Movie Recommend by Surprise with GridSearch

In this notebook, optimize hyperparameters of Surprise using GridSearchCV.

In [1]:

```
import sys
import random
from surprise import Dataset, Reader
from surprise import KNNBasic, SVD
from surprise import accuracy
from surprise.model_selection import train_test_split
from surprise.model_selection import cross_validate
from surprise.model_selection import GridSearchCV
from surprise.dataset import DatasetAutoFolds
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import operator
import requests
from zipfile import ZipFile
```

In [17]:

```
df = pd.read_csv('/kaggle/input/movies-and-ratings-for-recommendation-system/rating
s.csv',
                          error_bad_lines=False,
                          warn_bad_lines=False,
                          skiprows=lambda i: i>0 and random.random() > 0.1)
df.columns=['user_id','item_id','rating','timestamp']
print(len(df))
```

```
10043

/opt/conda/lib/python3.7/site-packages/IPython/core/interactiveshell.p
y:3552: FutureWarning: The warn_bad_lines argument has been deprecated
and will be removed in a future version.


  exec(code_obj, self.user_global_ns, self.user_ns)
/opt/conda/lib/python3.7/site-packages/IPython/core/interactiveshell.p
y:3552: FutureWarning: The error_bad_lines argument has been deprecate
d and will be removed in a future version.


  exec(code_obj, self.user_global_ns, self.user_ns)
```

In [3]:

```
reader = Reader(rating_scale=(1,5))
data = Dataset.load_from_df(df[['user_id','item_id','rating']], reader)
print(type(data))
```

```
<class 'surprise.dataset.DatasetAutoFolds'>
```

```
# cross_validate

# Run 5-fold cross-validation and print results.
results = cross_validate(algo, data, measures=['RMSE', 'MAE'], cv=5, verbose
=True)
print(results)
```

GridSearchCV - algo to best_algo

In [4]:

```
algo = KNNBasic

param_grid = { 'k': [5,10,15,20,50], 'min_k': [2,3,4,5] }
gs = GridSearchCV(algo, param_grid, cv=5)
gs.fit(data)
```

```
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.
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.
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.
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.
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.
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.
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.
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.
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.
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.
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.
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.
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.
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.
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.
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.
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.
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.
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.
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.
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.
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.
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.
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.
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.
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.
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.
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.
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.
```

```
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.
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.
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.
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.
Computing the msd similarity matrix...
Done computing similarity matrix.
```

In [5]:

```
print(gs.best_params)
best_algo = KNNBasic(k=gs.best_params['rmse']['k'], min_k=gs.best_params['rmse']['m
in_k'])
print(best_algo.k, best_algo.min_k)
```

```
{'rmse': {'k': 15, 'min_k': 5}, 'mae': {'k': 15, 'min_k': 5}}
15 5
```

