Recommendation Systems for Amazon Beauty Products

Overview

For the purpose of this project, we are a team of internal data scientists at Amazon. This project aims to create a recommendation system for the Amazon marketing team to utilize to send targeted recommendation e-mails to users who have purchased and rated products within 30 days. A collaborative approach was taken, meaning recommendations will be made by comparing similar reviewer profiles based on existing ratings.

Business Problem

Amazon's marketing team for beauty products has recognized a big opportunity to improve the emails they send to customers following a purchase. Customers open these post-purchase emails 17% more often than other types of automated emails. In addition to the visibility of post-purchase emails, their timing is critically important in the lifecyle of a customer. Succesfully re-engaging a customer at the post-purchase stage places them back into a consideration stage which will eventually lead to future purchases, increasing the customer's purchase frequency and lifetime value. However, re-engagment depends on these emails featuring content that customers want to engage with.

To help improve the engagment with their post-purchase emails, Amazon's beauty marketing team has pulled in an internal group of data scientists to create a recommendation system that will select personalized product recommendations for customers to include in post-purchase emails. Succesfull product recommendations must be personalized, relevant, and timely. Personalized in that they accurately predict products a given customer will enjoy, relevant because they don't recommend products the customer just purchased, and timely becuase they have the ability to tailor recommendations to seasonal events.

Data Understanding and Preparation

Data for this project was pulled from a compiled dataset of Amazon Beauty product reviews and meta data in two seperate JSON files. The datasets can be found here (LINK TO SITE). We utlized the smaller dataset known as 5-core which contained data for products and reviewers with at least 5 entries.

Our review data contained 198,502 reviews from 22,363 reviewers. The reviews spanned across 12,101 unique products. Reviews ranged on a scale of 1-5. A majority of reviews received an overall review of 5, which could be a limitation to our model.

Our data did not require much cleaning. We selected the appropriate columns of our model to utilize for surprise, which included 'reviewerID', 'asin', and 'overall'. This data contained our unique reviewer ID, unique product ID, and overall rating on a scale of 1-5.

In [1]:
```
#importing necessary imports
```

In [2]:
```
import pandas as pd
import pickle
import matplotlib.pyplot as plt

from surprise import Dataset, Reader, accuracy, NormalPredictor, KNNBasic,
from surprise.accuracy import rmse
from surprise.model_selection import cross_validate, train_test_split, Grid
from surprise.prediction_algorithms import SVD, SVDpp, NMF, BaselineOnly, N
from IPython.core.display import HTML

%matplotlib inline
```

Exploring Review Data

In [3]:
```
#reading in our data as a dataframe
```

In [4]:
```
df = pd.read_json("Data/reviews_Beauty_5.json.gz",lines=True)
```

In [5]:
```
df
```

Out[5]:

	reviewerID	asin	reviewerName	helpful	reviewText	overall	summary
0	A1YJEY40YUW4SE	7806397051	Andrea	[3, 4]	Very oily and creamy. Not at all what I expect...	1	Don't waste your money
1	A60XNB876KYML	7806397051	Jessica H.	[1, 1]	This palette was a decent price and I was look...	3	OK Palette!
2	A3G6XNM240RMWA	7806397051	Karen	[0, 1]	The texture of this concealer pallet is fantas...	4	great quality
3	A1PQFP6SAJ6D80	7806397051	Norah	[2, 2]	I really can't tell what exactly this thing is...	2	Do not work on my face
4	A38FVHZTNQ271F	7806397051	Nova Amor	[0, 0]	It was a little smaller than I expected, but t...	3	It's okay.
...	...	...	...	...	...	...	...
198497	A2BLFCOPSMBOZ9	B00LLPT4HI	Dave Edmiston	[0, 0]	Just a little dab of this shea butter should b...	5	A little dab...
198498	A1UQBFCERIP7VJ	B00LLPT4HI	Margaret Picky	[0, 0]	This shea butter is completely raw and unrefin...	5	Pure organic raw shea butter
198499	A35Q0RBM3YNQNF	B00LLPT4HI	M. Hill	[0, 0]	The skin is the body's largest organ and it ab...	5	One Pound Organic Grade A Unrefined Shea Butter
198500	A3LGT6UZL99IW1	B00LLPT4HI	Richard C. Drew "Anaal Nathra/Uthe vas Bethod...	[0, 0]	I have very dry elbows and knees. I have a to...	5	This stuff is amazing!

	reviewerID	asin	reviewerName	helpful	reviewText	overall	summary
198501	A3UJRNI8UR4871	B00LLPT4HI	Wulfstan "wulfstan"	[0, 1]	This is 100% pure Shea Butter. Do not mistake ...	5	The "Real Stuff"!

198502 rows × 9 columns

In [6]:
```
#checking for nulls
```

In [7]:
```
df.isna().sum()
```

Out[7]:
```
reviewerID           0
asin                 0
reviewerName      1386
helpful              0
reviewText           0
overall              0
summary              0
unixReviewTime       0
reviewTime           0
dtype: int64
```

In [8]:
```
#1386 reviewerNames left blank; we will not need reviewerName since utilizi
#checking that all needed information is provided when reviewerName is NaN
```

In [9]:
```
df[df['reviewerName'].isnull()]
```

Out[9]:

	reviewerID	asin	reviewerName	helpful	reviewText
8	A3LMILRM9OC3SA	9759091062	NaN	[0, 0]	Did nothing for me. Stings when I put it on. I...
1790	AK1H26O8DLMNN	B0000535UM	NaN	[0, 0]	The first thickening shampoo that works on my ...
2242	APTLHR9PHGPXN	B00005NAOD	NaN	[0, 0]	Kind of drying, not moisturizing. Kind of disa...
2304	AQWX644AFUFFK	B00005NFBD	NaN	[0, 0]	This is just ok. For one, I found this in a st...
3651	A43K5ZRQ87TO6	B00008PC1O	NaN	[0, 0]	Works well and easy to use!
...	...	...	...	...	...
197192	A1Z3AV93ONK5VF	B00KAL5JAU	NaN	[0, 0]	We already had the Dead Sea Shampoo by Adovia ...
197193	A184I8GT3BHZQV	B00KAL5JAU	NaN	[0, 1]	<a href="http://www.tomoson.com/?code=...
197194	A8C9EJORQD23	B00KAL5JAU	NaN	[0, 1]	I use this with the Adovia shampoo I mention a...
198446	A2PIGZCDGM4NJ7	B00L5JHZJO	NaN	[10, 11]	This is a horrible product, most of the review...
198447	A3M1ADU4JICQR2	B00L5JHZJO	NaN	[5, 6]	I bought this for my wife, as she loves using ...

1386 rows × 9 columns

In [10]:
```
#checking that all reviewers have completed at least 5 reviews
```

In [11]:
```
df.reviewerID.value_counts()
```

Out[11]:
```
A2V5R832QCSOMX    204
ALNFHVS3SC4FV     192
AKMEY1BSHSDG7     182
A3KEZLJ59C1JVH    154
ALQGOMOY1F5X9     150
                 ...
A2EAP2JZR5S1O6      5
A3RG1M07FULK14      5
AB6IV1YFCZKQH       5
A1U04ZOMR0F2HL      5
AAE64Q4WY6O4P       5
Name: reviewerID, Length: 22363, dtype: int64
```

In [12]:
```
#YAY! all reviewerIDs have value of at least 5, total of 22,363 reviewers
```

In [13]:
```
#checking that all products have been reviewed at least 5 times
```

In [14]:
```
df.asin.value_counts()
```

Out[14]:
```
B004OHQR1Q    431
B0043OYFKU    403
B0069FDR96    391
B000ZMBSPE    389
B00150LT40    329
             ...
B00005NAOJ      5
B001EJOPV6      5
B0087J0EQG      5
B000P27754      5
B00005375C      5
Name: asin, Length: 12101, dtype: int64
```

In [15]:
```
#YAY! all products have at least 5 reviews, total of 12,101 different produ
```

In [16]:
```
#looking at ratings distribution
```

In [17]:
```
import os

if not os.path.exists("images"):
    os.mkdir("images")
```

In [18]:

```
from plotly.offline import init_notebook_mode, plot, iplot
import plotly.graph_objs as go
from plotly.io import to_image
init_notebook_mode(connected=True)

data = df['overall'].value_counts().sort_index(ascending=False)
trace = go.Bar(x = data.index,
               text = ['{:.1f} %'.format(val) for val in (data.values / df.
               textposition = 'auto',
               textfont = dict(color = '#000000'),
               y = data.values,
               )
# Create layout
layout = dict(title = 'Distribution Of {} Reviews'.format(df.shape[0]),
              xaxis = dict(title = 'Rating'),
              yaxis = dict(title = 'Count'))
# Create plot
fig = go.Figure(data=[trace], layout=layout)
iplot(fig)

 # Do this first so we don't create a file if image conversion fails
img_data = to_image(fig,
        format='png',
        width=800,
        height=500,
        scale=5)

fig.write_image("images/reviews_distribution.png", scale=5)
```

