1_data_cleanning

December 10, 2023

0.0.1 Data cleanning on anime & anime_with_synopsis

[ ]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
import pickle
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
import re
from surprise import dump
from surprise import Dataset, Reader, SVD, KNNBasic
from surprise.model_selection import cross_validate
from sklearn.model_selection import train_test_split
from IPython.display import display
import ipywidgets as widgets
from surprise.model_selection import GridSearchCV

from proj_util import *
```

[ ]:
```
PATH = './Data'

anime = pd.read_csv(f'{PATH}/anime.csv')
anime_with_synopsis = pd.read_csv(f'{PATH}/anime_with_synopsis.csv')
```

[ ]:
```
replace_dict = {'Unknown': np.nan}
anime_with_synopsis = anime_with_synopsis.replace(replace_dict)
anime = anime.replace(replace_dict)
```

change genre into itemset

[ ]:
```
anime_with_synopsis['Genres'] = anime_with_synopsis['Genres'].apply(lambda x:␣
↪set((str(x)).strip().split(', ')))
anime['Genres'] = anime['Genres'].apply(lambda x: set((str(x)).strip().split(',␣
↪')))
```

[ ]:
```
genres = set()
for i in anime_with_synopsis['Genres'].to_numpy():
```

```
    genres = genres.union(i)
```

[ ]:
```
genres
```

[ ]:
```
{'Action',
 'Adventure',
 'Cars',
 'Comedy',
 'Dementia',
 'Demons',
 'Drama',
 'Ecchi',
 'Fantasy',
 'Game',
 'Harem',
 'Historical',
 'Horror',
 'Josei',
 'Kids',
 'Magic',
 'Martial Arts',
 'Mecha',
 'Military',
 'Music',
 'Mystery',
 'Parody',
 'Police',
 'Psychological',
 'Romance',
 'Samurai',
 'School',
 'Sci-Fi',
 'Seinen',
 'Shoujo',
 'Shoujo Ai',
 'Shounen',
 'Shounen Ai',
 'Slice of Life',
 'Space',
 'Sports',
 'Super Power',
 'Supernatural',
 'Thriller',
 'Vampire',
 'Yaoi',
 'nan'}
```

```
[ ]: anime_with_synopsis = anime_with_synopsis.replace({'sypnopsis': '^No synopsis␣
     ↪information.*$'}, {'sypnopsis': ''}, regex=True)
```

```
[ ]: display(anime_with_synopsis[anime_with_synopsis['Genres'] == {'nan'}])
     display(anime[anime['Genres'] == {'nan'}])
```

```
       MAL_ID                                Name Score Genres  \
8759    28487                           Ikite Iru   NaN  {nan}
8807    28653                                Maze   NaN  {nan}
8808    28655                           PiKA PiKA  5.12  {nan}
9049    29655                          Chanda Gou   NaN  {nan}
9101    29765                   Metropolis (2009)  5.93  {nan}
…         …                                   …    …     …
14684   40090  Sinbi Apateu: Ghost Ball X-ui Tansaeng   NaN  {nan}
15028   40717                        Kaiju Decode   NaN  {nan}
15963   43762                     Hula Fulla Dance   NaN  {nan}
15983   44041                 SD Gundam World Heroes   NaN  {nan}
16178   48171                        Summer Ghost   NaN  {nan}

                                               sypnopsis
8759   Tsuyoshi is 9 years old and had friends over t…
8807            stract stop motion animation by Tochka.
8808   stract short film, the first "lightning doodle…
9049   Independent animation by Yanagihara Ryouhei, m…
9101   ai Mizue's first time experimenting with geome…
…                                                    …
14684
15028
15963  Natsunagi Hiwa, a novice, jumps into the world…
15983  The balance of the worlds is maintained by her…
16178

[63 rows x 5 columns]

