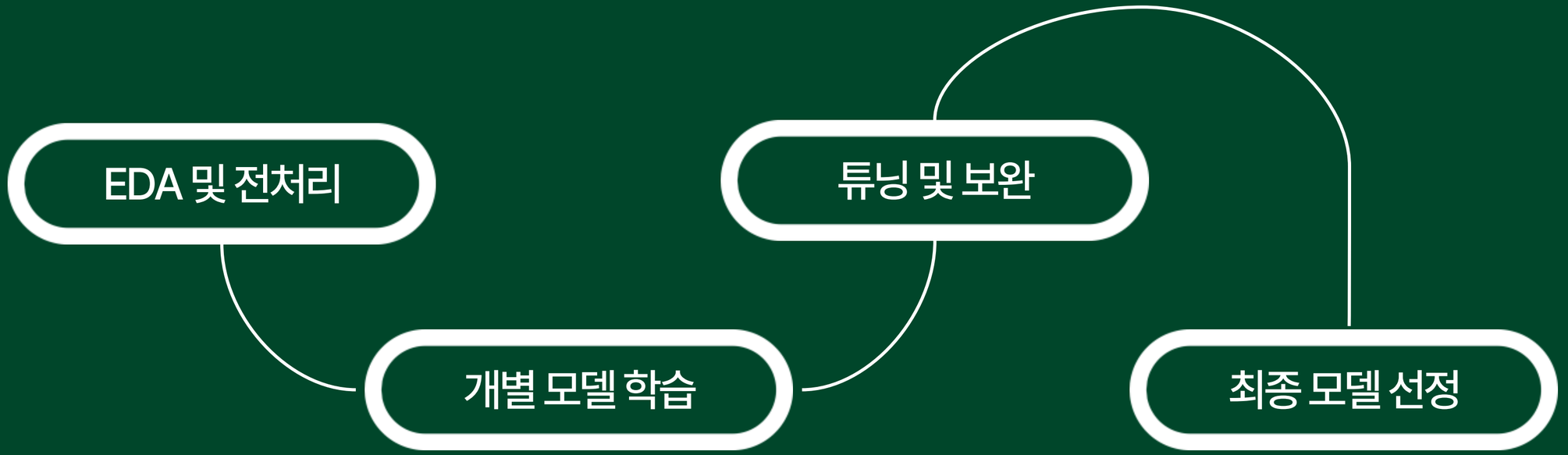
- **제 2회 코스포 x 데이콘 도서 추천 알고리즘 AI경진대회**




- **평가 산식 : RMSE**
  **(Root Mean Squared Error)**

$$\mathrm{RMSE} = \sqrt{MSE} = \sqrt{\frac{\sum(\hat{y}-y)^2}{n}}$$

# 01 Intro – 프로젝트 진행방식




- 총 프로젝트 소요 기간: 약 6주
- 개별 ML 모델 학습 후 가장 성능이 좋은 모델 최종 선정

# 02 EDA 및 전처리

- **칼럼 소개**

```
train.csv [파일]
ID : 샘플 고유 ID
User-ID : 유저 고유 ID
Book-ID : 도서 고유 ID
유저 정보
Age : 나이
Location : 지역
도서 정보
Book-Title : 도서 명
Book-Author : 도서 저자
Year-Of-Publication : 도서 출판 년도 (-1일 경우 결측 혹은 알 수 없음)
Publisher : 출판사
Book-Rating : 유저가 도서에 부여한 평점 (0점 ~ 10점)
단, 0점인 경우에는 유저가 해당 도서에 관심이 없고 관련이 없는 경우
```