In [19]:
```
#we see a majority of our ratings are 5s, which could impact our system
```

In [20]:
```
#exploring number of ratings per product
```

In [21]:
```
data = df.groupby('asin')['overall'].count().clip(upper=50)
```

In [22]:
```
data
```

Out[22]:
```
asin
7806397051     8
9759091062    11
9788072216     5
9790790961     6
9790794231     5
              ..
B00L5KTZ0K    15
B00L6Q3BH6     5
B00LCEROA2     9
B00LG63DOM    10
B00LLPT4HI     7
Name: overall, Length: 12101, dtype: int64
```

In [23]:
```
data = df.groupby('asin')['overall'].count()
```

In [24]:
```
data
```

Out[24]:
```
asin
7806397051     8
9759091062    11
9788072216     5
9790790961     6
9790794231     5
              ..
B00L5KTZ0K    15
B00L6Q3BH6     5
B00LCEROA2     9
B00LG63DOM    10
B00LLPT4HI     7
Name: overall, Length: 12101, dtype: int64
```

In [25]:

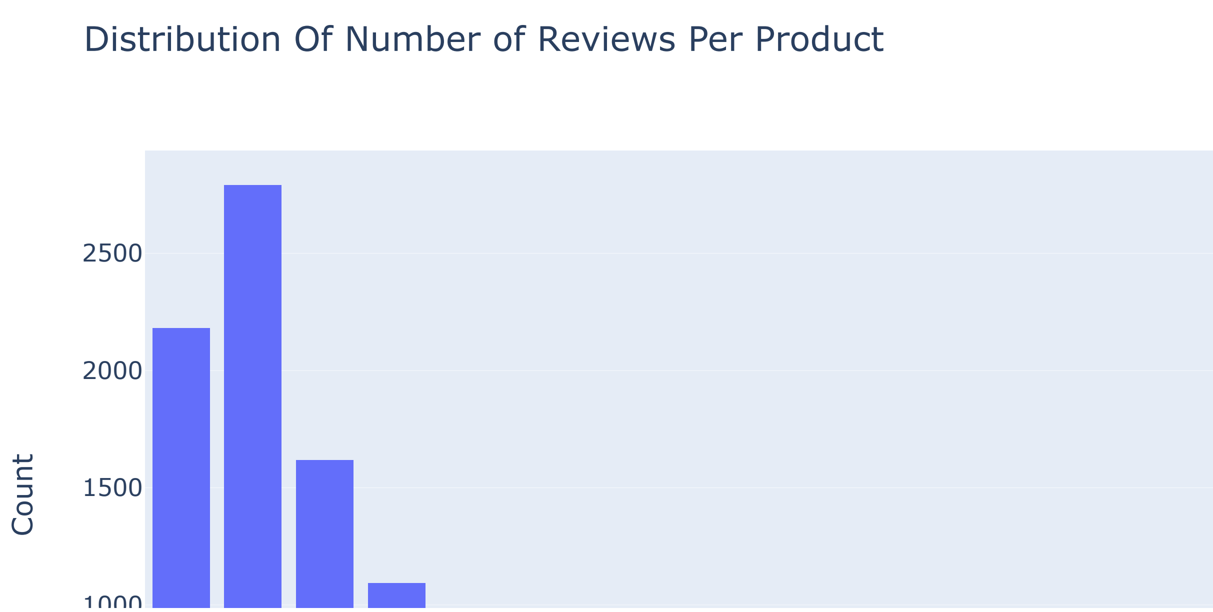
```
# Number of reviews per product
data = df.groupby('asin')['overall'].count()

# Create trace
trace = go.Histogram(x = data.values,
                     name = 'Ratings',
                     xbins = dict(start = 0,
                                  end = 50,
                                  size = 2))
# Create layout
layout = go.Layout(title = 'Distribution Of Number of Reviews Per Product',
                   xaxis = dict(title = 'Number of Reviews Per Product'),
                   yaxis = dict(title = 'Count'),
                   bargap = 0.2)

# Create plot
fig = go.Figure(data=[trace], layout=layout)
iplot(fig)

 # Do this first so we don't create a file if image conversion fails
img_data = to_image(fig,
        format='png',
        width=800,
        height=500,
        scale=5)

fig.write_image("images/reviews_per_product.png", scale=5)
```



In [26]:
```
#We see a majority of our projects have 10 or less ratings
```

In [27]:
```
#exploring ratings distribution by user
```

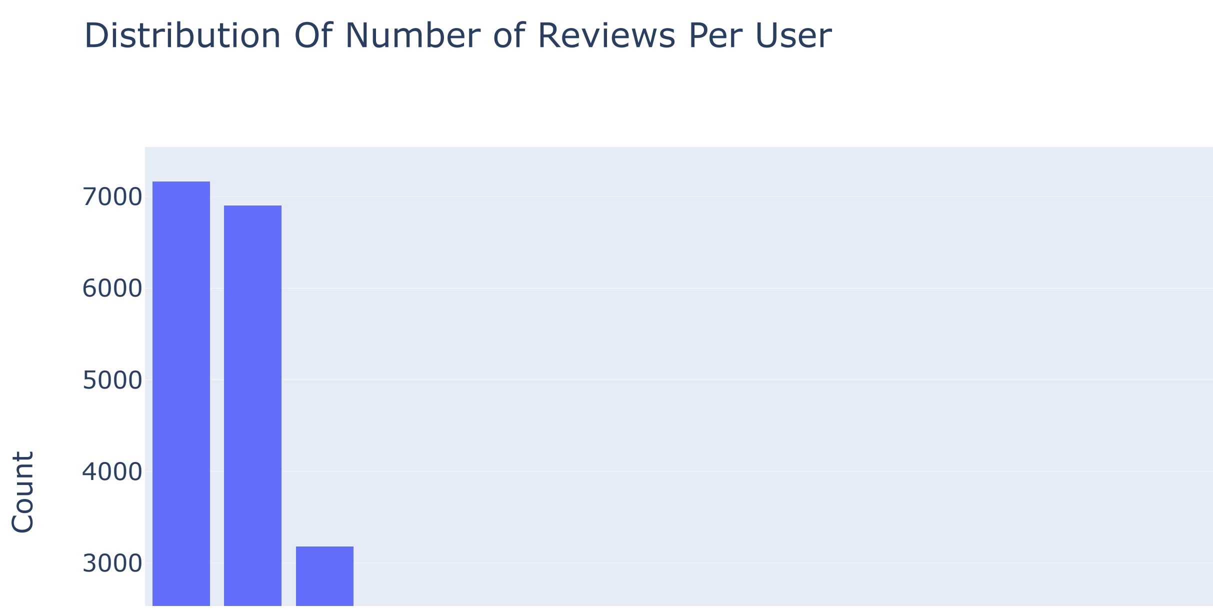
In [28]:
```
# Number of reviews per user
data = df.groupby('reviewerID')['overall'].count()

trace = go.Histogram(x = data.values,
                     name = 'Ratings',
                     xbins = dict(start = 0,
                                  end = 50,
                                  size = 2))
# Create layout
layout = go.Layout(title = 'Distribution Of Number of Reviews Per User',
                   xaxis = dict(title = 'Reviews Per User'),
                   yaxis = dict(title = 'Count'),
                   bargap = 0.2)

# Create plot
fig = go.Figure(data=[trace], layout=layout)
iplot(fig)

 # Do this first so we don't create a file if image conversion fails
img_data = to_image(fig,
        format='png',
        width=800,
        height=500,
        scale=5)

fig.write_image("images/reviews_per_user.png", scale=5)
```



In [29]:
```
#we see most users rated under 10 products
```

In [30]:
```
lower_rating = df.overall.min()
```

In [31]:
```
upper_rating = df.overall.max()
```

In [32]:
```
#Confirming our review range is 1 to 5
```

In [33]:
```
print('Review range: {0} to {1}'.format(lower_rating, upper_rating))
```

```
Review range: 1 to 5
```

In [34]:
```
#Creating dataframe with appropriate columns to run through surprise
```

In [35]:
```
surprise_df = df[['reviewerID', 'asin', 'overall']]
```

In [36]:
```
surprise_df
```

Out[36]:

	reviewerID	asin	overall
0	A1YJEY40YUW4SE	7806397051	1
1	A60XNB876KYML	7806397051	3
2	A3G6XNM240RMWA	7806397051	4
3	A1PQFP6SAJ6D80	7806397051	2
4	A38FVHZTNQ271F	7806397051	3
...	...	...	...
198497	A2BLFCOPSMBOZ9	B00LLPT4HI	5
198498	A1UQBFCERIP7VJ	B00LLPT4HI	5
198499	A35Q0RBM3YNQNF	B00LLPT4HI	5
198500	A3LGT6UZL99IW1	B00LLPT4HI	5
198501	A3UJRNI8UR4871	B00LLPT4HI	5

