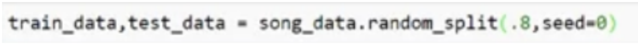
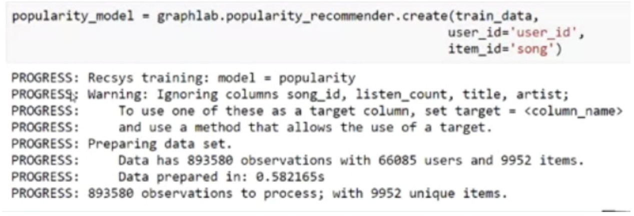
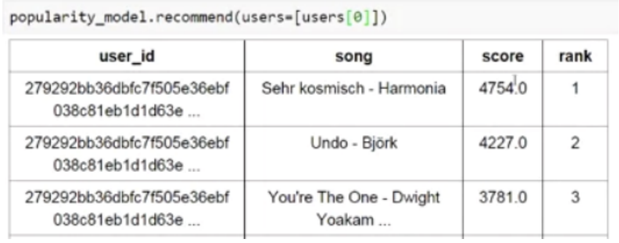
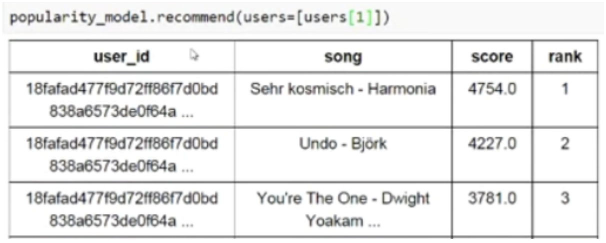
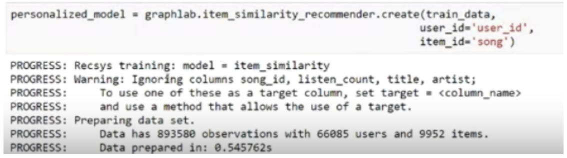
<table>
<tr><td colspan="2">Recommending Songs</td></tr>
<tr><td>Loading and Exploring Song Data</td><td>1. Import <i>.sframe(‘song’)</i> to load the data
2. <i>song_data.head()</i> to view the data
3. <i>song_data[‘song’].show()</i> to visualize data
4. <i>len(song_data)</i> to see how much data
5. <i>song_data[‘user_d’].unique()</i> to count number of users</td></tr>
<tr><td>Creating and Evaluating a Simple Popularity Recommender</td><td>1. Create train_data and test_data with random_split()

<pre>train_data,test_data = song_data.random_split(.8,seed=0)</pre>
2. Create popularity_model using popularity_recommender

<pre>popularity_model = graphlab.popularity_recommender.create(train_data,
                                                          user_id='user_id',
                                                          item_id='song')

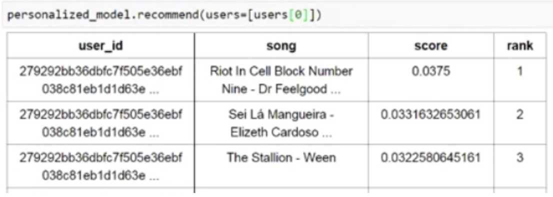
PROGRESS: Recsys training: model = popularity
PROGRESS: Warning: Ignoring columns song_id, listen_count, title, artist;
PROGRESS: To use one of these as a target column, set target = <column_name>
PROGRESS: and use a method that allows the use of a target.
PROGRESS: Preparing data set.
PROGRESS: Data has 893580 observations with 66085 users and 9952 items.
PROGRESS: Data prepared in: 0.582165s
PROGRESS: 893580 observations to process; with 9952 unique items.</pre>
3. Use popularity model to make some predictions

<pre>popularity_model.recommend(users=[users[0]])</pre>
<table>
<tr><th>user_id</th><th>song</th><th>score</th><th>rank</th></tr>
<tr><td>279292bb36dbfc7f505e36ebf 038c81eb1d1d63e ...</td><td>Sehr kosmisch - Harmonia</td><td>4754.0</td><td>1</td></tr>
<tr><td>279292bb36dbfc7f505e36ebf 038c81eb1d1d63e ...</td><td>Undo - Björk</td><td>4227.0</td><td>2</td></tr>
<tr><td>279292bb36dbfc7f505e36ebf 038c81eb1d1d63e ...</td><td>You're The One - Dwight Yoakam ...</td><td>3781.0</td><td>3</td></tr>
</table>