# 02 EDA 및 전처리

- 데이터 살펴보기
  - 특수문자 등 정돈되지 않은 상태

```
[ ] train.head()
```

| | ID | User-ID | Book-ID | Book-Rating | Age | Location | Book-Title | Book-Author | Year-Of-Publication | Publisher |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | TRAIN_000000 | USER_00000 | BOOK_044368 | 8 | 23.0 | sackville, new brunswick, canada | Road Taken | Rona Jaffe | 2001.0 | Mira |
| 1 | TRAIN_000001 | USER_00000 | BOOK_081205 | 8 | 23.0 | sackville, new brunswick, canada | Macbeth (New Penguin Shakespeare) | William Shakespeare | 1981.0 | Penguin Books |
| 2 | TRAIN_000002 | USER_00000 | BOOK_086781 | 0 | 23.0 | sackville, new brunswick, canada | Waverley (Penguin English Library) | Walter Scott | 1981.0 | Penguin Books |
| 3 | TRAIN_000003 | USER_00000 | BOOK_098622 | 0 | 23.0 | sackville, new brunswick, canada | Mother Earth Father Sky | Sue Harrison | 1991.0 | Avon |
| 4 | TRAIN_000004 | USER_00000 | BOOK_180810 | 8 | 23.0 | sackville, new brunswick, canada | She Who Remembers | Linda Lay Shuler | 1989.0 | Signet Book |

# 02 EDA 및 전처리

- **데이터 살펴보기**

```
[ ] train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 871393 entries, 0 to 871392
Data columns (total 10 columns):
 #   Column               Non-Null Count   Dtype  
---  ------               --------------   -----  
 0   ID                   871393 non-null  object 
 1   User-ID              871393 non-null  object 
 2   Book-ID              871393 non-null  object 
 3   Book-Rating          871393 non-null  int64  
 4   Age                  871393 non-null  float64
 5   Location             871393 non-null  object 
 6   Book-Title           871393 non-null  object 
 7   Book-Author          871393 non-null  object 
 8   Year-Of-Publication  871393 non-null  float64
 9   Publisher            871393 non-null  object 
dtypes: float64(2), int64(1), object(7)
memory usage: 66.5+ MB
```

- **데이터 개수: 871,393개**
- **피처 개수: 10개**
- **타겟: Book-Rating**

# 02 EDA 및 전처리

## • 칼럼 전처리 – Age(나이)

- train[' Age '].describe()

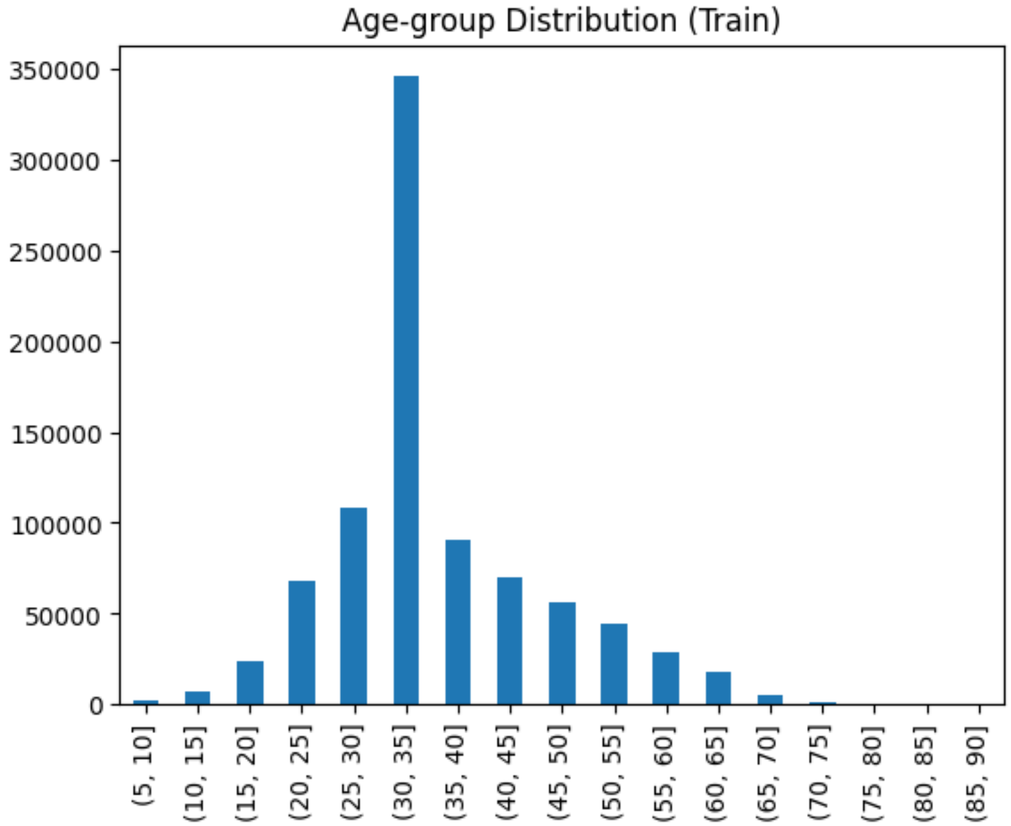
```
count    871393.000000
mean         36.799861
std          12.065509
min           0.000000
25%          31.000000
50%          35.000000
75%          41.000000
max         244.000000
Name: Age, dtype: float64
```