       MAL_ID                                Name Score Genres  \
9788    28487                           Ikite Iru   NaN  {nan}
9838    28653                                Maze   NaN  {nan}
9839    28655                           PiKA PiKA  5.12  {nan}
10090   29655                          Chanda Gou   NaN  {nan}
10145   29765                   Metropolis (2009)  5.93  {nan}
…         …                                   …    …     …
15964   40090  Sinbi Apateu: Ghost Ball X-ui Tansaeng   NaN  {nan}
16324   40717                        Kaiju Decode   NaN  {nan}
17304   43762                     Hula Fulla Dance   NaN  {nan}
17326   44041                 SD Gundam World Heroes   NaN  {nan}
17525   48171                        Summer Ghost   NaN  {nan}

                 English name     Japanese name   Type Episodes  \
```

```
9788                        NaN                   OVA         1
9838                        NaN             MAZE  Movie         1
9839                        NaN        PiKA PiKA  Movie         1
10090              M.S. Chanda                  Movie         1
10145                       NaN       METROPOLIS  Movie         1
…                         …                …     …       …
15964                       NaN      :  X        TV        23
16324                       NaN      KAIJU DECODE    NaN       NaN
17304          Hula Fulla Dance                 Movie         1
17326   SD GUNDAM WORLD HEROES   SD             TV       NaN
17525                       NaN                 Movie         1

                              Aired     Premiered  … Score-10 Score-9 Score-8  \
9788                           1996           NaN  …      4.0     NaN     2.0
9838                           2012           NaN  …      7.0     NaN     4.0
9839                           2006           NaN  …      4.0     2.0     5.0
10090                          1964           NaN  …      5.0     2.0     1.0
10145                          2009           NaN  …     18.0    11.0    31.0
…                                 …          …   …      …       …       …
15964  Nov 9, 2017 to Jan 24, 2019     Fall 2017  …      5.0     4.0     2.0
16324                           NaN           NaN  …      NaN     NaN     NaN
17304                          2021           NaN  …      NaN     NaN     NaN
17326              Apr, 2021 to ?  Spring 2021  …      NaN     NaN     NaN
17525                          2021           NaN  …      NaN     NaN     NaN

      Score-7 Score-6 Score-5 Score-4  Score-3  Score-2  Score-1
9788       1.0     2.0     8.0     NaN      NaN      NaN      3.0
9838      11.0    19.0    18.0    15.0      5.0     11.0     10.0
9839      27.0    45.0    55.0    47.0     18.0     23.0     35.0
10090      7.0    10.0    19.0    14.0      7.0     13.0     11.0
10145     33.0    31.0    39.0    12.0     12.0      9.0     13.0
…          …      …      …      …      …       …       …
15964      4.0     4.0     3.0     3.0      2.0      1.0      2.0
16324      NaN     NaN     NaN     NaN      NaN      NaN      NaN
17304      NaN     NaN     NaN     NaN      NaN      NaN      NaN
17326      NaN     NaN     NaN     NaN      NaN      NaN      NaN
17525      NaN     NaN     NaN     NaN      NaN      NaN      NaN

[63 rows x 35 columns]
```

[ ]:
```
anime_with_synopsis['Genres'] = anime_with_synopsis['Genres'].apply(lambda x: x␣
↪- {'nan'})
anime['Genres'] = anime['Genres'].apply(lambda x: x - {'nan'})
```

[ ]:
```
anime = anime.astype({k: 'float' for k in ['Score-10', 'Score-9', 'Score-8',␣
↪'Score-7', 'Score-6',
       'Score-5', 'Score-4', 'Score-3', 'Score-2', 'Score-1']})
```

```
[ ]: anime['English name'] = anime['English name'].apply(clean_string)
anime_with_synopsis['Name'] = anime_with_synopsis['Name'].apply(clean_string)
```

factor premired date into the year

```
[ ]: def get_year_from_string(season_str: str):
    '''Convert string of premired year into year'''
    r = re.compile('(\w+)\W*(\d+)')
    if type(season_str) != str:
        return np.nan
    splitted = r.findall(season_str)
    if len(splitted) != 1:
        return np.nan

    return int(splitted[0][1])
```

```
[ ]: anime['Premiered Year'] = anime['Premiered'].apply(get_year_from_string)
```

```
[ ]: anime['Premiered Year']
```

```
[ ]: 0        1998.0
1           NaN
2        1998.0
3        2002.0
4        2004.0
          …
17557       NaN
17558       NaN
17559    2021.0
17560       NaN
17561    2021.0
Name: Premiered Year, Length: 17562, dtype: float64
```

for seachable title

