빅데이터 기반 AI 응용 솔루션 개발자 전문과정

교과목명 : 머신러닝 응용

- 평가일 : 22.10.28
- 성명 : 신창훈
- 점수 : 80

```
import pandas as pd
ratings = pd.read_csv('회원별_소분류_구매건수_구매일자_완성.csv', encoding='cp949')
```

```
ratings.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8894978 entries, 0 to 8894977
Data columns (total 4 columns):
 #   Column   Dtype
---  ------   -----
 0   CUSTNO   int64
 1   SGROUP   object
 2   ITEM     int64
 3   PURDATE  int64
dtypes: int64(3), object(1)
memory usage: 271.5+ MB
```

```
import numpy as np
ratings.ITEM = round(np.log1p(ratings.ITEM),1)
```

```
# ratings.PURDATE = 1
```

```
ratings.ITEM.unique()
```

```
array([0.7, 1.9, 1.1, 2.4, 1.8, 1.4, 2.1, 2.8, 2.2, 2.6, 1.6, 3.5, 3.9,
       3. , 3.8, 2.5, 2.7, 3.1, 2.3, 2.9, 3.3, 3.7, 4.4, 3.2, 3.4, 3.6,
       5.7, 4.8, 4.5, 4.2, 5. , 4. , 4.1, 4.6, 5.4, 4.9, 4.7, 4.3, 5.3,
       5.1, 6.3, 5.5, 5.6, 5.9, 5.2, 5.8, 6.2, 6.1, 6. , 6.5, 6.4, 6.6])
```

```
ratings.ITEM.value_counts()
```

```
0.7    4650858
1.1    1527912
1.4     762096
1.6     457216
1.8     303900
1.9     215037
2.1     159786
2.2     122323
2.6      96991
2.3      96371
2.4      77534
```

```
2.5      63778
2.8      60042
2.9      45154
2.7      37899
3.0      34847
3.1      27580
3.3      25689
3.2      21805
3.4      19129
3.6      14867
3.5      14845
3.7      11033
3.8       8397
3.9       7953
4.0       6816
4.1       5198
4.2       4275
4.3       3060
4.4       2611
4.5       2119
4.6       1767
4.7       1371
4.8       1048
4.9        816
5.0        639
5.1        483
5.2        379
5.3        332
5.4        242
5.5        204
5.6        136
5.7        121
5.8         92
5.9         85
6.0         55
6.1         40
6.2         19
6.3         16
6.4          6
6.5          4
6.6          2
Name: ITEM, dtype: int64
```

```
ratings.shape
```

```
(8894978, 4)
```

```
ratings.head()
```

	CUSTNO	SGROUP	ITEM	PURDATE

```
# ratings_noh.csv 파일로 언로드 시 인덱스와 헤더를 모두 제거한 새로운 파일 생성.
ratings.to_csv('회원별_소분류_구매건수_구매일자_완성_noh.csv', index=False, header=False)
```

```
      2       1  A010402_3     2  20150520
```

```
from surprise.dataset import DatasetAutoFolds
from surprise import Reader, Dataset

reader = Reader(line_format='user item rating timestamp', sep=',', rating_scale=(0.7, 6.6)
# reader = Reader(line_format='user item rating', sep=',', rating_scale=(0.7, 6.3))
# DatasetAutoFolds 클래스를 회원별_소분류_구매건수_완성_A_noh.csv 파일 기반으로 생성.
data_folds = DatasetAutoFolds(ratings_file='회원별_소분류_구매건수_구매일자_완성_noh.csv', re

# 전체 데이터를 학습 데이터로 생성함.
trainset = data_folds.build_full_trainset()
```

```
import pandas as pd
from surprise import Reader, Dataset

from surprise import SVD
from surprise import accuracy
from surprise.model_selection import train_test_split
```

```
# algo = SVD(n_epochs=20, n_factors=50, random_state=0)
algo = SVD(n_factors=50, random_state=0)
algo.fit(trainset)
```

```
<surprise.prediction_algorithms.matrix_factorization.SVD at 0x286f35810a0>
```

```
# 상세 속성 정보 DataFrame 로딩
items = pd.read_csv('소분류코드_대분류명_중분류명_소분류명_상품등급.csv', encoding='cp949')
```

```
items.rename(columns={'소분류코드':'SGROUP'}, inplace=True)
items.head()
```