198502 rows × 3 columns

In [37]:
```
#Checking average rating user to see if there are users who rate everything
```

In [38]:
```
avg_rating_user = df.groupby("reviewerID")["overall","reviewerID"].mean().s
avg_rating_user
```

```
<ipython-input-38-8752c1212ea2>:1: FutureWarning:

Indexing with multiple keys (implicitly converted to a tuple of keys) wil
l be deprecated, use a list instead.
```

Out[38]:

	overall
reviewerID	
A1W522Z24EPBJB	1.0
A2DPSPXFJ507C0	1.0
A1GQLVT0SWAWU	1.0
A1KLA02LZXAT46	1.0
A2MHHSACEJANSX	1.0
...	...
A15QGN6UXJVW9G	5.0
ANOJX4RAUJ9HL	5.0
A2RJT3IE2T6KXJ	5.0
A1ORLBQV893JF0	5.0
A4UHZXSLMBWT2	5.0

22363 rows × 1 columns

In [39]:
```
low_rating_user = avg_rating_user[avg_rating_user["overall"]==1.0]
low_rating_user
```

Out[39]:

	overall
reviewerID	
A1W522Z24EPBJB	1.0
A2DPSPXFJ507C0	1.0
A1GQLVT0SWAWU	1.0
A1KLA02LZXAT46	1.0
A2MHHSACEJANSX	1.0
A2RJTIE73NPN3C	1.0
ASWIC85F71H4J	1.0
A2TBE0N8JN6H4K	1.0
A1GPPMHYM6SMEW	1.0

In [40]:
```
#only 9 users have rated every product a 1
```

In [41]:
```
high_rating_user = avg_rating_user[avg_rating_user["overall"]==5.0]
high_rating_user
```

Out[41]:

	overall
reviewerID	
A2FINIRQNXOTI	5.0
ATWS89FH6Y6S4	5.0
A16Q479PYT0G6N	5.0
A3OKW5VRXZG3OQ	5.0
A3O9Q3154FPZLL	5.0
...	...
A15QGN6UXJVW9G	5.0
ANOJX4RAUJ9HL	5.0
A2RJT3IE2T6KXJ	5.0
A1ORLBQV893JF0	5.0
A4UHZXSLMBWT2	5.0

2822 rows × 1 columns

In [42]:
```
#2822 users have rated every product a 5
```

In [43]:
```
#we decide to keep these users in our final dataset but will not use them t
```

In [44]:
```
#elaborate on justification for keeping these reviews
```

Exploring Meta Data

In [45]:
```
#Import our meta data
import gzip

def parse(path):
  g = gzip.open(path, 'rb')
  for l in g:
    yield eval(l)

def getDF(path):
  i = 0
  df = {}
  for d in parse(path):
    df[i] = d
    i += 1
  return pd.DataFrame.from_dict(df, orient='index')
```

In [46]:
```python
meta_data_df = getDF("Data/meta_Beauty.json.gz")
meta_data_df
```

Out[46]:

	asin	description	title	imUrl	sale
0	0205616461	As we age, our once youthful, healthy skin suc...	Bio-Active Anti-Aging Serum (Firming Ultra-Hyd...	http://ecx.images-amazon.com/images/I/41DecrGO...	{'H Pe 4
1	0558925278	Mineral Powder Brush--Apply powder or mineral ...	Eco Friendly Ecotools Quality Natural Bamboo C...	http://ecx.images-amazon.com/images/I/51L%2BzY...	{'B 4
2	0733001998	From the Greek island of Chios, this Mastiha b...	Mastiha Body Lotion	http://ecx.images-amazon.com/images/I/311WK5y1...	{'B 5
3	0737104473	Limited edition Hello Kitty Lipstick featuring...	Hello Kitty Lustre Lipstick (See sellers comme...	http://ecx.images-amazon.com/images/I/31u6Hrzk...	{'B 9
4	0762451459	The mermaid is an elusive (okay, mythical) cre...	Stephanie Johnson Mermaid Round Snap Mirror	http://ecx.images-amazon.com/images/I/41y2%2BF...	
...	...	...	...	...	
259199	B00LP2YB8E	Color: White\nFullness72 inches\nCenter Gather...	2t 2t Edge Crystal Rhinestones Bridal Wedding ...	http://ecx.images-amazon.com/images/I/41E630m-...	
259200	B00LOS7MEE	The secret to long lasting colors, healthy nai...	French Manicure Gel Nail Polish Set - "Se...	http://ecx.images-amazon.com/images/I/41skHL1O...	{'B 1
259201	B00LPVG6V0	ResQ Organics Face & Body Wash - With Aloe Ver...	ResQ Organics Face & Body Wash - Aloe Vera...	http://ecx.images-amazon.com/images/I/31C1w4Ku...	
259202	B00LTDUHJQ	Color: White\n2 Tier \nFullness 72 inches\nSew...	2 Tier Tulle Elbow Wedding Veil with Ribbon Ed...	http://ecx.images-amazon.com/images/I/51%2B%2B...	
259203	B00LU0LTOU	The bags produced by us are 100% ECO friendly ...	*ECOCRAFTWORLD* GENUINE BUFFALO LEATHER TRAVEL...	http://ecx.images-amazon.com/images/I/41kXSEch...	

259204 rows × 9 columns

```
In [47]: meta_data_df.head(100)
```

Out[47]:

	asin	description	title	imUrl	salesRank	categories
0	0205616461	As we age, our once youthful, healthy skin suc...	Bio-Active Anti-Aging Serum (Firming Ultra-Hyd...	http://ecx.images-amazon.com/images/I/41DecrGO...	{'Health & Personal Care': 461765}	[[Beauty, Skin Care, Face, Creams & Moisturize...
1	0558925278	Mineral Powder Brush--Apply powder or mineral ...	Eco Friendly Ecotools Quality Natural Bamboo C...	http://ecx.images-amazon.com/images/I/51L%2BzY...	{'Beauty': 402875}	[[Beauty, Tools & Accessories, Makeup Brushes ...
2	0733001998	From the Greek island of Chios, this Mastiha b...	Mastiha Body Lotion	http://ecx.images-amazon.com/images/I/311WK5y1...	{'Beauty': 540255}	[[Beauty, Skin Care, Body, Moisturizers, Lotio...
3	0737104473	Limited edition Hello Kitty Lipstick featuring...	Hello Kitty Lustre Lipstick (See sellers comme...	http://ecx.images-amazon.com/images/I/31u6Hrzk...	{'Beauty': 931125}	[[Beauty, Makeup, Lips, Lipstick]]
4	0762451459	The mermaid is an elusive (okay, mythical) cre...	Stephanie Johnson Mermaid Round Snap Mirror	http://ecx.images-amazon.com/images/I/41y2%2BF...	NaN	[[Beauty, Tools & Accessories, Mirrors, Makeup...
...	...	...	...	...	...	...
95	6041134473	Restore your skin's firmness and elasticity.\n...	Cellulite Massager Face Lift Face Massager 24k...	http://ecx.images-amazon.com/images/I/41Mv9hUf...	{'Beauty': 26646}	[[Beauty, Skin Care, Face]]
96	6040652705	Cure time: 3 minutes. Performs with ease when ...	Atnails Nail Uv Gel - Extreme White - French M...	http://ecx.images-amazon.com/images/I/41EPV9ft...	{'Beauty': 554040}	[[Beauty, Makeup, Nails, Nail Polish]]
97	6041134511	Gold has magic energy of resisting oxidation.\...	24k Gold Vibrating Face Lifting Tightening T S...	http://ecx.images-amazon.com/images/I/511oshir...	{'Beauty': 397584}	[[Beauty, Skin Care, Sets & Kits]]

	asin	description	title	imUrl	salesRank	categories
98	604113449X	The Extra 600 Titanium Micro Needles per rolle...	Derma Roller Titanium 1.0mm 600 Micro Needles ...	http://ecx.images-amazon.com/images/I/41kVoMvq...	{'Beauty': 80310}	[[Beauty, Skin Care, Face, Treatments & Masks]]
99	6053640972	Worried about your hair loss, tired of using p...	Toppik Hair Building Fibers Travel Size Small ...	http://ecx.images-amazon.com/images/I/419hAhz1...	{'Beauty': 319142}	[[Beauty, Hair Care, Hair Loss Products, Styli...

