분류 - 영화추천

#01. 패키지 참조

```
import warnings
warnings.filterwarnings('ignore')

import os
import numpy as np
from pandas import read_csv, DataFrame, pivot_table, merge
from matplotlib import pyplot as plt
import seaborn as sb

from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import cross_val_score
```

#02. 데이터 가져오기

jupyter가 참조하고 있는 현재 디렉토리 확인

```
print(os.getcwd())
```

```
c:\Users\leekh\J&Y Dropbox\메가IT수업자료\G. 머신러닝\02.Sklearn
```

영화 데이터 가져오기

실 분석용은 아니다. 분석 후 결과값을 맵핑시키기 위한 데이터이다.

```
origin_mv = read_csv("netflix\\Netflix_Dataset_Movie.csv", encoding='utf-8')
origin_mv.head()
```

	Movie_ID	Year	Name
0	1	2003	Dinosaur Planet
1	2	2004	Isle of Man TT 2004 Review
2	3	1997	Character
3	4	1994	Paula Abdul's Get Up & Dance
4	5	2004	The Rise and Fall of ECW

별점 데이터 가져오기

```
origin_rating = read_csv("netflix\\Netflix_Dataset_Rating.csv", encoding='utf-8')
origin_rating.head()
```

	User_ID	Rating	Movie_ID
0	712664	5	3
1	1331154	4	3
2	2632461	3	3
3	44937	5	3
4	656399	4	3

#03. 데이터 전처리

별점 데이터 재구조화

각 영화를 컬럼으로, 사용자 번호를 인덱스로 하는 피벗 테이블을 구성한다.

> 다소 시간이 오래 걸림

```
movie_users = pivot_table(origin_rating, index='Movie_ID', columns='User_ID', values=
movie_users.head()
```

User_ID	6	7	79	97	134	169	183	188	195	199	...	2649308	2
Movie_ID													
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	N
8	NaN	5.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	N
16	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	N
17	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	N
18	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	4.0	N

5 rows × 143458 columns

데이터 정제

결측치는 해당 영화를 보지 않은 것으로 간주하고 0으로 대체한다.

```
movie_users.fillna(0, inplace=True)
movie_users.head()
```

User_ID	6	7	79	97	134	169	183	188	195	199	...	2649308	2649328	2
Movie_ID														
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.
8	0.0	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.

User_ID	6	7	79	97	134	169	183	188	195	199	...	2649308	2649328	2
Movie_ID														
16	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.
17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.
18	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	4.0	0.0	0.

5 rows × 143458 columns

독립변수, 종속변수 분리

x는 데이터프레임 자체.

y는 데이터프레임의 인덱스

```
x = movie_users.copy()
y = movie_users.index
x.shape, y.shape
```

```
((1350, 143458), (1350,))
```

#04. 분류 모델 구축

단일 수행

```
k = 5
knn = KNeighborsClassifier(n_neighbors=k)
knn.fit(x, y)
y_pred = knn.predict(x)
score = accuracy_score(y, y_pred)
print("분류 정확도: {:.2f}%".format(score))
```

```
분류 정확도: 0.17%
```

최적의 k찾기

```
k_range = range(1, len(x.columns))
k_scores = []

for k in k_range:
    knn = KNeighborsClassifier(n_neighbors=k)
    #score = cross_val_score(knn, x, y, cv=100).mean()
    knn.fit(x, y)
    y_pred = knn.predict(x)
    score = accuracy_score(y, y_pred)

    if np.isnan(score):
        break
```

```
    k_scores.append(score)

k_scores
```