In [6]:

```
help(best_algo)
```

```
Help on KNNBasic in module surprise.prediction_algorithms.knns object:

class KNNBasic(SymmetricAlgo)
 |  KNNBasic(k=40, min_k=1, sim_options={'user_based': True}, verbose=
True, **kwargs)
 |  
 |  A basic collaborative filtering algorithm.
 |  
 |  The prediction :math:`\hat{r}_{ui}` is set as:
 |  
 |  .. math::
 |      \hat{r}_{ui} = \frac{
 |      \sum\limits_{v \in N^k_i(u)} \text{sim}(u, v) \cdot r_{vi}}
 |      {\sum\limits_{v \in N^k_i(u)} \text{sim}(u, v)}
 |  
 |  or
 |  
 |  .. math::
 |      \hat{r}_{ui} = \frac{
 |      \sum\limits_{j \in N^k_u(i)} \text{sim}(i, j) \cdot r_{uj}}
 |      {\sum\limits_{j \in N^k_u(i)} \text{sim}(i, j)}
 |  
 |  depending on the ``user_based`` field of the ``sim_options`` param
eter.
 |  
 |  Args:
 |      k(int): The (max) number of neighbors to take into account for
 |          aggregation (see :ref:`this note <actual_k_note>`). Defaul
t is
 |          ``40``.
 |      min_k(int): The minimum number of neighbors to take into accou
nt for
 |          aggregation. If there are not enough neighbors, the predic
tion is
 |          set to the global mean of all ratings. Default is ``1``.
 |      sim_options(dict): A dictionary of options for the similarity
 |          measure. See :ref:`similarity_measures_configuration` for
accepted
 |          options.
 |      verbose(bool): Whether to print trace messages of bias estimat
ion,
 |          similarity, etc.  Default is True.
 |  
 |  Method resolution order:
 |      KNNBasic
 |      SymmetricAlgo
 |      surprise.prediction_algorithms.algo_base.AlgoBase
 |      builtins.object
 |  
 |  Methods defined here:
 |  
 |  __init__(self, k=40, min_k=1, sim_options={'user_based': True}, ve
rbose=True, **kwargs)
 |      Initialize self.  See help(type(self)) for accurate signature.
 |  
 |  estimate(self, u, i)
 |  
 |  fit(self, trainset)
```

```
 |      Train an algorithm on a given training set.
 |      
 |      This method is called by every derived class as the first basi
c step
 |      for training an algorithm. It basically just initializes some 
internal
 |      structures and set the self.trainset attribute.
 |      
 |      Args:
 |          trainset(:obj:`Trainset <surprise.Trainset>`) : A training
 |              set, as returned by the :meth:`folds
 |              <surprise.dataset.Dataset.folds>` method.
 |      
 |      Returns:
 |          self
 |  
 |  ------------------------------------------------------------------
----
 |  Methods inherited from SymmetricAlgo:
 |  
 |  switch(self, u_stuff, i_stuff)
 |      Return x_stuff and y_stuff depending on the user_based field.
 |  
 |  ------------------------------------------------------------------
----
 |  Methods inherited from surprise.prediction_algorithms.algo_base.Al
goBase:
 |  
 |  compute_baselines(self)
 |      Compute users and items baselines.
 |      
 |      The way baselines are computed depends on the ``bsl_options`` 
parameter
 |      passed at the creation of the algorithm (see
 |      :ref:`baseline_estimates_configuration`).
 |      
 |      This method is only relevant for algorithms using :func:`Pears
on
 |      baseline similarty<surprise.similarities.pearson_baseline>` or 
the
 |      :class:`BaselineOnly
 |      <surprise.prediction_algorithms.baseline_only.BaselineOnly>` a
lgorithm.
 |      
 |      Returns:
 |          A tuple ``(bu, bi)``, which are users and items baselines.
 |  
 |  compute_similarities(self)
 |      Build the similarity matrix.
 |      
 |      The way the similarity matrix is computed depends on the
 |      ``sim_options`` parameter passed at the creation of the algori
thm (see
 |      :ref:`similarity_measures_configuration`).
 |      
 |      This method is only relevant for algorithms using a similarity 
measure,
 |      such as the :ref:`k-NN algorithms <pred_package_knn_inpired>`.
```

```
 |
 |      Returns:
 |          The similarity matrix.
 |
 |  default_prediction(self)
 |      Used when the ``PredictionImpossible`` exception is raised dur
ing a
 |      call to :meth:`predict()
 |      <surprise.prediction_algorithms.algo_base.AlgoBase.predict>`.
By
 |      default, return the global mean of all ratings (can be overrid
den in
 |      child classes).
 |
 |      Returns:
 |          (float): The mean of all ratings in the trainset.
 |
 |  get_neighbors(self, iid, k)
 |      Return the ``k`` nearest neighbors of ``iid``, which is the in
ner id
 |      of a user or an item, depending on the ``user_based`` field of
 |      ``sim_options`` (see :ref:`similarity_measures_configuration
`).
 |
 |      As the similarities are computed on the basis of a similarity
measure,
 |      this method is only relevant for algorithms using a similarity
measure,
 |      such as the :ref:`k-NN algorithms <pred_package_knn_inpired>`.
 |
 |      For a usage example, see the :ref:`FAQ <get_k_nearest_neighbor
s>`.
 |
 |      Args:
 |          iid(int): The (inner) id of the user (or item) for which w
e want
 |              the nearest neighbors. See :ref:`this note<raw_inner_n
ote>`.
 |
 |          k(int): The number of neighbors to retrieve.
 |
 |      Returns:
 |          The list of the ``k`` (inner) ids of the closest users (or
items)
 |          to ``iid``.
 |
 |  predict(self, uid, iid, r_ui=None, clip=True, verbose=False)
 |      Compute the rating prediction for given user and item.
 |
 |      The ``predict`` method converts raw ids to inner ids and then
calls the
 |      ``estimate`` method which is defined in every derived class. I
f the
 |      prediction is impossible (e.g. because the user and/or the ite
m is
 |      unknown), the prediction is set according to
 |      :meth:`default_prediction()
 |      <surprise.prediction_algorithms.algo_base.AlgoBase.default_pre
```

```
diction>`.
 |
 |      Args:
 |          uid: (Raw) id of the user. See :ref:`this note<raw_inner_n
ote>`.
 |          iid: (Raw) id of the item. See :ref:`this note<raw_inner_n
ote>`.
 |          r_ui(float): The true rating :math:`r_{ui}`. Optional, def
ault is
 |              ``None``.
 |          clip(bool): Whether to clip the estimation into the rating
scale.
 |              For example, if :math:`\hat{r}_{ui}` is :math:`5.5` wh
ile the
 |              rating scale is :math:`[1, 5]`, then :math:`\hat{r}_{u
i}` is
 |              set to :math:`5`. Same goes if :math:`\hat{r}_{ui} < 1
`.
 |              Default is ``True``.
 |          verbose(bool): Whether to print details of the prediction.
Default
 |              is False.
 |
 |      Returns:
 |          A :obj:`Prediction            <surprise.prediction_algorit
hms.predictions.Prediction>` object
 |          containing:
 |
 |          - The (raw) user id ``uid``.
 |          - The (raw) item id ``iid``.
 |          - The true rating ``r_ui`` (:math:`r_{ui}`).
 |          - The estimated rating (:math:`\hat{r}_{ui}`).
 |          - Some additional details about the prediction that might
be useful
 |            for later analysis.
 |
 |  test(self, testset, verbose=False)
 |      Test the algorithm on given testset, i.e. estimate all the rat
ings
 |      in the given testset.
 |
 |      Args:
 |          testset: A test set, as returned by a :ref:`cross-validati
on
 |              itertor<use_cross_validation_iterators>` or by the
 |              :meth:`build_testset() <surprise.Trainset.build_testse
t>`
 |              method.
 |          verbose(bool): Whether to print details for each predictio
ns.
 |              Default is False.
 |
 |      Returns:
 |          A list of :class:`Prediction            <surprise.predicti
on_algorithms.predictions.Prediction>` objects
 |          that contains all the estimated ratings.
 |
 |  ----------------------------------------------------------------------
```

```
 ----
 |  Data descriptors inherited from surprise.prediction_algorithms.alg
 o_base.AlgoBase:
 |
 |  __dict__
 |      dictionary for instance variables (if defined)
 |
 |  __weakref__
 |      list of weak references to the object (if defined)
```