100 rows × 9 columns

In [48]:
```
meta_data_df.isna().sum()
```

Out[48]:
```
asin                0
description     24707
title             444
imUrl              88
salesRank        5188
categories          0
price           69274
related         51350
brand          131038
dtype: int64
```

In [49]:
```
#exploring NaN and deciding which data is helpful to return to our users fo
```

In [50]:
```
meta_data_df.shape
```

Out[50]:
```
(259204, 9)
```

In [51]:
```
from IPython import display
display.Image(meta_data_df.loc[192]["imUrl"])
#display.Image(meta_data_df_cleaned.loc[259179]["imUrl"])
```

Out[51]:

In [52]:
```
meta_data_df.price.describe()
```

Out[52]:
```
count    189930.000000
mean         24.878165
std          33.431190
min           0.010000
25%           8.240000
50%          15.690000
75%          29.300000
max         999.990000
Name: price, dtype: float64
```

In [53]:
```
meta_data_df[meta_data_df['price'] == 999.99]
```

Out[53]:

	asin	description	title	imUrl	salesRank	ca
197364	B009PQIAL6	This beautifully sculpted and gracefully desig...	"Vernet" Black Dual Dryer Chair With...	http://ecx.images-amazon.com/images/I/41ks5sFA...	{'Beauty': 582815}	Acc N

In [54]:
```
display.Image(meta_data_df.loc[197364]["imUrl"])
```

Out[54]:

In [55]:
```
meta_data_df
```

Out[55]:

	asin	description	title	imUrl	sale
0	0205616461	As we age, our once youthful, healthy skin suc...	Bio-Active Anti-Aging Serum (Firming Ultra-Hyd...	http://ecx.images-amazon.com/images/I/41DecrGO...	{'H Pe 4
1	0558925278	Mineral Powder Brush--Apply powder or mineral ...	Eco Friendly Ecotools Quality Natural Bamboo C...	http://ecx.images-amazon.com/images/I/51L%2BzY...	{'B 4
2	0733001998	From the Greek island of Chios, this Mastiha b...	Mastiha Body Lotion	http://ecx.images-amazon.com/images/I/311WK5y1...	{'B 5
3	0737104473	Limited edition Hello Kitty Lipstick featuring...	Hello Kitty Lustre Lipstick (See sellers comme...	http://ecx.images-amazon.com/images/I/31u6Hrzk...	{'B 9
4	0762451459	The mermaid is an elusive (okay, mythical) cre...	Stephanie Johnson Mermaid Round Snap Mirror	http://ecx.images-amazon.com/images/I/41y2%2BF...	
...	...	...	...	...	
259199	B00LP2YB8E	Color: White\nFullness72 inches\nCenter Gather...	2t 2t Edge Crystal Rhinestones Bridal Wedding ...	http://ecx.images-amazon.com/images/I/41E630m-...	
259200	B00LOS7MEE	The secret to long lasting colors, healthy nai...	French Manicure Gel Nail Polish Set - "Se...	http://ecx.images-amazon.com/images/I/41skHL1O...	{'B 1
259201	B00LPVG6V0	ResQ Organics Face & Body Wash - With Aloe Ver...	ResQ Organics Face & Body Wash - Aloe Vera...	http://ecx.images-amazon.com/images/I/31C1w4Ku...	
259202	B00LTDUHJQ	Color: White\n2 Tier \nFullness 72 inches\nSew...	2 Tier Tulle Elbow Wedding Veil with Ribbon Ed...	http://ecx.images-amazon.com/images/I/51%2B%2B...	
259203	B00LU0LTOU	The bags produced by us are 100% ECO friendly ...	*ECOCRAFTWORLD* GENUINE BUFFALO LEATHER TRAVEL...	http://ecx.images-amazon.com/images/I/41kXSEch...	

259204 rows × 9 columns

In [56]:
```
#renaming columns we plan to return to users for improved aesthetics
```

In [57]:
```
meta_data_df.rename(columns={'description':'Description', 'title': 'Product
```

In [58]:
```
meta_data_df.head()
```

Out[58]:

	ASIN	Description	Product Name	Image	salesRank	categories	pri
0	0205616461	As we age, our once youthful, healthy skin suc...	Bio-Active Anti-Aging Serum (Firming Ultra-Hyd...	http://ecx.images-amazon.com/images/I/41DecrGO...	{'Health & Personal Care': 461765}	[[Beauty, Skin Care, Face, Creams & Moisturize...	N:
1	0558925278	Mineral Powder Brush--Apply powder or mineral ...	Eco Friendly Ecotools Quality Natural Bamboo C...	http://ecx.images-amazon.com/images/I/51L%2BzY...	{'Beauty': 402875}	[[Beauty, Tools & Accessories, Makeup Brushes ...	N:
2	0733001998	From the Greek island of Chios, this Mastiha b...	Mastiha Body Lotion	http://ecx.images-amazon.com/images/I/311WK5y1...	{'Beauty': 540255}	[[Beauty, Skin Care, Body, Moisturizers, Lotio...	N:
3	0737104473	Limited edition Hello Kitty Lipstick featuring...	Hello Kitty Lustre Lipstick (See sellers comme...	http://ecx.images-amazon.com/images/I/31u6Hrzk...	{'Beauty': 931125}	[[Beauty, Makeup, Lips, Lipstick]]	N:
4	0762451459	The mermaid is an elusive (okay, mythical) cre...	Stephanie Johnson Mermaid Round Snap Mirror	http://ecx.images-amazon.com/images/I/41y2%2BF...	NaN	[[Beauty, Tools & Accessories, Mirrors, Makeup...	19.

In [59]:
```
meta_data_df.isna().sum()
```

Out[59]:
```
ASIN                  0
Description       24707
Product Name        444
Image                88
salesRank          5188
categories            0
price             69274
related           51350
brand            131038
dtype: int64
```

In [60]:
```
#dropping brand due to large # of nulls
```

In [61]:
```
meta_data_df.drop(columns=['brand'], inplace=True)
```

Methods

We utilized a Normal Predictor model for our initial model, which returned an RMSE of 1.5. We iterated through the following model algorithms to assess which models to further explore: SVD(), SVDpp(), SlopeOne(), NMF(), NormalPredictor(), KNNBaseline(), KNNBasic(), KNNWithMeans(), KNNWithZScore(), BaselineOnly(), and CoClustering(). Our results were based on cross validation and returning the RMSE for each model, along with the fit time and test time. The top 3 models according to Test RMSE were SVDpp, SVD, and Baseline Only. Based on these results, we chose those 3 models to explore further.

We ran multiple grid searches to test hyperparameters for SVDpp and SVD. Our best model based on RMSE was an SVD model with the following paramenters specified: (n_factors=2, n_epochs=20, biased=True).

Setting Up Surprise

In [62]:
```
reader = Reader(rating_scale=(1, 5))
surprise_data = Dataset.load_from_df(surprise_df, reader)

trainset, testset = train_test_split(surprise_data, test_size=0.2, random_s
```

In [63]:
```
surprise_data
```

Out[63]:
```
<surprise.dataset.DatasetAutoFolds at 0x7fd8e984cf10>
```

In [64]:
```
# How many users and items are in the trainset
print('Number of users: ', trainset.n_users, '\n')
print('Number of items: ', trainset.n_items, '\n')
```

```
Number of users:  22359

Number of items:  12101
```

In [65]:
```
print('Type trainset :',type(trainset),'\n')
print('Type testset :',type(testset))
```

```
Type trainset : <class 'surprise.trainset.Trainset'>

Type testset : <class 'list'>
```

Dummy Model

In [66]:
```
baseline = NormalPredictor()
baseline.fit(trainset)
```

Out[66]:
```
<surprise.prediction_algorithms.random_pred.NormalPredictor at 0x7fd95eb0
3c40>
```

In [67]:
```
predictions = baseline.test(testset)
```

In [68]:
```
baseline = accuracy.rmse(predictions)
```

```
RMSE: 1.4993
```

Baseline Models

In [69]:
```
baseline2 = BaselineOnly()
baseline2.fit(trainset)
```

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

Out[69]:
```
<surprise.prediction_algorithms.baseline_only.BaselineOnly at 0x7fd8e9d95
160>
```

In [70]:
```
predictions2 = baseline2.test(testset)
```

In [71]:
```
baseline2 = accuracy.rmse(predictions2)
```

```
RMSE: 1.0890
```

In [72]:
```
#baseline RMSE of 1.089 utilizing BaselineOnly
```

In [73]:
```
als_options = {'method': 'als',
               }
als_baseline = BaselineOnly(bsl_options=als_options)
```

In [74]:
```
als_baseline.fit(trainset)
```

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

Out[74]:
```
<surprise.prediction_algorithms.baseline_only.BaselineOnly at 0x7fd8e687f
070>
```

In [75]:
```
predictions = als_baseline.test(testset)
```

In [76]:
```
als_baseline = accuracy.rmse(predictions)
```

```
RMSE: 1.0890
```

In [77]:
```
sgd_options = {'method': 'sgd',
               }
sgd_baseline = BaselineOnly(bsl_options=sgd_options)
```