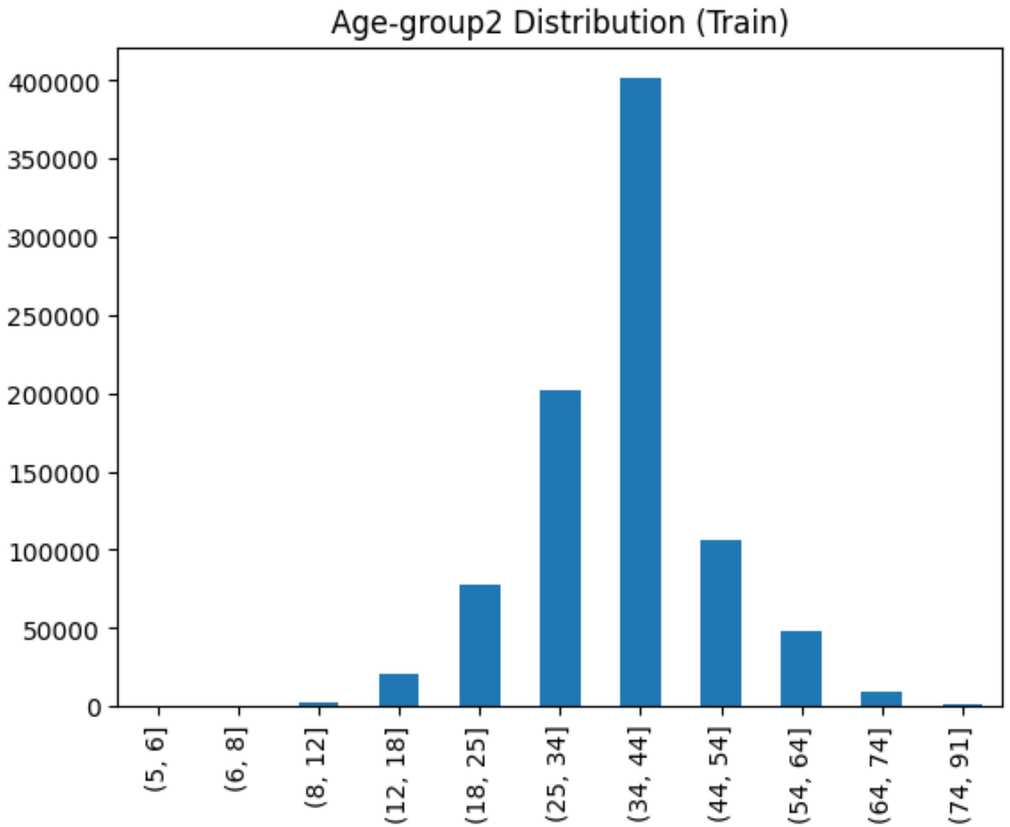
- 0-5세, 90세 이상은 이상치로 간주하고 평균값으로 대체

- 5세 이하 또는 90세 초과인 데이터: 4,497개
- Age (이상치 제거 후) 평균: 36.54세

# 02 EDA 및 전처리

- **칼럼 전처리 – Age(나이)**
  - **분포가 다양하기 때문에 2가지 기준으로 범주화 (5세 간격과 미국 노동통계국 자료 기반 그룹)**