best_algo - fit & test

In [7]:

```
trainset, testset = train_test_split(data, test_size=0.25)

best_algo.fit(trainset)

predictions = best_algo.test(testset)
print(predictions[0:2])
```

```
Computing the msd similarity matrix...
Done computing similarity matrix.
[Prediction(uid=292, iid=1291, r_ui=4.5, est=3.5127262828365318, detai
ls={'was_impossible': True, 'reason': 'Not enough neighbors.'}), Predi
ction(uid=182, iid=3326, r_ui=2.5, est=3.5127262828365318, details={'w
as_impossible': True, 'reason': 'Not enough neighbors.'})]
```

In [8]:

```
import itertools

for uid, iid, rating in itertools.islice(trainset.all_ratings(), 5):
    print(f"User {uid} rated item {iid} with a rating of {rating}")

print()
for uid, iid, rating in testset[:5]:
    print(f"User {uid} rated item {iid} with a rating of {rating}")

print()
print(trainset.n_ratings,len(testset))
```

```
User 0 rated item 0 with a rating of 5.0
User 0 rated item 213 with a rating of 3.0
User 0 rated item 801 with a rating of 4.0
User 0 rated item 1018 with a rating of 3.0
User 0 rated item 1414 with a rating of 4.0

User 292 rated item 1291 with a rating of 4.5
User 182 rated item 3326 with a rating of 2.5
User 89 rated item 56367 with a rating of 2.0
User 200 rated item 5617 with a rating of 3.5
User 232 rated item 37720 with a rating of 3.0

7347 2450
```

In [9]:

```
for uid, iid, rating in testset[:5]:
    print(f"User {uid} rated item {iid} with a rating of {rating}")
```

```
User 292 rated item 1291 with a rating of 4.5
User 182 rated item 3326 with a rating of 2.5
User 89 rated item 56367 with a rating of 2.0
User 200 rated item 5617 with a rating of 3.5
User 232 rated item 37720 with a rating of 3.0
```

The 'predictions' is a list of tuples of the form (user, item, actual_rating, predicted_rating, details). The predicted_rating is est value.