In [78]:
```
sgd_baseline.fit(trainset)
```

```
Estimating biases using sgd...
```

Out[78]:
```
<surprise.prediction_algorithms.baseline_only.BaselineOnly at 0x7fd96c692
d30>
```

In [79]:
```
predictions = sgd_baseline.test(testset)
```

In [80]:
```
sgd_baseline = accuracy.rmse(predictions)
```

```
RMSE: 1.0818
```

In [81]:
```
#our baseline model with sgd improved our RMSE to 1.0818
```

Iterating Over All Algorithms to Assess Which Models to Further Explore

In [82]:
```
benchmark = []
# Iterate over all algorithms
for algorithm in [SVD(), SVDpp(), SlopeOne(), NMF(), NormalPredictor(), KNN
    # Perform cross validation
    results = cross_validate(algorithm, surprise_data, measures=['RMSE'], c

    # Get results & append algorithm name
    tmp = pd.DataFrame.from_dict(results).mean(axis=0)
    tmp = tmp.append(pd.Series([str(algorithm).split(' ')[0].split('.')[-1]
    benchmark.append(tmp)

pd.DataFrame(benchmark).set_index('Algorithm').sort_values('test_rmse')
```

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

Out[82]:

Algorithm	test_rmse	fit_time	test_time
SVDpp	1.094490	23.848926	1.397009
SVD	1.094936	8.433316	0.488092
BaselineOnly	1.095902	0.336700	0.191650
KNNBaseline	1.170535	19.163846	2.919733

	test_rmse	fit_time	test_time
Algorithm			
CoClustering	1.203121	4.033110	0.484079
KNNWithMeans	1.213770	18.034438	2.814322
KNNWithZScore	1.216247	20.545019	2.854065
KNNBasic	1.233196	19.139106	2.495180
SlopeOne	1.239357	3.994641	1.064155
NMF	1.296316	8.622149	0.264613
NormalPredictor	1.497461	0.132576	0.299679

In [83]:
```
#given our results, we will further explor SVDpp and SVD
```

SVD Model Exploration

In [84]:
```
#Running an SVD model with defaults on trainset
```

In [85]:
```
svd = SVD(random_state=42)
svd.fit(trainset)
predictions = svd.test(testset)
print(accuracy.rmse(predictions))
```

```
RMSE: 1.0889
1.0889451149217502
```

In [86]:
```
#Checking to see estimated rating for 2 user/product combinations
```

In [87]:
```
svd.predict('A1YJEY40YUW4SE', 'B00LLPT4HI')
```

Out[87]:
```
Prediction(uid='A1YJEY40YUW4SE', iid='B00LLPT4HI', r_ui=None, est=4.41104
541567184, details={'was_impossible': False})
```

In [88]:
```
svd.predict('A2BLFCOPSMBOZ9', '7806397051')
```

Out[88]:
```
Prediction(uid='A2BLFCOPSMBOZ9', iid='7806397051', r_ui=None, est=3.77449
9861592997, details={'was_impossible': False})
```

In [89]:
```
#Cross validate the model
```

In [90]:
```
cv_svd_baseline = cross_validate(svd, surprise_data)
```

In [91]:
```
cv_svd_baseline
```

Out[91]:
```
{'test_rmse': array([1.08772928, 1.09045244, 1.08511667, 1.08844986, 1.09
410617]),
 'test_mae': array([0.83311391, 0.83453597, 0.83295605, 0.83554836, 0.837
51303]),
 'fit_time': (8.709398746490479,
  8.429296970367432,
  8.718079805374146,
  8.752227067947388,
  8.930760145187378),
 'test_time': (0.19809293746948242,
  0.21871113777160645,
  0.2233517169952392 6,
  0.21189093589782715,
  0.2131493091583252)}
```

Attempt on new split

In [92]:
```
#Hold out 10% of data for validation
#Create a new surpise data class
svd_data = Dataset.load_from_df(surprise_df, reader)
raw_ratings_svd = svd_data.raw_ratings
# A = 90% of the data, B = 10% of the data
threshold = int(.9 * len(raw_ratings_svd))
A_raw_ratings_svd = raw_ratings_svd[:threshold]
B_raw_ratings_svd = raw_ratings_svd[threshold:]
```

In [93]:
```
# svd_data is now the set A
svd_data.raw_ratings = A_raw_ratings_svd
```

In [94]:
```
#Create a param grid for grid search
SVD_parm_grid = {'n_factors':[20,50,100,150],'n_epochs':[10,20,30],'biased'
```

In [95]:
```
#Instantiate our grid search & fit to set A
svd_grid_search = GridSearchCV(algo_class=SVD,param_grid=SVD_parm_grid,meas
svd_grid_search.fit(svd_data)
```

In [96]:
```
best_svd_algo = svd_grid_search.best_estimator['rmse']
```

In [97]:
```
svd_grid_search.best_params
```

Out[97]:
```
{'rmse': {'n_factors': 20, 'n_epochs': 20, 'biased': True}}
```

In [98]:
```
#{'rmse': {'n_factors': 20, 'n_epochs': 20, 'biased': True}}
```

In [99]:
```
# retrain on the whole set A
trainset_svd = svd_data.build_full_trainset()
best_svd_algo.fit(trainset_svd)
```

Out[99]:
```
<surprise.prediction_algorithms.matrix_factorization.SVD at 0x7fd98ffebbe
0>
```

In [100]:
```
predictions = best_svd_algo.test(trainset_svd.build_testset())
print('Biased accuracy on A,', end='    ')
accuracy.rmse(predictions)
```

```
Biased accuracy on A,    RMSE: 0.9392
```

Out[100]:
```
0.9391517959433653
```

In [101]:
```
# Compute unbiased accuracy on B
testset_svd = svd_data.construct_testset(B_raw_ratings_svd)  # testset is n
predictions = best_svd_algo.test(testset_svd)
print('Unbiased accuracy on B,', end=' ')
accuracy.rmse(predictions)
```

```
Unbiased accuracy on B, RMSE: 0.9690
```

Out[101]:
```
0.9689750498995585
```

In [102]:
```
svd2 = SVD(n_factors=20, n_epochs=20, biased=True, random_state=42)
svd2.fit(trainset)
predictions = svd2.test(testset)
print(accuracy.rmse(predictions))
```

```
RMSE: 1.0840
1.0840257198509056
```

Attempt new grid search params with lower n_factors

In [103]:
```
SVD_parm_grid = {'n_factors':[2,5,10,20],'n_epochs':[10,20,30],'biased':[Tr
```

In [104]:
```
#Instantiate our grid search & fit to set A
svd_grid_search = GridSearchCV(algo_class=SVD,param_grid=SVD_parm_grid,meas
svd_grid_search.fit(svd_data)
```

In [105]:
```
best_svd_algo = svd_grid_search.best_estimator['rmse']
```

In [106]:
```
svd_grid_search.best_params
```

Out[106]:
```
{'rmse': {'n_factors': 2, 'n_epochs': 20, 'biased': True}}
```

In [107]:
```
#{'rmse': {'n_factors': 2, 'n_epochs': 20, 'biased': True}}
```

In [108]:
```
# retrain on the whole set A
trainset_svd = svd_data.build_full_trainset()
best_svd_algo.fit(trainset_svd)
```

Out[108]:
```
<surprise.prediction_algorithms.matrix_factorization.SVD at 0x7fd97a92197
0>
```

In [109]:
```
predictions = best_svd_algo.test(trainset_svd.build_testset())
print('Biased accuracy on A,', end='    ')
accuracy.rmse(predictions)
```

```
Biased accuracy on A,    RMSE: 0.9827
```

Out[109]:
```
0.9827370959416434
```

In [110]:
```
# Compute unbiased accuracy on B
testset_svd = svd_data.construct_testset(B_raw_ratings_svd)  # testset is n
predictions = best_svd_algo.test(testset_svd)
print('Unbiased accuracy on B,', end=' ')
accuracy.rmse(predictions)
```

```
Unbiased accuracy on B, RMSE: 0.9688
```

Out[110]:
```
0.9687683063536121
```

In [111]:
```
svd3 = SVD(n_factors=2, n_epochs=20, biased=True, random_state=23)
svd3.fit(trainset)
predictions = svd3.test(testset)
print(accuracy.rmse(predictions))
```

```
RMSE: 1.0820
1.08197848557543
```