<pre>popularity_model.recommend(users=[users[1]])</pre>
<table>
<tr><th>user_id</th><th>song</th><th>score</th><th>rank</th></tr>
<tr><td>18fafad477f9d72ff86f7d0bd 838a6573de0f64a ...</td><td>Sehr kosmisch - Harmonia</td><td>4754.0</td><td>1</td></tr>
<tr><td>18fafad477f9d72ff86f7d0bd 838a6573de0f64a ...</td><td>Undo - Björk</td><td>4227.0</td><td>2</td></tr>
<tr><td>18fafad477f9d72ff86f7d0bd 838a6573de0f64a ...</td><td>You're The One - Dwight Yoakam ...</td><td>3781.0</td><td>3</td></tr>
</table>
Notice that for both users, they are recommended the same songs since it’s based on global popularity</td></tr>
<tr><td>Creating and Evaluating a Personalized Song Recommender</td><td>1. Create popularity_model using item_ similarity_recommen</td></tr>
</table>



```
personalized_model = graphlab.item_similarity_recommender.create(train_data,
                                                                 user_id='user_id',
                                                                 item_id='song')
```

```
PROGRESS: Recsys training: model = item_similarity
PROGRESS: Warning: Ignoring columns song_id, listen_count, title, artist;
PROGRESS:     To use one of these as a target column, set target = <column_name>
PROGRESS:     and use a method that allows the use of a target.
PROGRESS: Preparing data set.
PROGRESS:     Data has 893580 observations with 66085 users and 9952 items.
PROGRESS:     Data prepared in: 0.545762s
```

2. Use the model to make some predictions

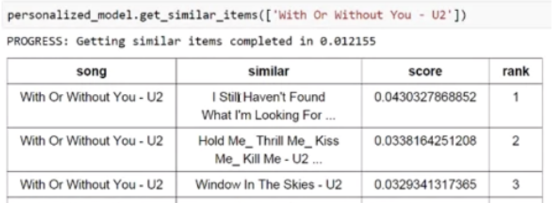


```
personalized_model.recommend(users=[users[0]])
```

user_id	song	score	rank
279292bb36dbfc7f505e36ebf 038c81eb1d1d63e ...	Riot In Cell Block Number Nine - Dr Feelgood ...	0.0375	1
279292bb36dbfc7f505e36ebf 038c81eb1d1d63e ...	Sei Lá Mangueira - Elizeth Cardoso ...	0.0331632653061	2
279292bb36dbfc7f505e36ebf 038c81eb1d1d63e ...	The Stallion - Ween	0.0322580645161	3

Notice that the recommended songs are different than before.

3. Use the model to be recommended similar songs using get_similar_items.()



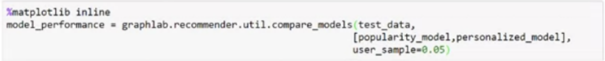
```
personalized_model.get_similar_items(['With Or Without You - U2'])
```

```
PROGRESS: Getting similar items completed in 0.012155
```

song	similar	score	rank
With Or Without You - U2	I Still Haven't Found What I'm Looking For ...	0.0430327868852	1
With Or Without You - U2	Hold Me_ Thrill Me_ Kiss Me_ Kill Me - U2 ...	0.0338164251208	2
With Or Without You - U2	Window In The Skies - U2	0.0329341317365	3

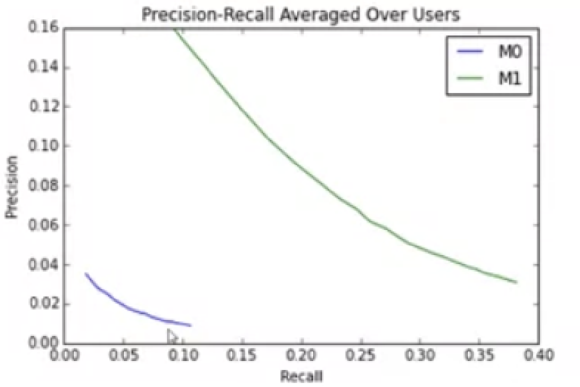
Using Precision-Recall to Compare Recommender Models

1. Use matplotlib



```
%matplotlib inline
model_performance = graphlab.recommender.util.compare_models(test_data,
                                                             [popularity_model,personalized_model],
                                                             user_sample=0.05)
```

2. Output - personalized M) is significantly better than simple model M1


Precision-Recall Averaged Over Users
M0
M1
Precision
Recall