In [10]:

```
for prediction in predictions[0:5]:
    print(prediction)
```

```
user: 292        item: 1291       r_ui = 4.50   est = 3.51   {'was_imp
ossible': True, 'reason': 'Not enough neighbors.'}
user: 182        item: 3326       r_ui = 2.50   est = 3.51   {'was_imp
ossible': True, 'reason': 'Not enough neighbors.'}
user: 89         item: 56367      r_ui = 2.00   est = 3.51   {'was_imp
ossible': True, 'reason': 'Not enough neighbors.'}
user: 200        item: 5617       r_ui = 3.50   est = 3.51   {'was_imp
ossible': True, 'reason': 'Not enough neighbors.'}
user: 232        item: 37720      r_ui = 3.00   est = 3.51   {'was_imp
ossible': True, 'reason': 'Not enough neighbors.'}
```

In [11]:

```
# Print the performance metrics
accuracy.rmse(predictions)
```

```
RMSE: 1.0626
```

Out[11]:

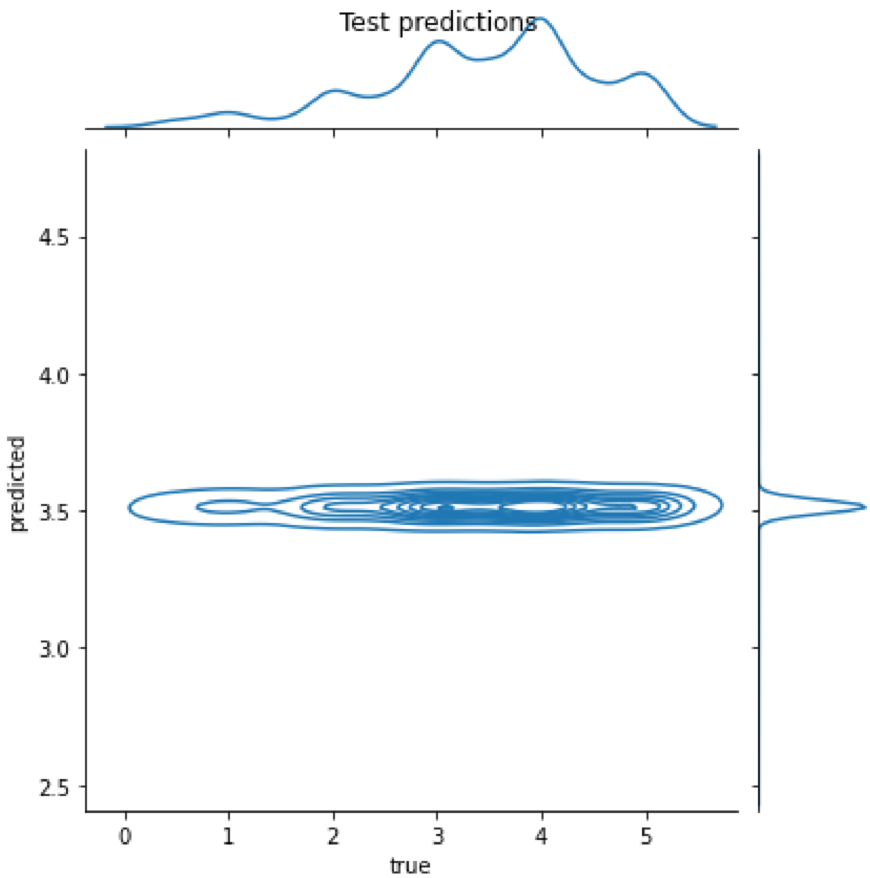
```
1.0626366532296367
```

In [12]:

```
true_ratings = [pred.r_ui for pred in predictions]
est_ratings = [pred.est for pred in predictions]
uids = [pred.uid for pred in predictions]
```

In [13]:

```
data=pd.DataFrame(columns=["true","predicted"])
data["true"]=true_ratings
data["predicted"]=est_ratings
g = sns.jointplot(data=data,x="true", y="predicted", kind="kde",)
g.fig.suptitle('Test predictions',fontsize=12)
plt.show()
```



Recommend unseen books for test set users

In [14]:

```
import pandas as pd
books=pd.read_csv('/kaggle/input/movies-and-ratings-for-recommendation-system/movie
s.csv')
#display(movies)
mapping = books.set_index("movieId")["title"].to_dict()
#print(mapping)
```