In [112]:
```
#Same RMSE as sgd_baseline
```

In [113]:
```
cv_svd3 = cross_validate(svd3, surprise_data)
```

In [114]:
```
cv_svd3
```

Out[114]:
```
{'test_rmse': array([1.09054859, 1.07979603, 1.08214753, 1.08071209, 1.08
448901]),
 'test_mae': array([0.83289907, 0.82617841, 0.82841238, 0.82549442, 0.829
95314]),
 'fit_time': (1.9939210414886475,
  1.9018058776855469,
  2.0698957443237305,
  2.0763142108917236,
  1.8815948963165283),
 'test_time': (0.18584513664245605,
  0.18175983428955078,
  0.1879570484161377,
  0.2019960880279541,
  0.16666698455810547)}
```

Attempt new grid search params with different lr_all values

In [115]:
```
#Param grid for SVD grid search
SVD_parm_grid = {'n_factors':[2,5,10,20],'n_epochs':[10,20,30],'biased':[Tr
#Best params
{'rmse': {'n_factors': 2, 'n_epochs': 20, 'biased': True, 'lr_all': 0.005}}
```

Out[115]:
```
{'rmse': {'n_factors': 2, 'n_epochs': 20, 'biased': True, 'lr_all': 0.00
5}}
```

In [116]:
```
#Best params
#{'rmse': {'n_factors': 2, 'n_epochs': 20, 'biased': True, 'lr_all': 0.005}
```

In [117]:
```
svd4 = SVD(n_factors=2, n_epochs=20, biased=True, lr_all=0.005, random_stat
svd4.fit(trainset)
predictions = svd4.test(testset)
print(accuracy.rmse(predictions))
```

```
RMSE: 1.0820
1.08197848557543
```

SVPpp Model Exploration

In [118]:
```
#Running an SVDpp model with defaults on train
```

In [119]:
```
svdpp = SVDpp(random_state=23)
svdpp.fit(trainset)
predictions = svdpp.test(testset)
print(accuracy.rmse(predictions))
```

```
RMSE: 1.0880
1.0879628428315302
```

In [120]:
```
cv_svdpp_baseline = cross_validate(svdpp, surprise_data)
```

In [121]:
```
cv_svdpp_baseline
```

Out[121]:
```
{'test_rmse': array([1.08945456, 1.09246937, 1.08849169, 1.09090713, 1.08
916246]),
 'test_mae': array([0.82466892, 0.82662864, 0.82357916, 0.82629973, 0.824
07994]),
 'fit_time': (30.680674076080322,
  29.895809173583984,
  29.96164894104004,
  30.057493925094604,
  29.788533926010132),
 'test_time': (0.8825788497924805,
  0.877150297164917,
  0.859544038772583,
  0.885124921798706,
  0.8451321125030518)}
```

In [122]:
```
# grid search for SVD++
svdpp_param_grid = {'n_factors':[10, 20],
                    'n_epochs':[20, 30],
                    'reg_all':[0.02, 0.05],
                    "lr_all": [0.007, 0.005]}
#svdpp_gs_model = GridSearchCV(SVDpp, param_grid=svdpp_param_grid, cv=3, jo

# Fit and return the best_params based on cross validation this will take a
#svdpp_gs_model.fit(surprise_data)
#svdpp_gs_model.best_params['rmse']
```

In [123]:
```
#{'n_factors': 10, 'n_epochs': 20, 'reg_all': 0.05, 'lr_all': 0.005}
```

In [124]:
```
# Instantiate - fit on trainset - score the model on testset
#SVDpp_model = SVDpp(n_factors=10, n_epochs=20, random_state=42, reg_all=0.
#SVDpp_model.fit(trainset)
#predictions = SVDpp_model.test(testset)
#SVDpp_gs = accuracy.rmse(predictions)
```

In [125]:
```
#RMSE: 1.0823
```

In [126]:
```
# New dictionary for SVD++
svdpp_param_grid = {'n_factors':[15, 20, 25],
                    'n_epochs':[10, 20 ],
                    'reg_all':[0.02, 0.05, .07],
                    "lr_all": [0.007, 0.005, .002]}
#svdpp_gs_model = GridSearchCV(SVDpp, param_grid=svdpp_param_grid, cv=3, jo

# Fit and return the best_params based on cross validation this will take a
#svdpp_gs_model.fit(surprise_data)
#svdpp_gs_model.best_params['rmse']
```

In [127]:
```
#{'n_factors': 15, 'n_epochs': 20, 'reg_all': 0.07, 'lr_all': 0.005}
```

In [128]:
```
# Instantiate - fit on trainset - score the model on testset
#SVDpp_model = SVDpp(n_factors=15, n_epochs=20, random_state=42, reg_all=0.
#SVDpp_model.fit(trainset)
#predictions = SVDpp_model.test(testset)
#SVDpp_gs = accuracy.rmse(predictions)
```

In [129]:
```
#RMSE: 1.0824
#still not as good as svd and very large fit time; will move forward with s
```

Final Collaborative Filtering Models

Our final model allows us to input the unique reviewerID and number of recommendations we would like the model to return. The model then returns the requested number of items, including the ASIN, Product Name, Description, Image, and predicted_rating. Recommended products are ordered from the highest predicted_rating to the lowest.

In [130]:
```
# Building our trainset_full to fit our final model on full trainset
```

In [131]:
```
trainset_full = surprise_data.build_full_trainset()
```

In [132]:
```
trainset_full
```

Out[132]:
```
<surprise.trainset.Trainset at 0x7fd8e984c730>
```

In [133]:
```
best_model = SVD(n_factors=2, n_epochs=20, biased=True, random_state=23)
best_model.fit(trainset_full)
```

Out[133]:
```
<surprise.prediction_algorithms.matrix_factorization.SVD at 0x7fd9137a3a6
0>
```

In [134]:
```
## Subset data frame to show reviewers the products they have rated
df_prior_ratings = pd.DataFrame(df.set_index("reviewerID"))
df_prior_ratings.drop(columns= ["reviewerName", "helpful", "reviewText", "o
df_prior_ratings.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 198502 entries, A1YJEY40YUW4SE to A3UJRNI8UR4871
Data columns (total 1 columns):
 #   Column  Non-Null Count   Dtype 
---  ------  --------------   ----- 
 0   asin    198502 non-null  object
dtypes: object(1)
memory usage: 3.0+ MB
```

In [135]:
```
pd.set_option('display.max_colwidth', None)
```

We utilize the same user demonstration purposes in the difference between general recommended products and products recommended by category:

- reviewerID: AYRYR6UGT2HAG

In [136]:
```
def buyer_recommended_products():
    pd.set_option('display.max_colwidth', None)
    buyer = input("reviewerID: ")
    n_recs = int(input("How many recommendations? "))

    already_reviewed = list(df_prior_ratings.loc[buyer, "asin"])
    not_reviewed = meta_data_df.copy()
    not_reviewed = not_reviewed[not_reviewed.ASIN.isin(already_reviewed) ==
    not_reviewed.reset_index(inplace=True)
    not_reviewed["predicted_rating"] = not_reviewed["ASIN"].apply(lambda x:
    not_reviewed.sort_values(by="predicted_rating", ascending=False, inplac
    not_reviewed = not_reviewed[['ASIN','Product Name', 'Description', 'Ima

 #Converting links to html tags
    def path_to_image_html(path):
        return '<img src="'+ path + '" width="60" >'


    return HTML(not_reviewed.to_html(escape=False, formatters=dict(Image=pa
```

In [137]:
```
buyer_recommended_products()
```

```
reviewerID: AYRYR6UGT2HAG
How many recommendations? 5
```

Creating a Recommendation System with an option to add Category of Product

Our additional final model allows us to input the unique reviewerID, the number of recommendations we would like the model to return, and the category of product we would like our recommended products to be. The model then returns the requested number of items, including the ASIN, Product Name, Description, and Image. This will be especially helpful when trying to promote certain items at certain times of year, like Fragrances around Valentine's Day or Skin Care products in the winter time.

In [138]:
```
meta_data_df.categories #which level do we want to go to?
```

Out[138]:
```
0                                       [[Beauty, Skin Care, Face, Creams &
Moisturizers]]
1          [[Beauty, Tools & Accessories, Makeup Brushes & Tools, Brushes
& Applicators]]
2                                       [[Beauty, Skin Care, Body, Moisturi
zers, Lotions]]
3                                                        [[Beauty, Makeup, L
ips, Lipstick]]
4                                 [[Beauty, Tools & Accessories, Mirrors, M
akeup Mirrors]]
                                                            ...
259199       [[Beauty, Hair Care, Styling Tools, Styling Accessories, Dec
orative Combs]]
259200                                               [[Beauty, Makeup, Nail
s, Nail Polish]]
259201                                  [[Beauty, Skin Care, Face, Creams &
Moisturizers]]
259202       [[Beauty, Hair Care, Styling Tools, Styling Accessories, Dec
orative Combs]]
259203                            [[Beauty, Tools & Accessories, Bags & Cases,
Toiletry Bags]]
Name: categories, Length: 259204, dtype: object
```

In [139]:
```
list(meta_data_df.categories)[:][643][0][1]
```

Out[139]:
```
'Makeup'
```

In [140]:
```
#return unique subcategories from meta deta to give user input options for
```

In [141]:
```
subcategories = []
for row in meta_data_df["categories"]:
    value = row[0][1]
    if value not in subcategories:
        subcategories.append(value)
subcategories
```

Out[141]:
```
['Skin Care',
 'Tools & Accessories',
 'Makeup',
 'Hair Care',
 'Bath & Body',
 'Fragrance',
 'Fan Shop',
 'Snow Sports',
 'Kitchen & Dining',
 'Health Care',
 'Stationery & Party Supplies',
 'Storage & Organization',
 'Baby & Child Care',
 'Personal Care',
 'Household Supplies',
 'Accessories',
 'Hardware']
```

We see subcategories beyond the Beauty category; we will focus on the Beauty subcategories of 'Skin Care', 'Tools and Accessories', 'Makeup', 'Hair Care', 'Bath & Body', and 'Fragrance'.

