final project

1 Business problem

Amazon is an amazing company. They are an important part to many people's lives around the world. Amazon has quickly taken the place of brick and mortars everywhere. The one thing Amazon does not have verses bricks and mortar is the ability to shop with friends. There is nothing like using inspiration from a someone else to inspire your next purchase. Connecting people with like minded individuals is the next big step of virtual shopping experience. Amazon already has user profiles that shows previous reviews in one place. They have also seen the benefit of influencers using affiliate links to drive traffic to their site. This recommendation system will recommend user instead of product to give Amazon more of a social media feel so you can buy directly from the site, eliminating the need for third party sites.

2 Import Data

All imports for this note book will be located below.

2.1 Import Libraries

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt #Draws graphes
import seaborn as sns #Draws intuitive graphs
import warnings #Removes warnings
from glob import glob
import os
import sys
import warnings #Removes warnings
from surprise import Dataset
from surprise.model_selection import train_test_split
from surprise import accuracy
from pyspark.sql import SparkSession
from surprise import Reader
from surprise.prediction_algorithms import SVD
from surprise.model_selection import cross_validate
from surprise import Prediction
from surprise.model_selection import GridSearchCV
from surprise.prediction_algorithms import *
from surprise.similarities import pearson_baseline
import gzip
import json


warnings.filterwarnings("ignore")
%matplotlib inline
```

executed in 2.61s, finished 15:42:13 2021-09-02

2.2 Data

In [2]:

```
cd \Users\laure\Flatiron\Final project\
```

executed in 15ms, finished 15:42:13 2021-09-02

```
C:\Users\laure\Flatiron\Final project
```

In [3]:

```
ls
```

executed in 42ms, finished 15:42:13 2021-09-02

```
 Volume in drive C is Local Disk
 Volume Serial Number is 402C-A0AD

 Directory of C:\Users\laure\Flatiron\Final project

09/02/2021  03:08 PM    <DIR>          .
09/02/2021  03:08 PM    <DIR>          ..
08/31/2021  10:03 PM    <DIR>          .ipynb_checkpoints
10/02/2019  03:56 AM    14,144,939,923 Clothing_Shoes_and_Jewelry.json
08/16/2021  11:20 PM     3,554,445,765 Clothing_Shoes_and_Jewelry.json.gz
10/02/2019  08:49 AM     5,088,375,908 Clothing_Shoes_and_Jewelry_5.json
08/10/2021  08:00 PM     1,262,892,731 Clothing_Shoes_and_Jewelry_5.json.gz
08/17/2021  08:37 PM     1,406,348,800 Clothing_Shoes_and_Jewelry_5_3.json
08/31/2021  02:06 PM         1,942,310 InstagramforAmazon.pptx
09/02/2021  03:08 PM         1,291,940 notebook.ipynb
09/02/2021  01:42 PM    <DIR>          photos
08/31/2021  02:05 PM         1,942,310 Title Lorem Ipsum.pptx
               8 File(s) 25,462,179,687 bytes
               4 Dir(s)  102,039,875,584 bytes free
```

In [4]:

```
data = pd.read_json('Clothing_Shoes_and_Jewelry_5_3.json', lines=True)
data.head()
```

executed in 31.3s, finished 15:42:45 2021-09-02

Out[4]:

	overall	vote	verified	reviewTime	reviewerID	asin	style	reviewer
0	5	2	True	05 4, 2014	A2IC3NZN488KWK	0871167042	{'Format:': ' Paperback'}	Rub
1	5	NaN	True	03 31, 2014	A30FG02C424EJ5	0871167042	{'Format:': ' Paperback'}	NWCance
2	5	NaN	True	05 30, 2015	A2G9GWQEWWNQUB	0871167042	{'Format:': ' Paperback'}	Pamela
3	5	NaN	True	02 21, 2015	A3NI5OGW35SLY2	0871167042	{'Format:': ' Paperback'}	
4	5	NaN	True	01 21, 2015	A1OPRA4NE56EV6	0871167042	{'Format:': ' Paperback'}	carol a

3 Data Exploration

In [5]:
```
data.info(verbose=True, null_counts=True)
```
executed in 2.22s, finished 15:42:47 2021-09-02

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3075277 entries, 0 to 3075276
Data columns (total 12 columns):
 #   Column          Non-Null Count    Dtype
---  ------          --------------    -----
 0   overall         3075277 non-null  int64
 1   vote            239699 non-null   object
 2   verified        3075277 non-null  bool
 3   reviewTime      3075277 non-null  object
 4   reviewerID      3075277 non-null  object
 5   asin            3075277 non-null  object
 6   style           2349881 non-null  object
 7   reviewerName    3075090 non-null  object
 8   reviewText      3073148 non-null  object
 9   summary         3074405 non-null  object
 10  unixReviewTime  3075277 non-null  int64
 11  image           31167 non-null    object
dtypes: bool(1), int64(2), object(9)
memory usage: 261.0+ MB
```