In [15]:

```
users=list(set(uids))
```

In [16]:

```
# items which the user not yet evaluate
items = trainset.build_anti_testset()
for user in users[0:30]:
    user_items = list(filter(lambda x: x[0] == user, items))
    # generate recommendation
    recommendations = best_algo.test(user_items)
    if len(recommendations)>0:
        recommendations.sort(key=operator.itemgetter(3), reverse=True)
        print(f"For User {user}:")
        for r in recommendations[0:5]:
            try:
                print(f"  {mapping[r[1]]} : [{round(r[3],3)}]")
            except:
                continue
```

```
For User 1:
  American Beauty (1999) : [4.442]
  Raiders of the Lost Ark (Indiana Jones and the Raiders of the Lost A
rk) (1981) : [4.286]
  Silence of the Lambs, The (1991) : [4.276]
  Ghostbusters (a.k.a. Ghost Busters) (1984) : [4.26]
  Shawshank Redemption, The (1994) : [4.247]
For User 2:
  Space Jam (1996) : [3.513]
  Mrs. Doubtfire (1993) : [3.513]
  Brazil (1985) : [3.513]
  Braveheart (1995) : [3.513]
  Ravenous (1999) : [3.513]
For User 3:
  Space Jam (1996) : [3.513]
  Mrs. Doubtfire (1993) : [3.513]
  Brazil (1985) : [3.513]
  Braveheart (1995) : [3.513]
  Ravenous (1999) : [3.513]
For User 4:
  Star Wars: Episode V - The Empire Strikes Back (1980) : [4.738]
  Alien (1979) : [4.337]
  Lord of the Rings: The Two Towers, The (2002) : [4.311]
  Bourne Identity, The (2002) : [4.22]
  Lord of the Rings: The Return of the King, The (2003) : [4.215]
For User 6:
  Fugitive, The (1993) : [4.184]
  Batman (1989) : [3.79]
  Space Jam (1996) : [3.513]
  Mrs. Doubtfire (1993) : [3.513]
  Brazil (1985) : [3.513]
For User 7:
  Forrest Gump (1994) : [4.702]
  Lord of the Rings: The Fellowship of the Ring, The (2001) : [4.56]
  Blade Runner (1982) : [4.285]
  Raiders of the Lost Ark (Indiana Jones and the Raiders of the Lost A
rk) (1981) : [4.244]
  Shawshank Redemption, The (1994) : [4.23]
For User 9:
  Space Jam (1996) : [3.513]
  Mrs. Doubtfire (1993) : [3.513]
  Brazil (1985) : [3.513]
  Braveheart (1995) : [3.513]
  Ravenous (1999) : [3.513]
For User 10:
  Space Jam (1996) : [3.513]
  Mrs. Doubtfire (1993) : [3.513]
  Brazil (1985) : [3.513]
  Braveheart (1995) : [3.513]
  Ravenous (1999) : [3.513]
For User 11:
  Space Jam (1996) : [3.513]
  Mrs. Doubtfire (1993) : [3.513]
  Brazil (1985) : [3.513]
  Braveheart (1995) : [3.513]
  Ravenous (1999) : [3.513]
For User 12:
  Space Jam (1996) : [3.513]
```

```
    Mrs. Doubtfire (1993) : [3.513]
    Brazil (1985) : [3.513]
    Braveheart (1995) : [3.513]
    Ravenous (1999) : [3.513]
  For User 14:
    Matrix, The (1999) : [3.925]
    Space Jam (1996) : [3.513]
    Mrs. Doubtfire (1993) : [3.513]
    Brazil (1985) : [3.513]
    Braveheart (1995) : [3.513]
  For User 17:
    Pulp Fiction (1994) : [4.65]
    Stargate (1994) : [4.343]
    Star Wars: Episode VI - Return of the Jedi (1983) : [4.195]
    Fifth Element, The (1997) : [4.103]
    Léon: The Professional (a.k.a. The Professional) (Léon) (1994) : [4.
  036]
  For User 18:
    Matrix, The (1999) : [4.487]
    Spirited Away (Sen to Chihiro no kamikakushi) (2001) : [4.471]
    Pulp Fiction (1994) : [4.453]
    Saving Private Ryan (1998) : [4.38]
    Beauty and the Beast (1991) : [4.329]