In [142]:
```
#create a function to extract subcategory level 1 from categories
def get_subcategory(cat):
    value = cat[0][1]
    return(value)
```

In [143]:
```
#Create a new column in our meta data df called "sub_cat" containing sub ca
meta_data_df["sub_cat"] = meta_data_df["categories"].apply(get_subcategory)
```

In [144]:

```
meta_data_df.head()
```

Out[144]:

	ASIN	Description	Product Name	Image
0	0205616461	As we age, our once youthful, healthy skin succumbs to an enzymatic imbalance that wears away the cellular network, resulting in skin thinning and aging. Combining the best of nature and cosmetic biotechnology, Bio-Active products are formulated with Enzymes that gently exfoliate the skin and stimulate regeneration for a youthful glow. Benefiting from fertile orchards in the Italian countryside, Bio-active formulas are rich in phytohormones, flavonoids and fatty acids from active extracts in Apple and Pear Seeds ,enzymatically modified and developed especially for the care of aging skin. This repairing fluid helps to nourish and firm by accelerating penetration and delivery of active principles to the skin, giving it a more youthful appearance. \n\nAdvanced "Probiotic" Complex from nourishing milk proteins regains the skin's natural equilibrium, boosts its immunities and protects it against environmental and biological stress.\n\nPeptides and Ceramides help to firm and	Bio-Active Anti-Aging Serum (Firming Ultra-Hydrating Serum)	http://ecx.images-amazon.com/images/I/41DecrGODDL._SY300_.jpg

ASIN	Description	Product Name	Image
	regenerate the skin by stimulating collagen production and strengthening the epidermis.\n\nA Calming Botanical Complex of Hyaluronic Acid and Wheat Germ Extract hydrates and restores the skin's protective barriers.\n\nA nutritive Vitamin Complex moisturizes and protects the skin from damaging environmental factors. \n\nParacress Extract, a natural alternative to cosmetic injections, limits and relaxes micro-contractions that create facial lines, producing immediate and long-term smoothing of the skin.\n\nTo Use: Apply a few pumps to Apply a few pumps to a clean and dried face, neck and dcollet.		

	ASIN	Description	Product Name	Image
1	0558925278	Mineral Powder Brush--Apply powder or mineral foundation all over face in a circular, buffing motion and work inward towards nose.\n\nConcealer Brush--Use with liquid or mineral powder concealer for more coverage on blemishes and under eyes. \n\nEye Shading Brush-- Expertly cut to apply and blend powder eye shadows.\n\nBaby Kabuki-- Buff powder over areas that need more coverage. \n\nCosmetic Brush Bag-- 55% hemp linen, 45% cotton	Eco Friendly Ecotools Quality Natural Bamboo Cosmetic Mineral Brush Set Kit of 4 Soft Brushes and 1 Pouch Baby Kabuki Eye Shading Brush Mineral Powder Brush Concealer Brush(travle Size)	http://ecx.images-amazon.com/images/I/51L%2BzYCQWSL._SX300_.jpg
2	0733001998	From the Greek island of Chios, this Mastiha body lotion is made from Mastic oil, a pure product derived from mastic. With organically grown: Olive oil, red grape leaves, Aloe vera, Rosemary, Bee's wax, Shea Butter.	Mastiha Body Lotion	http://ecx.images-amazon.com/images/I/311WK5y1dML._SY300_.jpg
3	0737104473	Limited edition Hello Kitty Lipstick featuring shiny black casing with Hello Kitty figure on a pop art pattern background. Cap features the logos of both MAC and Hello Kitty in this collection.	Hello Kitty Lustre Lipstick (See sellers comments for colors)	http://ecx.images-amazon.com/images/I/31u6Hrzk3WL._SY300_.jpg

	ASIN	Description	Product Name	Image
4	0762451459	The mermaid is an elusive (okay, mythical) creature, her beauty glimpsed only fleetingly by swimmers and sailors. With this mermaid-inspired iridescent compact, we landlubbers can capture a bit of that shimmering allure. Plus, having a mirror always on hand, we'll have no trouble catching a glimpse of our own gorgeousness (or doing a quick touchup when we need to).Within its slim silhouette, this colorful compact contains a double-sided mirror: Each mirror is large enough to show your entire face, and angling the two offers a side-angle view for hair fixes. The case snaps shut to keep the mirrors scuff-free.	Stephanie Johnson Mermaid Round Snap Mirror	http://ecx.images-amazon.com/images/I/41y2%2BFUdf1L._SY300_.jpg

In [145]:
```
def buyer_recommended_category_products():
    pd.set_option('display.max_colwidth', None)
    buyer = input("reviewerID: ")
    n_recs = int(input("How many recommendations? "))
    #request_category from subcategories
    request_category = input("Which category of beauty to recommend buyer?

    already_reviewed = list(df_prior_ratings.loc[buyer, "asin"])
    not_reviewed = meta_data_df.copy()
    not_reviewed = not_reviewed[not_reviewed.ASIN.isin(already_reviewed) ==
    not_reviewed.reset_index(inplace=True)
    not_reviewed["predicted_rating"] = not_reviewed["ASIN"].apply(lambda x:
    not_reviewed = not_reviewed[not_reviewed["sub_cat"]==request_category]
    not_reviewed.sort_values(by="predicted_rating", ascending=False, inplac
    not_reviewed = not_reviewed[['ASIN','Product Name', 'Description', 'Ima

    #Converting links to html tags
    def path_to_image_html(path):
        return '<img src="'+ path + '" width="60" >'


    return HTML(not_reviewed.to_html(escape=False, formatters=dict(Image=pa
```

In [146]:
```
buyer_recommended_category_products()
```

```
reviewerID: AYRYR6UGT2HAG
How many recommendations? 5
Which category of beauty to recommend buyer? Fragrance
```

Out[146]:

	ASIN	Product Name	Description	Image	predicted_rating
13421	B000C1VTOW	Curious by Britney Spears for Women, Eau De Parfum Spray, 1.7 Ounce	Introduced in 2004. Recommended use: casual.When applying any fragrance please consider that there are several factors which can affect the natural smell of your skin and, in turn, the way a scent smells on you. For instance, your mood, stress level, age, body chemistry, diet, and current medications may all alter the scents you wear. Similarly, factor such as dry or oily skin can even affect the amount of time a fragrance will last after being applied		4.279954
20884	B000HJT1CW	The Body Shop Vitamin E Face Mist, 3.3-Fluid Ounce	Vitamin E Face Mist is a quick skin pick-me up and excellent for setting make-up. Spritz it on for instant refreshment, moisture, and protection. With a delicate rosewater scent. Spray the moisturizing mist onto your face once or twice and allow it to dry. It is excellent for setting make–up; just spray over make–up to seal it after applying. Carry with you when traveling, especially when flying to protect your skin from dry air in the cabin and climate changes, so you’ll have healthy–looking, hydrated skin when you arrive. Try keeping face mist in the refrigerator during the summer for extra coolness.		4.267174
19288	B000GHYSVE	Fantasy by Britney Spears for Women - 1.7 Ounce EDP Spray	Launched by the design house of Britney Spears.When applying any fragrance please consider that there are several factors which can affect the natural smell of your skin and, in turn, the way a scent smells on you. For instance, your mood, stress level, age, body chemistry, diet, and current medications may all alter the scents you wear. Similarly, factor such as dry or oily skin can even affect the amount of time a fragrance will last after being applied		4.251050
10348	B0009OAHVO	L'eau D'issey (issey Miyake) by Issey Miyake for Men - EDT Spray	Introduced in 1994. Fragrance notes: citrus and spice combined with lower notes of musk, amber and woods. Recommended use: evening.When applying any fragrance please consider that there are several factors which can affect the natural smell of your skin and, in turn, the way a scent smells on you. For instance, your mood, stress level, age, body chemistry, diet, and current medications may all alter the scents you wear. Similarly, factor such as dry or oily skin can even affect the amount of time a fragrance will last after being applied		4.245603