# 02 EDA 및 전처리

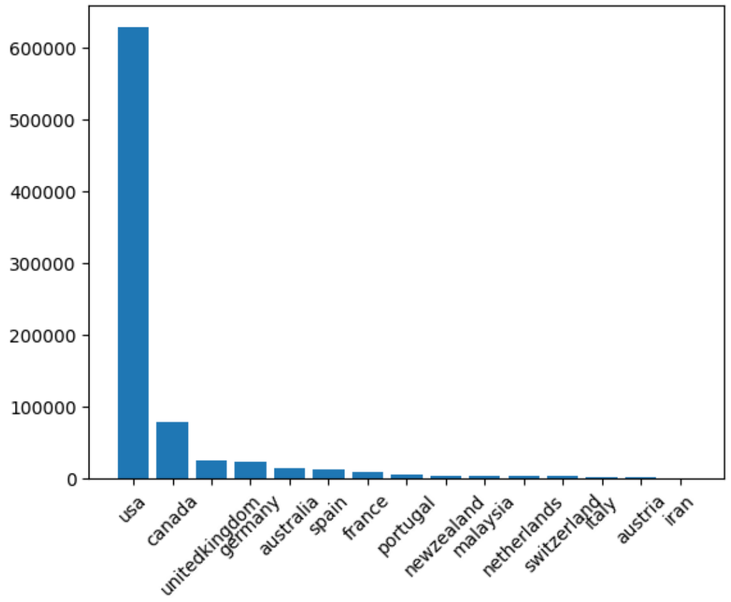
- **칼럼 전처리 – Location(지역)**
  - 쉼표를 기준으로 도시, 주, 나라로 구분되어 있었기 때문에 분리해서 각각 새로운 칼럼으로 추가
  - 분리 후 결측치가 생기는 문제, 도시에 나라 이름이 들어있는 등 제대로 분류되지 않는 문제 발생

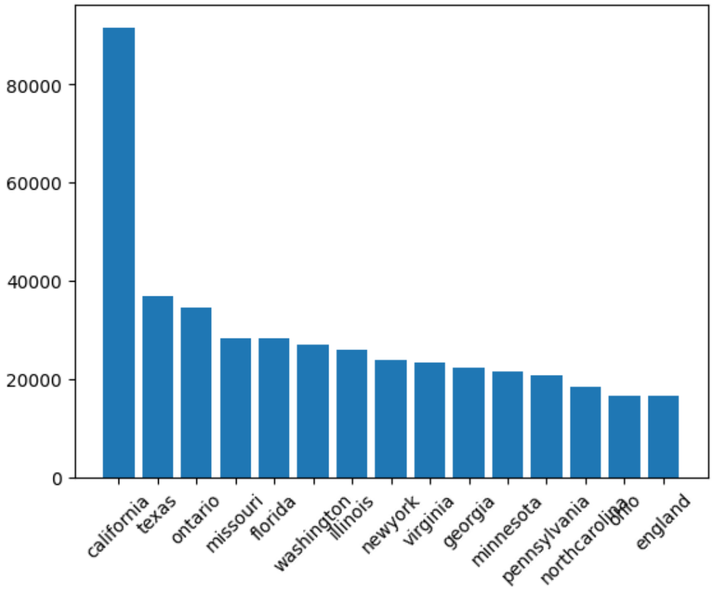
➡ 각 칼럼별 가장 많이 나타나는 최빈값으로 결측치 대체
분류 문제는 해결하지 X

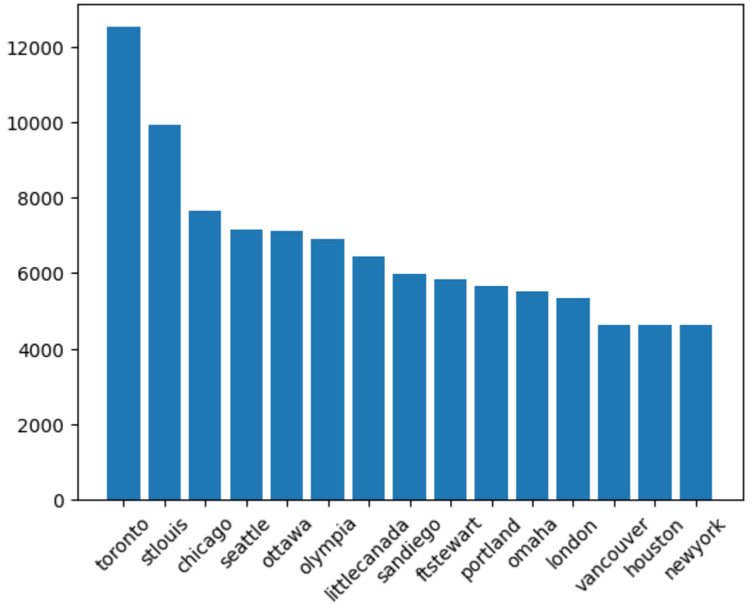
```
ID                        0
User-ID                   0
Book-ID                   0
Book-Rating               0
Age                       0
Book-Title                0
Book-Author               0
Year-Of-Publication       0
Publisher                 0
city                  14367
state                 37207
country               32327
dtype: int64
```