```
[ ]: anime_for_search = pd.DataFrame({
    'MAL_ID': anime['MAL_ID'],
    'Name': anime['Name'],
    'English name': anime['English name'],
    'Japanese name': anime['Japanese name'],
    'Genres': anime['Genres'],
    'Avg. Score': anime['Score']
    })
```

```
[ ]: vectorizer = TfidfVectorizer(ngram_range=(1, 2))
vectorizer_en = TfidfVectorizer(ngram_range=(1, 2))
tfidf = vectorizer.fit_transform(anime_for_search['Name'])
tfidf_en = vectorizer_en.fit_transform(anime_for_search['English name'])
```

[ ]:
```python
with open('./Model/search_TfidfVectorizer.pkl', '+wb') as f:
    pickle.dump(vectorizer, f)
with open('./Model/search_TfidfVectorizer_en.pkl', '+wb') as f:
    pickle.dump(vectorizer_en, f)

with open('./Data/search_tfidf.pkl', '+wb') as f:
    pickle.dump(tfidf, f)
with open('./Data/search_tfidf_en.pkl', '+wb') as f:
    pickle.dump(tfidf_en, f)
```

[ ]:
```python
# save file
anime_with_synopsis.to_feather('./Data/anime_with_synopsis.feather')
anime.to_feather('./Data/anime.feather')
anime_for_search.to_feather('./Data/anime_for_search.feather')
```

[ ]:
```python
with open('./Model/search_TfidfVectorizer.pkl', 'rb') as f:
    vectorizer = pickle.load(f)
    assert(isinstance(vectorizer, TfidfVectorizer))
with open('./Model/search_TfidfVectorizer_en.pkl', 'rb') as f:
    vectorizer_en = pickle.load(f)
    assert(isinstance(vectorizer_en, TfidfVectorizer))

with open('./Data/search_tfidf.pkl', 'rb') as f:
    tfidf = pickle.load(f)
with open('./Data/search_tfidf_en.pkl', 'rb') as f:
    tfidf_en = pickle.load(f)

anime_with_synopsis = pd.read_feather('./Data/anime_with_synopsis.feather')
anime = pd.read_feather('./Data/anime.feather')
anime_for_search = pd.read_feather('./Data/anime_for_search.feather')

def search(keyword: str):
    return search_util(keyword, vectorizer, vectorizer_en, tfidf, tfidf_en,␣
 ↪anime_for_search)
```

[ ]:
```python
search('One piece')
```

[ ]:
```
       MAL_ID                                              Name  \
11         21                                         One Piece
6842    12859                                 One Piece Film: Z
5260     8171                                  One Piece Recap
7385    16143  One Piece: Kinkyuu Kikaku One Piece Kanzen Kou…
14553   37902                      One Piece: Episode of Sorajima
433       462               One Piece Movie 4: Dead End no Bouken
3524     4155                       One Piece Film: Strong World
13502   36240                      Scratch x One Piece Film: Gold
4048     5252                        One Piece: Romance Dawn Story
```

```
431       460      One Piece Movie 2: Nejimaki-jima no Daibouken

                                                    English name  \
11                                                     One Piece
6842                                            One Piece Film Z
5260                                             One Piece Recap
7385   One Piece:Emergency Planning, A Perfect Strate…
14553                            One Piece:Episode of Skypiea
433                                        One Piece:Dead End
3524                               One Piece Film Strong World
13502                          Scratch x One Piece Film:Gold
4048                             One Piece:Romance Dawn Story
431                    One Piece:Clockwork Island Adventure

                        Japanese name  \
11                          ONE PIECE
6842
5260
7385
14553              ONE PIECE
433
3524
13502  SCRATCH × ONE PIECE FILM GOLD
4048
431