	ASIN	Product Name	Description	Image	predicted_rating
19211	B000GHWSG6	Euphoria by Calvin Klein for Women, Eau De Parfum Spray, 3.4 Ounce	Launched by the design house of Calvin Klein.When applying any fragrance please consider that there are several factors which can affect the natural smell of your skin and, in turn, the way a scent smells on you. For instance, your mood, stress level, age, body chemistry, diet, and current medications may all alter the scents you wear. Similarly, factor such as dry or oily skin can even affect the amount of time a fragrance will last after being applied		4.235487

In [147]:
```
#pulling out images for our recommendation e-mails
```

In [148]:
```
color_club = meta_data_df.loc[meta_data_df['ASIN'] == "B00A1Y177A"]
```

In [149]:
```
color_club
```

Out[149]:

	ASIN	Description	Product Name	Image	sale
200571	B00A1Y177A	A linear holographic nail polish that will take you to cloud nine. A halographic nail polish that will bring a touch of heaven to everything you do. Favorite nail polish for	Color Club Halographic Hues Nail Polish, Light Green, Cloud Nine, .05 Ounce	http://ecx.images-amazon.com/images/I/51J8BWjszoL._SX300_.jpg	{'E 1

In [150]:
```python
display.Image(color_club.loc[200571]["Image"])
```

Out[150]:

In [151]:
```python
bed_head = meta_data_df.loc[meta_data_df['ASIN'] == "B001EWF2SI"]
bed_head
```

Out[151]:

	ASIN	Description	Product Name	Image	sales
60398	B001EWF2SI	TIGI BedHead After the Party Smoothing Cream 3.4 Ounces	TIGI Bed Head After the Party Smoothing Cream, 3.4 Ounce(pack of 2)	http://ecx.images-amazon.com/images/I/31oiW3t2VBL._SY300_.jpg	{'Be 2

```
In [152]: display.Image(bed_head.loc[60398]["Image"])
```

Out[152]:


```
In [153]: curious = meta_data_df.loc[meta_data_df['ASIN'] == "B000C1VT0W"]
          curious
```

Out[153]:

	ASIN	Description	Product Name	
13422	B000C1VT0W	Introduced in 2004. Recommended use: casual.When applying any fragrance please consider that there are several factors which can affect the natural smell of your skin and, in turn, the way a scent smells on you. For instance, your mood, stress level, age, body chemistry, diet, and current medications may all alter the scents you wear. Similarly, factor such as dry or oily skin can even affect the amount of time a fragrance will last after being applied	Curious by Britney Spears for Women, Eau De Parfum Spray, 1.7 Ounce	h amazon.com/images/I/412S0EUS

In [154]:
```
display.Image(curious.loc[13422]["Image"])
```

Out[154]:

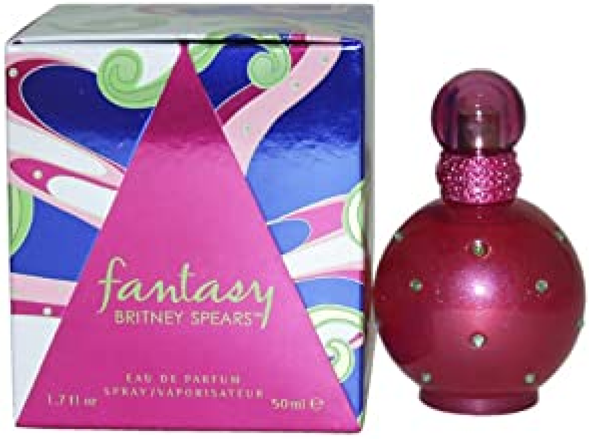
In [155]:
```
fantasy = meta_data_df.loc[meta_data_df['ASIN'] == "B000GHYSVE"]
fantasy
```

Out[155]:

	ASIN	Description	Product Name	
19289	B000GHYSVE	Launched by the design house of Britney Spears.When applying any fragrance please consider that there are several factors which can affect the natural smell of your skin and, in turn, the way a scent smells on you. For instance, your mood, stress level, age, body chemistry, diet, and current medications may all alter the scents you wear. Similarly, factor such as dry or oily skin can even affect the amount of time a fragrance will last after being applied	Fantasy by Britney Spears for Women - 1.7 Ounce EDP Spray	amazon.com/images/I/51MiFc%

```
In [156]: display.Image(fantasy.loc[19289]["Image"])
```

Out[156]:



Evaluation of Model

The following is code to take a random user, finds a product they should like based on what Amazon recommended to them in the bought together recommendations for their highest rated product. Then it uses our model to predict the user's rating for that product.

```
In [157]: #Join our review and meta data
          joined_df = surprise_df.join(meta_data_df.set_index("ASIN"),on="asin",how="
```

```
In [158]: #Confirming join worked
          len(joined_df) == len(surprise_df)
```

Out[158]: True

```
In [159]: #Select a random reviewer
          user_ID = "AYRYR6UGT2HAG"
```

In [160]:
```
#Inspect random user's reviews
user_ID_DF = joined_df[joined_df["reviewerID"] == user_ID]
user_ID_DF
```

In [161]:
```
#checking the items the users have reviewed; looking at asin with 5 review:
#in the related column, we see products that are purchased with Cure Natura
#pulling out the asin of the related product--we assume user should also li
#will check our system's predicted rating
```

In [162]:
```
#Find Amazon's bought together recommendation ASIN for user's top rated ite
bt_asin = user_ID_DF[user_ID_DF["overall"] == (user_ID_DF["overall"].max())
#We expect that this ASIN should recieve a high predicted rating from our s
from IPython import display
print(joined_df[joined_df["asin"]==bt_asin]["Description"].values[0])
display.Image(joined_df[joined_df["asin"]==bt_asin]["Image"].values[0])
```

```
Biore SARASARA UV Aqua Rich Waterly Essence Sunscreen 50g SPF50+ PA+++ fo
r Face and Body.SPF50+ PA+++ sunscreen for face and body.It is Water-base
Wash off with regular cleanser.Fresh fruit aroma
```

Out[162]:

In [163]:
```
#User our model to predict a rating given this product and user combination
best_model.predict(uid=user_ID,iid=bt_asin)
```

Out[163]:
```
Prediction(uid='AYRYR6UGT2HAG', iid='B004LRO7DO', r_ui=None, est=4.261800
862764359, details={'was_impossible': False})
```

In [164]:
```
#estimated rating of 4.26 for this product for this user
```

Final Model Evaluation

The final recommendation model using SVD yielded a RMSE of 1.0820 meaning that, on average, our predicted review scores for Amazon buyers were 1.0820 points off of the true value of review scores. This score is more than half a point drop from our baseline model. On a review scale of 1-5, we believe that is a significant improvement.

The model also has other features that greatly improves the personalization of a post-purchase marketing email:

- No repeat products. The model will not recommend items that the buyer has already purchased. This helps improve product discoverability.

- Prioritizes the best match for the buyer. Whether the model is recommended through the catalog of products, or through a sub category, it will always deliver N number of the top predicted reviews for a buyer.
- Subcategory filtering. The model allows for filtering beauty products based on six subcategories, allowing a more refined search.
- Image retrieval. The model converts the URL to an image and delivers this image alongside recommended titles and descriptions.

Limitations and Next Steps

While our model optimizes for minimizing the RMSE of predicted reviews, it has its limitations. First, our dataset was skewed towards higher ratings as nearly 60% of all reviews were rated 5 points. This in turn skews our predicted values higher. While this may not matter when grabbing the highest rated products, it certainly would affect our lower rated items. We suggest looking into whether these high ratings are a product of the dataset we used, or are consistent with Amazon buyer behavior.

Secondly, our model does not handle indiscriminate reviewers - or reviewers who rate all products the same. This means that our model does not capture their preferences well. We suggest a separate survey be sent to these buyers post-purchase in an effort to determine their preferences. We could then find a way to incorporate these preferences in a future model.

Third, our model does not address the cold start problem. Our model needs prior reviews from users in order to offer recommendations. Next steps would be to add a content-based approad to address this problem.

Finally, we noticed that the dataset often miscategorizes products in their subcategories (haircare, skincare, etc.), which can lead our subcategory predictor to recommend misclassified products. We recommend implementing a standardized classification of subcategories when new products are added to the marketplace.

Conclusion

The Amazon marketing team can implement our recommendation tools quickly and with ease in order to offer more individualized recommendations for users. This will increase user engagement and user purchases. Our model can also be used to market certain types of products that may be popular seasonally, such as sending out individualized Skin Care recommendations in the winter time/dry season, or Fragrance recommendations around gift-giving occassions such as Valentine's Day.