	SGROUP	대분류명	중분류명	소분류명	상품등급
0	A010101_1	일상용품	일용잡화	위생세제	저가
1	A010101_2	일상용품	일용잡화	위생세제	중가
2	A010101_3	일상용품	일용잡화	위생세제	고가
3	A010102_1	일상용품	일용잡화	휴지류	저가
4	A010102_2	일상용품	일용잡화	휴지류	중가

```
def get_not_buy_surprise(ratings, pro_df, userId):

    buying_pro = ratings[ratings['CUSTNO']==userId]['SGROUP'].tolist()
```

```
    # 모든 상품의 SGROUP를 리스트로 생성.
    total_pro = pro_df['SGROUP'].tolist()

    # 모든 상품의 SGROUP 중 이미 구매한 상품의 SGROUP를 제외한 후 리스트로 생성
    not_buy_pro = [ pro for pro in total_pro if pro not in buying_pro]
    print('구매했던 제품 수:', len(buying_pro), '추천 대상 제품 수:', len(not_buy_pro), '전체

    return not_buy_pro

not_buy_pro = get_not_buy_surprise(ratings, items, 9)
```

```
 구매했던 제품 수: 180 추천 대상 제품 수: 12978 전체 제품 수: 13158
```

```
def recomm_pro_by_surprise(algo, userId, not_buy_pro, top_n=10):
    # 알고리즘 객체의 predict() 메서드를 평점이 없는 영화에 반복 수행한 후 결과를 list 객체로 저
    predictions = [ algo.predict(str(userId), str(proid)) for proid in not_buy_pro ]

    # prediction list 객체는 surprise의 Predictions 객체를 원소로 가지고 있음.
    # [Prediction(uid='9', iid='1', est=3.69), Prediction(uid='9', iid='2', est=2.98),,,,]

    # 이를 est 값으로 정렬하기 위햇 아래의 sortkey_est 함수를 정의함.
    # sortkey_est 함수는 list 객체의 sort() 함수의 키 값으로 사용되어 정렬 수행.
    def sortkey_est(pred):
        return pred.est

    # sortkey_est() 반환값의 내림 차순으로 정렬 수행하고 top_n개의 최상위 값 추출.
    predictions.sort(key=sortkey_est, reverse=True)
    top_predictions = predictions[:top_n]

    # top_n으로 추출된 상품의 정보 추출. 상품 코드, 추천 예상 지수, 분류별 삼품명
    # est - 예상 평점
    top_pro_ids = [ pred.iid for pred in top_predictions ]
    top_pro_rating = [ pred.est for pred in top_predictions ]
    top_pro_bgroup = items[items.SGROUP.isin(top_pro_ids)]['대분류명']
    top_pro_mgroup = items[items.SGROUP.isin(top_pro_ids)]['중분류명']
    top_pro_sgroup = items[items.SGROUP.isin(top_pro_ids)]['소분류명']
    top_pro_scode = items[items.SGROUP.isin(top_pro_ids)]['상품등급']

    top_pro_preds = [ (id, rating, bgroup, mgroup, sgroup, scode) for id, rating, bgroup,

    return top_pro_preds

not_buy_pro = get_not_buy_surprise(ratings, items, 1000)
top_pro_preds = recomm_pro_by_surprise(algo, 1000, not_buy_pro, top_n=10)
```

```
print('#### Top-10 추천 제품 리스트 ####')
for top_pro in top_pro_preds:
    print(top_pro[0], ':', top_pro[1], top_pro[2], top_pro[3], top_pro[4], top_pro[5])
```

```