# 02 EDA 및 전처리

- 칼럼 전처리 – Location(지역)
  - 나라 최빈값: USA
  - 주 최빈값: California
  - 도시 최빈값: Toronto



# 02 EDA 및 전처리

- 칼럼 전처리 – 기타 칼럼

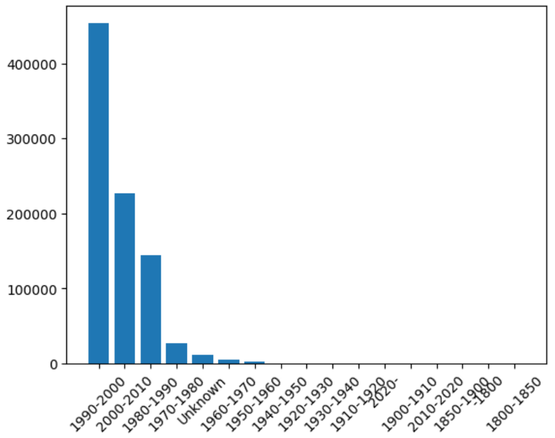
- Publisher(출판사), Book-Author(저자)
- 특정 출판사 또는 저자에게 쏠림 현상이 있었지만 따로 전처리 없이 소문자 통일, 특수문자 제거, 띄어쓰기 제거만 기본으로 진행

- 출판사: 15,505 → 14,840로 종류 감소
- 저자: 88,080으로 전처리 전후 변화 없음

# 02 EDA 및 전처리

- 칼럼 전처리 – 기타 칼럼

- Title(도서명), Year-Of-Publication(출판년도)
  - 도서명은 제목과 부제목이 혼재되어 있어 Main-Title과 Sub-Title로 분리
  - 정규표현식으로 대소문자 통일, 특수문자 제거
  - 출판 년도는 -1일 경우 결측 혹은 알 수 없음을 의미하는데 Unknown으로 대체시킴
  - 나머지 년도는 10년도씩 끊어서 범주화




1990-2000년대가 가장 많음

```
train['Pub_gb'].value_counts()
```

```
1990-2000    453785
2000-2010    227286
1980-1990    144594
1970-1980     26703
Unknown       11515
1960-1970      4591
1950-1960      2169
1940-1950       266
1920-1930       157
1930-1940       118
1910-1920       112
2020-            51
1900-1910        24
2010-2020        16
1850-1900         3
-1800             2
1800-1850         1
Name: Pub_gb, dtype: int64
```

# 개별 모델 선정

## 추천시스템끼리 앙상블

민소연

- 선형 회귀
- Surprise 라이브러리

다양한 모델 조합, 성능 향상을 위해 스태킹 앙상블 적용

## Catboost 단일모델

최유미

- Catboost + Optuna

boosting 앙상블 기법 사용

범주형 변수를 다룰 때 용이하기 때문에 선정

## 추천시스템 + Catboost

허성은

- Surprise + Catboost

RMSE 결과와 RAM 초과 문제를 고려하여 **BaselineOnly, CoClustering, SVD** 선정

# 03 개별 모델 분석

- 민소연 – 선형회귀
  - 유저별 평점 평균, 도서별 평점 평균을 구한 칼럼을 추가해 예측하면 성능이 2점대로 좋은 성능이 나와서 계속 시도해본 모델
  - User-ID, Book-ID를 제외한 나머지 피처도 선형 회귀를 통해 예측하고 싶었지만, 레이블 문제도 있고 피처와 타겟 간의 선형적인 관계를 찾지 못해서 폐기