                                                          Genres Avg. Score
11     [Action, Adventure, Fantasy, Shounen, Super Po…       8.52
6842   [Action, Adventure, Fantasy, Shounen, Comedy, …       8.18
5260   [Action, Adventure, Fantasy, Shounen, Super Po…       7.16
7385               [Comedy, Shounen, Adventure, Fantasy]       7.15
14553  [Action, Adventure, Fantasy, Shounen, Super Po…       7.11
433    [Action, Adventure, Fantasy, Shounen, Super Po…       7.59
3524   [Action, Adventure, Fantasy, Shounen, Comedy, …       8.17
13502                                [Shounen, Fantasy]       6.14
4048     [Action, Fantasy, Shounen, Super Power, Comedy]       7.39
431    [Action, Adventure, Fantasy, Shounen, Super Po…       7.17
```

[ ]:
```python
anime[anime['Genres'].apply(lambda x: 'Hentai' in x) | anime['Score'].isna()].
↪shape
```

[ ]: (6471, 36)

0.0.2 Data cleanning on Ratings

```
[ ]: # rating = pd.read_csv('./Data/animelist.csv')
     # rating_complete = pd.read_csv('./Data/rating_complete.csv')
     # rating.to_feather('./Data/animelist.feather')
     # rating_complete.to_feather('./Data/rating_complete.feather')
```

After save as feather binary file, load this instead

```
[ ]: rating = pd.read_feather('./Data/animelist.feather')
     rating_complete = pd.read_feather('./Data/rating_complete.feather')
```

```
[ ]: rating['rating'].unique()
```

```
[ ]: array([ 9,  7, 10,  0,  8,  6,  5,  4,  3,  2,  1])
```

I think the rating on `anime_id` is refer to `MAL_ID`

```
[ ]: anime_ids = rating_complete['anime_id'].unique()
     np.setdiff1d(anime_ids, anime['MAL_ID'].to_numpy())
```

```
[ ]: array([], dtype=int64)
```

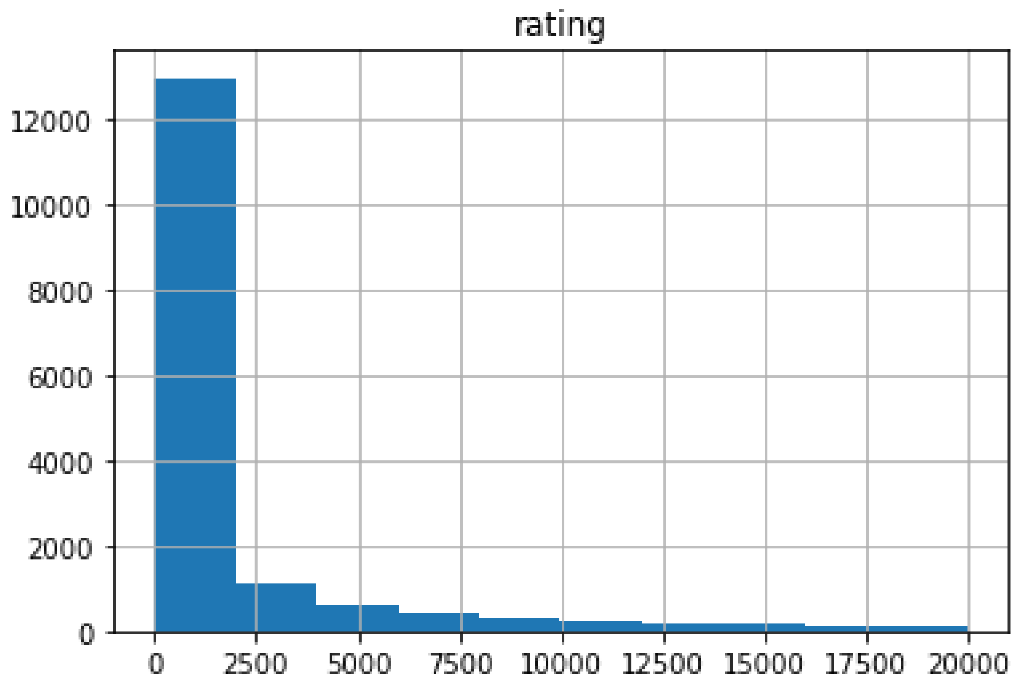
Just try to add count into the anime file

EDA (Ratings) Checking how many reviews for each anime, it appears that there are some anime which is so popular that there are more than 100,000 reviews on them.