 구매했던 제품 수: 519 추천 대상 제품 수: 12639 전체 제품 수: 13158
 #### Top-10 추천 제품 리스트 ####
 C070101_2 : 3.7405096608825334 신선식품 엽채류 쌈채소류 중가
 C070102_2 : 3.558504553489968 신선식품 과채류 파프리카/피망 중가
 C170701_2 : 3.0930825269597637 가공식품 계란류 가공계란 중가
 C030206_2 : 2.840582897001256 신선식품 기타수산 기타어류 중가
 C040601_2 : 2.824878233728178 가공식품 기타조리식품 기타냉장조리 중가
 C170701_1 : 2.746878285296976 가공식품 우유 일반우유 중가
 C030902_2 : 2.6685614105549047 가공식품 우유 기능성우유 중가
 C060504_2 : 2.4444130147919365 가공식품 생수 생수 고가
 C110101_3 : 2.3393128239193017 교육/문화용품 가정잡화 생활잡화균일가 저가
 C030107_2 : 2.2949184351229706 교육/문화용품 가정잡화 생활잡화균일가 중가
```

```
not_buy_pro = get_not_buy_surprise(ratings, items, 10)
top_pro_preds = recomm_pro_by_surprise(algo, 10, not_buy_pro, top_n=10)

print('#### Top-10 추천 제품 리스트 ####')
for top_pro in top_pro_preds:
    print(top_pro[0], ':', top_pro[1], top_pro[2], top_pro[3], top_pro[4], top_pro[5])
```

```
 구매했던 제품 수: 657 추천 대상 제품 수: 12501 전체 제품 수: 13158
 #### Top-10 추천 제품 리스트 ####
 C070101_2 : 3.6444584258501433 기타 주유소 주유소 중가
 C070102_2 : 3.2159710802912573 일상용품 종량제봉투 재사용봉투 중가
 C040601_2 : 2.6337098211267223 신선식품 과채류 파프리카/피망 중가
 C170701_2 : 2.621866320007357 가공식품 계란류 가공계란 중가
 C030206_2 : 2.4956129270306873 신선식품 기타수산 기타어류 중가
 B080601_2 : 2.349470363403591 가공식품 우유 일반우유 중가
 B210802_2 : 2.2787822307934773 가공식품 우유 기능성우유 중가
 C030902_2 : 2.2513919476615554 가공식품 생수 생수 고가
 C110101_3 : 2.2428117650306456 교육/문화용품 가정잡화 생활잡화균일가 저가
 C170701_1 : 2.23356641968076 교육/문화용품 가정잡화 생활잡화균일가 중가
```

```
not_buy_pro = get_not_buy_surprise(ratings, items, 1)
top_pro_preds = recomm_pro_by_surprise(algo, 1, not_buy_pro, top_n=10)

print('#### Top-10 추천 제품 리스트 ####')
for top_pro in top_pro_preds:
    print(top_pro[0], ':', top_pro[1], top_pro[2], top_pro[3], top_pro[4], top_pro[5])
```

```
 구매했던 제품 수: 374 추천 대상 제품 수: 12784 전체 제품 수: 13158
 #### Top-10 추천 제품 리스트 ####
 A010401_2 : 3.210493802947503 가공식품 축산가공 유제품 중가
 A010402_2 : 3.0180877601422793 신선식품 농산물 청과 중가
 A010404_2 : 2.97150176749285 신선식품 농산물 채소 중가
 A010302_2 : 2.94912843237212 신선식품 농산물 유기농채소 중가
 A020302_2 : 2.3858429737277187 가공식품 농산물 농산가공 중가
 A010608_2 : 2.368375827285293 가공식품 가공식품 일반가공식품 중가
 B080601_2 : 2.3375171012988316 가공식품 차/커피 수입식품 중가
 A010403_2 : 2.318122556779451 일상용품 화장품 기초 화장품 저가
 A020302_1 : 2.3101151154011905 일상용품 화장품 기초 화장품 중가
 A011004_2 : 2.27044330832176 기타 주유소 주유소 중가
```

Colab 유료 제품 - 여기에서 계약 취소