# 03 개별 모델 분석

- 민소연 – Surprise 라이브러리
  - 서프라이즈 라이브러리 중 다양한 모델 조합을 시도해서 스태킹 앙상블 기법 적용

시도 1

개별 모델: SlopeOne, SVD (2개)
메타 모델: LGBMRegressor 조합

- RMSE: 3.3949
- 데이콘 RMSE: 3.69511 / 185등
- SlopeOne의 경우 사용자가 일반적으로 평점을 높게 주거나 낮게 주는 경향이 있는 경우를 고려하여 평점 편향 보정 시도, 그러나 성능 낮음

# 03 개별 모델 분석

- 민소연 – Surprise 라이브러리

시도 2

개별 모델: BaselineOnly, SVD, CoClustering(3개)
메타 모델: LGBMRegressor 조합

- RMSE: 3.3366 (-0.0583)
- 데이콘 RMSE: 3.37900 / 64등 (121등 상승)

- 이후 하이퍼 파라미터 조정도 시도했으나 성능이 오히려 떨어짐
- 두 번째 시도 조합으로 결정

# 03 개별 모델 분석

- 허성은 – Surprise + Catboost
  - 추천 시스템의 경우, 신규 유저나 특정한 유저들에 대한 정보가 충분하지 않아 적합한 추천을 하지 못하는 cold start 문제가 발생함.

➡ 신규 사용자, 신규 아이템을 제외한 데이터(test_exist)에만 추천시스템을 적용하고 나머지(test_new)에 대해서는 CatBoost를 적용

# 03 개별 모델 분석

- 허성은 – Surprise + Catboost

| Algorithm | test_rmse | fit_time | test_time |
|---|---|---|---|
| BaselineOnly | 3.381113 | 0.123505 | 0.249944 |
| CoClustering | 3.478568 | 1.582240 | 0.327054 |
| SlopeOne | 3.483816 | 0.340898 | 1.689941 |
| KNNWithMeans | 3.495180 | 0.475094 | 2.774705 |
| KNNBaseline | 3.502716 | 0.505188 | 3.379397 |
| KNNWithZScore | 3.513496 | 0.484249 | 3.364993 |
| SVD | 3.524309 | 1.817457 | 0.552979 |
| KNNBasic | 3.726990 | 0.345575 | 2.677626 |
| SVDpp | 3.785625 | 10.210550 | 4.804407 |
| NMF | 3.916815 | 1.627800 | 0.278679 |
| NormalPredictor | 4.686601 | 0.076015 | 0.159820 |

- surprise 라이브러리 알고리즘 각각 cross validation 교차검증을 하여 RMSE를 구했음.
- rmse 결과와 RAM 초과 문제를 고려하여 BaselineOnly, CoClustering, SVD 세 알고리즘을 선정하여 예측을 수행함.

# 03 개별 모델 분석

- 허성은 – Surprise + Catboost
  - SVD
    - Public score: 3.4588290598
    - Private score: 3.4773677411
  - CoClustering
    - Public score: 3.6639490726
    - Private score: 3.6707599892
  - BaselineOnly
    - Public score: 3.3776050111
    - Private score: 3.3914113165

BaselineOnly + Catboost 조합

**3.37761 / 63등**

# 03 개별 모델 분석

- 최유미 – Catboost + Optuna

**시도 1**

데이터 전처리 X, feautres=['User-ID', 'Location', 'Book-Title', 'Book-Author', 'Publisher'] 만 이용함

- Best Score: 3.2622

**시도 2**

데이터 전처리 간단히 O, Label Encoding O, 전체 피처 사용

Best Score: 3.2541

**시도 3**

Book-Title까지 전처리, Label Encoding O, 전체 피처 사용

Best Score: 3.2499

- 이하 하이퍼 파라미터 조정, 교차 검증 등 다양한 시도

# 03 개별 모델 분석

- 최유미 – Catboost + Optuna

**Best Score: 3.24989**

**Best trial: {**

- **'n_estimators': 5411,**
- **'od_wait': 1699,**
- **'learning_rate': 0.07707,**
- **'reg_lambda': 85.29821,**
- **'random_strength': 44.31430,**
- **'depth': 12,**
- **'min_data_in_leaf': 24**