```
[ ]: rating_hist = rating_complete.loc[:, ['anime_id', 'rating']].
       ↪groupby(by='anime_id').count()
```

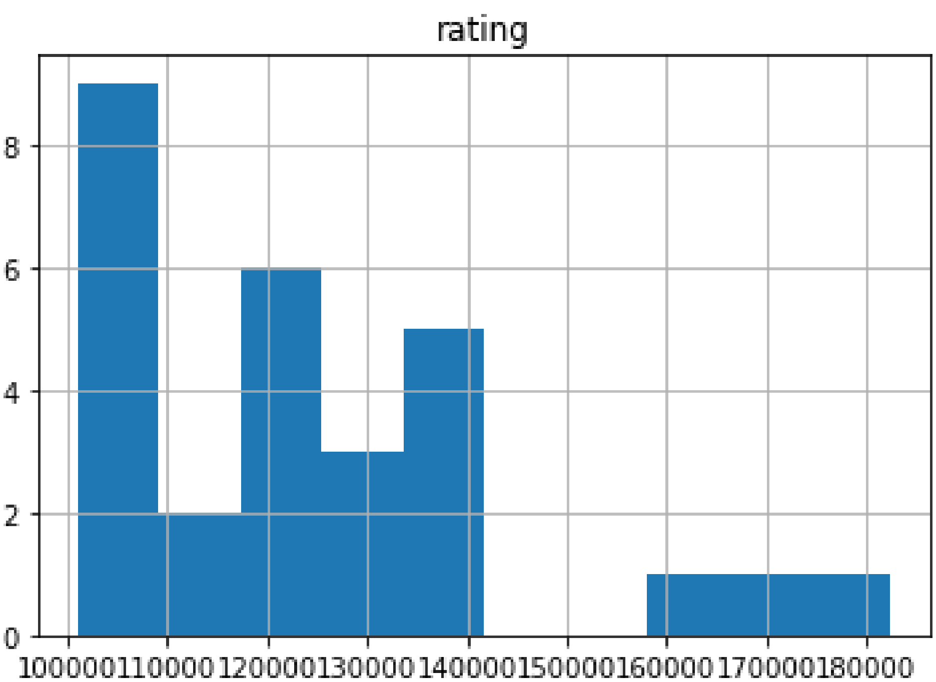
```
[ ]: rating_hist[rating_hist['rating'] < 2e4].hist()
```

```
[ ]: array([[<Axes: title={'center': 'rating'}>]], dtype=object)
```



[ ]:
```
rating_hist[rating_hist['rating'] > 1e5].hist()
```

[ ]: array([[<Axes: title={'center': 'rating'}>]], dtype=object)



Join the with rating with synopsis

```
[ ]: anime_with_synopsis_with_count = rating_hist.join(anime_with_synopsis.
      ↪set_index('MAL_ID'))
```

```
[ ]: anime_with_synopsis_with_count.columns = ['rating_completed_count', 'Name',␣
      ↪'Score', 'Genres', 'sypnopsis']
```

```
[ ]: anime_with_synopsis_with_count.sort_values(by='rating_completed_count',␣
      ↪ascending=False).head(10)
```

```
[ ]:           rating_completed_count                             Name Score  \
     anime_id
     1535                      182375                       Death Note  8.63
     16498                     169794               Shingeki no Kyojin  8.48
     11757                     161192                Sword Art Online  7.25
     6547                      141127                    Angel Beats!  8.15
     30276                     138924                   One Punch Man  8.57
     1575                      137291   Code Geass: Hangyaku no Lelouch  8.72
     4224                      135524                       Toradora!  8.24
     5114                      134197  Fullmetal Alchemist: Brotherhood  9.19
     19815                     129009                  No Game No Life   8.2
     22319                     128822                     Tokyo Ghoul  7.81

