

DEEP LEARNING WORKSHOP UP | Assessment | Desktop/Simplilearn Machine Le | Project - Jupyter Notebook

localhost:8888/notebooks/Desktop/Simplilearn%20Machine%20Learning/Project.ipynb#

jupyter Project Last Checkpoint: 04/23/2021 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
In [7]: # import numpy as np
import pandas as pd
import re
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import surprise
```

```
In [8]: df = pd.read_csv('Amazon - Movies and TV Ratings.csv')
```

```
In [9]: df.head()
```

Out[9]:

	user_id	Movie1	Movie2	Movie3	Movie4	Movie5	Movie6	Movie7	Movie8	Movie9	...	Movie197	Movie198	Movie199	Movie200	Movie201
0	A3R5OBKS7OM2IR	5.0	5.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN
1	AH3QC2PC1VTGP	NaN	NaN	2.0	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN
2	A3LKP6WMP9UKX	NaN	NaN	NaN	5.0	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN
3	AVY8KKEPQ5ZD	NaN	NaN	NaN	5.0	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN
4	A1CV1WROP5KTTW	NaN	NaN	NaN	NaN	5.0	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN

5 rows x 207 columns

```
In [10]: df.describe()
```

Out[10]:

DEEP LEARNING WORKSHOP UP | Assessment | Desktop/Simplilearn Machine Le | Project - Jupyter Notebook

localhost:8888/notebooks/Desktop/Simplilearn%20Machine%20Learning/Project.ipynb#

jupyter Project Last Checkpoint: 04/23/2021 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
In [10]: df.describe()
```

Out[10]:

	Movie1	Movie2	Movie3	Movie4	Movie5	Movie6	Movie7	Movie8	Movie9	Movie10	...	Movie197	Movie198	Movie199	Movie200	Movie201	Movie202
count	1.0	1.0	1.0	2.0	29.000000	1.0	1.0	1.0	1.0	1.0	...	5.000000	2.0	1.0	8.000000	3.000000	6.00
mean	5.0	5.0	2.0	5.0	4.103448	4.0	5.0	5.0	5.0	5.0	...	3.800000	5.0	5.0	4.625000	4.333333	4.33
std	NaN	NaN	NaN	0.0	1.496301	NaN	NaN	NaN	NaN	NaN	...	1.643168	0.0	NaN	0.517549	1.154701	1.63
min	5.0	5.0	2.0	5.0	1.000000	4.0	5.0	5.0	5.0	5.0	...	1.000000	5.0	5.0	4.000000	3.000000	1.00
25%	5.0	5.0	2.0	5.0	4.000000	4.0	5.0	5.0	5.0	5.0	...	4.000000	5.0	5.0	4.000000	4.000000	5.00
50%	5.0	5.0	2.0	5.0	5.000000	4.0	5.0	5.0	5.0	5.0	...	4.000000	5.0	5.0	5.000000	5.000000	5.00
75%	5.0	5.0	2.0	5.0	5.000000	4.0	5.0	5.0	5.0	5.0	...	5.000000	5.0	5.0	5.000000	5.000000	5.00
max	5.0	5.0	2.0	5.0	5.000000	4.0	5.0	5.0	5.0	5.0	...	5.000000	5.0	5.0	5.000000	5.000000	5.00

8 rows x 206 columns

```
In [11]: df.shape
```

Out[11]: (4848, 207)

```
In [12]: df_org = df.copy()
```

```
In [13]: df.describe().T
```

Out[13]:

DEEP LEARNING WORKSHOP UP x Assessment x Desktop/Simplilearn Machine Le x Project - Jupyter Notebook x

localhost:8888/notebooks/Desktop/Simplilearn%20Machine%20Learning/Project.ipynb#

jupyter Project Last Checkpoint: 04/23/2021 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [13]: `df.describe().T`

Out[13]:

	count	mean	std	min	25%	50%	75%	max
Movie1	1.0	5.000000	NaN	5.0	5.00	5.0	5.0	5.0
Movie2	1.0	5.000000	NaN	5.0	5.00	5.0	5.0	5.0
Movie3	1.0	2.000000	NaN	2.0	2.00	2.0	2.0	2.0
Movie4	2.0	5.000000	0.000000	5.0	5.00	5.0	5.0	5.0
Movie5	29.0	4.103448	1.496301	1.0	4.00	5.0	5.0	5.0
...
Movie202	6.0	4.333333	1.632993	1.0	5.00	5.0	5.0	5.0
Movie203	1.0	3.000000	NaN	3.0	3.00	3.0	3.0	3.0
Movie204	8.0	4.375000	1.407898	1.0	4.75	5.0	5.0	5.0
Movie205	35.0	4.628571	0.910259	1.0	5.00	5.0	5.0	5.0
Movie206	13.0	4.923077	0.277350	4.0	5.00	5.0	5.0	5.0

206 rows x 8 columns

In [14]: `#Movie with highest views
df.describe().T['count'].sort_values(ascending=False)[:1].to_frame()`

Out[14]:

	count
Movie127	2313.0

6:34 PM 08/05/2021

DEEP LEARNING WORKSHOP UP x Assessment x Desktop/Simplilearn Machine Le x Project - Jupyter Notebook x

localhost:8888/notebooks/Desktop/Simplilearn%20Machine%20Learning/Project.ipynb#

jupyter Project Last Checkpoint: 04/23/2021 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [15]: `#Movie with highest Ratings
df.drop('user_id',axis=1).sum().sort_values(ascending=False)[:1].to_frame()`

Out[15]:

	0
Movie127	9511.0

In [16]: `df.drop('user_id',axis=1).mean().sort_values(ascending=False)[:5].to_frame()`

Out[16]:

	0
Movie1	5.0
Movie55	5.0
Movie131	5.0
Movie132	5.0
Movie133	5.0

In [17]: `df.describe().T['count'].sort_values(ascending=True)[:5].to_frame()`

Out[17]:

	count
Movie1	1.0
Movie71	1.0
Movie145	1.0
Movie69	1.0

6:35 PM 08/05/2021

DEEP LEARNING WORKSHOP UP x Assessment x Desktop/Simplilearn Machine Le x Project - Jupyter Notebook x +

localhost:8888/notebooks/Desktop/Simplilearn%20Machine%20Learning/Project.ipynb#

jupyter Project Last Checkpoint: 04/23/2021 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [17]: `df.describe().T['count'].sort_values(ascending=True)[:5].to_frame()`

Out[17]:

	count
Movie1	1.0
Movie71	1.0
Movie145	1.0
Movie69	1.0
Movie68	1.0

Recommendation System

In [18]: `from surprise import Reader
from surprise import accuracy
from surprise import Dataset
from surprise.model_selection import train_test_split
from surprise import SVD
from surprise.model_selection import cross_validate`

In [19]: `df_melt = df.melt(id_vars = df.columns[0], value_vars=df.columns[1:], var_name="Movies", value_name="Rating")`

In [20]: `df_melt`

Out[20]:

	user_id	Movies	Rating
--	---------	--------	--------

6:35 PM 08/05/2021

DEEP LEARNING WORKSHOP UP x Assessment x Desktop/Simplilearn Machine Le x Project - Jupyter Notebook x +

localhost:8888/notebooks/Desktop/Simplilearn%20Machine%20Learning/Project.ipynb#

jupyter Project Last Checkpoint: 04/23/2021 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [18]: `from surprise import Reader
from surprise import accuracy
from surprise import Dataset
from surprise.model_selection import train_test_split
from surprise import SVD
from surprise.model_selection import cross_validate`

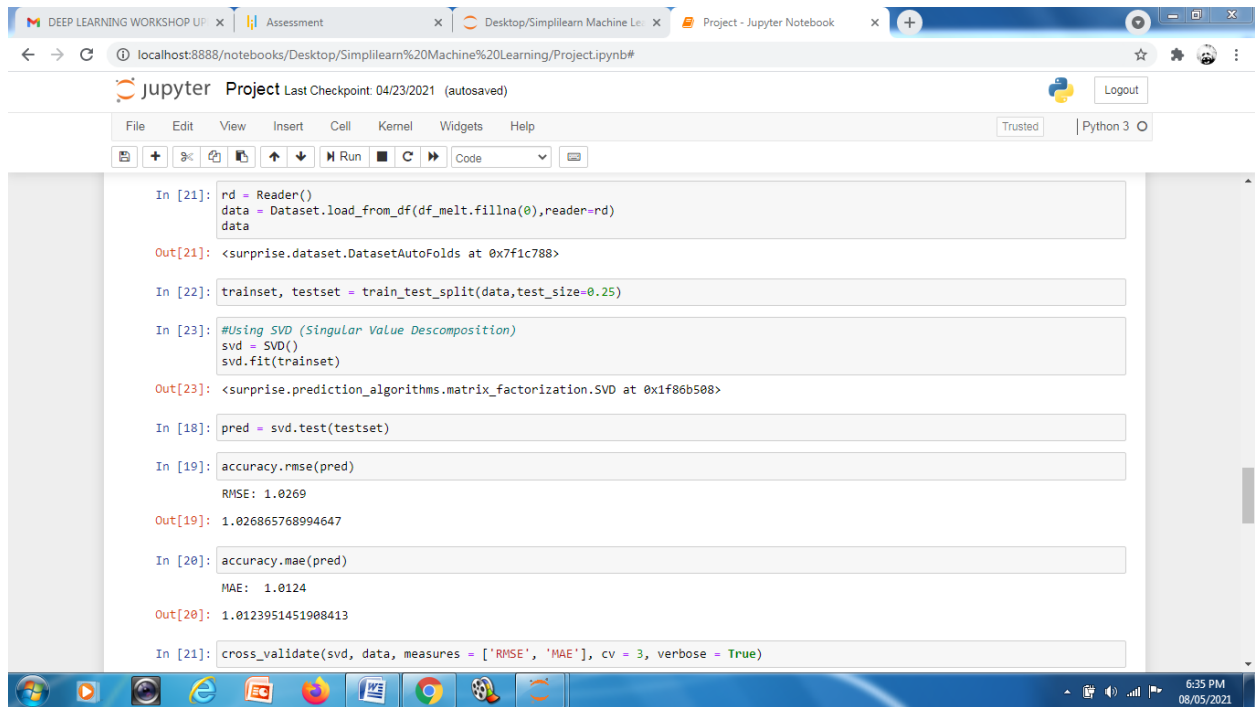
In [19]: `df_melt = df.melt(id_vars = df.columns[0], value_vars=df.columns[1:], var_name="Movies", value_name="Rating")`

In [20]: `df_melt`

Out[20]:

	user_id	Movies	Rating
0	A3R5OBKS7OM2IR	Movie1	5.0
1	AH3QC2PC1VTGP	Movie1	NaN
2	A3LKP6WMP9UKX	Movie1	NaN
3	AVY68KEPQ5ZD	Movie1	NaN
4	A1CV1WROP5KTTW	Movie1	NaN
...
998683	A1IMQ9WMFYKWH5	Movie206	5.0
998684	A1KLIKPUF5E88I	Movie206	5.0
998685	A5HG6WFZLO10D	Movie206	5.0
998686	A3UU690TWXCG1X	Movie206	5.0
998687	A14J76Z16S06	Movie206	5.0

6:35 PM 08/05/2021



```
In [21]: rd = Reader()
data = Dataset.load_from_df(df_melt.fillna(0), reader=rd)

Out[21]: <surprise.dataset.DatasetAutoFolds at 0x7f1c788>

In [22]: trainset, testset = train_test_split(data, test_size=0.25)

In [23]: #Using SVD (Singular Value Decomposition)
svd = SVD()
svd.fit(trainset)

Out[23]: <surprise.prediction_algorithms.matrix_factorization.SVD at 0x1f86b508>

In [18]: pred = svd.test(testset)

In [19]: accuracy.rmse(pred)

RMSE: 1.0269

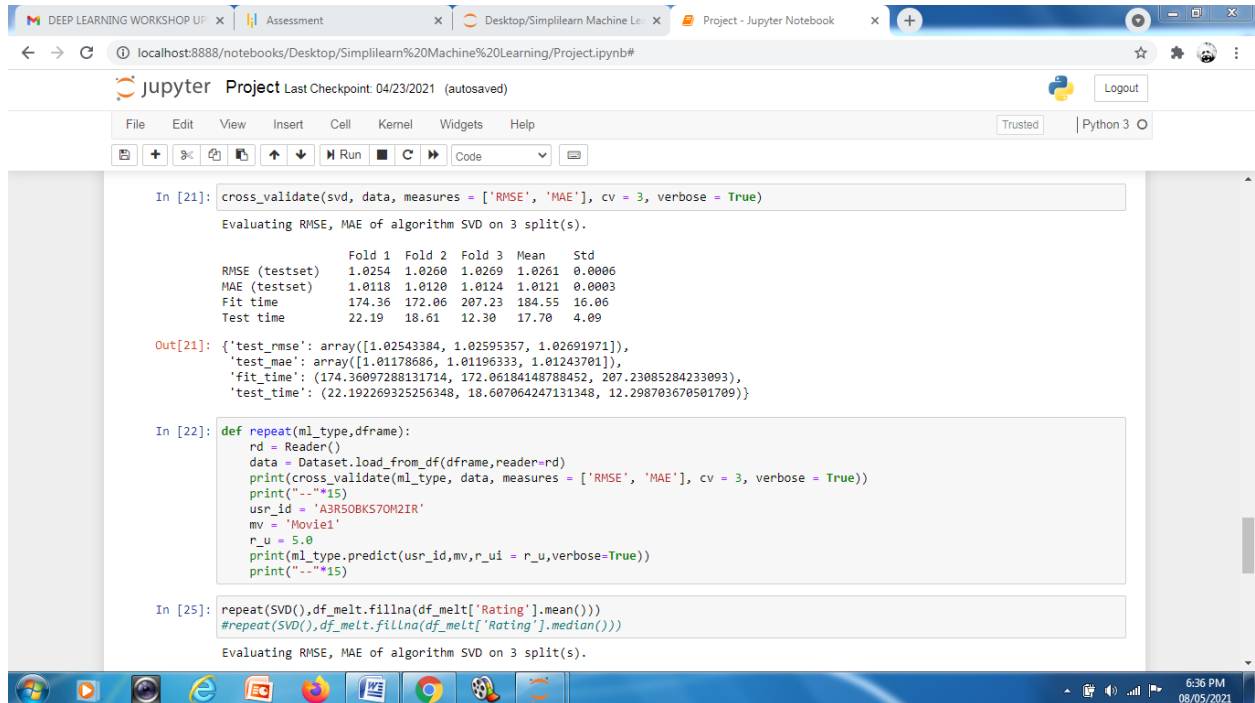
Out[19]: 1.026865768994647

In [20]: accuracy.mae(pred)

MAE: 1.0124

Out[20]: 1.0123951451908413

In [21]: cross_validate(svd, data, measures = ['RMSE', 'MAE'], cv = 3, verbose = True)
```



```
In [21]: cross_validate(svd, data, 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)  1.0254  1.0260  1.0269  1.0261  0.0006
MAE (testset)   1.0118  1.0120  1.0124  1.0121  0.0003
Fit time        174.36  172.06  207.23  184.55  16.06
Test time       22.19   18.61  12.30   17.70   4.09

Out[21]: {'test_rmse': array([1.02543384, 1.02595357, 1.02691971]),
          'test_mae': array([1.01178686, 1.01196333, 1.01243701]),
          'fit_time': (174.36097288131714, 172.06184148788452, 207.23085284233093),
          'test_time': (22.192269325256348, 18.607064247131348, 12.298703670501709)}

In [22]: def repeat(ml_type, df):
rd = Reader()
data = Dataset.load_from_df(df, reader=rd)
print(cross_validate(ml_type, data, measures = ['RMSE', 'MAE'], cv = 3, verbose = True))
print("--*15")
usr_id = 'A3R50BKS7QM2IR'
mv = 'Movie1'
r_u = 5.0
print(ml_type.predict(usr_id, mv, r_ui = r_u, verbose=True))
print("--*15")

In [25]: repeat(SVD(), df_melt.fillna(df_melt['Rating'].mean()))
#repeat(SVD(), df_melt.fillna(df_melt['Rating'].median()))

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

DEEP LEARNING WORKSHOP UP | Assessment | Desktop/Simplilearn Machine Learning | Project - Jupyter Notebook

localhost:8888/notebooks/Desktop/Simplilearn%20Machine%20Learning/Project.ipynb#

jupyter Project Last Checkpoint: 04/23/2021 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
'test_time': (22.192269325256348, 18.607064247131348, 12.298703670501709)}

In [22]: def repeat(ml_type,dfname):
         rd = Reader()
         data = Dataset.load_from_df(dfname,reader=rd)
         print(cross_validate(ml_type, data, measures = ['RMSE', 'MAE'], cv = 3, verbose = True))
         print("--*15)
         usr_id = 'A3R50BK570M2IR'
         mv = 'Movie1'
         r_u = 5.0
         print(ml_type.predict(usr_id,mv,r_ui = r_u,verbose=True))
         print("--*15)

In [25]: repeat(SVD(),df_melt.fillna(df_melt['Rating'].mean()))
         #repeat(SVD(),df_melt.fillna(df_melt['Rating'].median()))

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

RMSE (testset)   Fold 1   Fold 2   Fold 3   Mean   Std
MAE (testset)    0.0853   0.0850   0.0872   0.0858  0.0010
Fit time         130.29   127.28   127.39   128.32  1.39
Test time        11.31   10.09   8.87    10.09   1.00
{'test_rmse': array([0.08534581, 0.08498599, 0.08716732]), 'test_mae': array([0.00966222, 0.00999006, 0.00992907]), 'fit_time':
(130.28770685195923, 127.2842800617218, 127.3852858543396), 'test_time': (11.314647197723389, 10.09357762336731, 8.871507406234
741)}

-----
user: A3R50BK570M2IR item: Movie1   r_ui = 5.00   est = 4.39   {'was_impossible': False}
user: A3R50BK570M2IR item: Movie1   r_ui = 5.00   est = 4.39   {'was_impossible': False}
-----
```

6:36 PM 08/05/2021