**}**

Catboost + Optuna 조합

**3.31859 / 47등**

# 개별 모델 성능 비교

## 추천시스템끼리 앙상블

민소연

- Surprise 스태킹

  개별 모델: BaselineOnly, SVD, CoClustering(3개)
  메타 모델: LGBMRegressor

- 3.37900 / 64등

## Catboost 단일모델

최유미

- Catboost + Optuna
- 3.31859 / 46등

## 추천시스템 + Catboost

허성은

- Surprise + Catboost
- 3.37761 / 63등

# 04 최종 모델

- Catboost+Optuna
- Catboost 단일 모델을 최종 모델로 결정

- Optuna란?
  - 간단하고 빠르며 많은 머신러닝 프레임워크에 사용 가능해 널리 사용되는 하이퍼 파라미터 튜닝 라이브러리
  - 모델에서 objective_cat이라는 목적함수를 선정하여 튜닝 범위와 모델을 지정 후 사용하였습니다.
  - create_study 함수로 study 오브젝트를 생성하고 최적화를 수행합니다. 정학도 측정 함수에 따라 저희는 최소화하는 방향으로 수행하였습니다.

# 04 최종 모델

- Catboost란?
  - 다른 GBM에 비하여 과적합이 적고 범주형 변수가 많은 경우 효율적이라는 특징이 있습니다.

  - 따로 인코딩 작업을 하지 않아도 내장된 인코딩 방식을 사용하지만, 이번 변수의 경우 범위가 너무 많아 따로 label encoding을 수행 후 모델을 학습시켰습니다.
  - 다만 Null 데이터 처리가 되지 않고, 속도가 느리다는 단점이 있습니다

# 04 Discussion

## 최유미

- 범주형 변수를 알아서 처리해주고, 과적합 및 파라미터 최적화도 내부적으로 잘 되어있는 모델이여서 CatBoost를 잘 활용하면 더 매력적으로 활용할 수 있을 것 같다.

- 이번에는 범주형 변수의 범위가 너무 크고, 시간 및 용량 문제로 인하여 CatBoost를 100% 활용하지 못한 아쉬움이 있었다.

- 다만 CatBoost와 Optuna라는 새로운 라이브러리를 사용해 효율적으로 좋은 결과를 낼 수 있어 좋았다. 여러 오류들을 겪었는데, CatBoost의 고유한 특징을 제대로 이해하지 못한 것이였다.

# 04 Discussion

## 민소연

- Location 칼럼을 전처리 할 때, 제대로 분류되지 않은 것들을 다시 분류하지 못한 것과 Cold Start문제를 해결하지 못한 것이 아쉬웠지만 Surprise 라이브러리로 간단하게 모델을 학습시켜볼 수 있어서 좋았다.

- 그동안 배운 내용인 추천 시스템, 스태킹 앙상블, 하이퍼 파라미터 튜닝 등을 복습해볼 수 있던 시간이어서 좋았고 100위권 안이라는 성과를 얻어서 만족스럽다.

# 04 Discussion

## 허성은

- 뛰어난 예측성능을 가진 결과물은 만들지 못했지만, 정규세션 마지막에 다뤘던 추천시스템을 실제 데이터에 적용해볼 수 있어서 좋았다.

- 추천시스템의 단점인 cold start 문제를 단순히 성능이 비교적 좋은 CatBoost 결과로 대체하여 결과물을 만들었는데, 다른 해결방식(lightfm, cold start 문제의 원인들을 파악하고 추천 시스템의 종류에 따라 다르게 대응 등)을 적용해보지 못한 아쉬움이 있다.

# 04 Discussion

## Catboost 단일 모델 > Surprise + Catboost > Surprise만 스태킹

- 결론적으로 Catboost 단일 모델에 하이퍼 파라미터 튜닝을 거친 결과물이 가장 좋은 성능을 얻은 것을 보면서 데이터에 범주형 변수가 많았던만큼, Catboost 모델이 범주형 변수에 강하다는 사실을 다시 한 번 확인할 수 있었다.

- 모델 학습 과정에서 각자 다양한 아쉬움이 있었지만 팀원들 모두 47등, 63등, 64으로 100위권 안에 드는 성적이 나와서 의미 있는 시간이었다.

# Thank you

**발표를 들어주셔서 감사합니다.**

이화여자대학교 인공지능 동아리 Euron

입문초급팀 유런도서관
민소연, 최유미, 허성은