                                                          Genres  \
     anime_id
     1535      [Thriller, Mystery, Shounen, Psychological, Po…
     16498     [Military, Action, Mystery, Fantasy, Shounen, …
     11757           [Action, Adventure, Fantasy, Game, Romance]
     6547          [School, Action, Comedy, Drama, Supernatural]
     30276     [Parody, Action, Sci-Fi, Super Power, Comedy, …
     1575      [Military, School, Action, Sci-Fi, Mecha, Supe…
     4224                [Comedy, School, Slice of Life, Romance]
     5114      [Military, Action, Adventure, Fantasy, Shounen…
     19815     [Ecchi, Adventure, Fantasy, Game, Comedy, Supe…
     22319     [Action, Mystery, Horror, Seinen, Psychologica…

                                                       sypnopsis
     anime_id
     1535      shinigami, as a god of death, can kill any per…
     16498     Centuries ago, mankind was slaughtered to near…
     11757     In the year 2022, virtual reality has progress…
     6547      Otonashi awakens only to learn he is dead. A r…
     30276     The seemingly ordinary and unimpressive Saitam…
     1575      In the year 2010, the Holy Empire of Britannia…
     4224      uuji Takasu is a gentle high school student wi…
```

```
5114      "In order for something to be obtained, someth…
19815     No Game No Life is a surreal comedy that follo…
22319     Tokyo has become a cruel and merciless city-a …
```

0.0.3 Recommendation System (SVD)

Partition the dataset for performance reason

[ ]:
```
user_ids = rating_complete['user_id'].unique()
```

[ ]:
```
user_ids.shape
```

[ ]:
```
(310059,)
```

[ ]:
```
size = 0.05
focus_user_id, _ = train_test_split(user_ids, train_size=0.05, random_state=42)
focus_rating = rating_complete[rating_complete['user_id'].isin(focus_user_id)]
```

[ ]:
```
print(focus_user_id.shape)
print(focus_rating.shape)
```

```
(15502,)
(2862729, 3)
```

[ ]:
```
focus_rating.shape[0] / focus_user_id.shape[0]
```

[ ]:
```
184.66836537221005
```

[ ]:
```
reader = Reader(rating_scale=(0, 10))
dataset = Dataset.load_from_df(focus_rating[['user_id', 'anime_id', 'rating']],
 ↪reader)
```

[ ]:
```
# svd = SVD()
# cross_validate(svd, dataset, measures=['RMSE', 'MAE'], cv=3, verbose=True)
```

```
Evaluating RMSE, MAE of algorithm SVD on 3 split(s).

                  Fold 1  Fold 2  Fold 3  Mean    Std
RMSE (testset)    1.1765  1.1749  1.1763  1.1759  0.0007
MAE (testset)     0.8790  0.8777  0.8789  0.8785  0.0006
Fit time          12.93   12.49   12.95   12.79   0.21
Test time         6.14    6.15    6.26    6.18    0.05
{'test_rmse': array([1.1765492 , 1.17493892, 1.17632392]),
 'test_mae': array([0.87901695, 0.87771758, 0.87887208]),
 'fit_time': (12.930427074432373, 12.493419647216797, 12.945121049880981),
 'test_time': (6.144198894500732, 6.145112037658691, 6.2577879428863525)}
```

[ ]:
```
# knn = KNNBasic()
# cross_validate(knn, dataset, cv=3, verbose=True)
```

```
Computing the msd similarity matrix...
Done computing similarity matrix.
Computing the msd similarity matrix...
Done computing similarity matrix.
Computing the msd similarity matrix...
Done computing similarity matrix.
Evaluating RMSE, MAE of algorithm KNNBasic on 3 split(s).

                  Fold 1  Fold 2  Fold 3  Mean    Std
RMSE (testset)    1.2941  1.2918  1.2924  1.2928  0.0010
MAE (testset)     0.9587  0.9581  0.9584  0.9584  0.0002
Fit time          190.73  171.00  173.67  178.47  8.74
Test time         598.17  549.93  575.58  574.56  19.71
{'test_rmse': array([1.29408973, 1.29182792, 1.29243483]),
 'test_mae': array([0.95870405, 0.95810879, 0.9584189 ]),
 'fit_time': (190.73062705993652, 170.99515986442566, 173.67348313331604),
 'test_time': (598.1731100082397, 549.9259850978851, 575.5787467956543)}
```

[ ]:
```python
# sim_options = sim_options = {
#     "name": "cosine",
#     "user_based": False,  # compute  similarities between items
# }
# knn = KNNBasic(sim_options=sim_options)
# cross_validate(knn, dataset, cv=3, verbose=True)
```

```
Computing the cosine similarity matrix...
Done computing similarity matrix.
Computing the cosine similarity matrix...
Done computing similarity matrix.
Computing the cosine similarity matrix...
Done computing similarity matrix.
Evaluating RMSE, MAE of algorithm KNNBasic on 3 split(s).