  For User 19:
    Usual Suspects, The (1995) : [4.733]
    Princess Bride, The (1987) : [4.706]
    Braveheart (1995) : [4.508]
    Lord of the Rings: The Fellowship of the Ring, The (2001) : [4.493]
    Pulp Fiction (1994) : [4.438]
  For User 20:
    Space Jam (1996) : [3.513]
    Mrs. Doubtfire (1993) : [3.513]
    Brazil (1985) : [3.513]
    Braveheart (1995) : [3.513]
    Ravenous (1999) : [3.513]
  For User 21:
    Princess Bride, The (1987) : [4.743]
    Inception (2010) : [4.612]
    Silence of the Lambs, The (1991) : [4.448]
    Gladiator (2000) : [4.389]
    American Beauty (1999) : [4.38]
  For User 22:
    Lord of the Rings: The Fellowship of the Ring, The (2001) : [4.379]
    Star Wars: Episode V - The Empire Strikes Back (1980) : [3.691]
    Space Jam (1996) : [3.513]
    Mrs. Doubtfire (1993) : [3.513]
    Brazil (1985) : [3.513]
  For User 23:
    Jurassic Park (1993) : [4.004]
    Space Jam (1996) : [3.513]
    Mrs. Doubtfire (1993) : [3.513]
    Braveheart (1995) : [3.513]
    Ravenous (1999) : [3.513]
  For User 24:
    Space Jam (1996) : [3.513]
    Mrs. Doubtfire (1993) : [3.513]
    Brazil (1985) : [3.513]
    Braveheart (1995) : [3.513]
```

```
    Ravenous (1999) : [3.513]
  For User 25:
    Space Jam (1996) : [3.513]
    Mrs. Doubtfire (1993) : [3.513]
    Brazil (1985) : [3.513]
    Braveheart (1995) : [3.513]
    Ravenous (1999) : [3.513]
  For User 26:
    Space Jam (1996) : [3.513]
    Mrs. Doubtfire (1993) : [3.513]
    Brazil (1985) : [3.513]
    Ravenous (1999) : [3.513]
    Payback (1999) : [3.513]
  For User 27:
    Matrix, The (1999) : [4.204]
    Terminator 2: Judgment Day (1991) : [3.833]
    Space Jam (1996) : [3.513]
    Mrs. Doubtfire (1993) : [3.513]
    Brazil (1985) : [3.513]
  For User 28:
    Star Wars: Episode VI - Return of the Jedi (1983) : [4.58]
    Toy Story 2 (1999) : [4.53]
    Fight Club (1999) : [4.455]
    Silence of the Lambs, The (1991) : [4.446]
    Gladiator (2000) : [4.408]
  For User 29:
    Space Jam (1996) : [3.513]
    Mrs. Doubtfire (1993) : [3.513]
    Brazil (1985) : [3.513]
    Braveheart (1995) : [3.513]
    Ravenous (1999) : [3.513]
  For User 31:
    Silence of the Lambs, The (1991) : [4.657]
    Pulp Fiction (1994) : [4.42]
    Terminator 2: Judgment Day (1991) : [3.688]
    Space Jam (1996) : [3.513]
    Mrs. Doubtfire (1993) : [3.513]
  For User 32:
    Space Jam (1996) : [3.513]
    Mrs. Doubtfire (1993) : [3.513]
    Brazil (1985) : [3.513]
    Braveheart (1995) : [3.513]
    Ravenous (1999) : [3.513]
  For User 33:
    Space Jam (1996) : [3.513]
    Mrs. Doubtfire (1993) : [3.513]
    Brazil (1985) : [3.513]
    Braveheart (1995) : [3.513]
    Ravenous (1999) : [3.513]
  For User 34:
    Silence of the Lambs, The (1991) : [4.161]
    American History X (1998) : [4.1]
    Space Jam (1996) : [3.513]
    Mrs. Doubtfire (1993) : [3.513]
    Brazil (1985) : [3.513]
  For User 36:
    Space Jam (1996) : [3.513]
    Mrs. Doubtfire (1993) : [3.513]
```

```
  Brazil (1985) : [3.513]
  Braveheart (1995) : [3.513]
  Ravenous (1999) : [3.513]
For User 37:
  Fugitive, The (1993) : [4.196]
  Space Jam (1996) : [3.513]
  Mrs. Doubtfire (1993) : [3.513]
  Brazil (1985) : [3.513]
  Ravenous (1999) : [3.513]
```

In [ ]:

```

```