In [6]:
```
type(data)
```
executed in 14ms, finished 15:42:47 2021-09-02

Out[6]:
```
pandas.core.frame.DataFrame
```

In [7]:
```
data.columns
```
executed in 14ms, finished 15:42:47 2021-09-02

Out[7]:
```
Index(['overall', 'vote', 'verified', 'reviewTime', 'reviewerID', 'asin',
       'style', 'reviewerName', 'reviewText', 'summary', 'unixReviewTime',
       'image'],
      dtype='object')
```

In [8]:

```
data_columns = ['overall', 'vote', 'verified', 'reviewTime', 'reviewerID
                'asin','reviewerName', 'unixReviewTime']
for i in data_columns:
    print('\033[1m' + i.upper() + '\033[0m')
    print(data[[i]].value_counts(ascending=False))
    print('_____________________________________________________')
```

executed in 15.7s, finished 15:43:03 2021-09-02

```
OVERALL
overall
5          1946271
4           567679
3           275441
2           149263
1           136623
dtype: int64
_____________________________________________________
VOTE
vote
2        89616
3        43210
4        24667
5        16012
6        10923
         ...
330          1
327          1
317          1
312          1
1,009        1
Length: 413, dtype: int64
_____________________________________________________
VERIFIED
verified
True        2880130
False        195147
dtype: int64
_____________________________________________________
REVIEWTIME
reviewTime
01 15, 2016    9082
07 18, 2017    8133
04 18, 2016    7365
09 14, 2014    5183
12 2, 2015     4386
               ...
09 8, 2006        1
09 8, 2004        1
05 27, 2007       1
09 7, 2006        1
01 1, 2006        1
Length: 4482, dtype: int64
_____________________________________________________
REVIEWERID
reviewerID
A2RYWPOL4NN2KG    113
```

```
A3W4D8XOGLWUN5    111
A2QDOJFFLFGF18     97
A2T9EMIFA72AM6     95
ARTC13N4KKVPT      86
                 ...
A3FFMER98HCVN3      1
A3FFLCWI22BZLA      1
A1X1OKJKCW37UU      1
A3FFL77ZEV4MKM      1
AZZZY1W55XHZR       1
Length: 925678, dtype: int64
```

ASIN

```
asin
B000YXC2LI    19684
B00028AVDG    13888
B0001YRE04    13806
B000XDJ7LW    12454
B000XBM1L2    12432
              ...
B00172QZ40        1
B005BRV93Y        1
B004AGAUXM        1
B005BRF7SW        1
B004R6SNJ8        1
Length: 58329, dtype: int64
```

REVIEWERNAME

```
reviewerName
Amazon Customer        154433
Kindle Customer         17536
Mike                     2740
Chris                    2589
John                     2583
                        ...
Mom to Miss T               1
E&#039;Mari Coggins         1
Mom to Tut                  1
E!                          1
~~~IndianSummer~~~          1
Length: 609517, dtype: int64
```

UNIXREVIEWTIME

```
unixReviewTime
1452816000        9082
1500336000        8133
1460937600        7365
1410652800        5183
1449014400        4386
                  ...
1149292800           1
1149120000           1
1148515200           1
1148342400           1
1529798400           1
Length: 4482, dtype: int64
```

The verified column is an interesting feature to this dataframe. It lets you know whether a purchase was made on amazon or not giving extra authenticity to the review. Granted having an unverified review does not mean it is fake, but a lot of unverified reviews from one user might suggest the user is fake.

In [9]:

```
df_verified = data.loc[data['verified'] == True]
df_verified.head()
```

executed in 588ms, finished 15:43:03 2021-09-02

Out[9]:

	overall	vote	verified	reviewTime	reviewerID	asin	style	reviewer
0	5	2	True	05 4, 2014	A2IC3NZN488KWK	0871167042	{'Format:': ' Paperback'}	Rub
1	5	NaN	True	03 31, 2014	A30FG02C424EJ5	0871167042	{'Format:': ' Paperback'}	NWCance
2	5	NaN	True	05 30, 2015	A2G9GWQEWWNQUB	0871167042	{'Format:': ' Paperback'}	Pamela
3	5	NaN	True	02 21, 2015	A3NI5OGW35SLY2	0871167042	{'Format:': ' Paperback'}	
4	5	NaN	True	01 21, 2015	A1OPRA4NE56EV6	0871167042	{'Format:': ' Paperback'}	carol a

In [10]:

```
df_unverified = data.loc[data['verified'] == False]
df_unverified.head()
```

executed in 109ms, finished 15:43:03 2021-09-02

								Audiobook'}
33	5	NaN	False	12 20, 2015	A3P89EMO989X9D	1519588135	{'Format:': ' Kindle Edition'}	
89	1	NaN	False	11 21, 2014	A2NBLG314SMFLB	3979050432	NaN	
109	5	15	False	02 21, 2013	A1Y09QLADQYQJG	3979050432	NaN	Mic
114	5	5	False	03 1, 2016	A15PB5QABNV5NO	5120053084	{'Size:': ' Small', 'Color:': ' Berry'}	

In [11]:

```
df_unverified_vc= df_unverified['reviewerID'].value_counts(
    ascending=False).to_frame()
```

executed in 202ms, finished 15:43:03 2021-09-02

In [12]:

```
df_unverified_vc.reset_index(inplace= True)
df_unverified_vc.head(5)
```

executed in 31ms, finished 15:43:04 2021-09-02

Out[12]:

	index	reviewerID
0	A3W4D8XOGLWUN5	102
1	A2RYWPOL4NN2KG	97
2	A2QDOJFFLFGF18	89
3	AVU1ILDDYW301	60
4	A2J4XMWKR8PPD0	54

In [13]:

```
df_unverified_vc= df_unverified_vc.rename(columns={'reviewerID':'count'}
```

executed in 14ms, finished 15:43:04 2021-09-02

In [14]:

```
df_unverified_vc= df_unverified_vc.rename(columns={'index':'reviewerID'}
```

executed in 29ms, finished 15:43:04 2021-09-02

In [15]:
```
df_unverified_vc.index.name = "index"
```
executed in 14ms, finished 15:43:04 2021-09-02

In [16]:
```
df_unverified_vc.head()
```
executed in 14ms, finished 15:43:04 2021-09-02

Out[16]:

index	reviewerID	count
0	A3W4D8XOGLWUN5	102
1	A2RYWPOL4NN2KG	97
2	A2QDOJFFLFGF18	89
3	AVU1ILDDYW301	60
4	A2J4XMWKR8PPD0	54

In [17]:
```
df_unverified_vc.value_counts()
```
executed in 348ms, finished 15:43:04 2021-09-02

Out[17]:
```
reviewerID            count
AZZZHECGS8QGE         1        1
A29CLFLQFTQU8         1        1
A29C3QK7NGWIPS        1        1
A29C3SX2XV1ON0        1        1
A29C6RTMJT5BUV        1        1
                              ..
A3IQH7ZC7943DJ        1        1
A3IQJ8Z5314RGA        2        1
A3IQN0UCZTCWFI        4        1
A3IQPD829PJHKU        1        1
A0045558RLEOANWJ9H6A  2        1
Length: 94528, dtype: int64
```

In [18]:
```
df_unverified_vcc = df_unverified_vc.loc[df_unverified_vc['count'] < 50]
```
executed in 15ms, finished 15:43:04 2021-09-02

In [19]:

```
plt.figure(figsize= (100, 350))
sns.barplot(x=df_unverified_vcc['reviewerID']  ,
            y= df_unverified_vcc['count'],  alpha = 0.2)
plt.title(f"Reviewer ID", fontsize=50)
plt.ylabel("# of unverifed purchases", fontsize=100)
plt.xlabel("Reviewer ID", fontsize=50)
#plt.xticks(x, [str(i) for i in y], rotation=90)

#set parameters for tick labels
plt.tick_params(axis='x', which='major', labelsize=100)

plt.tight_layout()
```

executed in 36m 23s, finished 16:19:27 2021-09-02

In [20]:

```
df_unverified_fake = df_unverified_vc.loc[df_unverified_vc['count'] > 6]
```

executed in 13ms, finished 16:19:27 2021-09-02

3.1 review time

In [21]:

```
df_reviewtime_vc= data['reviewTime'].value_counts(ascending=False).to_fr
df_reviewtime_vc.head(5)
```

executed in 1.11s, finished 16:19:29 2021-09-02

Out[21]:

	reviewTime
01 15, 2016	9082
07 18, 2017	8133
04 18, 2016	7365
09 14, 2014	5183
12 2, 2015	4386

In [22]:

```
df_reviewtime_vc.index.name = "date"
df_reviewtime_vc.head(5)
```

executed in 14ms, finished 16:19:29 2021-09-02

Out[22]:

	reviewTime
date	
01 15, 2016	9082
07 18, 2017	8133
04 18, 2016	7365
09 14, 2014	5183
12 2, 2015	4386

In [23]:

```
df_reviewtime_vc= df_reviewtime_vc.rename(columns={'reviewTime':'count'}
df_reviewtime_vc.head(5)
```

executed in 8ms, finished 16:19:29 2021-09-02

Out[23]:

	count
date	
01 15, 2016	9082
07 18, 2017	8133
04 18, 2016	7365
09 14, 2014	5183
12 2, 2015	4386

In [24]:
```
df_reviewtime_vc.reset_index(inplace= True)
df_reviewtime_vc.head(5)
```
executed in 13ms, finished 16:19:29 2021-09-02

Out[24]:

	date	count
0	01 15, 2016	9082
1	07 18, 2017	8133
2	04 18, 2016	7365
3	09 14, 2014	5183
4	12 2, 2015	4386

In [25]:
```
df_reviewtime_vc['count'].value_counts()
```
executed in 14ms, finished 16:19:29 2021-09-02

Out[25]:
```
1       202
2       107
3        58
4        53
5        52
       ...
2627      1
586       1
2635      1
590       1
2049      1
Name: count, Length: 1629, dtype: int64
```

In [26]:
```
df_reviewtime_vc.info()
```
executed in 14ms, finished 16:19:29 2021-09-02

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4482 entries, 0 to 4481
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   date    4482 non-null   object
 1   count   4482 non-null   int64
dtypes: int64(1), object(1)
memory usage: 70.2+ KB
```

In [27]:
```
df_reviewtime_vc['date'] = df_reviewtime_vc['date'].str.replace(
    ',', '').str.replace(' ', '-')
df_reviewtime_vc.head()
```
executed in 30ms, finished 16:19:29 2021-09-02

Out[27]:

	date	count
0	01-15-2016	9082
1	07-18-2017	8133
2	04-18-2016	7365
3	09-14-2014	5183
4	12-2-2015	4386

In [28]:
```
df_reviewtime_vc['date'] = pd.to_datetime(df_reviewtime_vc['date'])
df_reviewtime_vc.head()
```
executed in 126ms, finished 16:19:29 2021-09-02

Out[28]:

	date	count
0	2016-01-15	9082
1	2017-07-18	8133
2	2016-04-18	7365
3	2014-09-14	5183
4	2015-12-02	4386

4 Cleaning

4.1 Dropping Columns

In [29]:
```
data.columns
```
executed in 14ms, finished 16:19:29 2021-09-02

Out[29]:
```
Index(['overall', 'vote', 'verified', 'reviewTime', 'reviewerID', 'asin',
       'style', 'reviewerName', 'reviewText', 'summary', 'unixReviewTime',
       'image'],
      dtype='object')
```

In [30]:
```
data_clean = data
```
executed in 14ms, finished 16:19:29 2021-09-02

In [31]:

```
data_clean
```

executed in 30ms, finished 16:19:29 2021-09-02

Out[31]:

	overall	vote	verified	reviewTime	reviewerID	asin	style
0	5	2	True	05 4, 2014	A2IC3NZN488KWK	0871167042	{'Format:': ' Paperback'}
1	5	NaN	True	03 31, 2014	A30FG02C424EJ5	0871167042	{'Format:': ' Paperback'}
2	5	NaN	True	05 30, 2015	A2G9GWQEWWNQUB	0871167042	{'Format:': ' Paperback'}
3	5	NaN	True	02 21, 2015	A3NI5OGW35SLY2	0871167042	{'Format:': ' Paperback'}
4	5	NaN	True	01 21, 2015	A1OPRA4NE56EV6	0871167042	{'Format:': ' Paperback'}
...	...	...	...	...	...	...	...
3075272	4	NaN	True	08 1, 2013	ASNEDRXLQLRFQ	B005UVTZEQ	{'Size:': ' 2X', 'Color:': ' Black'}
3075273	4	18	False	06 28, 2013	AMJS85VYM2IVU	B005UVTZEQ	{'Size:': ' 3X', 'Color:': ' Black'}
3075274	5	2	True	06 27, 2013	A1K0EIJ3C6BFVT	B005UVTZEQ	{'Size:': ' 3X', 'Color:': ' Beige'}
3075275	2	NaN	True	06 25, 2013	A28EJEA6G82BHM	B005UVTZEQ	{'Size:': ' 1X', 'Color:': ' Black'}
3075276	5	2	True	06 4, 2013	A8US2MLNYUMWJ	B005UVTZEQ	{'Size:': ' 2X', 'Color:': ' Black'}

3075277 rows × 12 columns

In [32]:
```
del data_clean['style']
del data_clean['reviewerName' ]
del data_clean['reviewText']
del data_clean['summary']
del data_clean['unixReviewTime']
del data_clean['image']

```
executed in 1.78s, finished 16:19:31 2021-09-02

In [33]:
```
data_clean.head()
```
executed in 15ms, finished 16:19:31 2021-09-02

Out[33]:

	overall	vote	verified	reviewTime	reviewerID	asin
0	5	2	True	05 4, 2014	A2IC3NZN488KWK	0871167042
1	5	NaN	True	03 31, 2014	A30FG02C424EJ5	0871167042
2	5	NaN	True	05 30, 2015	A2G9GWQEWWNQUB	0871167042
3	5	NaN	True	02 21, 2015	A3NI5OGW35SLY2	0871167042
4	5	NaN	True	01 21, 2015	A1OPRA4NE56EV6	0871167042

4.2 Drop Fake profiles

In [34]:
```
df_unverified_fake.head()
```
executed in 16ms, finished 16:19:31 2021-09-02

Out[34]:

index	reviewerID	count
0	A3W4D8XOGLWUN5	102
1	A2RYWPOL4NN2KG	97
2	A2QDOJFFLFGF18	89
3	AVU1ILDDYW301	60
4	A2J4XMWKR8PPD0	54

In [35]:
```
big_fakes = df_unverified_fake['reviewerID'].to_list()
```
executed in 16ms, finished 16:19:31 2021-09-02

In [36]:
```
data_clean.shape
```
executed in 15ms, finished 16:19:31 2021-09-02

Out[36]: (3075277, 6)

In [37]:
```
data_clean = data_clean[~data_clean['reviewerID'].isin(big_fakes)]
data_clean.shape
```
executed in 936ms, finished 16:19:32 2021-09-02

Out[37]:
```
(3017570, 6)
```

4.3 Drop reviews before 2012

Based on the time line of amazon reviews before 2012 would be less useful to this recommendation system.

In [38]:
```
data_clean.info()
```
executed in 14ms, finished 16:19:32 2021-09-02

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3017570 entries, 0 to 3075276
Data columns (total 6 columns):
 #   Column      Dtype
---  ------      -----
 0   overall     int64
 1   vote        object
 2   verified    bool
 3   reviewTime  object
 4   reviewerID  object
 5   asin        object
dtypes: bool(1), int64(1), object(4)
memory usage: 141.0+ MB
```

In [39]:
```
data_clean['reviewTime'] = data_clean['reviewTime'].str.replace(
    ',', '').str.replace(' ', '-')
data_clean['reviewTime'] = pd.to_datetime(data_clean['reviewTime'] )
data_clean.head()
```
executed in 53.7s, finished 16:20:25 2021-09-02

Out[39]:

	overall	vote	verified	reviewTime	reviewerID	asin
0	5	2	True	2014-05-04	A2IC3NZN488KWK	0871167042
1	5	NaN	True	2014-03-31	A30FG02C424EJ5	0871167042
2	5	NaN	True	2015-05-30	A2G9GWQEWWNQUB	0871167042
3	5	NaN	True	2015-02-21	A3NI5OGW35SLY2	0871167042
4	5	NaN	True	2015-01-21	A1OPRA4NE56EV6	0871167042

In [40]:
```
data_clean.shape
```
executed in 14ms, finished 16:20:25 2021-09-02

Out[40]:
```
(3017570, 6)
```

In [41]:
```
data_clean = data_clean.loc[data_clean['reviewTime'] >= '01-01-2013']
df_verified.shape
```
executed in 410ms, finished 16:20:26 2021-09-02

Out[41]:
```
(2880130, 12)
```

4.4 Dealing with missing data

In [42]:
```
data_clean.columns
```
executed in 16ms, finished 16:20:26 2021-09-02

Out[42]:
```
Index(['overall', 'vote', 'verified', 'reviewTime', 'reviewerID', 'asin'],
dtype='object')
```

In [43]:
```
data_clean_columns = data_clean.columns

for i in data_clean_columns:
    print(str(round(((((data_clean[i].isna().sum()))/len(data_clean))*10
    + '% Null in ' + str(i))

```
executed in 675ms, finished 16:20:27 2021-09-02

```
0.0% Null in overall
93.94% Null in vote
0.0% Null in verified
0.0% Null in reviewTime
0.0% Null in reviewerID
0.0% Null in asin
```

In [44]:
```
data_clean['vote'] = data_clean['vote'].fillna('1')
print(str(round(((((data_clean['vote'].isna().sum()))/
                  len(data_clean))*100),2))
    + '% Null in ' + str('vote'))
```
executed in 281ms, finished 16:20:27 2021-09-02

```
0.0% Null in vote
```

In [45]:

```
data_clean.sort_values(['vote'])
```

executed in 1.70s, finished 16:20:28 2021-09-02

1523896	5	1	True	2016-05-28	A1XDGIHUTX9KJP	B000IO8S0M
1924859	5	1	True	2014-11-13	AGRU4MUO8VPYA	B002B5VN76
1924860	5	1	True	2014-02-15	A3JXJCITB06V21	B002B5VN76
2985859	3	1	True	2013-09-13	A1I06UW9HTDXTX	B005JJ069C
1924862	5	1	True	2013-12-28	A26G99MRADXWB9	B002B5VN76
...	...	...	...	...	...	...
2351513	5	99	True	2016-08-22	A14B4MJ7KZE63B	B004154MAY
2759300	5	99	True	2013-05-30	AVF7R7527YPZ6	B0055J781A
2192430	5	99	True	2016-08-04	ACBK1EEEBQ7KF	B003DBEDF6
359735	2	99	True	2013-10-04	A10O18BTQL8QBE	B000FDSKZO
2462019	1	99	True	2016-09-24	A2ZFC2FR1Q0DAZ	B004CK739A

2805596 rows × 6 columns

In [46]:

```
set(data_clean['vote'].to_list())
```

executed in 61ms, finished 16:20:29 2021-09-02

```
'167',
'168',
'169',
'17',
'170',
'171',
'172',

'173',
'174',
'175',
'176',
'178',
'179',
'18',
'180',
'182',
'183',
'184',
'185',
```

In [47]:

```
data_clean['vote'] = data_clean['vote'].str.replace(',', '').astype('int
```

executed in 1.07s, finished 16:20:30 2021-09-02

In [48]:

```
data_clean.info()
```

executed in 16ms, finished 16:20:30 2021-09-02

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2805596 entries, 0 to 3075276
Data columns (total 6 columns):
 #   Column      Dtype
---  ------      -----
 0   overall     int64
 1   vote        int64
 2   verified    bool
 3   reviewTime  datetime64[ns]
 4   reviewerID  object
 5   asin        object
dtypes: bool(1), datetime64[ns](1), int64(2), object(2)
memory usage: 131.1+ MB
```

In [49]:

```
conditions = [
    data_clean['vote'] < 200,
    data_clean['vote'] < 400,
    data_clean['vote'] < 600,
    data_clean['vote'] < 800,
    True
]

outputs = [1, 2, 3, 4, 5]

data_clean['vote_scaled'] = np.select(conditions, outputs)

```

executed in 46ms, finished 16:20:30 2021-09-02

In [50]:

```
data_clean['vote_weighted']= data_clean['overall']*data_clean['vote_scale
```

executed in 91ms, finished 16:20:30 2021-09-02

In [51]:

```
data_clean.loc[data_clean['vote_scaled'] == 5]
```

executed in 78ms, finished 16:20:30 2021-09-02

Out[51]:

	overall	vote	verified	reviewTime	reviewerID	asin	vote_scaled
226932	5	1359	True	2014-11-25	A20RHN8THTPUZ9	B000A5APXM	5
628822	5	872	False	2015-03-29	A3O7NYI295LUJS	B000QW6LHI	5
774488	1	833	False	2013-10-22	A1QQB7U76YWOO4	B000XBM1L2	5
936724	5	1125	True	2015-02-08	ANWB7D10LPJFU	B00144MBWQ	5
1092481	1	833	False	2013-10-22	A1QQB7U76YWOO4	B000XBM1L2	5
1254717	5	1125	True	2015-02-08	ANWB7D10LPJFU	B00144MBWQ	5
2053418	4	1009	True	2015-06-07	A2LVZ9EWFODEPW	B002RAKPME	5
2188593	5	1066	False	2013-11-30	A1TM1BA7UE68A6	B003CMYTAA	5
2703902	1	886	True	2016-01-21	A3TW5AUCYCMQBM	B00503FUZW	5
2754995	5	823	True	2013-01-07	AESPPXJJOPPYH	B00553XM7K	5
2845180	5	1488	True	2016-05-15	ADQ54QZW2H32J	B005AIIPLS	5

In [52]:

```
data_clean.info()
```

executed in 19ms, finished 16:20:30 2021-09-02

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2805596 entries, 0 to 3075276
Data columns (total 8 columns):
 #   Column         Dtype
---  ------         -----
 0   overall        int64
 1   vote           int64
 2   verified       bool
 3   reviewTime     datetime64[ns]
 4   reviewerID     object
 5   asin           object
 6   vote_scaled    int32
 7   vote_weighted  int64
dtypes: bool(1), datetime64[ns](1), int32(1), int64(3), object(2)
memory usage: 163.2+ MB
```

In [53]:
```
data_clean['vote_weighted'].value_counts()
```
executed in 30ms, finished 16:20:30 2021-09-02

Out[53]:
```
5     1780736
4      515024
3      251014
2      135345
1      123317
10         68
8          32
15         20
6          12
25          7
12          7
20          6
16          4
9           4
Name: vote_weighted, dtype: int64
```

In [54]:
```
#df.label[df.label.eq(>10)].sample(50000).index
#data.loc[data['verified'] == False]
data_clean.loc[data_clean['vote_weighted'] > 7]
```
executed in 30ms, finished 16:20:30 2021-09-02

Out[54]:

	overall	vote	verified	reviewTime	reviewerID	asin	vote_scaled	v
3911	5	516	True	2015-02-09	A3CLWKX8RBR56V	B000074RL3	3	
6118	5	512	True	2016-06-04	A1P9FTJRE3KHDL	B00006XXGO	3	
10160	5	275	True	2013-12-20	A979D2KPVPLW7	B00006XXGO	2	
64694	4	310	True	2014-02-18	A1FSVT19B9MS1A	B00020BFSE	2	
72094	5	295	True	2014-10-11	A3ROYM48FRM3TU	B00028B4XW	2	
...	...	...	...	...	...	...	...	
2975082	5	200	True	2015-11-12	A4CVT63J027K3	B005I6F0RO	2	
2989657	5	242	True	2013-08-28	A17V6IWPTMUXXY	B00302HT86	2	
2991808	5	341	True	2017-02-13	AL3WKMCKNBX3L	B005KIWL0E	2	
2994935	5	346	True	2016-07-13	AEG4HF4461N0E	B005KQAZZO	2	
3035047	4	205	True	2013-06-10	A1GFH4ZNI2CVEQ	B005OVCF8U	2	

148 rows × 8 columns

In [55]:
```
data_clean_sp = data_clean.sample(70000)
```
executed in 167ms, finished 16:20:30 2021-09-02

5 Creating Model

In [56]:

```
reader = Reader(rating_scale=(0, 25))
data = Dataset.load_from_df(data_clean_sp[['asin', 'reviewerID',
                                           'vote_weighted']], reader)
benchmark = []

# Iterate over all algorithms
for algorithm in [SVD(), KNNBaseline(), KNNBasic(), KNNWithMeans(),
                  BaselineOnly()]:

# Perform cross validation
    results = cross_validate(algorithm, data, measures=['RMSE'],
                             cv=3, verbose=False)
    results
# Get results & append algorithm name
    tmp = pd.DataFrame.from_dict(results).mean(axis=0)
    tmp = tmp.append(pd.Series([str(algorithm).split(' ')[0].split('.')[
                               index=['Algorithm']))

    benchmark.append(tmp)

```

executed in 57.6s, finished 16:21:28 2021-09-02

```
Computing the msd similarity matrix...
Done computing similarity matrix.
Estimating biases using als...
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.
Estimating biases using als...
Estimating biases using als...
Estimating biases using als...
```

In [57]:
```
pd.DataFrame(benchmark).set_index('Algorithm').sort_values('test_rmse')
```
executed in 118ms, finished 16:21:28 2021-09-02

Out[57]:

Algorithm	test_rmse	fit_time	test_time
KNNBaseline	1.088620	5.503820	0.278020
BaselineOnly	1.094107	0.279843	0.182998
KNNBasic	1.094285	3.948528	0.236094
SVD	1.096636	2.649043	0.123083
KNNWithMeans	1.099975	4.414231	0.197663

```
BaselineOnly is the base algorithm for this set of data so it will be
used to train the set of data.
```

In [58]:
```
benchmark
```
executed in 28ms, finished 16:21:28 2021-09-02

Out[58]:
```
[test_rmse     1.09664
 fit_time      2.64904
 test_time    0.123083
 Algorithm         SVD
 dtype: object,
 test_rmse        1.08862
 fit_time         5.50382
 test_time        0.27802
 Algorithm    KNNBaseline
 dtype: object,
 test_rmse     1.09429
 fit_time      3.94853
 test_time    0.236094
 Algorithm    KNNBasic
 dtype: object,
 test_rmse         1.09998
 fit_time          4.41423
 test_time        0.197663
 Algorithm    KNNWithMeans
 dtype: object,
 test_rmse         1.09411
 fit_time         0.279843
 test_time        0.182998
 Algorithm    BaselineOnly
 dtype: object]
```

In [59]:

```
from pandas import DataFrame
ben_dict = [[1.09418, 2.65329, 0.124986, 'SVD'],[1.0862, 5.39907, 0.18298
                                                 'KNNBaseline'],
 [1.09226, 4.5777, 0.23998, 'KNNBasic'], [1.09684, 4.47567, 0.623204,
                                          'KNNWithMeans'],
 [1.09281, 0.282202, 0.119337, 'BaselineOnly']]


hhh =DataFrame(ben_dict,columns=['test_rmse','fit_time','test_time',
                                 'Algorithm'])
hhh.set_index('Algorithm')
print (hhh)
```

executed in 28ms, finished 16:21:28 2021-09-02

```
   test_rmse  fit_time  test_time     Algorithm
0    1.09418  2.653290   0.124986           SVD
1    1.08620  5.399070   0.182982   KNNBaseline
2    1.09226  4.577700   0.239980      KNNBasic
3    1.09684  4.475670   0.623204  KNNWithMeans
4    1.09281  0.282202   0.119337  BaselineOnly
```

In [60]:

```
#pickling a model. its a way to save model
```

executed in 12ms, finished 16:21:28 2021-09-02

In [61]:

```
trainset, testset = train_test_split(data, test_size=0.25)
algo = BaselineOnly()
predictions = algo.fit(trainset).test(testset)
accuracy.rmse(predictions)
```

executed in 732ms, finished 16:21:29 2021-09-02

```
Estimating biases using als...
RMSE: 1.1009
```

Out[61]: 1.100889880240353

6 Make Predictions

In [62]:

```
def get_Iu(uid):
    """ return the number of items rated by given user
    args:
      uid: the id of the user
    returns:
      the number of items rated by the user
    """
    try:
        return len(trainset.ur[trainset.to_inner_uid(uid)])
    except ValueError: # user was not part of the trainset
        return 0

def get_Ui(iid):
    """ return number of users that have rated given item
    args:
      iid: the raw id of the item
    returns:
      the number of users that have rated the item.
    """
    try:
        return len(trainset.ir[trainset.to_inner_iid(iid)])
    except ValueError:
        return 0

df = pd.DataFrame(predictions, columns=['uid', 'iid',
                                        'rui', 'est', 'details'])
df['Iu'] = df.uid.apply(get_Iu)
df['Ui'] = df.iid.apply(get_Ui)
df['err'] = abs(df.est - df.rui)
best_predictions = df.sort_values(by='err')[:10]
worst_predictions = df.sort_values(by='err')[-10:]
```

executed in 148ms, finished 16:21:29 2021-09-02

In [63]:

```
best_predictions
```

executed in 28ms, finished 16:21:29 2021-09-02

Out[63]:

	uid	iid	rui	est	details	Iu	Ui	err
1275	B000VK11DY	ALWATS9YVHR7F	4.0	4.001961	{'was_impossible': False}	51	0	0.001961
5577	B000VK11DY	A2BHE0T7G6I0XM	4.0	4.001961	{'was_impossible': False}	51	0	0.001961
13216	B000VK11DY	A1YNII97T16H5F	4.0	4.001961	{'was_impossible': False}	51	0	0.001961
16530	B000VK11DY	A5YGVOR6SYFRT	4.0	4.001961	{'was_impossible': False}	51	0	0.001961
5031	B000VK11DY	A1EBDALRZY29H8	4.0	4.001961	{'was_impossible': False}	51	0	0.001961
17272	B000VK11DY	AMN85MC0WA654	4.0	4.001961	{'was_impossible': False}	51	0	0.001961
9494	B0009GI0P2	A2E3CBQB52PTRZ	4.0	4.007524	{'was_impossible': False}	14	2	0.007524
2214	B000PCFNBY	A2SN61QOY0FL68	4.0	4.007602	{'was_impossible': False}	7	0	0.007602
2987	B000FXZV4W	A35ATZSOJN2F9W	4.0	4.008586	{'was_impossible': False}	13	0	0.008586
15734	B000FXZV4W	A2QX3NWJCYK5AU	4.0	4.008586	{'was_impossible': False}	13	0	0.008586

In [64]:
```
worst_predictions
```
executed in 32ms, finished 16:21:29 2021-09-02

Out[64]:

	uid	iid	rui	est	details	Iu	Ui	err
16071	B00016QPCU	A17S02TFWA79BK	1.0	4.624289	{'was_impossible': False}	35	0	3.624289
1562	B005H58ZWI	A3UI5ODGOLAXEI	8.0	4.372500	{'was_impossible': False}	3	0	3.627500
9349	B002G9UEG8	A2WIJH3N7PUAJF	1.0	4.654456	{'was_impossible': False}	51	0	3.654456
14876	B0058XISVW	A1OOVJB6JK3VNI	1.0	4.661079	{'was_impossible': False}	46	0	3.661079
15563	B00138VUXE	A31M39Y2KS83L9	1.0	4.671563	{'was_impossible': False}	18	0	3.671563
15440	B001AQ4AFY	A27FPKOKRNWKJM	1.0	4.709896	{'was_impossible': False}	53	0	3.709896
5311	B001LNCAPS	A1YE1DJBPQR2Y4	1.0	4.739374	{'was_impossible': False}	7	0	3.739374
14557	B000ZQ667A	A2YXE4V52LPMSS	8.0	4.212299	{'was_impossible': False}	8	0	3.787701
7310	B001DCEKXM	A2KFDAO3GXT07A	10.0	4.417839	{'was_impossible': False}	3	0	5.582161
3778	B00553XM7K	AESPPXJJOPPYH	25.0	4.296404	{'was_impossible': False}	21	0	20.703596

In [65]:
```
data_clean.columns
```
executed in 12ms, finished 16:21:29 2021-09-02

Out[65]:
```
Index(['overall', 'vote', 'verified', 'reviewTime', 'reviewerID', 'asin',
       'vote_scaled', 'vote_weighted'],
      dtype='object')
```

In [66]:

```
# user_id is the 13618
ratings = data_clean.loc[data_clean['asin'] == 'B000NZKD18']
# obtain the required data of this user
ratings=ratings[['asin', 'reviewerID', 'vote_weighted']]
ratings
```

executed in 3.20s, finished 16:21:32 2021-09-02

Out[66]:

	asin	reviewerID	vote_weighted
563708	B000NZKD18	A4VHNZO7MN2LM	5
563709	B000NZKD18	A3CN3AGZD6PY3K	5
563710	B000NZKD18	A1YHO9KQNI1MUD	5
563711	B000NZKD18	AKTQHVPAT5LF7	5
563712	B000NZKD18	A2LWYFKEMXDSK4	5
...	...	...	...
572208	B000NZKD18	A1L5K7EN5P0YK2	5
572209	B000NZKD18	AH13EGBR8LYC	5
572211	B000NZKD18	A39HQ0QURJ2AKS	5
572213	B000NZKD18	A2JRV85ZJCU48R	5
572214	B000NZKD18	ATQMAED8PBXTG	5

6671 rows × 3 columns

In [67]:

```
# get the list of the movie ids
unique_ids = data_clean['reviewerID'].unique()
# get the list of the ids that the userid B000NZKD18 has watched
iids1001 = data_clean.loc[data_clean['asin']=='B000NZKD18', 'reviewerID'
# remove the rated movies for the recommendations
movies_to_predict = np.setdiff1d(unique_ids,iids1001)
```

executed in 13m 20s, finished 16:34:52 2021-09-02

In [68]:

```
algo = BaselineOnly()
algo.fit(data.build_full_trainset())
my_recs = []
for iid in movies_to_predict:
 my_recs.append((iid, algo.predict(uid='B000NZKD18',iid=iid).est))
pd.DataFrame(my_recs, columns=['iid', 'predictions']).sort_values(
    'predictions', ascending=False).head(10)
```

executed in 7.31s, finished 16:35:00 2021-09-02

```
Estimating biases using als...
```

Out[68]:

	iid	predictions
752113	AESPPXJJOPPYH	6.230255
518484	A37BAUTPCU2R18	5.390690
651547	A3ROYM48FRM3TU	4.920149
72347	A1B2LEW6RAFIKN	4.880262
369323	A2KFDAO3GXT07A	4.871148
19605	A12YSW2P8SOLNX	4.870464
463946	A2YXE4V52LPMSS	4.719618
670090	A3UI5ODGOLAXEI	4.699560
12946	A11Y35JOOKDOLK	4.625046
234033	A1ZQVNX62VJADT	4.621580