                  Fold 1  Fold 2  Fold 3  Mean    Std
RMSE (testset)    1.3754  1.3752  1.3734  1.3747  0.0009
MAE (testset)     1.0304  1.0302  1.0292  1.0299  0.0005
Fit time          88.27   103.83  104.06  98.72   7.39
Test time         159.00  161.31  167.64  162.65  3.65
{'test_rmse': array([1.37540845, 1.37515701, 1.37341346]),
 'test_mae': array([1.03036109, 1.0302266 , 1.02920084]),
 'fit_time': (88.27486371994019, 103.82556104660034, 104.05858612060547),
 'test_time': (159.00390911102295, 161.3134388923645, 167.64093279838562)}
```

[ ]:
```python
# trainset = dataset.build_full_trainset()
# svd.fit(trainset)
```

[ ]:
```python
# dump.dump('./Model/svd.pkl', svd)
```

Model selection

[ ]:
```
# param_grid = {
#   'n_factors': [20, 50, 100],
#   'n_epochs': [5, 10, 20]
# }

# gs = GridSearchCV(SVD, param_grid, measures=['rmse', 'mae'], cv=5)
# gs.fit(dataset)

# print(gs.best_score['rmse'])
# print(gs.best_params['rmse'])
```

```
1.1458925132330084
{'n_factors': 20, 'n_epochs': 20}
```

[ ]:
```
svd = SVD(n_factors=20, n_epochs=20)
svd.fit(dataset.build_full_trainset())
dump.dump('./Model/svd.pkl', svd)
```

[ ]:
```
focus_rating[focus_rating['user_id'] == focus_user_id[100]].
 ↪set_index('anime_id').join(anime_for_search.set_index('MAL_ID'))
x = search('Kara no Kyoukai 4')
x['Predict Score'] = x['MAL_ID'].apply(lambda id: svd.
 ↪predict(uid=focus_user_id[0], iid=id).est)
x
```

[ ]:
```
      MAL_ID                                               Name  \
4838    6954                          Kara no Kyoukai 8: Shuushou
735      814  Tennis no Ouji-sama: Atobe kara no Okurimono -…
7177   14807                         Kara no Kyoukai: Mirai Fukuin
8990   23697                        Kara no Kyoukai: Manner Movies
3578    4282                        Kara no Kyoukai 5: Mujun Rasen
4017    5204                    Kara no Kyoukai 6: Boukyaku Rokuon
2379    2593                       Kara no Kyoukai 1: Fukan Fuukei
3277    3783                  Kara no Kyoukai 3: Tsuukaku Zanryuu
6091   10161                                                 No.6
3577    4280                       Kara no Kyoukai 4: Garan no Dou

                                           English name  \
4838  the Garden of sinners Chapter 8:The Final Chapter
735            Prince of Tennis:Atobe Kara no Okurimono
7177         the Garden of sinners -recalled out summer-
8990             the Garden of sinners Pre-show Reminder
3578   the Garden of sinners Chapter 5:Paradox Paradigm
4017  the Garden of sinners Chapter 6:Oblivion Recor…
2379   the Garden of sinners Chapter 1:Overlooking View
3277  the Garden of sinners Chapter 3:Remaining Sens…
6091                                              No. 6
3577  the Garden of sinners Chapter 4:The Hollow Shrine
```

```
                               Japanese name  \
4838                 the Garden of sinners
735
7177
8990                                    CM
3578         the Garden of sinners
4017         the Garden of sinners
2379         the Garden of sinners
3277         the Garden of sinners
6091                              NO.6
3577         the Garden of sinners

                                                 Genres Avg. Score  \
4838                                          [Mystery]       7.21
735                           [Comedy, Shounen, Sports]       7.49
7177              [Mystery, Drama, Seinen, Supernatural]       8.03
8990                                   [Comedy, Action]       6.41
3578  [Thriller, Action, Mystery, Romance, Drama, Su…       8.56
4017  [Thriller, Action, Mystery, Magic, Romance, Su…       7.53
2379           [Mystery, Thriller, Action, Supernatural]       7.64
3277   [Thriller, Action, Mystery, Drama, Supernatural]       8.07
6091                    [Mystery, Drama, Action, Sci-Fi]       7.58
3577           [Mystery, Thriller, Action, Supernatural]        7.9

      Predict Score
4838       5.993547
735        5.934506
7177       7.389680
8990       5.895363
3578       8.300138
4017       6.357418
2379       6.761795
3277       7.314412
6091       6.212659
3577       7.072808
```

[ ]:
```
focus_rating.reset_index().to_feather('./Data/For_SVD/focus_rating.feather')
```

0.0.4 Actual Model for recommender system

[ ]:
```
size = 0.2
focus_user_id_2, _ = train_test_split(user_ids, train_size=size,␣
 ↪random_state=42)
focus_rating_2 = rating_complete[rating_complete['user_id'].
 ↪isin(focus_user_id_2)]
focus_rating_2 = focus_rating_2.reset_index()
```

```
focus_rating_2.reset_index().to_feather('./Data/For_SVD/focus_rating_2.feather')
```

[ ]:
```
print(f'# of users: {len(focus_user_id_2)}')
print(f'# of ratings: {len(focus_rating_2)}')
print(f'# anime titles: {len(anime_for_search)}')
print(f'sparce ratio: {len(focus_rating_2) / (len(focus_user_id_2) *␣
 ↪len(focus_rating_2))}')
```

```
# of users: 62011
# of ratings: 11474169
# anime titles: 17562
sparce ratio: 1.6126171163180727e-05
```

[ ]:
```
svd_big = SVD(n_factors=20, n_epochs=20)
reader_big = Reader(rating_scale=(0, 10))
dataset_big = Dataset.load_from_df(focus_rating_2[['user_id', 'anime_id',␣
 ↪'rating']], reader_big)
svd_big.fit(dataset_big.build_full_trainset())
```

[ ]:
```
<surprise.prediction_algorithms.matrix_factorization.SVD at 0x7fd01d2eab80>
```

[ ]:
```
dump.dump('./Model/svd_big.pkl', svd_big)
```

0.0.5 Interactive session (See interactive session.pdf)

[ ]:
```
search_widget = widgets.Text(
    value = '', description = 'Title'
)

search_btn = widgets.Button(
    description='Search',
    disabled=False,
    button_style='info',
    tooltip='Search',
    icon='search'
)

search_output = widgets.Output()

def search_event(sender):
    search_output.clear_output()
    x = search_widget.value
    if len(x) < 3:
            return
    with search_output:
        display(search(x))

search_btn.on_click(search_event)
```

```
display(widgets.VBox([widgets.HBox([search_widget, search_btn]),␣
 ↪search_output]))
```

```
VBox(children=(HBox(children=(Text(value='', description='Title'),␣
 ↪Button(button_style='info', description='Se…
```

[ ]:
```
from sklearn.metrics import jaccard_score
def search_similar_users(given_review: dict, top_k = 1):
    rank = pd.Series()
    v = pd.DataFrame({'anime_id': given_review.keys(), 'rating': given_review.
 ↪values()})\
        .pivot(columns='anime_id', values='rating')
    for i in user_ids[:2000]:
        u = focus_rating_2[focus_rating_2['user_id'] == i].
 ↪pivot(index='user_id', columns='anime_id', values='rating')
        cos_sim = cosine_similarity(pd.concat([u, v]).fillna(0))[0][1]

        rank[i] = cos_sim
        if len(rank) > top_k:
            rank.sort_values(inplace=True, ascending=False)
            rank = rank[:top_k]

    return rank

a = {235: 8, 21:10}
x = pd.DataFrame({'anime_id': a.keys(), 'rating':a.values()})
search_similar_users(a, 3)
```

[ ]:
```
0    0.0
1    0.0
2    0.0
dtype: float64
```

[ ]: