Project Solution Lab 2:

Problem: Recommend Movies or Shows to Users

Modified from:

- Implementing a Recommender System with SageMaker, MXNet, and Gluon (https://github.com/awslabs/amazon-sagemaker-examples/blob/master/introduction_to_applying_machine_learning/gluon_recommender_system/gluon_recom
- An Introduction to Factorization Machines with MNIST (https://github.com/awslabs/amazon-sagemaker-examples/blob/master/introduction_to_amazon_algorithms/factorization_machines_mnist/factorization_machir
- Extending Amazon SageMaker Factorization Machines Algorithm to Predict Top X Recommendations (https://aws.amazon.com/blogs/machine-learning/extending-amazon-sagemaker-factorization-machines-algorithm-to-predict-top-x-recommendations/)

Introduction to business scenario

You work for a startup that focuses on delivering on-demand video streaming services to users. The company wants to introduce movie/show recommendations for their users based on their viewing history.

You are tasked with solving part of this problem by leveraging machine learning to create a recommendation engine to be used on the user website. You are given access to the dataset of historical user preferences and the movies they watched. You can use this to train a machine learning model to recommend movies/shows to watch.

About this dataset

The Amazon Customer Reviews Dataset is a collection of reviews on different products from the Amazon.com marketplace from 1995 until 2015. Customer reviews are one of the most important data types at Amazon. Collecting and showing reviews has been part of the Amazon culture since the beginning of the company and is arguably one important source of innovation. For more details on this dataset, see Amazon Customer Reviews Dataset (https://s3.amazonaws.com/amazon-reviews-pds/readme.html).

This exercise focuses on reviews of videos. The videos dataset contains 1- to 5-star ratings from over 2M Amazon customers on 160K digital videos.

Features

Data columns

- `marketplace` : Two-letter country code (in this case, all "US")
- `customer_id` : Random identifier that can be used to aggregate reviews written by a single author
- `review_id` : Unique ID for the review
- `product_id` : Amazon Standard Identification Number (ASIN). http://www.amazon.com/dp/ (http://www.amazon.com/dp/)<ASIN> links to the product's detail page.
- `product_parent` : The parent of that ASIN. Multiple ASINs (color or format variations of the same product) can roll up into a single parent.
- `product_title` : Title description of the product
- `product_category` : Broad product category that can be used to group reviews (in this case, digital videos)
- `star_rating` : Product's rating (1 to 5 stars)
- `helpful_votes` : Number of helpful votes for the review
- `total_votes` : Number of total votes the review received
- `vine` : Was the review written as part of the Vine program?
- `verified_purchase` : Was the review from a verified purchase?
- `review_headline` : Title of the review itself
- `review_body` : Text of the review
- `review_date` : Date the review was written

Data format

- Tab `\t` separated text file, without quote or escape characters
- First line in each file is header; 1 line corresponds to 1 record

Dataset attributions

Website: https://s3.amazonaws.com/amazon-reviews-pds/readme.html (https://s3.amazonaws.com/amazon-reviews-pds/readme.html)

This dataset is being provided to you by permission of Amazon and is subject to the terms of the AWS Digital Training Service Agreement (available at https://aws.amazon.com/training/digital-training-agreement (https://aws.amazon.com/training/digital-training-agreement)). You are expressly prohibited from copying, modifying, selling, exporting, or using this dataset in any way other than for the purpose of completing this lab.

Brainstorming and designing a question...

...That you can answer with machine learning.

The first step in most projects is to think about the question you want to ask, how the data available supports this question, and which tool (in this case, machine learning model) you are going to use to answer the question. This is an important step because it helps narrow the scope of exploration and gives clarity on the features that you are going to use.

Take a moment to write your thoughts regarding the dataset in the cell below. What are the things you can predict with machine learning? Why may that be relevant from a business/client perspective? Explain why you consider these thoughts important.

In [1]:
```
# Write your thoughts here
```

There might be several ideas about what to do with the data, but for now we are all going to work on recommending a video to a particular user.

Recommendation and factorization machines

In many ways, recommender systems were a catalyst for the current popularity of machine learning. One of Amazon's earliest successes was the "Customers who bought this, also bought..." feature. The million dollar Netflix Prize spurred research, raised public awareness, and inspired numerous other data science competitions.

Recommender systems can utilize a multitude of data sources and machine learning algorithms. Most combine various unsupervised, supervised, and reinforcement learning techniques into a holistic framework. However, the core component is almost always a model that predicts a user's rating (or purchase) for a certain item based on that user's historical ratings of similar items as well as the behavior of other similar users. The minimal required dataset for this is a history of user item ratings (which we have).

The method that you'll use is a factorization machine. A factorization machine is a general-purpose supervised learning algorithm that you can use for both classification and regression tasks. It is an extension of a linear model and is designed to parsimoniously (simply) capture interactions between features in high-dimensional sparse datasets. This makes it a good candidate to handle data patterns with features such as click prediction and item recommendation.

Step 1: Problem formulation and data collection

Start this project off by writing a few sentences below that summarize the business problem and the business goal you're trying to achieve in this scenario. Include a business metric you would like your team to aspire toward. With that information defined, clearly write out the machine learning problem statement. Finally, add a comment or two about the type of machine learning this represents.

Project presentation: Include a summary of these details in your project presentations.

Read through a business scenario and:

1. Determine if and why ML is an appropriate solution to deploy.

Write your answer here

2. Formulate the business problem, success metrics, and desired ML output.

Write your answer here

3. Identify the type of ML problem you're dealing with.

Write your answer here

4. Analyze the appropriateness of the data you're working with.

Write your answer here

Setup

Now that we have decided where to focus our energy, let's set things up so you can start working on solving the problem.

Note: This notebook was created and tested on an `ml.m4.xlarge` notebook instance.

Start by specifying:

- The Amazon Simple Storage Service (Amazon S3) bucket and prefix(?) that you want to use for training and model data. This should be within the same Region as the Notebook Instance, training, and hosting.
- The AWS Identity and Access Management (IAM) role [Amazon Resource Name (ARN) (https://docs.aws.amazon.com/general/latest/gr/aws-arns-and-namespaces.html)](https://docs.aws.amazon.com/general/latest/gr/aws-arns-and-namespaces.html) used to give training and hosting access to your data. See the documentation for how to create these.

Note: If more than one role is required for notebook instances, training, and/or hosting, replace the `get_execution_role()` call with the appropriate full IAM role ARN string(s).

Replace **<LabBucketName>** with the resource name that was provided with your lab account.

In [2]:
```
# Change the bucket and prefix according to your information
#bucket = '<LabBucketName>'
bucket = 'ml-pipeline-bucket'
prefix = 'sagemaker-fm'

import sagemaker
role = sagemaker.get_execution_role()
```

Now, load some Python libraries you'll need for the remainder of this example notebook.

In [3]:
```
import os, subprocess
import warnings
import pandas as pd
import numpy as np
import sagemaker
from sagemaker.mxnet import MXNet
import boto3
import json
import matplotlib.pyplot as plt
import seaborn as sns

# Add this to display all the outputs in the cell and not just the last one
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"

# Ignore warnings
warnings.filterwarnings("ignore")
```

Step 2: Data preprocessing and visualization

In this data preprocessing phase, you should take the opportunity to explore and visualize your data to better understand it. First, import the necessary libraries and read the data into a Pandas dataframe. After that, explore your data. Look for the shape of the dataset and explore your columns and the types of columns you're working with (numerical, categorical). Consider performing basic statistics on the features to get a sense of feature means and ranges. Take a close look at your target column and determine its distribution.

Specific questions to consider

1. What can you deduce from the basic statistics you ran on the features?
2. What can you deduce from the distributions of the target classes?
3. Is there anything else you deduced from exploring the data?

Project presentation: Include a summary of your answers to these and other similar questions in your project presentations.

Start by bringing in the dataset from an Amazon S3 public bucket to this notebook environment.

In [4]:
```
# Check whether the file is already in the desired path or if it needs to be d
ownloaded

base_path = '/home/ec2-user/SageMaker/project/data/AmazonReviews'
file_path = '/amazon_reviews_us_Digital_Video_Download_v1_00.tsv.gz'

if not os.path.isfile(base_path + file_path):
    subprocess.run(['mkdir', '-p', base_path])
    subprocess.run(['aws', 's3', 'cp', 's3://amazon-reviews-pds/tsv' + file_pa
th, base_path])
else:
    print('File already downloaded!')
```

Out[4]:
```
CompletedProcess(args=['mkdir', '-p', '/home/ec2-user/SageMaker/project/data/
AmazonReviews'], returncode=0)
```

Out[4]:
```
CompletedProcess(args=['aws', 's3', 'cp', 's3://amazon-reviews-pds/tsv/amazon
_reviews_us_Digital_Video_Download_v1_00.tsv.gz', '/home/ec2-user/SageMaker/p
roject/data/AmazonReviews'], returncode=0)
```

Reading the dataset

Read the data into a Pandas dataframe so that you can know what you are dealing with.

Note: You'll set `error_bad_lines=False` when reading the file in, because there appear to be a very small number of records that would create a problem otherwise.

Hint: You can use the built-in Python `read_csv` function ([documentation (https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_csv.html)](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_csv.html)). You can use the file path directly with Pandas `read_csv` with `delimiter='\t'`.

For example: `pd.read_csv('filename.tar.gz', delimiter = '\t', error_bad_lines=False)`

In [5]:
```
df = pd.read_csv(base_path + file_path,
                 delimiter='\t',
                 error_bad_lines=False)
```

```
b'Skipping line 92523: expected 15 fields, saw 22\n'
b'Skipping line 343254: expected 15 fields, saw 22\n'
b'Skipping line 524626: expected 15 fields, saw 22\n'
b'Skipping line 623024: expected 15 fields, saw 22\n'
b'Skipping line 977412: expected 15 fields, saw 22\n'
b'Skipping line 1496867: expected 15 fields, saw 22\n'
b'Skipping line 1711638: expected 15 fields, saw 22\n'
b'Skipping line 1787213: expected 15 fields, saw 22\n'
b'Skipping line 2395306: expected 15 fields, saw 22\n'
b'Skipping line 2527690: expected 15 fields, saw 22\n'
```

Print the first few rows of your dataset.

Hint: Use the `pandas.head(<number>)` function to print the rows.

In [6]:
```
df.head(3)
```

Out[6]:

	marketplace	customer_id	review_id	product_id	product_parent	product_title	
0	US	12190288	R3FU16928EP5TC	B00AYB1482	668895143	Enlightened: Season 1	Di
1	US	30549954	R1IZHHS1MH3AQ4	B00KQD28OM	246219280	Vicious	Di
2	US	52895410	R52R85WC6TIAH	B01489L5LQ	534732318	After Words	Di

Now what is the information contained in all the columns?

Anatomy of the dataset

Get a little more comfortable with the data and see what features are at hand.

- `marketplace` : Two-letter country code (in this case, all "US")
- `customer_id` : Random identifier that can be used to aggregate reviews written by a single author
- `review_id` : Unique ID for the review
- `product_id` : Amazon Standard Identification Number (ASIN). http://www.amazon.com/dp/ (http://www.amazon.com/dp/)<ASIN> links to the product's detail page.
- `product_parent` : The parent of that ASIN. Multiple ASINs (color or format variations of the same product) can roll up into a single parent.
- `product_title` : Title description of the product
- `product_category` : Broad product category that can be used to group reviews (in this case, digital videos)
- `star_rating` : Product's rating (1 to 5 stars)
- `helpful_votes` : Number of helpful votes for the review
- `total_votes` : Number of total votes the review received
- `vine` : Was the review written as part of the Vine program?
- `verified_purchase` : Was the review from a verified purchase?
- `review_headline` : Title of the review itself
- `review_body` : Text of the review
- `review_date` : Date the review was written

Analyzing and processing the dataset

Exploring the data

Question: How many rows and columns do you have in the dataset?

Check the size of the dataset.

Hint: Use the `<dataframe>.shape` function to check the size of your dataframe

In [7]:
```
df.shape
```

Out[7]:
```
(3998345, 15)
```

Answer: (3998345,15)

Question: Which columns contain null values, and how many null values do they contain?

Print a summary of the dataset.

Hint: Use `<dataframe>.info` function using the keyword arguments `null_counts = True`

In [8]:
```
df.info(null_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3998345 entries, 0 to 3998344
Data columns (total 15 columns):
 #   Column             Non-Null Count    Dtype
---  ------             --------------    -----
 0   marketplace        3998345 non-null  object
 1   customer_id        3998345 non-null  int64
 2   review_id          3998345 non-null  object
 3   product_id         3998345 non-null  object
 4   product_parent     3998345 non-null  int64
 5   product_title      3998345 non-null  object
 6   product_category   3998345 non-null  object
 7   star_rating        3998345 non-null  int64
 8   helpful_votes      3998345 non-null  int64
 9   total_votes        3998345 non-null  int64
 10  vine               3998345 non-null  object
 11  verified_purchase  3998345 non-null  object
 12  review_headline    3998320 non-null  object
 13  review_body        3998267 non-null  object
 14  review_date        3998207 non-null  object
dtypes: int64(5), object(10)
memory usage: 457.6+ MB
```

Answer: Review headline: 25, Review_body: 78, Review_date: 138

Question: Are there any duplicate rows? If yes, how many are there?

Hint: Filter the dataframe using `dataframe.duplicated()` ([documentation (https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.duplicated.html#pandas.DataFrame.duplicated)](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.duplicated.html#pandas.DataFrame.duplicated)) and check the length of the new dataframe.

In [9]:
```
duplicates = df[df.duplicated()]

len(duplicates)
```

Out[9]: 0

Answer: There are no duplicated rows.

Data preprocessing

Now it's time to decide what features you are going to use and how you are going to prepare them for your model. For this example, limit yourself to `customer_id`, `product_id`, `product_title`, and `star_rating`. Including additional features in the recommendation system could be beneficial but would require substantial processing (particularly the text data), which would be beyond the scope of this notebook.

Reduce this dataset and only use the columns mentioned.

Hint: Select multiple columns as a dataframe by passing the columns as a list. For example: `df[['column_name 1', 'column_name 2']]`

In [10]:
```
df_reduced = df[['customer_id', 'product_id', 'star_rating', 'product_title']]
```

Check if you have duplicates after reducing the dataset.

In [11]:
```
duplicates = df_reduced[df_reduced.duplicated()]

len(duplicates)
```

Out[11]: 131

Answer: 131

Question: Why do you have duplicates in your dataset now? What changed after you reduced the dataset? Review the first 20 lines of the duplicates.

Hint: Use the `pandas.head(<number>)` function to print the rows.

In [12]:
```
duplicates.head(20)
```

Out[12]:

	customer_id	product_id	star_rating	product_title
565194	41454255	B00Y2UYRFS	1	unseen 2
594322	17570065	B00R3EEO2G	2	The Maze Runner
611264	15703996	B00I3MQNWG	5	Bosch Season 1
612471	28456429	B008Y6W7J4	5	Rabbit Hole
613791	52388381	B00YORA25I	5	McFarland, USA (Theatrical)
685156	31828958	B00TT53YSW	5	Bates Motel, Season 3
1204110	19462	B00QWUL4AW	5	Exodus: Gods and Kings
1564816	24892653	B00L2GPYKW	5	The Escape Artist Season 1
1601662	44513234	B00NY4UIKG	5	The Equalizer
1616429	43345475	B00P5968FC	3	The Babadook
1616584	20779855	B00O99GXPE	5	Max Lucado's The Christmas Candle
1617034	52861226	B00Q4FMCDS	2	The November Man
1618497	735926	B00IQBE0BU	3	Once Upon A Forest
1618598	3283400	B000NPQ0II	4	Creator
1618822	50380731	B00C58TGQ4	3	Hemingway & Gellhorn
1618896	5465295	B009AP2B9Y	5	Kickin' It Volume 3
1619053	39849456	B0035LN5YO	5	Gallipoli
1620635	22158610	B00M0HXNPK	5	Transcendence (2014)
1621358	46543213	B005OPTSIQ	5	Workaholics Season 2
1621751	25315183	B00NEFWXNK	5	Arrow: Season 3

Hint: Take a look at the first two elements in the duplicates dataframe, and query the original dataframe df to see what the data looks like. You can use the `query` function ([documentation (https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.query.html)](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.query.html)).

For example:

```
df_eg = pd.DataFrame({
            'A': [1,2,3,4],
            'B': [
        })
df_eg.query('A > 1 & B > 0')
```

In [13]:
```
df.query("customer_id == 17570065 & product_id == 'B00R3EEO2G'")
```

Out[13]:

	marketplace	customer_id	review_id	product_id	product_parent	product_tit
588857	US	17570065	RZ0MXSMTR2WQP	B00R3EEO2G	226574237	The Maz Runn
594322	US	17570065	R26IBV6N1BV1VN	B00R3EEO2G	226574237	The Maz Runn

Answer: The dataset has duplicates because there are products with the same information but different `review_id` or `product_id` .

Before continuing, remove the duplicate rows.

Hint: Use the `~` operator to select all the rows that aren't duplicated. For example:

```
df_eg = pd.DataFrame({
            'A': [1,2,3,4],
            'B': [2,0,5,2]
        })
df_eg[~(df_eg['B'] > 0)]
```

In [14]:
```
df_reduced = df_reduced[~df_reduced.duplicated()]
```

Visualize some of the rows in the dataset

If you haven't done so in the above, you can use the space below to further visualize some of your data. Look specifically at the distribution of features like `star_rating` , `customer_id` , and `product_id` .

Specific questions to consider

1. After looking at the distributions of features, to what extent might those features help your model? Is there anything you can deduce from those distributions that might be helpful in better understanding your data?
2. Should you use all the data? What features should you use?
3. What month has the highest count of user ratings?

Use the cells below to visualize your data and answer these and other questions that might be of interest to you. Insert and delete cells where needed.

Project presentation: Include a summary of your answers to these and similar questions in your project presentations.

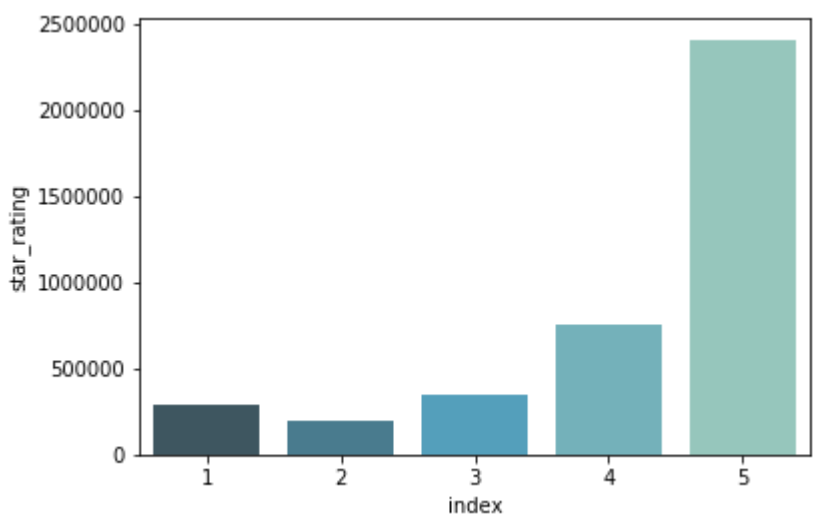
Use `sns.barplot` (documentation (https://seaborn.pydata.org/generated/seaborn.barplot.html)) to plot the `star_rating` density and distribution.

In [15]:
```
# Count the number of reviews with a specific rating
df['star_rating'].value_counts().reset_index()
sns.barplot(
    x='index',
    y='star_rating',
    data=_,  # The underscore symbol in Python is used to store the output of
 the last operation
    palette='GnBu_d'
)
```

Out[15]:

	index	star_rating
0	5	2410533
1	4	756423
2	3	345891
3	1	289731
4	2	195767

Out[15]: `<matplotlib.axes._subplots.AxesSubplot at 0x7fa88bfbd978>`



Question: What month contains the highest count of user ratings?

Hint:

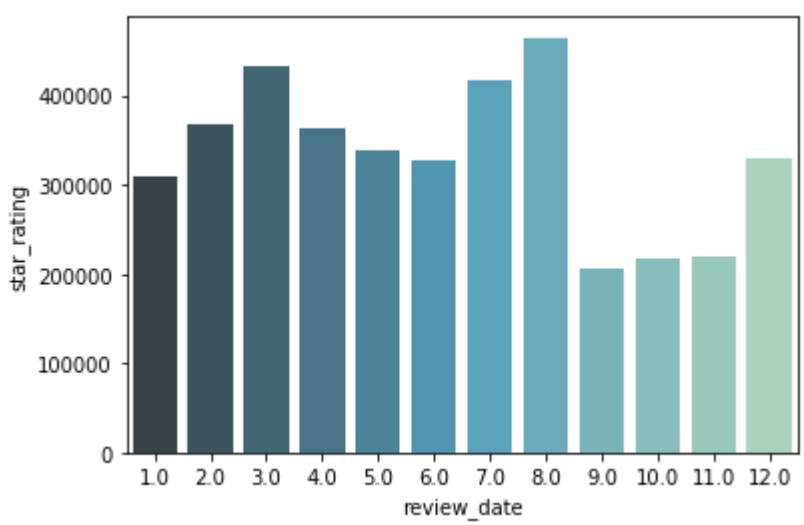
1. Use `pd.to_datetime` to convert the `review_date` column to a datetime column.
2. Use the month from the `review_date` column. You can access it for a datetime column using `<column_name>.dt.month` .
3. Use the `groupby` function using `idxmax` .

In [16]:
```
# Convert the review date to a datetime type and count the number of ratings b
y month
df['review_date'] = pd.to_datetime(df.review_date)
df.groupby(df.review_date.dt.month).star_rating.count().reset_index()
sns.barplot(x='review_date', y='star_rating', data=_, palette='GnBu_d')
```

Out[16]:

	review_date	star_rating
0	1.0	309083
1	2.0	369059
2	3.0	432084
3	4.0	363611
4	5.0	339285
5	6.0	328114
6	7.0	417844
7	8.0	464912
8	9.0	206453
9	10.0	218583
10	11.0	219316
11	12.0	329863

Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x7fa889155780>



In [17]:
```
max_month = df.groupby(df.review_date.dt.month).star_rating.count().idxmax()
print(f'The month with the most reviews is: {max_month}')
```

```
The month with the most reviews is: 8.0
```

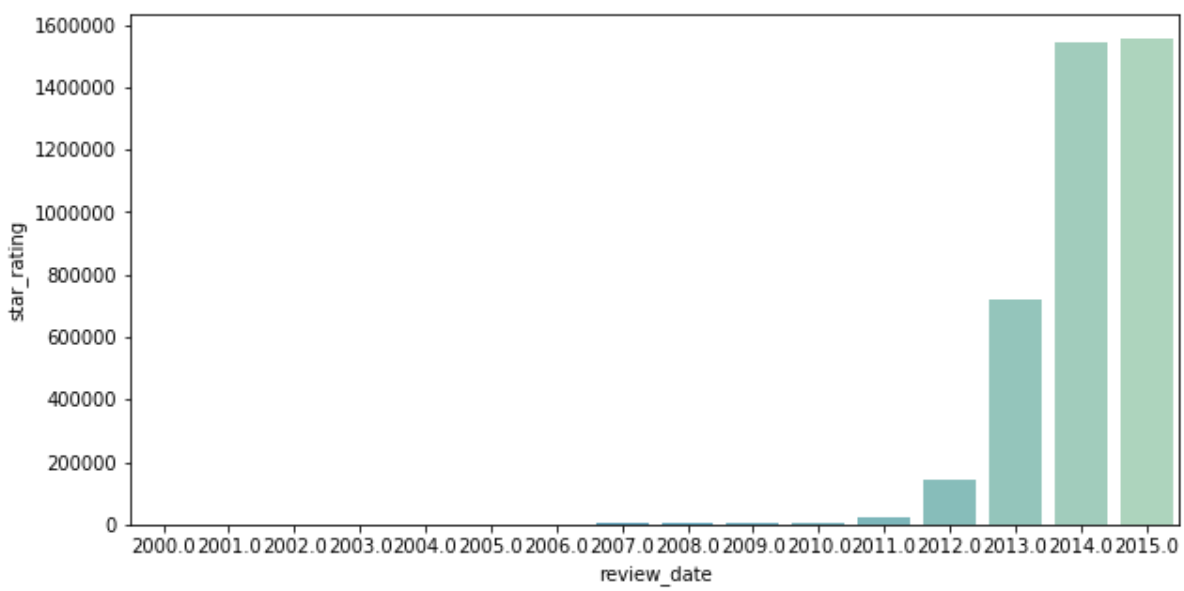
Bonus question (optional): Which years have the most and least reviews?

In [18]:
```
df.groupby(df.review_date.dt.year).star_rating.count().reset_index()
fig = plt.gcf()
fig.set_size_inches(10, 5)
sns.barplot(x='review_date', y='star_rating', data=_, palette='GnBu_d')
```

Out[18]:

	review_date	star_rating
0	2000.0	1
1	2001.0	1
2	2002.0	5
3	2003.0	4
4	2004.0	6
5	2005.0	11
6	2006.0	185
7	2007.0	2597
8	2008.0	3079
9	2009.0	3262
10	2010.0	6091
11	2011.0	20890
12	2012.0	142466
13	2013.0	721748
14	2014.0	1543049
15	2015.0	1554812

Out[18]: <matplotlib.axes._subplots.AxesSubplot at 0x7fa8890770f0>



Answer: The years with the least amount of data are 2000 and 2001, with only 1 review. The year with the highest number of reviews is 2015.

Cleaning data

Question: How heterogeneous are the number of reviews per customer and reviews per video? Use quantiles to find out.

Hint: Use `<dataframe>['columns_name'].value_counts()` for the customers and products dataframe, and use `<dataframe>.quantile(<list>)` to find the relationship.

In [19]:
```
customers = df['customer_id'].value_counts()
products = df['product_id'].value_counts()

quantiles = [0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.1, 0.25, 0.5,
             0.75, 0.9, 0.95, 0.96, 0.97, 0.98, 0.99, 0.995,
             0.999, 1]
print('customers\n', customers.quantile(quantiles))
print('products\n', products.quantile(quantiles))
```

```
customers
 0.000       1.0
0.010       1.0
0.020       1.0
0.030       1.0
0.040       1.0
0.050       1.0
0.100       1.0
0.250       1.0
0.500       1.0
0.750       2.0
0.900       4.0
0.950       5.0
0.960       6.0
0.970       7.0
0.980       9.0
0.990      13.0
0.995      18.0
0.999      37.0
1.000    2704.0
Name: customer_id, dtype: float64
products
 0.000        1.000
0.010        1.000
0.020        1.000
0.030        1.000
0.040        1.000
0.050        1.000
0.100        1.000
0.250        1.000
0.500        3.000
0.750        9.000
0.900       31.000
0.950       73.000
0.960       95.000
0.970      130.000
0.980      199.000
0.990      386.670
0.995      699.000
0.999     1993.901
1.000    32790.000
Name: product_id, dtype: float64
```

Answer: Only about 5% of customers have rated 5 or more videos, and only 25% of videos have been rated by 9+ customers.

Filter out this long tail. Select the customers that have rated 18 or more videos and the products that have more than 95 reviews.

In [20]:
```
customers1 = customers[customers >= 18]
products1 = products[products >= 95]

reduced_df = (
    df_reduced.merge(pd.DataFrame({'customer_id': customers1.index}))
              .merge(pd.DataFrame({'product_id': products1.index}))
)
```

Question: What is the shape of `customers1` , `products1` , and the new dataframe reduced_df?

Note: Use f-strings for this:

```
x= 3
print(f'X = {x}')
```

In [21]:
```
print(f'Number of users is {customers1.shape[0]} and number of items is {produ
cts1.shape[0]}.')
print(f'Length of reduced df is {reduced_df.shape[0]}.')
```

```
Number of users is 10538 and number of items is 6683.
Length of reduced df is 173500.
```

Print the first 5 columns of the dataframe.

In [22]:
```
reduced_df.head()
```

Out[22]:

	customer_id	product_id	star_rating	product_title
0	11763902	B00PSLQYWE	4	Downton Abbey Season 5
1	1411480	B00PSLQYWE	5	Downton Abbey Season 5
2	35303629	B00PSLQYWE	5	Downton Abbey Season 5
3	21285980	B00PSLQYWE	5	Downton Abbey Season 5
4	29260449	B00PSLQYWE	5	Downton Abbey Season 5

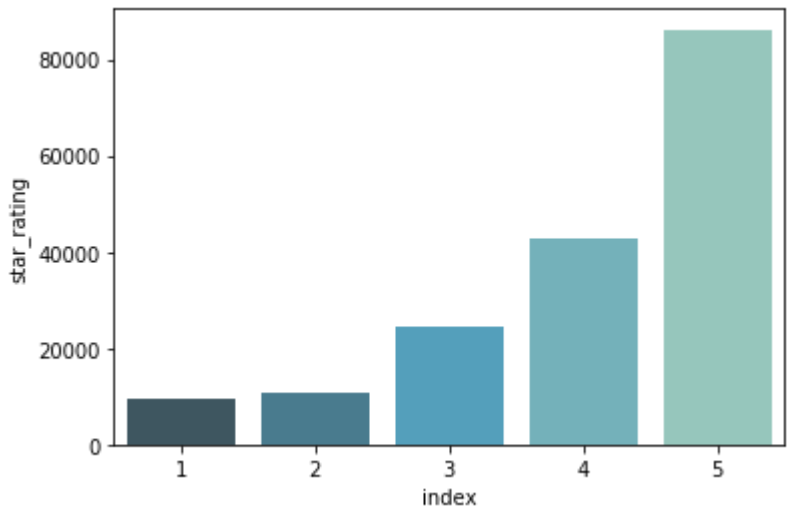
Question: Does `reduced_df` maintain the same ratio of ratings?

In [23]:
```
reduced_df['star_rating'].value_counts().reset_index()
sns.barplot(x='index', y='star_rating', data=_, palette='GnBu_d')
```

Out[23]:

	index	star_rating
0	5	86200
1	4	42668
2	3	24470
3	2	10803
4	1	9359

Out[23]: `<matplotlib.axes._subplots.AxesSubplot at 0x7fa888f81f98>`



Answer: The number of reviews with a 1-star rating decreased in proportion to 2-star ratings. That was not the case with the original data.

Now, recreate the customer and product distributions of count per customer and product.

Hint: Use the `value_counts()` function on the `customer_id` and `product_id` columns.

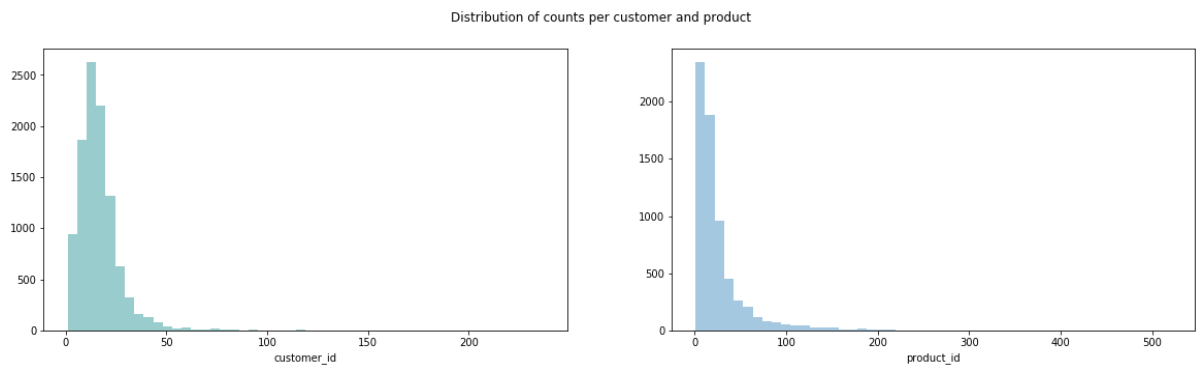
In [24]:
```
customers = reduced_df['customer_id'].value_counts()
products = reduced_df['product_id'].value_counts()

fig, axs = plt.subplots(1, 2, figsize=(20, 5))
fig.suptitle('Distribution of counts per customer and product')
sns.distplot(customers, kde=False, ax=axs[0], color='teal')
sns.distplot(products, kde=False, ax=axs[1])
```

Out[24]: Text(0.5, 0.98, 'Distribution of counts per customer and product')

Out[24]: <matplotlib.axes._subplots.AxesSubplot at 0x7fa888eecf28>

Out[24]: <matplotlib.axes._subplots.AxesSubplot at 0x7fa888eb0668>



Next, number each user and item, giving them their own sequential index. This will allow you to hold the information in a sparse format where the sequential indices indicate the row and column in the ratings matrix.

To create the `customer_index` and `product_index` , create a new dataframe with `customer_id` as the index value and a sequential counter/values for the user and item number. Once you are finished creating both indexes, use the Pandas `merge` function to merge `customer_index` with `product_index` .

Hint: Use the `shape` function to generate the total number of customers and products. Use `np.arange` to generate a list of numbers from 0 to the number of customers and products.

In [25]:
```
customer_index = pd.DataFrame({'customer_id': customers.index,
                               'user': np.arange(customers.shape[0])})
product_index = pd.DataFrame({'product_id': products.index,
                              'item': np.arange(products.shape[0])})

reduced_df = reduced_df.merge(customer_index).merge(product_index)
reduced_df.head()
```

Out[25]:

	customer_id	product_id	star_rating	product_title	user	item
0	11763902	B00PSLQYWE	4	Downton Abbey Season 5	3065	102
1	1411480	B00PSLQYWE	5	Downton Abbey Season 5	130	102
2	35303629	B00PSLQYWE	5	Downton Abbey Season 5	4682	102
3	21285980	B00PSLQYWE	5	Downton Abbey Season 5	449	102
4	29260449	B00PSLQYWE	5	Downton Abbey Season 5	131	102

Sample answer:

	customer_id	product_id	star_rating	product_title	user	item
0	11763902	B00PSLQYWE	4	Downton Abbey Season 5	3065	103
1	1411480	B00PSLQYWE	5	Downton Abbey Season 5	130	103
2	35303629	B00PSLQYWE	5	Downton Abbey Season 5	4683	103
3	21285980	B00PSLQYWE	5	Downton Abbey Season 5	449	103
4	29260449	B00PSLQYWE	5	Downton Abbey Season 5	131	103

Project Solution Lab 3:

Step 3: Model training and evaluation

There are some preliminary steps that you must include when converting the dataset from a dataframe to a format that a machine learning algorithm can use. For Amazon SageMaker, here are the steps you need to take:

1. Split the data into `train_data` and `test_data` .
2. Convert the dataset to an appropriate file format that the Amazon SageMaker training job can use. This can be either a CSV file or record protobuf. For more information, see [Common Data Formats for Training (https://docs.aws.amazon.com/sagemaker/latest/dg/cdf-training.html)](https://docs.aws.amazon.com/sagemaker/latest/dg/cdf-training.html). For this problem, the data will be sparse, so you can use the `scipy.sparse.lilmatrix` function and then convert the function to the `RecordIO protobuf` format using `sagemaker.amazon.common.write_spmatrix_to_sparse_tensor` .
3. Upload the data to your Amazon S3 bucket. If you have not created one before, see [Create a Bucket (https://docs.aws.amazon.com/AmazonS3/latest/gsg/CreatingABucket.html)](https://docs.aws.amazon.com/AmazonS3/latest/gsg/CreatingABucket.html).

Use the following cells to complete these steps. Insert and delete cells where needed.

Project presentation: Take note of the key decisions you've made in this phase in your project presentations.

Prepare the data

You are at a point where you can start preparing the dataset as input for your model. Every model has different input needs. Some of the algorithms implemented in Amazon SageMaker require the data to be in the recordIO-wrapped protobuf form. You will take care of that in the following cells.

First, split the dataset into training and test sets. This will allow you to estimate the model's accuracy on videos that customers rated but that weren't included in the training.

Start with creation of the `test_df` dataframe. Create the dataframe by grouping the dataframe on `customer_id` and using the `last` function, similar to `pd.groupby('  ').last()` .

In [26]:
```
test_df = reduced_df.groupby('customer_id').last().reset_index()
```

To create the training data, remove the values present in `test_df` from the `reduced_df` dataframe.

Hint: Merge the `reduced_df` dataframe with the `test_df` dataset with `customer_id` and `product_id` columns as an outer join.

In [27]:
```
train_df = reduced_df.merge(test_df[['customer_id', 'product_id']],
                            on=['customer_id', 'product_id'],
                            how='outer',
                            indicator=True)
train_df = train_df[(train_df['_merge'] == 'left_only')].reset_index()
```

In [28]:
```
test_df.head()
```

Out[28]:

	customer_id	product_id	star_rating	product_title	user	item
0	10981	B00D6BQG1W	5	Warm Bodies	1859	3776
1	12783	B001GQK56G	4	Inspector Morse Season 1	1951	2718
2	17106	B00RPNBLWG	5	A Most Violent Year	10337	2042
3	19462	B00DFFI7K2	1	G.I. Joe: Retaliation	9030	628
4	54160	B005OPTSIQ	2	Workaholics Season 2	7309	3679

Now you can look at some basic characteristics of the data that will later help you convert the features to an appropriate format for training your model.

Create two variables `nb_rating_test` and `nb_ratings_train` for the length of the test and training datasets.

In [29]:
```
nb_ratings_test = len(test_df.index)
nb_ratings_train = len(train_df.index)
print(f" Training Count: {nb_ratings_train}")
print(f" Test Count: {nb_ratings_test}")
```

```
 Training Count: 163064
 Test Count: 10436
```

Data conversion

Now, you can convert your Pandas dataframes into a sparse matrix. This process is the same for both train and test. The Amazon SageMaker implementation of factorization machines takes recordIO-wrapped protobuf, where the data you have today is a Pandas dataframe on disk. Therefore, you are going to convert the data to a sparse matrix to express the relationships between each user and each movie.

In [30]:
```
# First convert data to sparse fomat with scipy lil_matrix

from scipy.sparse import lil_matrix

def loadDataset(df, lines, columns, regressor=True):
    """
    Convert the pandas dataframe into sparse matrix

    Args:
        df: DataFrame
        lines: number of rows of the final sparse matrix
        columns: number of columns of final sparse matrix
        regressor: Boolean value to check if we are using regression
                   or classification
    Returns:
        X: Feature vector
        Y: Label vector
    """
    # Features are one-hot encoded in a sparse matrix
    # Use scipy.sparse.lil_matrix to create the feature vector X of type float
32
    X = lil_matrix((len(df), lines + columns)).astype('float32')

    # Labels are stored in a vector. Instantiate an empty label vector Y.
    Y = []

    line = 0

    # For each row in the dataframe, use 1 for the item and product number
    for index, row in df.iterrows():
        X[line,row['user']] = 1
        X[line, lines + (row['item'])] = 1
        line += 1

        if regressor:
            # If using regression, append the star_rating
            Y.append(row['star_rating'])
        else:
            # Use 1 for star_rating 5 else use 0
            if int(row['star_rating']) >= 5:
                Y.append(1)
            else:
                Y.append(0)

    # Convert the list into NumPy array of type float32
    Y = np.array(Y).astype('float32')
    return X, Y
```

Use the `loadDataset` function to create the training and test sets.

In [31]:
```
print(customers.shape[0],
      products.shape[0],
      customers.shape[0] + products.shape[0])

X_train, Y_train = loadDataset(train_df, customers.shape[0],
                               products.shape[0])
X_test, Y_test = loadDataset(test_df, customers.shape[0],
                             products.shape[0])
```

```
10436 6659 17095
```

Now that your data is in a sparse format, save it as a protobuf format and upload it to Amazon S3. This step might look intimidating, but most of the conversion effort is handled by the Amazon SageMaker Python SDK, imported as SageMaker below.

In [32]:
```
import io
import sagemaker.amazon.common as smac

def writeDatasetToProtobuf(X, bucket, prefix, key, d_type, Y=None):
    buf = io.BytesIO()
    if d_type == "sparse":
        smac.write_spmatrix_to_sparse_tensor(buf, X, labels=Y)
    else:
        smac.write_numpy_to_dense_tensor(buf, X, labels=Y)

    buf.seek(0)
    obj = '{}/{}'.format(prefix, key)
    boto3.resource('s3').Bucket(bucket).Object(obj).upload_fileobj(buf)
    return 's3://{}/{}'.format(bucket,obj)

fm_train_data_path = writeDatasetToProtobuf(X_train, bucket, prefix, 'train',
"sparse", Y_train)
fm_test_data_path  = writeDatasetToProtobuf(X_test, bucket, prefix, 'test', "s
parse", Y_test)

print("Training data S3 path: ", fm_train_data_path)
print("Test data S3 path: ", fm_test_data_path)
```

```
Training data S3 path:  s3://qls-163754-04de04c0c698e64b-labbucket-q4zl2sa0cq
9l/sagemaker-fm/train
Test data S3 path:  s3://qls-163754-04de04c0c698e64b-labbucket-q4zl2sa0cq9l/s
agemaker-fm/test
```

You are finally finished with data preparation. Hooray! As you can see, it takes a lot of time and effort to clean and prepare the data for modeling. This is true for every single data science project, and this step has a high impact on the outcome. Make sure you spend enough time understanding and preparing your data for training in all future machine learning dventures!

Training the model

Now it's time to train the model. You will use an Amazon SageMaker training job for that. Amazon SageMaker training jobs are an easy way to create models, as you don't really have to write all the code for training. That is already handled for you in a nice container format.

The general workflow for creating training jobs from the notebook is to instantiate the predictor, pass some hyperparameters, and then pass the data in the correct format. This is what happens in the following cell.

For more more information about FM estimator, see [FactorizationMachines (https://sagemaker.readthedocs.io/en/stable/factorization_machines.html)](https://sagemaker.readthedocs.io/en/stable/factorization_machines.html).

For more information about hyperparameters, see [Factorization Machines Hyperparameters (https://docs.aws.amazon.com/sagemaker/latest/dg/fact-machines-hyperparameters.html)](https://docs.aws.amazon.com/sagemaker/latest/dg/fact-machines-hyperparameters.html).

Hint: Example:

```
sess = sagemaker.Session()

pca = sagemaker.estimator.Estimator(containers[boto3.Session().region_name],
                                    role,
                                    instance_count=1,
                                    instance_type='ml.m4.xlarge',
                                    output_path=output_location,
                                    sagemaker_session=sess)

pca.set_hyperparameters(featuer_dim=50000,
                        num_components=10,
                        subtract_mean=True,
                        algorithm_mode='randomized',
                        mini_batch_size=200)

pca.fit({'train': s3_train_data})
```

In [33]:
```
from sagemaker import get_execution_role
from sagemaker import image_uris

output_prefix = 's3://' + bucket + '/sagemaker-fm/model'
instance_type='ml.m4.xlarge'
batch_size = 128

fm = sagemaker.estimator.Estimator(
    image_uris.retrieve("factorization-machines",boto3.Session().region_name),
    role,
    instance_count=1,
    instance_type=instance_type,
    output_path=output_prefix,
    sagemaker_session=sagemaker.Session()
)

fm.set_hyperparameters(feature_dim=X_train.shape[1],
                       predictor_type='regressor',
                     # predictor_type='binary_classifier',
                       mini_batch_size=batch_size,
                       num_factors=64,
                       epochs=25,
                       clip_gradient=5.0,
                       rescale_grad=1.0/batch_size)

fm.fit({'train': fm_train_data_path, 'test': fm_test_data_path})
```

```
2020-11-11 15:51:17 Starting - Starting the training job...
2020-11-11 15:51:19 Starting - Launching requested ML instances......
2020-11-11 15:52:29 Starting - Preparing the instances for training......
2020-11-11 15:53:32 Downloading - Downloading input data...
2020-11-11 15:54:05 Training - Downloading the training image..Docker entrypo
int called with argument(s): train
Running default environment configuration script
/opt/amazon/lib/python2.7/site-packages/pandas/util/nosetester.py:13: Depreca
tionWarning: Importing from numpy.testing.nosetester is deprecated, import fr
om numpy.testing instead.
  from numpy.testing import nosetester
[11/11/2020 15:54:29 INFO 140334359897920] Reading default configuration from
/opt/amazon/lib/python2.7/site-packages/algorithm/resources/default-conf.jso
n: {u'factors_lr': u'0.0001', u'linear_init_sigma': u'0.01', u'epochs': 1,
 u'_wd': u'1.0', u'_num_kv_servers': u'auto', u'use_bias': u'true', u'factors
_init_sigma': u'0.001', u'_log_level': u'info', u'bias_init_method': u'norma
l', u'linear_init_method': u'normal', u'linear_lr': u'0.001', u'factors_init_
method': u'normal', u'_tuning_objective_metric': u'', u'bias_wd': u'0.01',
 u'use_linear': u'true', u'bias_lr': u'0.1', u'mini_batch_size': u'1000', u'_
use_full_symbolic': u'true', u'batch_metrics_publish_interval': u'500', u'bia
s_init_sigma': u'0.01', u'_num_gpus': u'auto', u'_data_format': u'record',
 u'factors_wd': u'0.00001', u'linear_wd': u'0.001', u'_kvstore': u'auto', u'_
learning_rate': u'1.0', u'_optimizer': u'adam'}
[11/11/2020 15:54:29 INFO 140334359897920] Reading provided configuration fro
m /opt/ml/input/config/hyperparameters.json: {u'epochs': u'25', u'predictor_t
ype': u'regressor', u'rescale_grad': u'0.0078125', u'clip_gradient': u'5.0',
 u'feature_dim': u'17095', u'num_factors': u'64', u'mini_batch_size': u'128'}
[11/11/2020 15:54:29 INFO 140334359897920] Final configuration: {u'factors_l
r': u'0.0001', u'linear_init_sigma': u'0.01', u'epochs': u'25', u'feature_di
m': u'17095', u'num_factors': u'64', u'_wd': u'1.0', u'_num_kv_servers': u'au
to', u'use_bias': u'true', u'factors_init_sigma': u'0.001', u'_log_level':
 u'info', u'bias_init_method': u'normal', u'linear_init_method': u'normal',
 u'rescale_grad': u'0.0078125', u'linear_lr': u'0.001', u'factors_init_metho
d': u'normal', u'_tuning_objective_metric': u'', u'bias_wd': u'0.01', u'use_l
inear': u'true', u'bias_lr': u'0.1', u'mini_batch_size': u'128', u'_use_full_
symbolic': u'true', u'clip_gradient': u'5.0', u'batch_metrics_publish_interva
l': u'500', u'predictor_type': u'regressor', u'bias_init_sigma': u'0.01', u'_
num_gpus': u'auto', u'_data_format': u'record', u'factors_wd': u'0.00001',
 u'linear_wd': u'0.001', u'_kvstore': u'auto', u'_learning_rate': u'1.0', u'_
optimizer': u'adam'}
[11/11/2020 15:54:29 WARNING 140334359897920] Loggers have already been setu
p.
Process 1 is a worker.
[11/11/2020 15:54:29 INFO 140334359897920] Using default worker.
[2020-11-11 15:54:29.390] [tensorio] [warning] TensorIO is already initialize
d; ignoring the initialization routine.
[2020-11-11 15:54:29.394] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 0, "duration": 6, "num_examples": 1, "num_b
ytes": 8192}
[11/11/2020 15:54:29 INFO 140334359897920] nvidia-smi took: 0.025162935257 se
cs to identify 0 gpus
[11/11/2020 15:54:29 INFO 140334359897920] Number of GPUs being used: 0
[11/11/2020 15:54:29 INFO 140334359897920] [Sparse network] Building a sparse
network.
[11/11/2020 15:54:29 INFO 140334359897920] Create Store: local
```

```
#metrics {"Metrics": {"initialize.time": {"count": 1, "max": 38.8560295104980
5, "sum": 38.85602951049805, "min": 38.85602951049805}}, "EndTime": 160511006
9.431333, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605110069.383226}

#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 1, "sum": 1.0, "min": 1}, "Number of Batches Since Last Reset": {"count": 1,
"max": 0, "sum": 0.0, "min": 0}, "Number of Records Since Last Reset": {"coun
t": 1, "max": 0, "sum": 0.0, "min": 0}, "Total Batches Seen": {"count": 1, "m
ax": 1, "sum": 1.0, "min": 1}, "Total Records Seen": {"count": 1, "max": 128,
"sum": 128.0, "min": 128}, "Max Records Seen Between Resets": {"count": 1, "m
ax": 128, "sum": 128.0, "min": 128}, "Reset Count": {"count": 1, "max": 1, "s
um": 1.0, "min": 1}}, "EndTime": 1605110069.431568, "Dimensions": {"Host": "a
lgo-1", "Meta": "init_train_data_iter", "Operation": "training", "Algorithm":
"factorization-machines"}, "StartTime": 1605110069.431502}

[15:54:29] /opt/brazil-pkg-cache/packages/AIAlgorithmsMXNet/AIAlgorithmsMXNet
-1.1.x.203343.0/AL2012/generic-flavor/src/src/kvstore/./kvstore_local.h:280:
 Warning: non-default weights detected during kvstore pull. This call has bee
n ignored. Please make sure to use row_sparse_pull with row_ids.
[15:54:29] /opt/brazil-pkg-cache/packages/AIAlgorithmsMXNet/AIAlgorithmsMXNet
-1.1.x.203343.0/AL2012/generic-flavor/src/src/kvstore/./kvstore_local.h:280:
 Warning: non-default weights detected during kvstore pull. This call has bee
n ignored. Please make sure to use row_sparse_pull with row_ids.
[11/11/2020 15:54:29 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=0, batch=0 train rmse <loss>=4.8314414871
[11/11/2020 15:54:29 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=0, batch=0 train mse <loss>=23.3428268433
[11/11/2020 15:54:29 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=0, batch=0 train absolute_loss <loss>=4.80321931839
[11/11/2020 15:54:30 INFO 140334359897920] Iter[0] Batch [500]#011Speed: 4548
2.13 samples/sec
[11/11/2020 15:54:30 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=0, batch=500 train rmse <loss>=1.33602978251
[11/11/2020 15:54:30 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=0, batch=500 train mse <loss>=1.78497557976
[11/11/2020 15:54:30 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=0, batch=500 train absolute_loss <loss>=1.02332682334
[11/11/2020 15:54:32 INFO 140334359897920] Iter[0] Batch [1000]#011Speed: 365
71.28 samples/sec
[11/11/2020 15:54:32 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=0, batch=1000 train rmse <loss>=1.26037772916
[11/11/2020 15:54:32 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=0, batch=1000 train mse <loss>=1.58855202017
[11/11/2020 15:54:32 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=0, batch=1000 train absolute_loss <loss>=0.985341296568
[2020-11-11 15:54:33.524] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 2, "duration": 4030, "num_examples": 1274,
 "num_bytes": 10436096}
[11/11/2020 15:54:33 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=0, train rmse <loss>=1.24960825905
[11/11/2020 15:54:33 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=0, train mse <loss>=1.56152080109
[11/11/2020 15:54:33 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=0, train absolute_loss <loss>=0.984034941606
#metrics {"Metrics": {"epochs": {"count": 1, "max": 25, "sum": 25.0, "min": 2
```

```
5}, "update.time": {"count": 1, "max": 4093.13702583313, "sum": 4093.13702583
313, "min": 4093.13702583313}}, "EndTime": 1605110073.524987, "Dimensions":
 {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-mach
ines"}, "StartTime": 1605110069.431421}

[11/11/2020 15:54:33 INFO 140334359897920] #progress_metric: host=algo-1, com
pleted 4 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 1274, "sum": 1274.0, "min": 1274}, "Number of Batches Since Last Reset": {"c
ount": 1, "max": 1274, "sum": 1274.0, "min": 1274}, "Number of Records Since
 Last Reset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "T
otal Batches Seen": {"count": 1, "max": 1275, "sum": 1275.0, "min": 1275}, "T
otal Records Seen": {"count": 1, "max": 163192, "sum": 163192.0, "min": 16319
2}, "Max Records Seen Between Resets": {"count": 1, "max": 163064, "sum": 163
064.0, "min": 163064}, "Reset Count": {"count": 1, "max": 2, "sum": 2.0, "mi
n": 2}}, "EndTime": 1605110073.525315, "Dimensions": {"Host": "algo-1", "Met
a": "training_data_iter", "Operation": "training", "Algorithm": "factorizatio
n-machines", "epoch": 0}, "StartTime": 1605110069.4318}

[11/11/2020 15:54:33 INFO 140334359897920] #throughput_metric: host=algo-1, t
rain throughput=39833.1883986 records/second
[11/11/2020 15:54:33 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=1, batch=0 train rmse <loss>=0.8809152319
[11/11/2020 15:54:33 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=1, batch=0 train mse <loss>=0.776011645794
[11/11/2020 15:54:33 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=1, batch=0 train absolute_loss <loss>=0.832543611526
[11/11/2020 15:54:34 INFO 140334359897920] Iter[1] Batch [500]#011Speed: 5246
5.29 samples/sec
[11/11/2020 15:54:34 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=1, batch=500 train rmse <loss>=1.10744924717
[11/11/2020 15:54:34 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=1, batch=500 train mse <loss>=1.22644383506
[11/11/2020 15:54:34 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=1, batch=500 train absolute_loss <loss>=0.88313777028

2020-11-11 15:54:26 Training - Training image download completed. Training in
progress.[11/11/2020 15:54:35 INFO 140334359897920] Iter[1] Batch [1000]#011S
peed: 51972.61 samples/sec
[11/11/2020 15:54:35 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=1, batch=1000 train rmse <loss>=1.13422066056
[11/11/2020 15:54:35 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=1, batch=1000 train mse <loss>=1.28645650684
[11/11/2020 15:54:35 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=1, batch=1000 train absolute_loss <loss>=0.908082547424
[2020-11-11 15:54:36.721] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 4, "duration": 3193, "num_examples": 1274,
 "num_bytes": 10436096}
[11/11/2020 15:54:36 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=1, train rmse <loss>=1.14598292619
[11/11/2020 15:54:36 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=1, train mse <loss>=1.31327686713
[11/11/2020 15:54:36 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=1, train absolute_loss <loss>=0.919676214176
#metrics {"Metrics": {"update.time": {"count": 1, "max": 3195.8959102630615,
 "sum": 3195.8959102630615, "min": 3195.8959102630615}}, "EndTime": 160511007
```

```
6.7222, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605110073.525106}

[11/11/2020 15:54:36 INFO 140334359897920] #progress_metric: host=algo-1, com
pleted 8 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 1274, "sum": 1274.0, "min": 1274}, "Number of Batches Since Last Reset": {"c
ount": 1, "max": 1274, "sum": 1274.0, "min": 1274}, "Number of Records Since
 Last Reset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "T
otal Batches Seen": {"count": 1, "max": 2549, "sum": 2549.0, "min": 2549}, "T
otal Records Seen": {"count": 1, "max": 326256, "sum": 326256.0, "min": 32625
6}, "Max Records Seen Between Resets": {"count": 1, "max": 163064, "sum": 163
064.0, "min": 163064}, "Reset Count": {"count": 1, "max": 3, "sum": 3.0, "mi
n": 3}}, "EndTime": 1605110076.72246, "Dimensions": {"Host": "algo-1", "Met
a": "training_data_iter", "Operation": "training", "Algorithm": "factorizatio
n-machines", "epoch": 1}, "StartTime": 1605110073.526264}

[11/11/2020 15:54:36 INFO 140334359897920] #throughput_metric: host=algo-1, t
rain throughput=51015.973612 records/second
[11/11/2020 15:54:36 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=2, batch=0 train rmse <loss>=0.887766282336
[11/11/2020 15:54:36 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=2, batch=0 train mse <loss>=0.788128972054
[11/11/2020 15:54:36 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=2, batch=0 train absolute_loss <loss>=0.83714312315
[11/11/2020 15:54:38 INFO 140334359897920] Iter[2] Batch [500]#011Speed: 5010
2.24 samples/sec
[11/11/2020 15:54:38 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=2, batch=500 train rmse <loss>=1.09028215903
[11/11/2020 15:54:38 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=2, batch=500 train mse <loss>=1.1887151863
[11/11/2020 15:54:38 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=2, batch=500 train absolute_loss <loss>=0.867635217969
[11/11/2020 15:54:39 INFO 140334359897920] Iter[2] Batch [1000]#011Speed: 503
91.75 samples/sec
[11/11/2020 15:54:39 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=2, batch=1000 train rmse <loss>=1.11658644539
[11/11/2020 15:54:39 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=2, batch=1000 train mse <loss>=1.24676529002
[11/11/2020 15:54:39 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=2, batch=1000 train absolute_loss <loss>=0.892808036549
[2020-11-11 15:54:39.943] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 6, "duration": 3217, "num_examples": 1274,
 "num_bytes": 10436096}
[11/11/2020 15:54:39 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=2, train rmse <loss>=1.12826089537
[11/11/2020 15:54:39 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=2, train mse <loss>=1.27297264803
[11/11/2020 15:54:39 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=2, train absolute_loss <loss>=0.904543336183
#metrics {"Metrics": {"update.time": {"count": 1, "max": 3220.2799320220947,
 "sum": 3220.2799320220947, "min": 3220.2799320220947}}, "EndTime": 160511007
9.943738, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605110076.722295}

[11/11/2020 15:54:39 INFO 140334359897920] #progress_metric: host=algo-1, com
```

```
pleted 12 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 1274, "sum": 1274.0, "min": 1274}, "Number of Batches Since Last Reset": {"c
ount": 1, "max": 1274, "sum": 1274.0, "min": 1274}, "Number of Records Since
 Last Reset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "T
otal Batches Seen": {"count": 1, "max": 3823, "sum": 3823.0, "min": 3823}, "T
otal Records Seen": {"count": 1, "max": 489320, "sum": 489320.0, "min": 48932
0}, "Max Records Seen Between Resets": {"count": 1, "max": 163064, "sum": 163
064.0, "min": 163064}, "Reset Count": {"count": 1, "max": 4, "sum": 4.0, "mi
n": 4}}, "EndTime": 1605110079.944025, "Dimensions": {"Host": "algo-1", "Met
a": "training_data_iter", "Operation": "training", "Algorithm": "factorizatio
n-machines", "epoch": 2}, "StartTime": 1605110076.723413}

[11/11/2020 15:54:39 INFO 140334359897920] #throughput_metric: host=algo-1, t
rain throughput=50628.7762984 records/second
[11/11/2020 15:54:39 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=3, batch=0 train rmse <loss>=0.883823625932
[11/11/2020 15:54:39 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=3, batch=0 train mse <loss>=0.781144201756
[11/11/2020 15:54:39 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=3, batch=0 train absolute_loss <loss>=0.8312099576
[11/11/2020 15:54:41 INFO 140334359897920] Iter[3] Batch [500]#011Speed: 5197
4.84 samples/sec
[11/11/2020 15:54:41 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=3, batch=500 train rmse <loss>=1.07473813385
[11/11/2020 15:54:41 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=3, batch=500 train mse <loss>=1.15506205635
[11/11/2020 15:54:41 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=3, batch=500 train absolute_loss <loss>=0.853333925059
[11/11/2020 15:54:42 INFO 140334359897920] Iter[3] Batch [1000]#011Speed: 489
53.02 samples/sec
[11/11/2020 15:54:42 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=3, batch=1000 train rmse <loss>=1.1011050623
[11/11/2020 15:54:42 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=3, batch=1000 train mse <loss>=1.21243235823
[11/11/2020 15:54:42 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=3, batch=1000 train absolute_loss <loss>=0.879178321147
[2020-11-11 15:54:43.177] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 8, "duration": 3230, "num_examples": 1274,
 "num_bytes": 10436096}
[11/11/2020 15:54:43 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=3, train rmse <loss>=1.11292184622
[11/11/2020 15:54:43 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=3, train mse <loss>=1.2385950358
[11/11/2020 15:54:43 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=3, train absolute_loss <loss>=0.891243942164
#metrics {"Metrics": {"update.time": {"count": 1, "max": 3232.9108715057373,
 "sum": 3232.9108715057373, "min": 3232.9108715057373}}, "EndTime": 160511008
3.177873, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605110079.943835}

[11/11/2020 15:54:43 INFO 140334359897920] #progress_metric: host=algo-1, com
pleted 16 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 1274, "sum": 1274.0, "min": 1274}, "Number of Batches Since Last Reset": {"c
ount": 1, "max": 1274, "sum": 1274.0, "min": 1274}, "Number of Records Since
```

```
 Last Reset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "T
otal Batches Seen": {"count": 1, "max": 5097, "sum": 5097.0, "min": 5097}, "T
otal Records Seen": {"count": 1, "max": 652384, "sum": 652384.0, "min": 65238
4}, "Max Records Seen Between Resets": {"count": 1, "max": 163064, "sum": 163
064.0, "min": 163064}, "Reset Count": {"count": 1, "max": 5, "sum": 5.0, "mi
n": 5}}, "EndTime": 1605110083.178152, "Dimensions": {"Host": "algo-1", "Met
a": "training_data_iter", "Operation": "training", "Algorithm": "factorizatio
n-machines", "epoch": 3}, "StartTime": 1605110079.944927}

[11/11/2020 15:54:43 INFO 140334359897920] #throughput_metric: host=algo-1, t
rain throughput=50431.7184542 records/second
[11/11/2020 15:54:43 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=4, batch=0 train rmse <loss>=0.878184280567
[11/11/2020 15:54:43 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=4, batch=0 train mse <loss>=0.771207630634
[11/11/2020 15:54:43 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=4, batch=0 train absolute_loss <loss>=0.824202656746
[11/11/2020 15:54:44 INFO 140334359897920] Iter[4] Batch [500]#011Speed: 5193
3.47 samples/sec
[11/11/2020 15:54:44 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=4, batch=500 train rmse <loss>=1.06142055422
[11/11/2020 15:54:44 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=4, batch=500 train mse <loss>=1.12661359291
[11/11/2020 15:54:44 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=4, batch=500 train absolute_loss <loss>=0.841020489524
[11/11/2020 15:54:45 INFO 140334359897920] Iter[4] Batch [1000]#011Speed: 524
51.06 samples/sec
[11/11/2020 15:54:45 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=4, batch=1000 train rmse <loss>=1.08786765234
[11/11/2020 15:54:45 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=4, batch=1000 train mse <loss>=1.183456029
[11/11/2020 15:54:45 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=4, batch=1000 train absolute_loss <loss>=0.867482604442
[2020-11-11 15:54:46.293] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 10, "duration": 3112, "num_examples": 1274,
"num_bytes": 10436096}
[11/11/2020 15:54:46 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=4, train rmse <loss>=1.09983062834
[11/11/2020 15:54:46 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=4, train mse <loss>=1.20962741103
[11/11/2020 15:54:46 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=4, train absolute_loss <loss>=0.879827954331
#metrics {"Metrics": {"update.time": {"count": 1, "max": 3115.149974822998,
 "sum": 3115.149974822998, "min": 3115.149974822998}}, "EndTime": 1605110086.
29421, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm":
"factorization-machines"}, "StartTime": 1605110083.177951}

[11/11/2020 15:54:46 INFO 140334359897920] #progress_metric: host=algo-1, com
pleted 20 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 1274, "sum": 1274.0, "min": 1274}, "Number of Batches Since Last Reset": {"c
ount": 1, "max": 1274, "sum": 1274.0, "min": 1274}, "Number of Records Since
 Last Reset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "T
otal Batches Seen": {"count": 1, "max": 6371, "sum": 6371.0, "min": 6371}, "T
otal Records Seen": {"count": 1, "max": 815448, "sum": 815448.0, "min": 81544
8}, "Max Records Seen Between Resets": {"count": 1, "max": 163064, "sum": 163
```

```
064.0, "min": 163064}, "Reset Count": {"count": 1, "max": 6, "sum": 6.0, "mi
n": 6}}, "EndTime": 1605110086.294484, "Dimensions": {"Host": "algo-1", "Met
a": "training_data_iter", "Operation": "training", "Algorithm": "factorizatio
n-machines", "epoch": 4}, "StartTime": 1605110083.179019}

[11/11/2020 15:54:46 INFO 140334359897920] #throughput_metric: host=algo-1, t
rain throughput=52337.8625466 records/second
[11/11/2020 15:54:46 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=5, batch=0 train rmse <loss>=0.872543941091
[11/11/2020 15:54:46 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=5, batch=0 train mse <loss>=0.761332929134
[11/11/2020 15:54:46 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=5, batch=0 train absolute_loss <loss>=0.817543208599
[11/11/2020 15:54:47 INFO 140334359897920] Iter[5] Batch [500]#011Speed: 4977
3.03 samples/sec
[11/11/2020 15:54:47 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=5, batch=500 train rmse <loss>=1.04987547663
[11/11/2020 15:54:47 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=5, batch=500 train mse <loss>=1.10223851643
[11/11/2020 15:54:47 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=5, batch=500 train absolute_loss <loss>=0.830332175462
[11/11/2020 15:54:48 INFO 140334359897920] Iter[5] Batch [1000]#011Speed: 513
75.87 samples/sec
[11/11/2020 15:54:48 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=5, batch=1000 train rmse <loss>=1.07637128847
[11/11/2020 15:54:48 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=5, batch=1000 train mse <loss>=1.15857515064
[11/11/2020 15:54:48 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=5, batch=1000 train absolute_loss <loss>=0.857297107771
[2020-11-11 15:54:49.493] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 12, "duration": 3196, "num_examples": 1274,
"num_bytes": 10436096}
[11/11/2020 15:54:49 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=5, train rmse <loss>=1.08845090798
[11/11/2020 15:54:49 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=5, train mse <loss>=1.18472537908
[11/11/2020 15:54:49 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=5, train absolute_loss <loss>=0.869849221252
#metrics {"Metrics": {"update.time": {"count": 1, "max": 3199.115037918091,
 "sum": 3199.115037918091, "min": 3199.115037918091}}, "EndTime": 1605110089.
494653, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605110086.294318}

[11/11/2020 15:54:49 INFO 140334359897920] #progress_metric: host=algo-1, com
pleted 24 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 1274, "sum": 1274.0, "min": 1274}, "Number of Batches Since Last Reset": {"c
ount": 1, "max": 1274, "sum": 1274.0, "min": 1274}, "Number of Records Since
 Last Reset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "T
otal Batches Seen": {"count": 1, "max": 7645, "sum": 7645.0, "min": 7645}, "T
otal Records Seen": {"count": 1, "max": 978512, "sum": 978512.0, "min": 97851
2}, "Max Records Seen Between Resets": {"count": 1, "max": 163064, "sum": 163
064.0, "min": 163064}, "Reset Count": {"count": 1, "max": 7, "sum": 7.0, "mi
n": 7}}, "EndTime": 1605110089.494976, "Dimensions": {"Host": "algo-1", "Met
a": "training_data_iter", "Operation": "training", "Algorithm": "factorizatio
n-machines", "epoch": 5}, "StartTime": 1605110086.295493}
```

```
[11/11/2020 15:54:49 INFO 140334359897920] #throughput_metric: host=algo-1, t
rain throughput=50963.2327963 records/second
[11/11/2020 15:54:49 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=6, batch=0 train rmse <loss>=0.867150383232
[11/11/2020 15:54:49 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=6, batch=0 train mse <loss>=0.75194978714
[11/11/2020 15:54:49 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=6, batch=0 train absolute_loss <loss>=0.811504602432
[11/11/2020 15:54:50 INFO 140334359897920] Iter[6] Batch [500]#011Speed: 5195
2.98 samples/sec
[11/11/2020 15:54:50 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=6, batch=500 train rmse <loss>=1.03967721476
[11/11/2020 15:54:50 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=6, batch=500 train mse <loss>=1.08092871088
[11/11/2020 15:54:50 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=6, batch=500 train absolute_loss <loss>=0.820884611911
[11/11/2020 15:54:52 INFO 140334359897920] Iter[6] Batch [1000]#011Speed: 496
50.58 samples/sec
[11/11/2020 15:54:52 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=6, batch=1000 train rmse <loss>=1.06619360672
[11/11/2020 15:54:52 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=6, batch=1000 train mse <loss>=1.136768807
[11/11/2020 15:54:52 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=6, batch=1000 train absolute_loss <loss>=0.848224952743
[2020-11-11 15:54:52.682] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 14, "duration": 3184, "num_examples": 1274,
"num_bytes": 10436096}
[11/11/2020 15:54:52 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=6, train rmse <loss>=1.07836110122
[11/11/2020 15:54:52 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=6, train mse <loss>=1.16286266463
[11/11/2020 15:54:52 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=6, train absolute_loss <loss>=0.860927643965
#metrics {"Metrics": {"update.time": {"count": 1, "max": 3187.09397315979, "s
um": 3187.09397315979, "min": 3187.09397315979}}, "EndTime": 1605110092.6830
2, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "fa
ctorization-machines"}, "StartTime": 1605110089.494766}

[11/11/2020 15:54:52 INFO 140334359897920] #progress_metric: host=algo-1, com
pleted 28 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 1274, "sum": 1274.0, "min": 1274}, "Number of Batches Since Last Reset": {"c
ount": 1, "max": 1274, "sum": 1274.0, "min": 1274}, "Number of Records Since
 Last Reset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "T
otal Batches Seen": {"count": 1, "max": 8919, "sum": 8919.0, "min": 8919}, "T
otal Records Seen": {"count": 1, "max": 1141576, "sum": 1141576.0, "min": 114
1576}, "Max Records Seen Between Resets": {"count": 1, "max": 163064, "sum":
 163064.0, "min": 163064}, "Reset Count": {"count": 1, "max": 8, "sum": 8.0,
 "min": 8}}, "EndTime": 1605110092.683339, "Dimensions": {"Host": "algo-1",
 "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factori
zation-machines", "epoch": 6}, "StartTime": 1605110089.495883}

[11/11/2020 15:54:52 INFO 140334359897920] #throughput_metric: host=algo-1, t
rain throughput=51155.7147896 records/second
[11/11/2020 15:54:52 INFO 140334359897920] #quality_metric: host=algo-1, epoc
```

```
h=7, batch=0 train rmse <loss>=0.862029397813
[11/11/2020 15:54:52 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=7, batch=0 train mse <loss>=0.743094682693
[11/11/2020 15:54:52 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=7, batch=0 train absolute_loss <loss>=0.805572330952
[11/11/2020 15:54:53 INFO 140334359897920] Iter[7] Batch [500]#011Speed: 5193
3.72 samples/sec
[11/11/2020 15:54:53 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=7, batch=500 train rmse <loss>=1.03052124527
[11/11/2020 15:54:53 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=7, batch=500 train mse <loss>=1.06197403696
[11/11/2020 15:54:53 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=7, batch=500 train absolute_loss <loss>=0.812373859618
[11/11/2020 15:54:55 INFO 140334359897920] Iter[7] Batch [1000]#011Speed: 496
20.87 samples/sec
[11/11/2020 15:54:55 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=7, batch=1000 train rmse <loss>=1.0570373095
[11/11/2020 15:54:55 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=7, batch=1000 train mse <loss>=1.11732787367
[11/11/2020 15:54:55 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=7, batch=1000 train absolute_loss <loss>=0.839997416937
[2020-11-11 15:54:55.901] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 16, "duration": 3216, "num_examples": 1274,
"num_bytes": 10436096}
[11/11/2020 15:54:55 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=7, train rmse <loss>=1.06926899417
[11/11/2020 15:54:55 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=7, train mse <loss>=1.1433361819
[11/11/2020 15:54:55 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=7, train absolute_loss <loss>=0.852811083057
#metrics {"Metrics": {"update.time": {"count": 1, "max": 3218.402147293091,
 "sum": 3218.402147293091, "min": 3218.402147293091}}, "EndTime": 1605110095.
902647, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605110092.683127}

[11/11/2020 15:54:55 INFO 140334359897920] #progress_metric: host=algo-1, com
pleted 32 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 1274, "sum": 1274.0, "min": 1274}, "Number of Batches Since Last Reset": {"c
ount": 1, "max": 1274, "sum": 1274.0, "min": 1274}, "Number of Records Since
 Last Reset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "T
otal Batches Seen": {"count": 1, "max": 10193, "sum": 10193.0, "min": 10193},
"Total Records Seen": {"count": 1, "max": 1304640, "sum": 1304640.0, "min": 1
304640}, "Max Records Seen Between Resets": {"count": 1, "max": 163064, "su
m": 163064.0, "min": 163064}, "Reset Count": {"count": 1, "max": 9, "sum": 9.
0, "min": 9}}, "EndTime": 1605110095.902967, "Dimensions": {"Host": "algo-1",
"Meta": "training_data_iter", "Operation": "training", "Algorithm": "factoriz
ation-machines", "epoch": 7}, "StartTime": 1605110092.684199}

[11/11/2020 15:54:55 INFO 140334359897920] #throughput_metric: host=algo-1, t
rain throughput=50657.9059899 records/second
[11/11/2020 15:54:55 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=8, batch=0 train rmse <loss>=0.857168312206
[11/11/2020 15:54:55 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=8, batch=0 train mse <loss>=0.73473751545
[11/11/2020 15:54:55 INFO 140334359897920] #quality_metric: host=algo-1, epoc
```

```
h=8, batch=0 train absolute_loss <loss>=0.799885749817
[11/11/2020 15:54:57 INFO 140334359897920] Iter[8] Batch [500]#011Speed: 4941
4.13 samples/sec
[11/11/2020 15:54:57 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=8, batch=500 train rmse <loss>=1.02219512142
[11/11/2020 15:54:57 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=8, batch=500 train mse <loss>=1.04488286626
[11/11/2020 15:54:57 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=8, batch=500 train absolute_loss <loss>=0.804606426619
[11/11/2020 15:54:58 INFO 140334359897920] Iter[8] Batch [1000]#011Speed: 512
50.64 samples/sec
[11/11/2020 15:54:58 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=8, batch=1000 train rmse <loss>=1.0486955256
[11/11/2020 15:54:58 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=8, batch=1000 train mse <loss>=1.09976230542
[11/11/2020 15:54:58 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=8, batch=1000 train absolute_loss <loss>=0.832443417339
[2020-11-11 15:54:59.126] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 18, "duration": 3220, "num_examples": 1274,
"num_bytes": 10436096}
[11/11/2020 15:54:59 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=8, train rmse <loss>=1.06097204129
[11/11/2020 15:54:59 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=8, train mse <loss>=1.1256616724
[11/11/2020 15:54:59 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=8, train absolute_loss <loss>=0.845334018816
#metrics {"Metrics": {"update.time": {"count": 1, "max": 3222.9790687561035,
 "sum": 3222.9790687561035, "min": 3222.9790687561035}}, "EndTime": 160511009
9.126918, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605110095.902774}

[11/11/2020 15:54:59 INFO 140334359897920] #progress_metric: host=algo-1, com
pleted 36 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 1274, "sum": 1274.0, "min": 1274}, "Number of Batches Since Last Reset": {"c
ount": 1, "max": 1274, "sum": 1274.0, "min": 1274}, "Number of Records Since
 Last Reset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "T
otal Batches Seen": {"count": 1, "max": 11467, "sum": 11467.0, "min": 11467},
"Total Records Seen": {"count": 1, "max": 1467704, "sum": 1467704.0, "min": 1
467704}, "Max Records Seen Between Resets": {"count": 1, "max": 163064, "su
m": 163064.0, "min": 163064}, "Reset Count": {"count": 1, "max": 10, "sum": 1
0.0, "min": 10}}, "EndTime": 1605110099.12724, "Dimensions": {"Host": "algo-
1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "fact
orization-machines", "epoch": 8}, "StartTime": 1605110095.903896}

[11/11/2020 15:54:59 INFO 140334359897920] #throughput_metric: host=algo-1, t
rain throughput=50586.0237528 records/second
[11/11/2020 15:54:59 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=9, batch=0 train rmse <loss>=0.852546892862
[11/11/2020 15:54:59 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=9, batch=0 train mse <loss>=0.726836204529
[11/11/2020 15:54:59 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=9, batch=0 train absolute_loss <loss>=0.794544637203
[11/11/2020 15:55:00 INFO 140334359897920] Iter[9] Batch [500]#011Speed: 5216
6.41 samples/sec
[11/11/2020 15:55:00 INFO 140334359897920] #quality_metric: host=algo-1, epoc
```

```
h=9, batch=500 train rmse <loss>=1.0145469405
[11/11/2020 15:55:00 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=9, batch=500 train mse <loss>=1.02930549447
[11/11/2020 15:55:00 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=9, batch=500 train absolute_loss <loss>=0.79744535524
[11/11/2020 15:55:01 INFO 140334359897920] Iter[9] Batch [1000]#011Speed: 491
28.70 samples/sec
[11/11/2020 15:55:01 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=9, batch=1000 train rmse <loss>=1.04102036197
[11/11/2020 15:55:01 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=9, batch=1000 train mse <loss>=1.08372339404
[11/11/2020 15:55:01 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=9, batch=1000 train absolute_loss <loss>=0.825436949254
[2020-11-11 15:55:02.428] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 20, "duration": 3298, "num_examples": 1274,
"num_bytes": 10436096}
[11/11/2020 15:55:02 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=9, train rmse <loss>=1.05332556672
[11/11/2020 15:55:02 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=9, train mse <loss>=1.1094947495
[11/11/2020 15:55:02 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=9, train absolute_loss <loss>=0.838377923943
#metrics {"Metrics": {"update.time": {"count": 1, "max": 3300.9448051452637,
 "sum": 3300.9448051452637, "min": 3300.9448051452637}}, "EndTime": 160511010
2.429139, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605110099.127025}

[11/11/2020 15:55:02 INFO 140334359897920] #progress_metric: host=algo-1, com
pleted 40 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 1274, "sum": 1274.0, "min": 1274}, "Number of Batches Since Last Reset": {"c
ount": 1, "max": 1274, "sum": 1274.0, "min": 1274}, "Number of Records Since
 Last Reset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "T
otal Batches Seen": {"count": 1, "max": 12741, "sum": 12741.0, "min": 12741},
"Total Records Seen": {"count": 1, "max": 1630768, "sum": 1630768.0, "min": 1
630768}, "Max Records Seen Between Resets": {"count": 1, "max": 163064, "su
m": 163064.0, "min": 163064}, "Reset Count": {"count": 1, "max": 11, "sum": 1
1.0, "min": 11}}, "EndTime": 1605110102.429411, "Dimensions": {"Host": "algo-
1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "fact
orization-machines", "epoch": 9}, "StartTime": 1605110099.128154}

[11/11/2020 15:55:02 INFO 140334359897920] #throughput_metric: host=algo-1, t
rain throughput=49392.5117744 records/second
[11/11/2020 15:55:02 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=10, batch=0 train rmse <loss>=0.848141228247
[11/11/2020 15:55:02 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=10, batch=0 train mse <loss>=0.719343543053
[11/11/2020 15:55:02 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=10, batch=0 train absolute_loss <loss>=0.789300560951
[11/11/2020 15:55:03 INFO 140334359897920] Iter[10] Batch [500]#011Speed: 518
43.61 samples/sec
[11/11/2020 15:55:03 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=10, batch=500 train rmse <loss>=1.00746450847
[11/11/2020 15:55:03 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=10, batch=500 train mse <loss>=1.01498473583
[11/11/2020 15:55:03 INFO 140334359897920] #quality_metric: host=algo-1, epoc
```

```
h=10, batch=500 train absolute_loss <loss>=0.790778390424
[11/11/2020 15:55:04 INFO 140334359897920] Iter[10] Batch [1000]#011Speed: 52
424.21 samples/sec
[11/11/2020 15:55:04 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=10, batch=1000 train rmse <loss>=1.03390250878
[11/11/2020 15:55:04 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=10, batch=1000 train mse <loss>=1.06895439766
[11/11/2020 15:55:04 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=10, batch=1000 train absolute_loss <loss>=0.818888032174
[2020-11-11 15:55:05.577] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 22, "duration": 3145, "num_examples": 1274,
"num_bytes": 10436096}
[11/11/2020 15:55:05 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=10, train rmse <loss>=1.04622266377
[11/11/2020 15:55:05 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=10, train mse <loss>=1.09458186219
[11/11/2020 15:55:05 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=10, train absolute_loss <loss>=0.831858618948
#metrics {"Metrics": {"update.time": {"count": 1, "max": 3147.8540897369385,
 "sum": 3147.8540897369385, "min": 3147.8540897369385}}, "EndTime": 160511010
5.578197, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605110102.429245}

[11/11/2020 15:55:05 INFO 140334359897920] #progress_metric: host=algo-1, com
pleted 44 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 1274, "sum": 1274.0, "min": 1274}, "Number of Batches Since Last Reset": {"c
ount": 1, "max": 1274, "sum": 1274.0, "min": 1274}, "Number of Records Since
 Last Reset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "T
otal Batches Seen": {"count": 1, "max": 14015, "sum": 14015.0, "min": 14015},
"Total Records Seen": {"count": 1, "max": 1793832, "sum": 1793832.0, "min": 1
793832}, "Max Records Seen Between Resets": {"count": 1, "max": 163064, "su
m": 163064.0, "min": 163064}, "Reset Count": {"count": 1, "max": 12, "sum": 1
2.0, "min": 12}}, "EndTime": 1605110105.578504, "Dimensions": {"Host": "algo-
1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "fact
orization-machines", "epoch": 10}, "StartTime": 1605110102.430296}

[11/11/2020 15:55:05 INFO 140334359897920] #throughput_metric: host=algo-1, t
rain throughput=51793.3300737 records/second
[11/11/2020 15:55:05 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=11, batch=0 train rmse <loss>=0.843928071058
[11/11/2020 15:55:05 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=11, batch=0 train mse <loss>=0.712214589119
[11/11/2020 15:55:05 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=11, batch=0 train absolute_loss <loss>=0.784216105938
[11/11/2020 15:55:06 INFO 140334359897920] Iter[11] Batch [500]#011Speed: 518
29.92 samples/sec
[11/11/2020 15:55:06 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=11, batch=500 train rmse <loss>=1.00086225458
[11/11/2020 15:55:06 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=11, batch=500 train mse <loss>=1.00172525264
[11/11/2020 15:55:06 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=11, batch=500 train absolute_loss <loss>=0.78453022307
[11/11/2020 15:55:08 INFO 140334359897920] Iter[11] Batch [1000]#011Speed: 49
552.78 samples/sec
[11/11/2020 15:55:08 INFO 140334359897920] #quality_metric: host=algo-1, epoc
```

```
h=11, batch=1000 train rmse <loss>=1.02725847573
[11/11/2020 15:55:08 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=11, batch=1000 train mse <loss>=1.05525997597
[11/11/2020 15:55:08 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=11, batch=1000 train absolute_loss <loss>=0.812728731782
[2020-11-11 15:55:08.781] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 24, "duration": 3200, "num_examples": 1274,
"num_bytes": 10436096}
[11/11/2020 15:55:08 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=11, train rmse <loss>=1.03958172532
[11/11/2020 15:55:08 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=11, train mse <loss>=1.08073016363
[11/11/2020 15:55:08 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=11, train absolute_loss <loss>=0.825711312062
#metrics {"Metrics": {"update.time": {"count": 1, "max": 3202.6309967041016,
 "sum": 3202.6309967041016, "min": 3202.6309967041016}}, "EndTime": 160511010
8.78211, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605110105.578292}

[11/11/2020 15:55:08 INFO 140334359897920] #progress_metric: host=algo-1, com
pleted 48 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 1274, "sum": 1274.0, "min": 1274}, "Number of Batches Since Last Reset": {"c
ount": 1, "max": 1274, "sum": 1274.0, "min": 1274}, "Number of Records Since
 Last Reset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "T
otal Batches Seen": {"count": 1, "max": 15289, "sum": 15289.0, "min": 15289},
"Total Records Seen": {"count": 1, "max": 1956896, "sum": 1956896.0, "min": 1
956896}, "Max Records Seen Between Resets": {"count": 1, "max": 163064, "su
m": 163064.0, "min": 163064}, "Reset Count": {"count": 1, "max": 13, "sum": 1
3.0, "min": 13}}, "EndTime": 1605110108.782398, "Dimensions": {"Host": "algo-
1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "fact
orization-machines", "epoch": 11}, "StartTime": 1605110105.579437}

[11/11/2020 15:55:08 INFO 140334359897920] #throughput_metric: host=algo-1, t
rain throughput=50908.1983947 records/second
[11/11/2020 15:55:08 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=12, batch=0 train rmse <loss>=0.839884273527
[11/11/2020 15:55:08 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=12, batch=0 train mse <loss>=0.705405592918
[11/11/2020 15:55:08 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=12, batch=0 train absolute_loss <loss>=0.779221594334
[11/11/2020 15:55:10 INFO 140334359897920] Iter[12] Batch [500]#011Speed: 526
23.92 samples/sec
[11/11/2020 15:55:10 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=12, batch=500 train rmse <loss>=0.994673017975
[11/11/2020 15:55:10 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=12, batch=500 train mse <loss>=0.989374412688
[11/11/2020 15:55:10 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=12, batch=500 train absolute_loss <loss>=0.778649260422
[11/11/2020 15:55:11 INFO 140334359897920] Iter[12] Batch [1000]#011Speed: 52
529.03 samples/sec
[11/11/2020 15:55:11 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=12, batch=1000 train rmse <loss>=1.02102272528
[11/11/2020 15:55:11 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=12, batch=1000 train mse <loss>=1.04248740555
[11/11/2020 15:55:11 INFO 140334359897920] #quality_metric: host=algo-1, epoc
```

```
h=12, batch=1000 train absolute_loss <loss>=0.8069068944
[2020-11-11 15:55:11.897] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 26, "duration": 3112, "num_examples": 1274,
"num_bytes": 10436096}
[11/11/2020 15:55:11 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=12, train rmse <loss>=1.03333875275
[11/11/2020 15:55:11 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=12, train mse <loss>=1.06778897794
[11/11/2020 15:55:11 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=12, train absolute_loss <loss>=0.819887958642
#metrics {"Metrics": {"update.time": {"count": 1, "max": 3115.0639057159424,
 "sum": 3115.0639057159424, "min": 3115.0639057159424}}, "EndTime": 160511011
1.898444, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605110108.782217}

[11/11/2020 15:55:11 INFO 140334359897920] #progress_metric: host=algo-1, com
pleted 52 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 1274, "sum": 1274.0, "min": 1274}, "Number of Batches Since Last Reset": {"c
ount": 1, "max": 1274, "sum": 1274.0, "min": 1274}, "Number of Records Since
 Last Reset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "T
otal Batches Seen": {"count": 1, "max": 16563, "sum": 16563.0, "min": 16563},
"Total Records Seen": {"count": 1, "max": 2119960, "sum": 2119960.0, "min": 2
119960}, "Max Records Seen Between Resets": {"count": 1, "max": 163064, "su
m": 163064.0, "min": 163064}, "Reset Count": {"count": 1, "max": 14, "sum": 1
4.0, "min": 14}}, "EndTime": 1605110111.898768, "Dimensions": {"Host": "algo-
1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "fact
orization-machines", "epoch": 12}, "StartTime": 1605110108.783334}

[11/11/2020 15:55:11 INFO 140334359897920] #throughput_metric: host=algo-1, t
rain throughput=52337.9106079 records/second
[11/11/2020 15:55:11 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=13, batch=0 train rmse <loss>=0.835986127063
[11/11/2020 15:55:11 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=13, batch=0 train mse <loss>=0.698872804642
[11/11/2020 15:55:11 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=13, batch=0 train absolute_loss <loss>=0.774308860302
[11/11/2020 15:55:13 INFO 140334359897920] Iter[13] Batch [500]#011Speed: 501
49.43 samples/sec
[11/11/2020 15:55:13 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=13, batch=500 train rmse <loss>=0.988842844929
[11/11/2020 15:55:13 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=13, batch=500 train mse <loss>=0.977810171966
[11/11/2020 15:55:13 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=13, batch=500 train absolute_loss <loss>=0.773086047577
[11/11/2020 15:55:14 INFO 140334359897920] Iter[13] Batch [1000]#011Speed: 53
060.64 samples/sec
[11/11/2020 15:55:14 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=13, batch=1000 train rmse <loss>=1.0151425465
[11/11/2020 15:55:14 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=13, batch=1000 train mse <loss>=1.03051438972
[11/11/2020 15:55:14 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=13, batch=1000 train absolute_loss <loss>=0.801379469993
[2020-11-11 15:55:15.037] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 28, "duration": 3135, "num_examples": 1274,
"num_bytes": 10436096}
```

```
[11/11/2020 15:55:15 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=13, train rmse <loss>=1.02744233617
[11/11/2020 15:55:15 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=13, train mse <loss>=1.05563775416
[11/11/2020 15:55:15 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=13, train absolute_loss <loss>=0.814348729219
#metrics {"Metrics": {"update.time": {"count": 1, "max": 3138.579845428467,
 "sum": 3138.579845428467, "min": 3138.579845428467}}, "EndTime": 1605110115.
038359, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605110111.898551}

[11/11/2020 15:55:15 INFO 140334359897920] #progress_metric: host=algo-1, com
pleted 56 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 1274, "sum": 1274.0, "min": 1274}, "Number of Batches Since Last Reset": {"c
ount": 1, "max": 1274, "sum": 1274.0, "min": 1274}, "Number of Records Since
 Last Reset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "T
otal Batches Seen": {"count": 1, "max": 17837, "sum": 17837.0, "min": 17837},
"Total Records Seen": {"count": 1, "max": 2283024, "sum": 2283024.0, "min": 2
283024}, "Max Records Seen Between Resets": {"count": 1, "max": 163064, "su
m": 163064.0, "min": 163064}, "Reset Count": {"count": 1, "max": 15, "sum": 1
5.0, "min": 15}}, "EndTime": 1605110115.038625, "Dimensions": {"Host": "algo-
1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "fact
orization-machines", "epoch": 13}, "StartTime": 1605110111.899739}

[11/11/2020 15:55:15 INFO 140334359897920] #throughput_metric: host=algo-1, t
rain throughput=51947.439424 records/second
[11/11/2020 15:55:15 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=14, batch=0 train rmse <loss>=0.832213046133
[11/11/2020 15:55:15 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=14, batch=0 train mse <loss>=0.692578554153
[11/11/2020 15:55:15 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=14, batch=0 train absolute_loss <loss>=0.769470572472
[11/11/2020 15:55:16 INFO 140334359897920] Iter[14] Batch [500]#011Speed: 519
84.42 samples/sec
[11/11/2020 15:55:16 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=14, batch=500 train rmse <loss>=0.98332766363
[11/11/2020 15:55:16 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=14, batch=500 train mse <loss>=0.96693329406
[11/11/2020 15:55:16 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=14, batch=500 train absolute_loss <loss>=0.767795609857
[11/11/2020 15:55:17 INFO 140334359897920] Iter[14] Batch [1000]#011Speed: 50
141.80 samples/sec
[11/11/2020 15:55:17 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=14, batch=1000 train rmse <loss>=1.0095748347
[11/11/2020 15:55:17 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=14, batch=1000 train mse <loss>=1.01924134687
[11/11/2020 15:55:17 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=14, batch=1000 train absolute_loss <loss>=0.796111279762
[2020-11-11 15:55:18.239] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 30, "duration": 3197, "num_examples": 1274,
"num_bytes": 10436096}
[11/11/2020 15:55:18 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=14, train rmse <loss>=1.02185047585
[11/11/2020 15:55:18 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=14, train mse <loss>=1.044178395
```

```
[11/11/2020 15:55:18 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=14, train absolute_loss <loss>=0.809059959496
#metrics {"Metrics": {"update.time": {"count": 1, "max": 3200.3979682922363,
 "sum": 3200.3979682922363, "min": 3200.3979682922363}}, "EndTime": 160511011
8.240026, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605110115.03846}

[11/11/2020 15:55:18 INFO 140334359897920] #progress_metric: host=algo-1, com
pleted 60 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 1274, "sum": 1274.0, "min": 1274}, "Number of Batches Since Last Reset": {"c
ount": 1, "max": 1274, "sum": 1274.0, "min": 1274}, "Number of Records Since
 Last Reset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "T
otal Batches Seen": {"count": 1, "max": 19111, "sum": 19111.0, "min": 19111},
"Total Records Seen": {"count": 1, "max": 2446088, "sum": 2446088.0, "min": 2
446088}, "Max Records Seen Between Resets": {"count": 1, "max": 163064, "su
m": 163064.0, "min": 163064}, "Reset Count": {"count": 1, "max": 16, "sum": 1
6.0, "min": 16}}, "EndTime": 1605110118.240323, "Dimensions": {"Host": "algo-
1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "fact
orization-machines", "epoch": 14}, "StartTime": 1605110115.039582}

[11/11/2020 15:55:18 INFO 140334359897920] #throughput_metric: host=algo-1, t
rain throughput=50942.8219032 records/second
[11/11/2020 15:55:18 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=15, batch=0 train rmse <loss>=0.828544762228
[11/11/2020 15:55:18 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=15, batch=0 train mse <loss>=0.686486423016
[11/11/2020 15:55:18 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=15, batch=0 train absolute_loss <loss>=0.764696776867
[11/11/2020 15:55:19 INFO 140334359897920] Iter[15] Batch [500]#011Speed: 530
47.10 samples/sec
[11/11/2020 15:55:19 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=15, batch=500 train rmse <loss>=0.978090751254
[11/11/2020 15:55:19 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=15, batch=500 train mse <loss>=0.956661517689
[11/11/2020 15:55:19 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=15, batch=500 train absolute_loss <loss>=0.762760946851
[11/11/2020 15:55:20 INFO 140334359897920] Iter[15] Batch [1000]#011Speed: 53
137.22 samples/sec
[11/11/2020 15:55:20 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=15, batch=1000 train rmse <loss>=1.00428366054
[11/11/2020 15:55:20 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=15, batch=1000 train mse <loss>=1.00858567082
[11/11/2020 15:55:20 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=15, batch=1000 train absolute_loss <loss>=0.791077592394
[2020-11-11 15:55:21.339] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 32, "duration": 3095, "num_examples": 1274,
"num_bytes": 10436096}
[11/11/2020 15:55:21 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=15, train rmse <loss>=1.01652823122
[11/11/2020 15:55:21 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=15, train mse <loss>=1.03332964486
[11/11/2020 15:55:21 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=15, train absolute_loss <loss>=0.803999086938
#metrics {"Metrics": {"update.time": {"count": 1, "max": 3098.510980606079,
 "sum": 3098.510980606079, "min": 3098.510980606079}}, "EndTime": 1605110121.
```

```
339856, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605110118.240121}

[11/11/2020 15:55:21 INFO 140334359897920] #progress_metric: host=algo-1, com
pleted 64 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 1274, "sum": 1274.0, "min": 1274}, "Number of Batches Since Last Reset": {"c
ount": 1, "max": 1274, "sum": 1274.0, "min": 1274}, "Number of Records Since
 Last Reset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "T
otal Batches Seen": {"count": 1, "max": 20385, "sum": 20385.0, "min": 20385},
"Total Records Seen": {"count": 1, "max": 2609152, "sum": 2609152.0, "min": 2
609152}, "Max Records Seen Between Resets": {"count": 1, "max": 163064, "su
m": 163064.0, "min": 163064}, "Reset Count": {"count": 1, "max": 17, "sum": 1
7.0, "min": 17}}, "EndTime": 1605110121.340152, "Dimensions": {"Host": "algo-
1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "fact
orization-machines", "epoch": 15}, "StartTime": 1605110118.241304}

[11/11/2020 15:55:21 INFO 140334359897920] #throughput_metric: host=algo-1, t
rain throughput=52618.2481522 records/second
[11/11/2020 15:55:21 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=16, batch=0 train rmse <loss>=0.824964863578
[11/11/2020 15:55:21 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=16, batch=0 train mse <loss>=0.680567026138
[11/11/2020 15:55:21 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=16, batch=0 train absolute_loss <loss>=0.759979665279
[11/11/2020 15:55:22 INFO 140334359897920] Iter[16] Batch [500]#011Speed: 507
55.26 samples/sec
[11/11/2020 15:55:22 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=16, batch=500 train rmse <loss>=0.973101155179
[11/11/2020 15:55:22 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=16, batch=500 train mse <loss>=0.94692585821
[11/11/2020 15:55:22 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=16, batch=500 train absolute_loss <loss>=0.75795083322
[11/11/2020 15:55:23 INFO 140334359897920] Iter[16] Batch [1000]#011Speed: 50
811.31 samples/sec
[11/11/2020 15:55:23 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=16, batch=1000 train rmse <loss>=0.999238763312
[11/11/2020 15:55:23 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=16, batch=1000 train mse <loss>=0.998478106105
[11/11/2020 15:55:23 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=16, batch=1000 train absolute_loss <loss>=0.786253587468
[2020-11-11 15:55:24.544] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 34, "duration": 3201, "num_examples": 1274,
"num_bytes": 10436096}
[11/11/2020 15:55:24 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=16, train rmse <loss>=1.01144620105
[11/11/2020 15:55:24 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=16, train mse <loss>=1.02302341761
[11/11/2020 15:55:24 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=16, train absolute_loss <loss>=0.799141140944
#metrics {"Metrics": {"update.time": {"count": 1, "max": 3203.881025314331,
 "sum": 3203.881025314331, "min": 3203.881025314331}}, "EndTime": 1605110124.
545036, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605110121.339961}

[11/11/2020 15:55:24 INFO 140334359897920] #progress_metric: host=algo-1, com
```

```
pleted 68 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 1274, "sum": 1274.0, "min": 1274}, "Number of Batches Since Last Reset": {"c
ount": 1, "max": 1274, "sum": 1274.0, "min": 1274}, "Number of Records Since
 Last Reset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "T
otal Batches Seen": {"count": 1, "max": 21659, "sum": 21659.0, "min": 21659},
"Total Records Seen": {"count": 1, "max": 2772216, "sum": 2772216.0, "min": 2
772216}, "Max Records Seen Between Resets": {"count": 1, "max": 163064, "su
m": 163064.0, "min": 163064}, "Reset Count": {"count": 1, "max": 18, "sum": 1
8.0, "min": 18}}, "EndTime": 1605110124.54528, "Dimensions": {"Host": "algo-
1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "fact
orization-machines", "epoch": 16}, "StartTime": 1605110121.341115}

[11/11/2020 15:55:24 INFO 140334359897920] #throughput_metric: host=algo-1, t
rain throughput=50889.2438643 records/second
[11/11/2020 15:55:24 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=17, batch=0 train rmse <loss>=0.821457164131
[11/11/2020 15:55:24 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=17, batch=0 train mse <loss>=0.674791872501
[11/11/2020 15:55:24 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=17, batch=0 train absolute_loss <loss>=0.755309641361
[11/11/2020 15:55:25 INFO 140334359897920] Iter[17] Batch [500]#011Speed: 503
58.06 samples/sec
[11/11/2020 15:55:25 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=17, batch=500 train rmse <loss>=0.968332529657
[11/11/2020 15:55:25 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=17, batch=500 train mse <loss>=0.937667887992
[11/11/2020 15:55:25 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=17, batch=500 train absolute_loss <loss>=0.753338944472
[11/11/2020 15:55:27 INFO 140334359897920] Iter[17] Batch [1000]#011Speed: 51
635.73 samples/sec
[11/11/2020 15:55:27 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=17, batch=1000 train rmse <loss>=0.994414303195
[11/11/2020 15:55:27 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=17, batch=1000 train mse <loss>=0.9888598064
[11/11/2020 15:55:27 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=17, batch=1000 train absolute_loss <loss>=0.781619698226
[2020-11-11 15:55:27.789] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 36, "duration": 3241, "num_examples": 1274,
"num_bytes": 10436096}
[11/11/2020 15:55:27 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=17, train rmse <loss>=1.0065793172
[11/11/2020 15:55:27 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=17, train mse <loss>=1.01320192182
[11/11/2020 15:55:27 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=17, train absolute_loss <loss>=0.794467437702
#metrics {"Metrics": {"update.time": {"count": 1, "max": 3243.872880935669,
 "sum": 3243.872880935669, "min": 3243.872880935669}}, "EndTime": 1605110127.
790187, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605110124.545122}

[11/11/2020 15:55:27 INFO 140334359897920] #progress_metric: host=algo-1, com
pleted 72 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 1274, "sum": 1274.0, "min": 1274}, "Number of Batches Since Last Reset": {"c
ount": 1, "max": 1274, "sum": 1274.0, "min": 1274}, "Number of Records Since
```

```
 Last Reset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "T
otal Batches Seen": {"count": 1, "max": 22933, "sum": 22933.0, "min": 22933},
"Total Records Seen": {"count": 1, "max": 2935280, "sum": 2935280.0, "min": 2
935280}, "Max Records Seen Between Resets": {"count": 1, "max": 163064, "su
m": 163064.0, "min": 163064}, "Reset Count": {"count": 1, "max": 19, "sum": 1
9.0, "min": 19}}, "EndTime": 1605110127.790445, "Dimensions": {"Host": "algo-
1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "fact
orization-machines", "epoch": 17}, "StartTime": 1605110124.546277}

[11/11/2020 15:55:27 INFO 140334359897920] #throughput_metric: host=algo-1, t
rain throughput=50261.4883083 records/second
[11/11/2020 15:55:27 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=18, batch=0 train rmse <loss>=0.81800998704
[11/11/2020 15:55:27 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=18, batch=0 train mse <loss>=0.669140338898
[11/11/2020 15:55:27 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=18, batch=0 train absolute_loss <loss>=0.750680863857
[11/11/2020 15:55:29 INFO 140334359897920] Iter[18] Batch [500]#011Speed: 517
34.94 samples/sec
[11/11/2020 15:55:29 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=18, batch=500 train rmse <loss>=0.963762202982
[11/11/2020 15:55:29 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=18, batch=500 train mse <loss>=0.928837583897
[11/11/2020 15:55:29 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=18, batch=500 train absolute_loss <loss>=0.748905224893
[11/11/2020 15:55:30 INFO 140334359897920] Iter[18] Batch [1000]#011Speed: 51
602.00 samples/sec
[11/11/2020 15:55:30 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=18, batch=1000 train rmse <loss>=0.989788080269
[11/11/2020 15:55:30 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=18, batch=1000 train mse <loss>=0.979680443843
[11/11/2020 15:55:30 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=18, batch=1000 train absolute_loss <loss>=0.777158362763
[2020-11-11 15:55:30.946] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 38, "duration": 3153, "num_examples": 1274,
"num_bytes": 10436096}
[11/11/2020 15:55:30 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=18, train rmse <loss>=1.0019060603
[11/11/2020 15:55:30 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=18, train mse <loss>=1.00381575367
[11/11/2020 15:55:30 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=18, train absolute_loss <loss>=0.78996197236
#metrics {"Metrics": {"update.time": {"count": 1, "max": 3155.890941619873,
 "sum": 3155.890941619873, "min": 3155.890941619873}}, "EndTime": 1605110130.
947303, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605110127.790255}

[11/11/2020 15:55:30 INFO 140334359897920] #progress_metric: host=algo-1, com
pleted 76 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 1274, "sum": 1274.0, "min": 1274}, "Number of Batches Since Last Reset": {"c
ount": 1, "max": 1274, "sum": 1274.0, "min": 1274}, "Number of Records Since
 Last Reset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "T
otal Batches Seen": {"count": 1, "max": 24207, "sum": 24207.0, "min": 24207},
"Total Records Seen": {"count": 1, "max": 3098344, "sum": 3098344.0, "min": 3
098344}, "Max Records Seen Between Resets": {"count": 1, "max": 163064, "su
```

```
m": 163064.0, "min": 163064}, "Reset Count": {"count": 1, "max": 20, "sum": 2
0.0, "min": 20}}, "EndTime": 1605110130.947635, "Dimensions": {"Host": "algo-
1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "fact
orization-machines", "epoch": 18}, "StartTime": 1605110127.791368}

[11/11/2020 15:55:30 INFO 140334359897920] #throughput_metric: host=algo-1, t
rain throughput=51660.9928344 records/second
[11/11/2020 15:55:30 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=19, batch=0 train rmse <loss>=0.814612796052
[11/11/2020 15:55:30 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=19, batch=0 train mse <loss>=0.663594007492
[11/11/2020 15:55:30 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=19, batch=0 train absolute_loss <loss>=0.746087372303
[11/11/2020 15:55:32 INFO 140334359897920] Iter[19] Batch [500]#011Speed: 530
90.44 samples/sec
[11/11/2020 15:55:32 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=19, batch=500 train rmse <loss>=0.959370523402
[11/11/2020 15:55:32 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=19, batch=500 train mse <loss>=0.920391801172
[11/11/2020 15:55:32 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=19, batch=500 train absolute_loss <loss>=0.744637038655
[11/11/2020 15:55:33 INFO 140334359897920] Iter[19] Batch [1000]#011Speed: 50
496.76 samples/sec
[11/11/2020 15:55:33 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=19, batch=1000 train rmse <loss>=0.985340836462
[11/11/2020 15:55:33 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=19, batch=1000 train mse <loss>=0.970896564
[11/11/2020 15:55:33 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=19, batch=1000 train absolute_loss <loss>=0.772858306155
[2020-11-11 15:55:34.110] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 40, "duration": 3160, "num_examples": 1274,
"num_bytes": 10436096}
[11/11/2020 15:55:34 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=19, train rmse <loss>=0.997407773285
[11/11/2020 15:55:34 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=19, train mse <loss>=0.994822266209
[11/11/2020 15:55:34 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=19, train absolute_loss <loss>=0.785613738082
#metrics {"Metrics": {"update.time": {"count": 1, "max": 3162.7840995788574,
 "sum": 3162.7840995788574, "min": 3162.7840995788574}}, "EndTime": 160511013
4.111403, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605110130.947411}

[11/11/2020 15:55:34 INFO 140334359897920] #progress_metric: host=algo-1, com
pleted 80 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 1274, "sum": 1274.0, "min": 1274}, "Number of Batches Since Last Reset": {"c
ount": 1, "max": 1274, "sum": 1274.0, "min": 1274}, "Number of Records Since
 Last Reset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "T
otal Batches Seen": {"count": 1, "max": 25481, "sum": 25481.0, "min": 25481},
"Total Records Seen": {"count": 1, "max": 3261408, "sum": 3261408.0, "min": 3
261408}, "Max Records Seen Between Resets": {"count": 1, "max": 163064, "su
m": 163064.0, "min": 163064}, "Reset Count": {"count": 1, "max": 21, "sum": 2
1.0, "min": 21}}, "EndTime": 1605110134.111654, "Dimensions": {"Host": "algo-
1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "fact
orization-machines", "epoch": 19}, "StartTime": 1605110130.948579}
```

```
[11/11/2020 15:55:34 INFO 140334359897920] #throughput_metric: host=algo-1, t
rain throughput=51550.0506317 records/second
[11/11/2020 15:55:34 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=20, batch=0 train rmse <loss>=0.811257292328
[11/11/2020 15:55:34 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=20, batch=0 train mse <loss>=0.658138394356
[11/11/2020 15:55:34 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=20, batch=0 train absolute_loss <loss>=0.741524815559
[11/11/2020 15:55:35 INFO 140334359897920] Iter[20] Batch [500]#011Speed: 507
45.92 samples/sec
[11/11/2020 15:55:35 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=20, batch=500 train rmse <loss>=0.955140419585
[11/11/2020 15:55:35 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=20, batch=500 train mse <loss>=0.912293221125
[11/11/2020 15:55:35 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=20, batch=500 train absolute_loss <loss>=0.740522784863
[11/11/2020 15:55:36 INFO 140334359897920] Iter[20] Batch [1000]#011Speed: 52
875.73 samples/sec
[11/11/2020 15:55:36 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=20, batch=1000 train rmse <loss>=0.981055789817
[11/11/2020 15:55:36 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=20, batch=1000 train mse <loss>=0.962470462734
[11/11/2020 15:55:36 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=20, batch=1000 train absolute_loss <loss>=0.768709583001
[2020-11-11 15:55:37.258] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 42, "duration": 3143, "num_examples": 1274,
"num_bytes": 10436096}
[11/11/2020 15:55:37 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=20, train rmse <loss>=0.993068205518
[11/11/2020 15:55:37 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=20, train mse <loss>=0.986184460811
[11/11/2020 15:55:37 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=20, train absolute_loss <loss>=0.781413494324
#metrics {"Metrics": {"update.time": {"count": 1, "max": 3146.1288928985596,
 "sum": 3146.1288928985596, "min": 3146.1288928985596}}, "EndTime": 160511013
7.258917, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605110134.111494}

[11/11/2020 15:55:37 INFO 140334359897920] #progress_metric: host=algo-1, com
pleted 84 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 1274, "sum": 1274.0, "min": 1274}, "Number of Batches Since Last Reset": {"c
ount": 1, "max": 1274, "sum": 1274.0, "min": 1274}, "Number of Records Since
 Last Reset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "T
otal Batches Seen": {"count": 1, "max": 26755, "sum": 26755.0, "min": 26755},
"Total Records Seen": {"count": 1, "max": 3424472, "sum": 3424472.0, "min": 3
424472}, "Max Records Seen Between Resets": {"count": 1, "max": 163064, "su
m": 163064.0, "min": 163064}, "Reset Count": {"count": 1, "max": 22, "sum": 2
2.0, "min": 22}}, "EndTime": 1605110137.259204, "Dimensions": {"Host": "algo-
1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "fact
orization-machines", "epoch": 20}, "StartTime": 1605110134.112664}

[11/11/2020 15:55:37 INFO 140334359897920] #throughput_metric: host=algo-1, t
rain throughput=51820.6704619 records/second
[11/11/2020 15:55:37 INFO 140334359897920] #quality_metric: host=algo-1, epoc
```

```
h=21, batch=0 train rmse <loss>=0.807937429423
[11/11/2020 15:55:37 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=21, batch=0 train mse <loss>=0.652762889862
[11/11/2020 15:55:37 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=21, batch=0 train absolute_loss <loss>=0.736990809441
[11/11/2020 15:55:38 INFO 140334359897920] Iter[21] Batch [500]#011Speed: 481
30.94 samples/sec
[11/11/2020 15:55:38 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=21, batch=500 train rmse <loss>=0.951056862351
[11/11/2020 15:55:38 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=21, batch=500 train mse <loss>=0.904509155425
[11/11/2020 15:55:38 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=21, batch=500 train absolute_loss <loss>=0.736552469685
[11/11/2020 15:55:39 INFO 140334359897920] Iter[21] Batch [1000]#011Speed: 52
809.16 samples/sec
[11/11/2020 15:55:39 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=21, batch=1000 train rmse <loss>=0.976918186857
[11/11/2020 15:55:39 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=21, batch=1000 train mse <loss>=0.954369143813
[11/11/2020 15:55:39 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=21, batch=1000 train absolute_loss <loss>=0.764698756027
[2020-11-11 15:55:40.475] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 44, "duration": 3213, "num_examples": 1274,
"num_bytes": 10436096}
[11/11/2020 15:55:40 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=21, train rmse <loss>=0.98887307381
[11/11/2020 15:55:40 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=21, train mse <loss>=0.977869956106
[11/11/2020 15:55:40 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=21, train absolute_loss <loss>=0.777347897435
#metrics {"Metrics": {"update.time": {"count": 1, "max": 3215.7931327819824,
 "sum": 3215.7931327819824, "min": 3215.7931327819824}}, "EndTime": 160511014
0.475994, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605110137.259022}

[11/11/2020 15:55:40 INFO 140334359897920] #progress_metric: host=algo-1, com
pleted 88 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 1274, "sum": 1274.0, "min": 1274}, "Number of Batches Since Last Reset": {"c
ount": 1, "max": 1274, "sum": 1274.0, "min": 1274}, "Number of Records Since
 Last Reset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "T
otal Batches Seen": {"count": 1, "max": 28029, "sum": 28029.0, "min": 28029},
"Total Records Seen": {"count": 1, "max": 3587536, "sum": 3587536.0, "min": 3
587536}, "Max Records Seen Between Resets": {"count": 1, "max": 163064, "su
m": 163064.0, "min": 163064}, "Reset Count": {"count": 1, "max": 23, "sum": 2
3.0, "min": 23}}, "EndTime": 1605110140.476246, "Dimensions": {"Host": "algo-
1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "fact
orization-machines", "epoch": 21}, "StartTime": 1605110137.260162}

[11/11/2020 15:55:40 INFO 140334359897920] #throughput_metric: host=algo-1, t
rain throughput=50700.4644656 records/second
[11/11/2020 15:55:40 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=22, batch=0 train rmse <loss>=0.804648648407
[11/11/2020 15:55:40 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=22, batch=0 train mse <loss>=0.647459447384
[11/11/2020 15:55:40 INFO 140334359897920] #quality_metric: host=algo-1, epoc
```

```
h=22, batch=0 train absolute_loss <loss>=0.732483625412
[11/11/2020 15:55:41 INFO 140334359897920] Iter[22] Batch [500]#011Speed: 516
86.06 samples/sec
[11/11/2020 15:55:41 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=22, batch=500 train rmse <loss>=0.947106693783
[11/11/2020 15:55:41 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=22, batch=500 train mse <loss>=0.897011089408
[11/11/2020 15:55:41 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=22, batch=500 train absolute_loss <loss>=0.732712092811
[11/11/2020 15:55:42 INFO 140334359897920] Iter[22] Batch [1000]#011Speed: 51
327.17 samples/sec
[11/11/2020 15:55:42 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=22, batch=1000 train rmse <loss>=0.97291508061
[11/11/2020 15:55:42 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=22, batch=1000 train mse <loss>=0.946563754078
[11/11/2020 15:55:42 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=22, batch=1000 train absolute_loss <loss>=0.760814901058
[2020-11-11 15:55:43.621] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 46, "duration": 3142, "num_examples": 1274,
"num_bytes": 10436096}
[11/11/2020 15:55:43 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=22, train rmse <loss>=0.984809831397
[11/11/2020 15:55:43 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=22, train mse <loss>=0.969850404016
[11/11/2020 15:55:43 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=22, train absolute_loss <loss>=0.773406506099
#metrics {"Metrics": {"update.time": {"count": 1, "max": 3145.188093185425,
 "sum": 3145.188093185425, "min": 3145.188093185425}}, "EndTime": 1605110143.
622464, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605110140.476082}

[11/11/2020 15:55:43 INFO 140334359897920] #progress_metric: host=algo-1, com
pleted 92 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 1274, "sum": 1274.0, "min": 1274}, "Number of Batches Since Last Reset": {"c
ount": 1, "max": 1274, "sum": 1274.0, "min": 1274}, "Number of Records Since
 Last Reset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "T
otal Batches Seen": {"count": 1, "max": 29303, "sum": 29303.0, "min": 29303},
"Total Records Seen": {"count": 1, "max": 3750600, "sum": 3750600.0, "min": 3
750600}, "Max Records Seen Between Resets": {"count": 1, "max": 163064, "su
m": 163064.0, "min": 163064}, "Reset Count": {"count": 1, "max": 24, "sum": 2
4.0, "min": 24}}, "EndTime": 1605110143.622772, "Dimensions": {"Host": "algo-
1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "fact
orization-machines", "epoch": 22}, "StartTime": 1605110140.477229}

[11/11/2020 15:55:43 INFO 140334359897920] #throughput_metric: host=algo-1, t
rain throughput=51837.1427722 records/second
[11/11/2020 15:55:43 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=23, batch=0 train rmse <loss>=0.801388580014
[11/11/2020 15:55:43 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=23, batch=0 train mse <loss>=0.642223656178
[11/11/2020 15:55:43 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=23, batch=0 train absolute_loss <loss>=0.728003621101
[11/11/2020 15:55:44 INFO 140334359897920] Iter[23] Batch [500]#011Speed: 518
12.80 samples/sec
[11/11/2020 15:55:44 INFO 140334359897920] #quality_metric: host=algo-1, epoc
```

```
h=23, batch=500 train rmse <loss>=0.943278229769
[11/11/2020 15:55:44 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=23, batch=500 train mse <loss>=0.889773818755
[11/11/2020 15:55:44 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=23, batch=500 train absolute_loss <loss>=0.728992368111
[11/11/2020 15:55:46 INFO 140334359897920] Iter[23] Batch [1000]#011Speed: 50
891.06 samples/sec
[11/11/2020 15:55:46 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=23, batch=1000 train rmse <loss>=0.969034963094
[11/11/2020 15:55:46 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=23, batch=1000 train mse <loss>=0.939028759698
[11/11/2020 15:55:46 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=23, batch=1000 train absolute_loss <loss>=0.757049942975
[2020-11-11 15:55:46.804] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 48, "duration": 3179, "num_examples": 1274,
"num_bytes": 10436096}
[11/11/2020 15:55:46 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=23, train rmse <loss>=0.980867332083
[11/11/2020 15:55:46 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=23, train mse <loss>=0.962100723147
[11/11/2020 15:55:46 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=23, train absolute_loss <loss>=0.769582282173
#metrics {"Metrics": {"update.time": {"count": 1, "max": 3182.2638511657715,
 "sum": 3182.2638511657715, "min": 3182.2638511657715}}, "EndTime": 160511014
6.805985, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605110143.622562}

[11/11/2020 15:55:46 INFO 140334359897920] #progress_metric: host=algo-1, com
pleted 96 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 1274, "sum": 1274.0, "min": 1274}, "Number of Batches Since Last Reset": {"c
ount": 1, "max": 1274, "sum": 1274.0, "min": 1274}, "Number of Records Since
 Last Reset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "T
otal Batches Seen": {"count": 1, "max": 30577, "sum": 30577.0, "min": 30577},
"Total Records Seen": {"count": 1, "max": 3913664, "sum": 3913664.0, "min": 3
913664}, "Max Records Seen Between Resets": {"count": 1, "max": 163064, "su
m": 163064.0, "min": 163064}, "Reset Count": {"count": 1, "max": 25, "sum": 2
5.0, "min": 25}}, "EndTime": 1605110146.806319, "Dimensions": {"Host": "algo-
1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "fact
orization-machines", "epoch": 23}, "StartTime": 1605110143.623671}

[11/11/2020 15:55:46 INFO 140334359897920] #throughput_metric: host=algo-1, t
rain throughput=51232.8451321 records/second
[11/11/2020 15:55:46 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=24, batch=0 train rmse <loss>=0.798155709126
[11/11/2020 15:55:46 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=24, batch=0 train mse <loss>=0.637052536011
[11/11/2020 15:55:46 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=24, batch=0 train absolute_loss <loss>=0.723551869392
[11/11/2020 15:55:48 INFO 140334359897920] Iter[24] Batch [500]#011Speed: 507
11.26 samples/sec
[11/11/2020 15:55:48 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=24, batch=500 train rmse <loss>=0.939561114011
[11/11/2020 15:55:48 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=24, batch=500 train mse <loss>=0.882775086962
[11/11/2020 15:55:48 INFO 140334359897920] #quality_metric: host=algo-1, epoc
```

```
h=24, batch=500 train absolute_loss <loss>=0.72538643427
[11/11/2020 15:55:49 INFO 140334359897920] Iter[24] Batch [1000]#011Speed: 51
847.39 samples/sec
[11/11/2020 15:55:49 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=24, batch=1000 train rmse <loss>=0.965267626186
[11/11/2020 15:55:49 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=24, batch=1000 train mse <loss>=0.931741590162
[11/11/2020 15:55:49 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=24, batch=1000 train absolute_loss <loss>=0.753396290106
[2020-11-11 15:55:49.975] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 50, "duration": 3165, "num_examples": 1274,
"num_bytes": 10436096}
[11/11/2020 15:55:49 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=24, train rmse <loss>=0.977035674922
[11/11/2020 15:55:49 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=24, train mse <loss>=0.95459871007
[11/11/2020 15:55:49 INFO 140334359897920] #quality_metric: host=algo-1, epoc
h=24, train absolute_loss <loss>=0.765867047304
[11/11/2020 15:55:49 INFO 140334359897920] #quality_metric: host=algo-1, trai
n rmse <loss>=0.977035674922
[11/11/2020 15:55:49 INFO 140334359897920] #quality_metric: host=algo-1, trai
n mse <loss>=0.95459871007
[11/11/2020 15:55:49 INFO 140334359897920] #quality_metric: host=algo-1, trai
n absolute_loss <loss>=0.765867047304
#metrics {"Metrics": {"update.time": {"count": 1, "max": 3168.9820289611816,
 "sum": 3168.9820289611816, "min": 3168.9820289611816}}, "EndTime": 160511014
9.976448, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605110146.806091}

[11/11/2020 15:55:49 INFO 140334359897920] #progress_metric: host=algo-1, com
pleted 100 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 1274, "sum": 1274.0, "min": 1274}, "Number of Batches Since Last Reset": {"c
ount": 1, "max": 1274, "sum": 1274.0, "min": 1274}, "Number of Records Since
 Last Reset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "T
otal Batches Seen": {"count": 1, "max": 31851, "sum": 31851.0, "min": 31851},
"Total Records Seen": {"count": 1, "max": 4076728, "sum": 4076728.0, "min": 4
076728}, "Max Records Seen Between Resets": {"count": 1, "max": 163064, "su
m": 163064.0, "min": 163064}, "Reset Count": {"count": 1, "max": 26, "sum": 2
6.0, "min": 26}}, "EndTime": 1605110149.976749, "Dimensions": {"Host": "algo-
1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "fact
orization-machines", "epoch": 24}, "StartTime": 1605110146.807421}

[11/11/2020 15:55:49 INFO 140334359897920] #throughput_metric: host=algo-1, t
rain throughput=51447.8760924 records/second
[11/11/2020 15:55:49 WARNING 140334359897920] wait_for_all_workers will not s
ync workers since the kv store is not running distributed
[11/11/2020 15:55:49 INFO 140334359897920] Pulling entire model from kvstore
 to finalize
#metrics {"Metrics": {"finalize.time": {"count": 1, "max": 5.388021469116211,
"sum": 5.388021469116211, "min": 5.388021469116211}}, "EndTime": 1605110149.9
82556, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm":
"factorization-machines"}, "StartTime": 1605110149.976555}

[11/11/2020 15:55:49 INFO 140334359897920] Saved checkpoint to "/tmp/tmpTJ2MQ
T/state-0001.params"
```

```
[2020-11-11 15:55:50.012] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/test", "epoch": 0, "duration": 80622, "num_examples": 1, "nu
m_bytes": 8192}
[2020-11-11 15:55:50.397] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/test", "epoch": 1, "duration": 384, "num_examples": 82, "num
_bytes": 667904}
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 82, "sum": 82.0, "min": 82}, "Number of Batches Since Last Reset": {"count":
1, "max": 82, "sum": 82.0, "min": 82}, "Number of Records Since Last Reset":
 {"count": 1, "max": 10436, "sum": 10436.0, "min": 10436}, "Total Batches See
n": {"count": 1, "max": 82, "sum": 82.0, "min": 82}, "Total Records Seen":
 {"count": 1, "max": 10436, "sum": 10436.0, "min": 10436}, "Max Records Seen
 Between Resets": {"count": 1, "max": 10436, "sum": 10436.0, "min": 10436},
 "Reset Count": {"count": 1, "max": 1, "sum": 1.0, "min": 1}}, "EndTime": 160
5110150.397605, "Dimensions": {"Host": "algo-1", "Meta": "test_data_iter", "O
peration": "training", "Algorithm": "factorization-machines"}, "StartTime": 1
605110150.012392}

[11/11/2020 15:55:50 INFO 140334359897920] #test_score (algo-1) : ('rmse', 1.
1145810041623063)
[11/11/2020 15:55:50 INFO 140334359897920] #test_score (algo-1) : ('mse', 1.2
42290814839455)
[11/11/2020 15:55:50 INFO 140334359897920] #test_score (algo-1) : ('absolute_
loss', 0.8778366887190188)
[11/11/2020 15:55:50 INFO 140334359897920] #quality_metric: host=algo-1, test
rmse <loss>=1.11458100416
[11/11/2020 15:55:50 INFO 140334359897920] #quality_metric: host=algo-1, test
mse <loss>=1.24229081484
[11/11/2020 15:55:50 INFO 140334359897920] #quality_metric: host=algo-1, test
absolute_loss <loss>=0.877836688719
#metrics {"Metrics": {"totaltime": {"count": 1, "max": 81110.06093025208, "su
m": 81110.06093025208, "min": 81110.06093025208}, "setuptime": {"count": 1,
 "max": 83.79387855529785, "sum": 83.79387855529785, "min": 83.7938785552978
5}}, "EndTime": 1605110150.39904, "Dimensions": {"Host": "algo-1", "Operatio
n": "training", "Algorithm": "factorization-machines"}, "StartTime": 16051101
49.983049}


2020-11-11 15:56:00 Uploading - Uploading generated training model
2020-11-11 15:56:00 Completed - Training job completed
Training seconds: 148
Billable seconds: 148
```

Question: What does changing the `batch_size` and `epochs` do to the final metric?

Answer: There are a couple of things to notice here. The `predictor_type` is set to `'regressor'` because you are trying to predict the star rating. The `batch_size` is set to 128. This value can be tuned for relatively minor improvements in fit and speed, but selecting a reasonable value relative to the dataset is appropriate in most cases. `num_factors` is set to 64. As mentioned initially, factorization machines find a lower dimensional representation of the interactions for all features. Making this value smaller provides a more parsimonious (simple) model, closer to a linear model, but may sacrifice information about interactions. Making it larger provides a higher dimensional representation of feature interactions but adds computational complexity and can lead to overfitting. In a practical application, time should be invested to tune this parameter to the appropriate value.

Question: Check the output of the model. What is the meaning of the metrics used? Is there a difference between the training and testing sets? If yes, what is the meaning of that?

Answer: The **MSE**, **RMSE**, and **Absolute loss** are considerably lower in the training set than in the testing dataset. This might indicate that the model is being molded to the pattern of the training set and not the general pattern on the reviews. This phenomenon is called overfitting.

Evaluate

Congratulations! You have successfully launched an Amazon SageMaker training job. Now what? Well, you need a way to verify that your model is actually predicting coherent values. How do you do this?

Start by calculating a naive baseline to approximate how well your model is doing. The simplest estimate would be to assume every user item rating is just the average rating over all ratings. This is basically saying that you have a model that only learned to output the mean value of all reviews.

Note: You could do better by using each individual video's average; however, in this case, it doesn't really matter because the same conclusions would hold.

Calculate the mean of `star_rating` to get the `naive_guess`. Then, calculate the naive MSE by squaring the naive guess from the test `star_rating` and getting an average.

$$average(test(star_rating) - naive_guess)^2)$$

In [34]:
```
naive_guess = np.mean(train_df['star_rating'])
print('Naive MSE:', np.mean((test_df['star_rating'] - naive_guess) ** 2))
```

```
Naive MSE: 1.5018689895554682
```

Now, calculate predictions for your test dataset. To this end, you'll need to *deploy* the model you just trained.

Note: This will align closely to your CloudWatch output above but may differ slightly due to skipping partial mini-batches in the `eval_net` function.

Use `<estimator_name>.deploy` with `initial_instance_count=1, instance_type=ml.m4.xlarge` .

In [35]:
```
fm_predictor = fm.deploy(initial_instance_count=1, instance_type='ml.m4.xlarg
e')
```

```
---------------!
```

Now that your endpoint is 'InService', evaluate how your model performs on the test set. Compare that test set performance to the performance on the training set.

Key questions to consider:

1. How does your model's performance on the test set compare to the training set? What can you deduce from this comparison?
2. Are there obvious differences between the outcomes of metrics like accuracy, precision, and recall? If so, why might you be seeing those differences?
3. Given your business situation and goals, which metric(s) is most important for you to consider here? Why?
4. Is the outcome for the metric(s) you consider most important sufficient for what you need from a business standpoint? If not, what are some things you might change in your next iteration (in the feature engineering section, which is coming up next)?

Use the cells below to answer these and other questions. Insert and delete cells where needed.

Project presentation: Record questions to these and other similar questions you might answer in this section in your project presentations. Record key details and decisions you've made in your project presentations.

The deployment process involves creating an instance of the specified size, in this case `ml.m4.xlarge` , with the model you trained and saved on Amazon S3. To get a prediction, you need to pass your data in a serialized form of JSON. The output you get from the inference will be in serialized JSON form as well, so you also need to deserialize it to get the predicted values.

import json from sagemaker.deserializers import JSONDeserializer from sagemaker.serializers import BaseSerializer class fm_serializer(BaseSerializer): CONTENT_TYPE='application/json' def serialize(data): js = {'instances': []} for row in data: js['instances'].append({'features': row.tolist()}) return json.dumps(js) fm_predictor.serializer = fm_serializer fm_predictor.deserializer = JSONDeserializer() print(f"Accepted content type: {fm_predictor.content_type}")

In [36]:
```
# Create a serializer function for the predictor
import json
from sagemaker.deserializers import JSONDeserializer
from sagemaker.serializers import BaseSerializer

class fm_serializer(BaseSerializer):
    CONTENT_TYPE='application/json'
    def serialize(data):
            js = {'instances': []}
            for row in data:
                js['instances'].append({'features': row.tolist()})
            return json.dumps(js)
fm_predictor.serializer = fm_serializer
fm_predictor.deserializer = JSONDeserializer()
print(f"Accepted content type: {fm_predictor.content_type}")
```

```
Accepted content type: application/json
```

Check how your training set did. Use the endpoint to get predictions from your model.

First, look at what a single prediction looks like.

Amazon SageMaker model containers must respond to requests within 60 seconds. The model itself can have a maximum processing time of 60 seconds before responding to the /invocations. To do that, call the `predict` function for 5 rows at a time and then add those rows to a list.

In [37]:
```
# Pass the X_train data to the predictor deployed
ytrain_p = []
for i in range(0, 1000, 5):
    preds = fm_predictor.predict(X_train[i:i + 5].toarray())['predictions']
    p = [ytrain_p.append(x['score']) for x in preds]
```

Question: Now that you have inferences, do a sanity check. What are the minimum and maximum values predicted in the inferences? Do those correspond to the minimum and maximum values in the training data?

In [38]:
```
print('The minimum rating predicted is: ', min(ytrain_p), 'and the maximum is:
',max(ytrain_p))
```

```
The minimum rating predicted is:  3.1462368965148926 and the maximum is:  4.8
64139556884766
```

Now, check your test dataset.

In [39]:
```
Y_pred = []
for i in range(0, X_test.shape[0], 5):
    preds = fm_predictor.predict(X_test[i:i+5].toarray())['predictions']
    p = [Y_pred.append(x['score']) for x in preds]
```

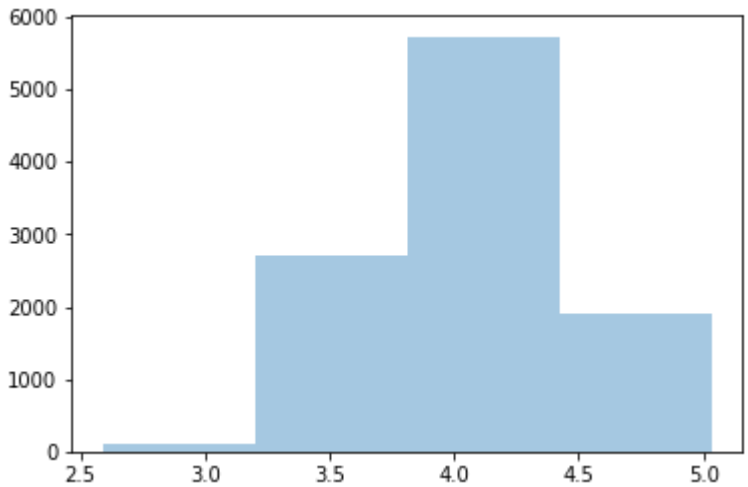
Question: How are the min and max values alike in the predictions? Bonus point if you check the entire distribution (histogram).

In [40]:
```
max(Y_pred), min(Y_pred)
```

Out[40]:
```
(5.033811092376709, 2.58854341506958)
```

In [41]:
```
sns.distplot(Y_pred, kde=False, bins=4)
```

Out[41]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa8439bc7b8>
```



Finally, calculate the mean squared error for the test set and see how much of an improvement it is from the baseline.

In [42]:
```
print('MSE:', np.mean((test_df['star_rating'] - Y_pred) ** 2))
```

```
MSE: 1.2422908140030924
```

For recommender systems, subjective accuracy also matters. Get some recommendations for a random user to see if they make intuitive sense.

Try using user number 200, and see what they have watched and rated highly.

In [43]:
```
reduced_df[reduced_df['user'] == 200].sort_values(
    ['star_rating', 'item'], ascending=[False, True])
```

Out[43]:

	customer_id	product_id	star_rating	product_title	user	item
60486	43080952	B00DTOYJJ2	5	Breaking Bad The Final Season	200	79
14249	43080952	B00VFTA70I	5	The Imitation Game	200	88
28407	43080952	B00KG2NGPS	5	The Monuments Men	200	149
71978	43080952	B00TPJHW7Q	5	The Theory of Everything	200	157
6753	43080952	B00IIVJU3Q	5	Philomena	200	180
84419	43080952	B0047WJ12K	5	Inception	200	467
161384	43080952	B00DGNO3WE	5	Quartet	200	1094
119440	43080952	B0059R5Q08	5	Jane Eyre	200	2617
167594	43080952	B001VLKWUA	5	Out of Africa	200	3119
136597	43080952	B00TF87WQI	5	Rosewater	200	3606
118790	43080952	B001P4NZ9I	5	Louis CK: Chewed Up	200	3840
99775	43080952	B00SH00N3Y	4	Lucy	200	26
102549	43080952	B00S65TBHY	4	Gone Girl	200	40
40897	43080952	B00I8H6MGS	4	Dallas Buyers Club	200	41
40443	43080952	B00J8CG6KE	4	Wolf Of Wall Street	200	45
27118	43080952	B00TPJE3ZK	4	Birdman	200	70
3368	43080952	B00XVPL8G6	4	American Sniper	200	96
45654	43080952	B00J2PEU3W	4	Saving Mr. Banks	200	111
28279	43080952	B00IMY9TNK	4	Nebraska	200	193
93385	43080952	B00TF874XE	4	Nightcrawler	200	277
34018	43080952	B00RNRD8RA	4	The Interview	200	368
20026	43080952	B00XJDYITO	4	Still Alice	200	383
12546	43080952	B006IGZWB2	4	The Help	200	450
123745	43080952	B001Q2J8BS	4	Burn After Reading	200	3221
26130	43080952	B00797MB0O	4	Young Adult	200	3546
150894	43080952	B001AITGWE	4	Falling Down	200	4992
28625	43080952	B00IMYSCR4	3	Gravity	200	15
55116	43080952	B00HYTSTNU	3	Captain Phillips	200	47
142138	43080952	B00HZ4ASEC	3	Last Vegas	200	107
6913	43080952	B00JP5ONY0	3	The Secret Life of Walter Mitty	200	117
112562	43080952	B00J2PRBNS	3	American Hustle	200	137
56580	43080952	B00QGKVB4W	3	The Hundred-Foot Journey (Theatrical)	200	198
50080	43080952	B00JJH3HCS	3	August: Osage County	200	207
74850	43080952	B00H8C4HZ2	3	Lee Daniels' The Butler	200	229
40084	43080952	B00L23RCPU	3	The Grand Budapest Hotel	200	234

	customer_id	product_id	star_rating	product_title	user	item
97041	43080952	B00HAQAQME	3	Blue Jasmine	200	270
45571	43080952	B00VFTBPSG	3	Wild	200	351
43350	43080952	B00HNBS8MQ	3	Thanks For Sharing	200	720
121649	43080952	B00WG1FMCW	3	Cake	200	1169
155565	43080952	B00G32OMF8	3	Before Midnight	200	2437
74333	43080952	B00ST1OBS4	2	The Judge (2014)	200	34
97168	43080952	B00RVC3YKI	2	Boyhood	200	563
72804	43080952	B00U7B21A4	2	Foxcatcher	200	860
57296	43080952	B001DM1VXE	2	Pi	200	1789
29821	43080952	B005KL3R6I	1	Bridesmaids	200	559
115522	43080952	B001H6BIOS	1	Then She Found Me	200	1762

As you can see, this user likes to watch comedies, romance, and light-hearted movies and dislikes drama and fantasy movies. Let's see how your model predicts movie ratings for this user.

In [44]:
```
def prepare_predictions(user_id, number_movies, columns):
    # Create the sparse matrix
    X = lil_matrix((number_movies, columns)).astype('float32')
    movie_index_start = columns - number_movies

    # Fill out the matrix. Each row will be the same user with every possible
 movie.
    for row in range(number_movies):
        X[row, user_id - 1] = 1
        X[row, movie_index_start + row] = 1

    return X

user_200 = prepare_predictions(200, products.shape[0], customers.shape[0] + pr
oducts.shape[0])
```

Now create a list of all the ratings that the model would predict for user 200 for all movies.

In [45]:
```
pred_200 = []
for i in range(0, user_200.shape[0], 5):
    preds = fm_predictor.predict(user_200[i:i+5].toarray())['predictions']
    p = [pred_200.append(x['score']) for x in preds]
```

Now loop through and predict user 200's ratings for every common video in the catalog to see which ones to recommend or not recommend.

Create a new dataframe `titles` by using the `reduced_df` dataframe to group by the items. Use the `product_title` column and create another column `score` and add the values from `pred_200` to it.

In [46]:
```
titles = reduced_df.groupby('item')['product_title'].first().reset_index()
titles['score'] = pred_200
```

Question: What products got the highest score?

Hint: Use the `sort_values` function to sort columns `score` and `item` and use parameter `asecnding=[False,True]`

In [47]:
```
titles.sort_values(['score', 'item'], ascending=[False, True])
```

Out[47]:

	item	product_title	score
25	25	John Adams Season 1	4.048313
5	5	Downton Abbey Season 3	4.040651
6	6	Justified Season 4	4.003147
12	12	Justified Season 3	3.997310
315	315	The Man In the High Castle - Season 1	3.989603
...	...	...	...
2602	2602	A Most Violent Year	3.424680
786	786	Inside Amy Schumer Season 1	3.424536
2112	2112	Snatch (Feature)	3.423681
13	13	Under The Dome, Season 2	3.422001
571	571	Transparent Season 1	3.418923

6659 rows × 3 columns

Question: What can you conclude from the highly rated and lowest rated shows for the user?

Answer: The predicted highly rated shows have some well-reviewed TV dramas and some book adaptations. The lowest scores are from comedies, child, and teenage movies.

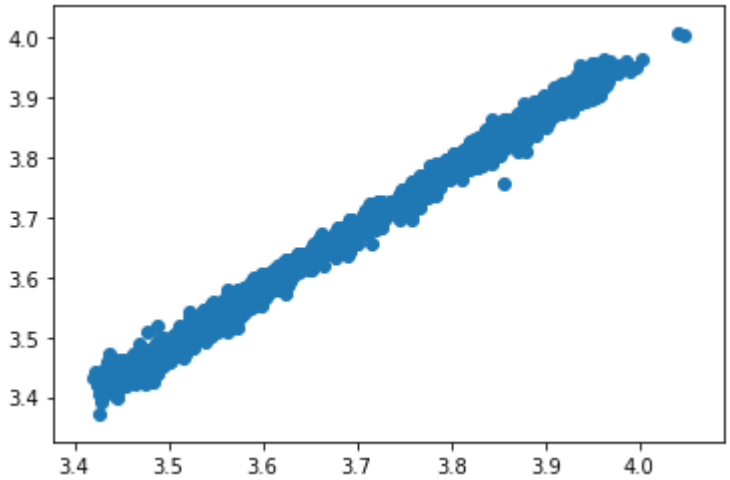
See if your recommendations have correlations with other users. Try user 201. Perform the same operations as you did for user 200.

In [48]:
```
user_201 = prepare_predictions(201, products.shape[0], customers.shape[0] + pr
oducts.shape[0])

pred_201 = []
for i in range(0, user_201.shape[0], 5):
    preds = fm_predictor.predict(user_201[i:i+5].toarray())['predictions']
    p = [pred_201.append(x['score']) for x in preds]
```

In [49]:
```
plt.scatter(pred_200, pred_201)
plt.show()
```

Out[49]:
```
<matplotlib.collections.PathCollection at 0x7fa843997c18>
```



Question: What can you conclude from the scatter plot between the two users?

Answer: This correlation is nearly perfect. Essentially, the average rating of items dominates across users, and our system will recommend the same well-reviewed items to everyone. That basically means that the model is only finding the best rated shows and presenting them to all users. Now we need to see if we can do better by shifting the paradigm from regression to classification.

Delete the endpoint you created for inference because you won't be using it anymore.

In [50]:
```
sagemaker.Session().delete_endpoint(fm_predictor.endpoint_name)
```

Project Solution Lab 4:

Iteration II

Step 4: Feature engineering

You've now gone through one iteration of training and evaluating your model. Given that the outcome you reached for your model the first time probably wasn't sufficient for solving your business problem, what are some things you could change about your data to possibly improve model performance?

Key questions to consider:

1. How might changing the machine learning problem help your dataset? You tried to use regression to solve the problem; can classification help?
2. What do you need to do to change the machine learning problem to a machine learning classification problem? Write down the new problem statement for classification.

Project presentation: Record key decisions and methods you use in this section in your project presentations, as well as any new performance metrics you obtain after evaluating your model again.

Now change the training datasets to have a binary output depending on the rating they get. You will consider recommending something to a user when the rating is 5 stars, and you will save again as a protobuf format in Amazon S3. Do the following:

1. Use the `loadDataset` function with the option `regression=False` to create your training datasets.
2. Write the dataset as a protobuf format.
3. Retrain the model using `predictor_type='binary_classifier'` .
4. Deploy your model to an endpoint and evaluate the model, similar to how you did before on the test set.
5. Inspect how you did on the test set using a confusion matrix.

In [51]:
```
X_train_class, Y_train_class = loadDataset(train_df, customers.shape[0],
                                products.shape[0], regressor=False)
X_test_class, Y_test_class = loadDataset(test_df, customers.shape[0],
                              products.shape[0], regressor=False)
```

In [52]:
```
# Write dataset as a protobuf

fm_train_data_path = writeDatasetToProtobuf(X_train_class, bucket, prefix, 'tr
ain_class', "sparse", Y_train_class)
fm_test_data_path  = writeDatasetToProtobuf(X_test_class, bucket, prefix, 'tes
t_class', "sparse", Y_test_class)

print("Training data S3 path: ", fm_train_data_path)
print("Test data S3 path: ", fm_test_data_path)
```

```
Training data S3 path:  s3://qls-163754-04de04c0c698e64b-labbucket-q4zl2sa0cq
9l/sagemaker-fm/train_class
Test data S3 path:  s3://qls-163754-04de04c0c698e64b-labbucket-q4zl2sa0cq9l/s
agemaker-fm/test_class
```

Finally, retrain the model, changing from regression to binary classification. Use the same code and settings that you did when you trained your model previously, but change the `predictor_type='binary_classifier` .

In [53]:
```
from sagemaker import get_execution_role
from sagemaker import image_uris

#output_prefix= 's3://<LabBucketName>/sagemaker-fm/model'

output_prefix = 's3://' + bucket + '/sagemaker-fm/model'
instance_type='ml.m4.xlarge'
batch_size = 512

fm = sagemaker.estimator.Estimator(
    image_uris.retrieve("factorization-machines",boto3.Session().region_name),
    role,
    instance_count=1,
    instance_type=instance_type,
    output_path=output_prefix,
    sagemaker_session=sagemaker.Session()
)

fm.set_hyperparameters(feature_dim=X_train.shape[1],
                      # predictor_type='regressor',
                        predictor_type='binary_classifier',
                        mini_batch_size=batch_size,
                        num_factors=128,
                        epochs=25,
                        clip_gradient=5.0,
                        rescale_grad=1.0/batch_size
                        )

fm.fit({'train': fm_train_data_path, 'test': fm_test_data_path})
```

```
2020-11-11 16:08:57 Starting - Starting the training job...
2020-11-11 16:08:59 Starting - Launching requested ML instances.........
2020-11-11 16:10:28 Starting - Preparing the instances for training...
2020-11-11 16:11:25 Downloading - Downloading input data...
2020-11-11 16:11:57 Training - Downloading the training image...
2020-11-11 16:12:21 Training - Training image download completed. Training in
progress.Docker entrypoint called with argument(s): train
Running default environment configuration script
/opt/amazon/lib/python2.7/site-packages/pandas/util/nosetester.py:13: Depreca
tionWarning: Importing from numpy.testing.nosetester is deprecated, import fr
om numpy.testing instead.
  from numpy.testing import nosetester
[11/11/2020 16:12:24 INFO 139996470646592] Reading default configuration from
/opt/amazon/lib/python2.7/site-packages/algorithm/resources/default-conf.jso
n: {u'factors_lr': u'0.0001', u'linear_init_sigma': u'0.01', u'epochs': 1,
 u'_wd': u'1.0', u'_num_kv_servers': u'auto', u'use_bias': u'true', u'factors
_init_sigma': u'0.001', u'_log_level': u'info', u'bias_init_method': u'norma
l', u'linear_init_method': u'normal', u'linear_lr': u'0.001', u'factors_init_
method': u'normal', u'_tuning_objective_metric': u'', u'bias_wd': u'0.01',
 u'use_linear': u'true', u'bias_lr': u'0.1', u'mini_batch_size': u'1000', u'_
use_full_symbolic': u'true', u'batch_metrics_publish_interval': u'500', u'bia
s_init_sigma': u'0.01', u'_num_gpus': u'auto', u'_data_format': u'record',
 u'factors_wd': u'0.00001', u'linear_wd': u'0.001', u'_kvstore': u'auto', u'_
learning_rate': u'1.0', u'_optimizer': u'adam'}
[11/11/2020 16:12:24 INFO 139996470646592] Reading provided configuration fro
m /opt/ml/input/config/hyperparameters.json: {u'epochs': u'25', u'predictor_t
ype': u'binary_classifier', u'rescale_grad': u'0.001953125', u'clip_gradien
t': u'5.0', u'feature_dim': u'17095', u'num_factors': u'128', u'mini_batch_si
ze': u'512'}
[11/11/2020 16:12:24 INFO 139996470646592] Final configuration: {u'factors_l
r': u'0.0001', u'linear_init_sigma': u'0.01', u'epochs': u'25', u'feature_di
m': u'17095', u'num_factors': u'128', u'_wd': u'1.0', u'_num_kv_servers': u'a
uto', u'use_bias': u'true', u'factors_init_sigma': u'0.001', u'_log_level':
 u'info', u'bias_init_method': u'normal', u'linear_init_method': u'normal',
 u'rescale_grad': u'0.001953125', u'linear_lr': u'0.001', u'factors_init_meth
od': u'normal', u'_tuning_objective_metric': u'', u'bias_wd': u'0.01', u'use_
linear': u'true', u'bias_lr': u'0.1', u'mini_batch_size': u'512', u'_use_full
_symbolic': u'true', u'clip_gradient': u'5.0', u'batch_metrics_publish_interv
al': u'500', u'predictor_type': u'binary_classifier', u'bias_init_sigma':
 u'0.01', u'_num_gpus': u'auto', u'_data_format': u'record', u'factors_wd':
 u'0.00001', u'linear_wd': u'0.001', u'_kvstore': u'auto', u'_learning_rate':
u'1.0', u'_optimizer': u'adam'}
[11/11/2020 16:12:24 WARNING 139996470646592] Loggers have already been setu
p.
Process 1 is a worker.
[11/11/2020 16:12:24 INFO 139996470646592] Using default worker.
[2020-11-11 16:12:24.676] [tensorio] [warning] TensorIO is already initialize
d; ignoring the initialization routine.
[2020-11-11 16:12:24.683] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 0, "duration": 10, "num_examples": 1, "num_
bytes": 32768}
[11/11/2020 16:12:24 INFO 139996470646592] nvidia-smi took: 0.0251250267029 s
ecs to identify 0 gpus
[11/11/2020 16:12:24 INFO 139996470646592] Number of GPUs being used: 0
[11/11/2020 16:12:24 INFO 139996470646592] [Sparse network] Building a sparse
```

```
network.
[11/11/2020 16:12:24 INFO 139996470646592] Create Store: local
#metrics {"Metrics": {"initialize.time": {"count": 1, "max": 37.6939773559570
3, "sum": 37.69397735595703, "min": 37.69397735595703}}, "EndTime": 160511114
4.719135, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605111144.672124}

#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 1, "sum": 1.0, "min": 1}, "Number of Batches Since Last Reset": {"count": 1,
"max": 0, "sum": 0.0, "min": 0}, "Number of Records Since Last Reset": {"coun
t": 1, "max": 0, "sum": 0.0, "min": 0}, "Total Batches Seen": {"count": 1, "m
ax": 1, "sum": 1.0, "min": 1}, "Total Records Seen": {"count": 1, "max": 512,
"sum": 512.0, "min": 512}, "Max Records Seen Between Resets": {"count": 1, "m
ax": 512, "sum": 512.0, "min": 512}, "Reset Count": {"count": 1, "max": 1, "s
um": 1.0, "min": 1}}, "EndTime": 1605111144.719328, "Dimensions": {"Host": "a
lgo-1", "Meta": "init_train_data_iter", "Operation": "training", "Algorithm":
"factorization-machines"}, "StartTime": 1605111144.71929}

[16:12:24] /opt/brazil-pkg-cache/packages/AIAlgorithmsMXNet/AIAlgorithmsMXNet
-1.1.x.203343.0/AL2012/generic-flavor/src/src/kvstore/./kvstore_local.h:280:
 Warning: non-default weights detected during kvstore pull. This call has bee
n ignored. Please make sure to use row_sparse_pull with row_ids.
[16:12:24] /opt/brazil-pkg-cache/packages/AIAlgorithmsMXNet/AIAlgorithmsMXNet
-1.1.x.203343.0/AL2012/generic-flavor/src/src/kvstore/./kvstore_local.h:280:
 Warning: non-default weights detected during kvstore pull. This call has bee
n ignored. Please make sure to use row_sparse_pull with row_ids.
[11/11/2020 16:12:24 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=0, batch=0 train binary_classification_accuracy <score>=0.693359375
[11/11/2020 16:12:24 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=0, batch=0 train binary_classification_cross_entropy <loss>=0.689488351345
[11/11/2020 16:12:24 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=0, batch=0 train binary_f_1.000 <score>=0.816374269006
[2020-11-11 16:12:27.226] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 2, "duration": 2410, "num_examples": 319,
 "num_bytes": 10436096}
[11/11/2020 16:12:27 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=0, train binary_classification_accuracy <score>=0.524662029781
[11/11/2020 16:12:27 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=0, train binary_classification_cross_entropy <loss>=0.692112501886
[11/11/2020 16:12:27 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=0, train binary_f_1.000 <score>=0.49222337044
#metrics {"Metrics": {"epochs": {"count": 1, "max": 25, "sum": 25.0, "min": 2
5}, "update.time": {"count": 1, "max": 2507.1980953216553, "sum": 2507.198095
3216553, "min": 2507.1980953216553}}, "EndTime": 1605111147.226763, "Dimensio
ns": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-
machines"}, "StartTime": 1605111144.719216}

[11/11/2020 16:12:27 INFO 139996470646592] #progress_metric: host=algo-1, com
pleted 4 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 319, "sum": 319.0, "min": 319}, "Number of Batches Since Last Reset": {"coun
t": 1, "max": 319, "sum": 319.0, "min": 319}, "Number of Records Since Last R
eset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "Total Ba
tches Seen": {"count": 1, "max": 320, "sum": 320.0, "min": 320}, "Total Recor
ds Seen": {"count": 1, "max": 163576, "sum": 163576.0, "min": 163576}, "Max R
ecords Seen Between Resets": {"count": 1, "max": 163064, "sum": 163064.0, "mi
```

```
n": 163064}, "Reset Count": {"count": 1, "max": 2, "sum": 2.0, "min": 2}}, "E
ndTime": 1605111147.227043, "Dimensions": {"Host": "algo-1", "Meta": "trainin
g_data_iter", "Operation": "training", "Algorithm": "factorization-machines",
"epoch": 0}, "StartTime": 1605111144.719529}

[11/11/2020 16:12:27 INFO 139996470646592] #throughput_metric: host=algo-1, t
rain throughput=65027.1157826 records/second
[11/11/2020 16:12:27 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=1, batch=0 train binary_classification_accuracy <score>=0.279296875
[11/11/2020 16:12:27 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=1, batch=0 train binary_classification_cross_entropy <loss>=0.714410603046
[11/11/2020 16:12:27 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=1, batch=0 train binary_f_1.000 <score>=0.0
[2020-11-11 16:12:29.550] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 4, "duration": 2321, "num_examples": 319,
 "num_bytes": 10436096}
[11/11/2020 16:12:29 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=1, train binary_classification_accuracy <score>=0.555097717476
[11/11/2020 16:12:29 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=1, train binary_classification_cross_entropy <loss>=0.684296471011
[11/11/2020 16:12:29 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=1, train binary_f_1.000 <score>=0.517570357781
#metrics {"Metrics": {"update.time": {"count": 1, "max": 2323.8959312438965,
 "sum": 2323.8959312438965, "min": 2323.8959312438965}}, "EndTime": 160511114
9.55127, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605111147.226858}

[11/11/2020 16:12:29 INFO 139996470646592] #progress_metric: host=algo-1, com
pleted 8 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 319, "sum": 319.0, "min": 319}, "Number of Batches Since Last Reset": {"coun
t": 1, "max": 319, "sum": 319.0, "min": 319}, "Number of Records Since Last R
eset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "Total Ba
tches Seen": {"count": 1, "max": 639, "sum": 639.0, "min": 639}, "Total Recor
ds Seen": {"count": 1, "max": 326640, "sum": 326640.0, "min": 326640}, "Max R
ecords Seen Between Resets": {"count": 1, "max": 163064, "sum": 163064.0, "mi
n": 163064}, "Reset Count": {"count": 1, "max": 3, "sum": 3.0, "min": 3}}, "E
ndTime": 1605111149.551506, "Dimensions": {"Host": "algo-1", "Meta": "trainin
g_data_iter", "Operation": "training", "Algorithm": "factorization-machines",
"epoch": 1}, "StartTime": 1605111147.22734}

[11/11/2020 16:12:29 INFO 139996470646592] #throughput_metric: host=algo-1, t
rain throughput=70156.0881652 records/second
[11/11/2020 16:12:29 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=2, batch=0 train binary_classification_accuracy <score>=0.287109375
[11/11/2020 16:12:29 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=2, batch=0 train binary_classification_cross_entropy <loss>=0.706026315689
[11/11/2020 16:12:29 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=2, batch=0 train binary_f_1.000 <score>=0.0214477211796
[2020-11-11 16:12:31.553] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 6, "duration": 2000, "num_examples": 319,
 "num_bytes": 10436096}
[11/11/2020 16:12:31 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=2, train binary_classification_accuracy <score>=0.586911001176
[11/11/2020 16:12:31 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=2, train binary_classification_cross_entropy <loss>=0.675965565873
```

```
[11/11/2020 16:12:31 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=2, train binary_f_1.000 <score>=0.552969316292
#metrics {"Metrics": {"update.time": {"count": 1, "max": 2002.5949478149414,
 "sum": 2002.5949478149414, "min": 2002.5949478149414}}, "EndTime": 160511115
1.554391, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605111149.551353}

[11/11/2020 16:12:31 INFO 139996470646592] #progress_metric: host=algo-1, com
pleted 12 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 319, "sum": 319.0, "min": 319}, "Number of Batches Since Last Reset": {"coun
t": 1, "max": 319, "sum": 319.0, "min": 319}, "Number of Records Since Last R
eset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "Total Ba
tches Seen": {"count": 1, "max": 958, "sum": 958.0, "min": 958}, "Total Recor
ds Seen": {"count": 1, "max": 489704, "sum": 489704.0, "min": 489704}, "Max R
ecords Seen Between Resets": {"count": 1, "max": 163064, "sum": 163064.0, "mi
n": 163064}, "Reset Count": {"count": 1, "max": 4, "sum": 4.0, "min": 4}}, "E
ndTime": 1605111151.554673, "Dimensions": {"Host": "algo-1", "Meta": "trainin
g_data_iter", "Operation": "training", "Algorithm": "factorization-machines",
"epoch": 2}, "StartTime": 1605111149.551758}

[11/11/2020 16:12:31 INFO 139996470646592] #throughput_metric: host=algo-1, t
rain throughput=81407.0839252 records/second
[11/11/2020 16:12:31 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=3, batch=0 train binary_classification_accuracy <score>=0.365234375
[11/11/2020 16:12:31 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=3, batch=0 train binary_classification_cross_entropy <loss>=0.69769436121
[11/11/2020 16:12:31 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=3, batch=0 train binary_f_1.000 <score>=0.213075060533
[2020-11-11 16:12:33.607] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 8, "duration": 2050, "num_examples": 319,
 "num_bytes": 10436096}
[11/11/2020 16:12:33 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=3, train binary_classification_accuracy <score>=0.613973109326
[11/11/2020 16:12:33 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=3, train binary_classification_cross_entropy <loss>=0.668364912552
[11/11/2020 16:12:33 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=3, train binary_f_1.000 <score>=0.584501426754
#metrics {"Metrics": {"update.time": {"count": 1, "max": 2052.5999069213867,
 "sum": 2052.5999069213867, "min": 2052.5999069213867}}, "EndTime": 160511115
3.607628, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605111151.55449}

[11/11/2020 16:12:33 INFO 139996470646592] #progress_metric: host=algo-1, com
pleted 16 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 319, "sum": 319.0, "min": 319}, "Number of Batches Since Last Reset": {"coun
t": 1, "max": 319, "sum": 319.0, "min": 319}, "Number of Records Since Last R
eset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "Total Ba
tches Seen": {"count": 1, "max": 1277, "sum": 1277.0, "min": 1277}, "Total Re
cords Seen": {"count": 1, "max": 652768, "sum": 652768.0, "min": 652768}, "Ma
x Records Seen Between Resets": {"count": 1, "max": 163064, "sum": 163064.0,
 "min": 163064}, "Reset Count": {"count": 1, "max": 5, "sum": 5.0, "min":
 5}}, "EndTime": 1605111153.607864, "Dimensions": {"Host": "algo-1", "Meta":
 "training_data_iter", "Operation": "training", "Algorithm": "factorization-m
achines", "epoch": 3}, "StartTime": 1605111151.554991}
```

```
[11/11/2020 16:12:33 INFO 139996470646592] #throughput_metric: host=algo-1, t
rain throughput=79426.9501702 records/second
[11/11/2020 16:12:33 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=4, batch=0 train binary_classification_accuracy <score>=0.498046875
[11/11/2020 16:12:33 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=4, batch=0 train binary_classification_cross_entropy <loss>=0.689783751965
[11/11/2020 16:12:33 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=4, batch=0 train binary_f_1.000 <score>=0.48496993988
[2020-11-11 16:12:35.637] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 10, "duration": 2027, "num_examples": 319,
 "num_bytes": 10436096}
[11/11/2020 16:12:35 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=4, train binary_classification_accuracy <score>=0.638267780172
[11/11/2020 16:12:35 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=4, train binary_classification_cross_entropy <loss>=0.661561312533
[11/11/2020 16:12:35 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=4, train binary_f_1.000 <score>=0.613640079259
#metrics {"Metrics": {"update.time": {"count": 1, "max": 2029.6108722686768,
 "sum": 2029.6108722686768, "min": 2029.6108722686768}}, "EndTime": 160511115
5.637776, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605111153.607712}

[11/11/2020 16:12:35 INFO 139996470646592] #progress_metric: host=algo-1, com
pleted 20 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 319, "sum": 319.0, "min": 319}, "Number of Batches Since Last Reset": {"coun
t": 1, "max": 319, "sum": 319.0, "min": 319}, "Number of Records Since Last R
eset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "Total Ba
tches Seen": {"count": 1, "max": 1596, "sum": 1596.0, "min": 1596}, "Total Re
cords Seen": {"count": 1, "max": 815832, "sum": 815832.0, "min": 815832}, "Ma
x Records Seen Between Resets": {"count": 1, "max": 163064, "sum": 163064.0,
 "min": 163064}, "Reset Count": {"count": 1, "max": 6, "sum": 6.0, "min":
 6}}, "EndTime": 1605111155.638057, "Dimensions": {"Host": "algo-1", "Meta":
 "training_data_iter", "Operation": "training", "Algorithm": "factorization-m
achines", "epoch": 4}, "StartTime": 1605111153.608124}

[11/11/2020 16:12:35 INFO 139996470646592] #throughput_metric: host=algo-1, t
rain throughput=80323.8117529 records/second
[11/11/2020 16:12:35 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=5, batch=0 train binary_classification_accuracy <score>=0.58984375
[11/11/2020 16:12:35 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=5, batch=0 train binary_classification_cross_entropy <loss>=0.682355940342
[11/11/2020 16:12:35 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=5, batch=0 train binary_f_1.000 <score>=0.623655913978
[2020-11-11 16:12:37.689] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 12, "duration": 2049, "num_examples": 319,
 "num_bytes": 10436096}
[11/11/2020 16:12:37 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=5, train binary_classification_accuracy <score>=0.658301087382
[11/11/2020 16:12:37 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=5, train binary_classification_cross_entropy <loss>=0.655454463906
[11/11/2020 16:12:37 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=5, train binary_f_1.000 <score>=0.637714463768
#metrics {"Metrics": {"update.time": {"count": 1, "max": 2052.121877670288,
 "sum": 2052.121877670288, "min": 2052.121877670288}}, "EndTime": 1605111157.
```

```
690537, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605111155.637871}

[11/11/2020 16:12:37 INFO 139996470646592] #progress_metric: host=algo-1, com
pleted 24 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 319, "sum": 319.0, "min": 319}, "Number of Batches Since Last Reset": {"coun
t": 1, "max": 319, "sum": 319.0, "min": 319}, "Number of Records Since Last R
eset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "Total Ba
tches Seen": {"count": 1, "max": 1915, "sum": 1915.0, "min": 1915}, "Total Re
cords Seen": {"count": 1, "max": 978896, "sum": 978896.0, "min": 978896}, "Ma
x Records Seen Between Resets": {"count": 1, "max": 163064, "sum": 163064.0,
 "min": 163064}, "Reset Count": {"count": 1, "max": 7, "sum": 7.0, "min":
 7}}, "EndTime": 1605111157.690811, "Dimensions": {"Host": "algo-1", "Meta":
 "training_data_iter", "Operation": "training", "Algorithm": "factorization-m
achines", "epoch": 5}, "StartTime": 1605111155.638374}

[11/11/2020 16:12:37 INFO 139996470646592] #throughput_metric: host=algo-1, t
rain throughput=79442.8185644 records/second
[11/11/2020 16:12:37 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=6, batch=0 train binary_classification_accuracy <score>=0.646484375
[11/11/2020 16:12:37 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=6, batch=0 train binary_classification_cross_entropy <loss>=0.675384640694
[11/11/2020 16:12:37 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=6, batch=0 train binary_f_1.000 <score>=0.695798319328
[2020-11-11 16:12:39.672] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 14, "duration": 1979, "num_examples": 319,
 "num_bytes": 10436096}
[11/11/2020 16:12:39 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=6, train binary_classification_accuracy <score>=0.673877106191
[11/11/2020 16:12:39 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=6, train binary_classification_cross_entropy <loss>=0.649930583459
[11/11/2020 16:12:39 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=6, train binary_f_1.000 <score>=0.656450146089
#metrics {"Metrics": {"update.time": {"count": 1, "max": 1982.1529388427734,
 "sum": 1982.1529388427734, "min": 1982.1529388427734}}, "EndTime": 160511115
9.673311, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605111157.690626}

[11/11/2020 16:12:39 INFO 139996470646592] #progress_metric: host=algo-1, com
pleted 28 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 319, "sum": 319.0, "min": 319}, "Number of Batches Since Last Reset": {"coun
t": 1, "max": 319, "sum": 319.0, "min": 319}, "Number of Records Since Last R
eset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "Total Ba
tches Seen": {"count": 1, "max": 2234, "sum": 2234.0, "min": 2234}, "Total Re
cords Seen": {"count": 1, "max": 1141960, "sum": 1141960.0, "min": 1141960},
 "Max Records Seen Between Resets": {"count": 1, "max": 163064, "sum": 16306
4.0, "min": 163064}, "Reset Count": {"count": 1, "max": 8, "sum": 8.0, "min":
8}}, "EndTime": 1605111159.673583, "Dimensions": {"Host": "algo-1", "Meta":
 "training_data_iter", "Operation": "training", "Algorithm": "factorization-m
achines", "epoch": 6}, "StartTime": 1605111157.691123}

[11/11/2020 16:12:39 INFO 139996470646592] #throughput_metric: host=algo-1, t
rain throughput=82247.4657929 records/second
[11/11/2020 16:12:39 INFO 139996470646592] #quality_metric: host=algo-1, epoc
```

```
h=7, batch=0 train binary_classification_accuracy <score>=0.669921875
[11/11/2020 16:12:39 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=7, batch=0 train binary_classification_cross_entropy <loss>=0.668821871281
[11/11/2020 16:12:39 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=7, batch=0 train binary_f_1.000 <score>=0.724306688418
[2020-11-11 16:12:41.707] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 16, "duration": 2032, "num_examples": 319,
 "num_bytes": 10436096}
[11/11/2020 16:12:41 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=7, train binary_classification_accuracy <score>=0.685963215125
[11/11/2020 16:12:41 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=7, train binary_classification_cross_entropy <loss>=0.644890655731
[11/11/2020 16:12:41 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=7, train binary_f_1.000 <score>=0.670791586704
#metrics {"Metrics": {"update.time": {"count": 1, "max": 2034.6260070800781,
 "sum": 2034.6260070800781, "min": 2034.6260070800781}}, "EndTime": 160511116
1.708511, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605111159.673401}

[11/11/2020 16:12:41 INFO 139996470646592] #progress_metric: host=algo-1, com
pleted 32 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 319, "sum": 319.0, "min": 319}, "Number of Batches Since Last Reset": {"coun
t": 1, "max": 319, "sum": 319.0, "min": 319}, "Number of Records Since Last R
eset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "Total Ba
tches Seen": {"count": 1, "max": 2553, "sum": 2553.0, "min": 2553}, "Total Re
cords Seen": {"count": 1, "max": 1305024, "sum": 1305024.0, "min": 1305024},
 "Max Records Seen Between Resets": {"count": 1, "max": 163064, "sum": 16306
4.0, "min": 163064}, "Reset Count": {"count": 1, "max": 9, "sum": 9.0, "min":
9}}, "EndTime": 1605111161.708772, "Dimensions": {"Host": "algo-1", "Meta":
 "training_data_iter", "Operation": "training", "Algorithm": "factorization-m
achines", "epoch": 7}, "StartTime": 1605111159.673848}

[11/11/2020 16:12:41 INFO 139996470646592] #throughput_metric: host=algo-1, t
rain throughput=80127.801873 records/second
[11/11/2020 16:12:41 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=8, batch=0 train binary_classification_accuracy <score>=0.689453125
[11/11/2020 16:12:41 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=8, batch=0 train binary_classification_cross_entropy <loss>=0.662618756294
[11/11/2020 16:12:41 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=8, batch=0 train binary_f_1.000 <score>=0.747217806041
[2020-11-11 16:12:43.781] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 18, "duration": 2070, "num_examples": 319,
 "num_bytes": 10436096}
[11/11/2020 16:12:43 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=8, train binary_classification_accuracy <score>=0.69570434953
[11/11/2020 16:12:43 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=8, train binary_classification_cross_entropy <loss>=0.640253493405
[11/11/2020 16:12:43 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=8, train binary_f_1.000 <score>=0.682342865178
#metrics {"Metrics": {"update.time": {"count": 1, "max": 2072.9598999023438,
 "sum": 2072.9598999023438, "min": 2072.9598999023438}}, "EndTime": 160511116
3.782053, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605111161.708607}

[11/11/2020 16:12:43 INFO 139996470646592] #progress_metric: host=algo-1, com
```

```
pleted 36 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 319, "sum": 319.0, "min": 319}, "Number of Batches Since Last Reset": {"coun
t": 1, "max": 319, "sum": 319.0, "min": 319}, "Number of Records Since Last R
eset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "Total Ba
tches Seen": {"count": 1, "max": 2872, "sum": 2872.0, "min": 2872}, "Total Re
cords Seen": {"count": 1, "max": 1468088, "sum": 1468088.0, "min": 1468088},
 "Max Records Seen Between Resets": {"count": 1, "max": 163064, "sum": 16306
4.0, "min": 163064}, "Reset Count": {"count": 1, "max": 10, "sum": 10.0, "mi
n": 10}}, "EndTime": 1605111163.782295, "Dimensions": {"Host": "algo-1", "Met
a": "training_data_iter", "Operation": "training", "Algorithm": "factorizatio
n-machines", "epoch": 8}, "StartTime": 1605111161.709058}

[11/11/2020 16:12:43 INFO 139996470646592] #throughput_metric: host=algo-1, t
rain throughput=78646.5791493 records/second
[11/11/2020 16:12:43 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=9, batch=0 train binary_classification_accuracy <score>=0.708984375
[11/11/2020 16:12:43 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=9, batch=0 train binary_classification_cross_entropy <loss>=0.656731307507
[11/11/2020 16:12:43 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=9, batch=0 train binary_f_1.000 <score>=0.767550702028
[2020-11-11 16:12:45.833] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 20, "duration": 2049, "num_examples": 319,
 "num_bytes": 10436096}
[11/11/2020 16:12:45 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=9, train binary_classification_accuracy <score>=0.704398510972
[11/11/2020 16:12:45 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=9, train binary_classification_cross_entropy <loss>=0.635953472513
[11/11/2020 16:12:45 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=9, train binary_f_1.000 <score>=0.692142884471
#metrics {"Metrics": {"update.time": {"count": 1, "max": 2051.3060092926025,
 "sum": 2051.3060092926025, "min": 2051.3060092926025}}, "EndTime": 160511116
5.8339, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605111163.782142}

[11/11/2020 16:12:45 INFO 139996470646592] #progress_metric: host=algo-1, com
pleted 40 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 319, "sum": 319.0, "min": 319}, "Number of Batches Since Last Reset": {"coun
t": 1, "max": 319, "sum": 319.0, "min": 319}, "Number of Records Since Last R
eset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "Total Ba
tches Seen": {"count": 1, "max": 3191, "sum": 3191.0, "min": 3191}, "Total Re
cords Seen": {"count": 1, "max": 1631152, "sum": 1631152.0, "min": 1631152},
 "Max Records Seen Between Resets": {"count": 1, "max": 163064, "sum": 16306
4.0, "min": 163064}, "Reset Count": {"count": 1, "max": 11, "sum": 11.0, "mi
n": 11}}, "EndTime": 1605111165.834135, "Dimensions": {"Host": "algo-1", "Met
a": "training_data_iter", "Operation": "training", "Algorithm": "factorizatio
n-machines", "epoch": 9}, "StartTime": 1605111163.78256}

[11/11/2020 16:12:45 INFO 139996470646592] #throughput_metric: host=algo-1, t
rain throughput=79477.00327 records/second
[11/11/2020 16:12:45 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=10, batch=0 train binary_classification_accuracy <score>=0.720703125
[11/11/2020 16:12:45 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=10, batch=0 train binary_classification_cross_entropy <loss>=0.651121854782
[11/11/2020 16:12:45 INFO 139996470646592] #quality_metric: host=algo-1, epoc
```

```
h=10, batch=0 train binary_f_1.000 <score>=0.779661016949
[2020-11-11 16:12:47.858] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 22, "duration": 2022, "num_examples": 319,
 "num_bytes": 10436096}
[11/11/2020 16:12:47 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=10, train binary_classification_accuracy <score>=0.71127424569
[11/11/2020 16:12:47 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=10, train binary_classification_cross_entropy <loss>=0.631937614616
[11/11/2020 16:12:47 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=10, train binary_f_1.000 <score>=0.699944642755
#metrics {"Metrics": {"update.time": {"count": 1, "max": 2024.9691009521484,
 "sum": 2024.9691009521484, "min": 2024.9691009521484}}, "EndTime": 160511116
7.859401, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605111165.833983}

[11/11/2020 16:12:47 INFO 139996470646592] #progress_metric: host=algo-1, com
pleted 44 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 319, "sum": 319.0, "min": 319}, "Number of Batches Since Last Reset": {"coun
t": 1, "max": 319, "sum": 319.0, "min": 319}, "Number of Records Since Last R
eset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "Total Ba
tches Seen": {"count": 1, "max": 3510, "sum": 3510.0, "min": 3510}, "Total Re
cords Seen": {"count": 1, "max": 1794216, "sum": 1794216.0, "min": 1794216},
 "Max Records Seen Between Resets": {"count": 1, "max": 163064, "sum": 16306
4.0, "min": 163064}, "Reset Count": {"count": 1, "max": 12, "sum": 12.0, "mi
n": 12}}, "EndTime": 1605111167.859637, "Dimensions": {"Host": "algo-1", "Met
a": "training_data_iter", "Operation": "training", "Algorithm": "factorizatio
n-machines", "epoch": 10}, "StartTime": 1605111165.834397}

[11/11/2020 16:12:47 INFO 139996470646592] #throughput_metric: host=algo-1, t
rain throughput=80510.4039347 records/second
[11/11/2020 16:12:47 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=11, batch=0 train binary_classification_accuracy <score>=0.728515625
[11/11/2020 16:12:47 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=11, batch=0 train binary_classification_cross_entropy <loss>=0.645758569241
[11/11/2020 16:12:47 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=11, batch=0 train binary_f_1.000 <score>=0.787136294028
[2020-11-11 16:12:49.956] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 24, "duration": 2094, "num_examples": 319,
 "num_bytes": 10436096}
[11/11/2020 16:12:49 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=11, train binary_classification_accuracy <score>=0.717109130094
[11/11/2020 16:12:49 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=11, train binary_classification_cross_entropy <loss>=0.628162963824
[11/11/2020 16:12:49 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=11, train binary_f_1.000 <score>=0.706384006304
#metrics {"Metrics": {"update.time": {"count": 1, "max": 2096.641778945923,
 "sum": 2096.641778945923, "min": 2096.641778945923}}, "EndTime": 1605111169.
956585, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605111167.859483}

[11/11/2020 16:12:49 INFO 139996470646592] #progress_metric: host=algo-1, com
pleted 48 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 319, "sum": 319.0, "min": 319}, "Number of Batches Since Last Reset": {"coun
t": 1, "max": 319, "sum": 319.0, "min": 319}, "Number of Records Since Last R
```

```
eset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "Total Ba
tches Seen": {"count": 1, "max": 3829, "sum": 3829.0, "min": 3829}, "Total Re
cords Seen": {"count": 1, "max": 1957280, "sum": 1957280.0, "min": 1957280},
 "Max Records Seen Between Resets": {"count": 1, "max": 163064, "sum": 16306
4.0, "min": 163064}, "Reset Count": {"count": 1, "max": 13, "sum": 13.0, "mi
n": 13}}, "EndTime": 1605111169.956848, "Dimensions": {"Host": "algo-1", "Met
a": "training_data_iter", "Operation": "training", "Algorithm": "factorizatio
n-machines", "epoch": 11}, "StartTime": 1605111167.859909}

[11/11/2020 16:12:49 INFO 139996470646592] #throughput_metric: host=algo-1, t
rain throughput=77757.8740969 records/second
[11/11/2020 16:12:49 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=12, batch=0 train binary_classification_accuracy <score>=0.740234375
[11/11/2020 16:12:49 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=12, batch=0 train binary_classification_cross_entropy <loss>=0.640614449978
[11/11/2020 16:12:49 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=12, batch=0 train binary_f_1.000 <score>=0.798789712557
[2020-11-11 16:12:52.001] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 26, "duration": 2042, "num_examples": 319,
 "num_bytes": 10436096}
[11/11/2020 16:12:52 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=12, train binary_classification_accuracy <score>=0.722521551724
[11/11/2020 16:12:52 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=12, train binary_classification_cross_entropy <loss>=0.624594498204
[11/11/2020 16:12:52 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=12, train binary_f_1.000 <score>=0.712374497036
#metrics {"Metrics": {"update.time": {"count": 1, "max": 2044.9011325836182,
 "sum": 2044.9011325836182, "min": 2044.9011325836182}}, "EndTime": 160511117
2.00204, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605111169.956686}

[11/11/2020 16:12:52 INFO 139996470646592] #progress_metric: host=algo-1, com
pleted 52 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 319, "sum": 319.0, "min": 319}, "Number of Batches Since Last Reset": {"coun
t": 1, "max": 319, "sum": 319.0, "min": 319}, "Number of Records Since Last R
eset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "Total Ba
tches Seen": {"count": 1, "max": 4148, "sum": 4148.0, "min": 4148}, "Total Re
cords Seen": {"count": 1, "max": 2120344, "sum": 2120344.0, "min": 2120344},
 "Max Records Seen Between Resets": {"count": 1, "max": 163064, "sum": 16306
4.0, "min": 163064}, "Reset Count": {"count": 1, "max": 14, "sum": 14.0, "mi
n": 14}}, "EndTime": 1605111172.002302, "Dimensions": {"Host": "algo-1", "Met
a": "training_data_iter", "Operation": "training", "Algorithm": "factorizatio
n-machines", "epoch": 12}, "StartTime": 1605111169.957108}

[11/11/2020 16:12:52 INFO 139996470646592] #throughput_metric: host=algo-1, t
rain throughput=79724.760785 records/second
[11/11/2020 16:12:52 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=13, batch=0 train binary_classification_accuracy <score>=0.7421875
[11/11/2020 16:12:52 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=13, batch=0 train binary_classification_cross_entropy <loss>=0.635666489601
[11/11/2020 16:12:52 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=13, batch=0 train binary_f_1.000 <score>=0.801204819277
[2020-11-11 16:12:54.038] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 28, "duration": 2033, "num_examples": 319,
 "num_bytes": 10436096}
```

```
[11/11/2020 16:12:54 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=13, train binary_classification_accuracy <score>=0.727860501567
[11/11/2020 16:12:54 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=13, train binary_classification_cross_entropy <loss>=0.621203468324
[11/11/2020 16:12:54 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=13, train binary_f_1.000 <score>=0.718266293118
#metrics {"Metrics": {"update.time": {"count": 1, "max": 2035.965919494629,
 "sum": 2035.965919494629, "min": 2035.965919494629}}, "EndTime": 1605111174.
038619, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605111172.00211}

[11/11/2020 16:12:54 INFO 139996470646592] #progress_metric: host=algo-1, com
pleted 56 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 319, "sum": 319.0, "min": 319}, "Number of Batches Since Last Reset": {"coun
t": 1, "max": 319, "sum": 319.0, "min": 319}, "Number of Records Since Last R
eset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "Total Ba
tches Seen": {"count": 1, "max": 4467, "sum": 4467.0, "min": 4467}, "Total Re
cords Seen": {"count": 1, "max": 2283408, "sum": 2283408.0, "min": 2283408},
 "Max Records Seen Between Resets": {"count": 1, "max": 163064, "sum": 16306
4.0, "min": 163064}, "Reset Count": {"count": 1, "max": 15, "sum": 15.0, "mi
n": 15}}, "EndTime": 1605111174.038871, "Dimensions": {"Host": "algo-1", "Met
a": "training_data_iter", "Operation": "training", "Algorithm": "factorizatio
n-machines", "epoch": 13}, "StartTime": 1605111172.002608}

[11/11/2020 16:12:54 INFO 139996470646592] #throughput_metric: host=algo-1, t
rain throughput=80074.5258892 records/second
[11/11/2020 16:12:54 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=14, batch=0 train binary_classification_accuracy <score>=0.75390625
[11/11/2020 16:12:54 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=14, batch=0 train binary_classification_cross_entropy <loss>=0.630895256996
[11/11/2020 16:12:54 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=14, batch=0 train binary_f_1.000 <score>=0.811940298507
[2020-11-11 16:12:56.109] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 30, "duration": 2068, "num_examples": 319,
 "num_bytes": 10436096}
[11/11/2020 16:12:56 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=14, train binary_classification_accuracy <score>=0.732611677116
[11/11/2020 16:12:56 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=14, train binary_classification_cross_entropy <loss>=0.617966106319
[11/11/2020 16:12:56 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=14, train binary_f_1.000 <score>=0.723479428116
#metrics {"Metrics": {"update.time": {"count": 1, "max": 2070.6448554992676,
 "sum": 2070.6448554992676, "min": 2070.6448554992676}}, "EndTime": 160511117
6.109807, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605111174.038708}

[11/11/2020 16:12:56 INFO 139996470646592] #progress_metric: host=algo-1, com
pleted 60 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 319, "sum": 319.0, "min": 319}, "Number of Batches Since Last Reset": {"coun
t": 1, "max": 319, "sum": 319.0, "min": 319}, "Number of Records Since Last R
eset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "Total Ba
tches Seen": {"count": 1, "max": 4786, "sum": 4786.0, "min": 4786}, "Total Re
cords Seen": {"count": 1, "max": 2446472, "sum": 2446472.0, "min": 2446472},
 "Max Records Seen Between Resets": {"count": 1, "max": 163064, "sum": 16306
```

```
4.0, "min": 163064}, "Reset Count": {"count": 1, "max": 16, "sum": 16.0, "mi
n": 16}}, "EndTime": 1605111176.110105, "Dimensions": {"Host": "algo-1", "Met
a": "training_data_iter", "Operation": "training", "Algorithm": "factorizatio
n-machines", "epoch": 14}, "StartTime": 1605111174.039124}

[11/11/2020 16:12:56 INFO 139996470646592] #throughput_metric: host=algo-1, t
rain throughput=78731.4091472 records/second
[11/11/2020 16:12:56 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=15, batch=0 train binary_classification_accuracy <score>=0.7578125
[11/11/2020 16:12:56 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=15, batch=0 train binary_classification_cross_entropy <loss>=0.626283764839
[11/11/2020 16:12:56 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=15, batch=0 train binary_f_1.000 <score>=0.816023738872
[2020-11-11 16:12:58.160] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 32, "duration": 2048, "num_examples": 319,
 "num_bytes": 10436096}
[11/11/2020 16:12:58 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=15, train binary_classification_accuracy <score>=0.737203663793
[11/11/2020 16:12:58 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=15, train binary_classification_cross_entropy <loss>=0.614862639889
[11/11/2020 16:12:58 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=15, train binary_f_1.000 <score>=0.728482686201
#metrics {"Metrics": {"update.time": {"count": 1, "max": 2050.997018814087,
 "sum": 2050.997018814087, "min": 2050.997018814087}}, "EndTime": 1605111178.
161471, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605111176.109902}

[11/11/2020 16:12:58 INFO 139996470646592] #progress_metric: host=algo-1, com
pleted 64 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 319, "sum": 319.0, "min": 319}, "Number of Batches Since Last Reset": {"coun
t": 1, "max": 319, "sum": 319.0, "min": 319}, "Number of Records Since Last R
eset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "Total Ba
tches Seen": {"count": 1, "max": 5105, "sum": 5105.0, "min": 5105}, "Total Re
cords Seen": {"count": 1, "max": 2609536, "sum": 2609536.0, "min": 2609536},
 "Max Records Seen Between Resets": {"count": 1, "max": 163064, "sum": 16306
4.0, "min": 163064}, "Reset Count": {"count": 1, "max": 17, "sum": 17.0, "mi
n": 17}}, "EndTime": 1605111178.161695, "Dimensions": {"Host": "algo-1", "Met
a": "training_data_iter", "Operation": "training", "Algorithm": "factorizatio
n-machines", "epoch": 15}, "StartTime": 1605111176.110439}

[11/11/2020 16:12:58 INFO 139996470646592] #throughput_metric: host=algo-1, t
rain throughput=79489.3624203 records/second
[11/11/2020 16:12:58 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=16, batch=0 train binary_classification_accuracy <score>=0.765625
[11/11/2020 16:12:58 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=16, batch=0 train binary_classification_cross_entropy <loss>=0.621817708015
[11/11/2020 16:12:58 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=16, batch=0 train binary_f_1.000 <score>=0.823008849558
[2020-11-11 16:13:00.188] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 34, "duration": 2025, "num_examples": 319,
 "num_bytes": 10436096}
[11/11/2020 16:13:00 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=16, train binary_classification_accuracy <score>=0.741244612069
[11/11/2020 16:13:00 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=16, train binary_classification_cross_entropy <loss>=0.611876536313
```

```
[11/11/2020 16:13:00 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=16, train binary_f_1.000 <score>=0.732907792454
#metrics {"Metrics": {"update.time": {"count": 1, "max": 2027.4839401245117,
 "sum": 2027.4839401245117, "min": 2027.4839401245117}}, "EndTime": 160511118
0.189464, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605111178.161544}

[11/11/2020 16:13:00 INFO 139996470646592] #progress_metric: host=algo-1, com
pleted 68 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 319, "sum": 319.0, "min": 319}, "Number of Batches Since Last Reset": {"coun
t": 1, "max": 319, "sum": 319.0, "min": 319}, "Number of Records Since Last R
eset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "Total Ba
tches Seen": {"count": 1, "max": 5424, "sum": 5424.0, "min": 5424}, "Total Re
cords Seen": {"count": 1, "max": 2772600, "sum": 2772600.0, "min": 2772600},
 "Max Records Seen Between Resets": {"count": 1, "max": 163064, "sum": 16306
4.0, "min": 163064}, "Reset Count": {"count": 1, "max": 18, "sum": 18.0, "mi
n": 18}}, "EndTime": 1605111180.189801, "Dimensions": {"Host": "algo-1", "Met
a": "training_data_iter", "Operation": "training", "Algorithm": "factorizatio
n-machines", "epoch": 16}, "StartTime": 1605111178.161946}

[11/11/2020 16:13:00 INFO 139996470646592] #throughput_metric: host=algo-1, t
rain throughput=80406.2408543 records/second
[11/11/2020 16:13:00 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=17, batch=0 train binary_classification_accuracy <score>=0.775390625
[11/11/2020 16:13:00 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=17, batch=0 train binary_classification_cross_entropy <loss>=0.61748456955
[11/11/2020 16:13:00 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=17, batch=0 train binary_f_1.000 <score>=0.831625183016
[2020-11-11 16:13:02.308] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 36, "duration": 2116, "num_examples": 319,
 "num_bytes": 10436096}
[11/11/2020 16:13:02 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=17, train binary_classification_accuracy <score>=0.745181475313
[11/11/2020 16:13:02 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=17, train binary_classification_cross_entropy <loss>=0.608993841376
[11/11/2020 16:13:02 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=17, train binary_f_1.000 <score>=0.737126001907
#metrics {"Metrics": {"update.time": {"count": 1, "max": 2118.8809871673584,
 "sum": 2118.8809871673584, "min": 2118.8809871673584}}, "EndTime": 160511118
2.308999, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605111180.189603}

[11/11/2020 16:13:02 INFO 139996470646592] #progress_metric: host=algo-1, com
pleted 72 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 319, "sum": 319.0, "min": 319}, "Number of Batches Since Last Reset": {"coun
t": 1, "max": 319, "sum": 319.0, "min": 319}, "Number of Records Since Last R
eset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "Total Ba
tches Seen": {"count": 1, "max": 5743, "sum": 5743.0, "min": 5743}, "Total Re
cords Seen": {"count": 1, "max": 2935664, "sum": 2935664.0, "min": 2935664},
 "Max Records Seen Between Resets": {"count": 1, "max": 163064, "sum": 16306
4.0, "min": 163064}, "Reset Count": {"count": 1, "max": 19, "sum": 19.0, "mi
n": 19}}, "EndTime": 1605111182.309269, "Dimensions": {"Host": "algo-1", "Met
a": "training_data_iter", "Operation": "training", "Algorithm": "factorizatio
n-machines", "epoch": 17}, "StartTime": 1605111180.19008}
```

```
[11/11/2020 16:13:02 INFO 139996470646592] #throughput_metric: host=algo-1, t
rain throughput=76941.3346731 records/second
[11/11/2020 16:13:02 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=18, batch=0 train binary_classification_accuracy <score>=0.78515625
[11/11/2020 16:13:02 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=18, batch=0 train binary_classification_cross_entropy <loss>=0.613273382187
[11/11/2020 16:13:02 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=18, batch=0 train binary_f_1.000 <score>=0.84011627907
[2020-11-11 16:13:04.332] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 38, "duration": 2021, "num_examples": 319,
 "num_bytes": 10436096}
[11/11/2020 16:13:04 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=18, train binary_classification_accuracy <score>=0.748989762931
[11/11/2020 16:13:04 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=18, train binary_classification_cross_entropy <loss>=0.60620273673
[11/11/2020 16:13:04 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=18, train binary_f_1.000 <score>=0.74116584907
#metrics {"Metrics": {"update.time": {"count": 1, "max": 2023.772954940796,
 "sum": 2023.772954940796, "min": 2023.772954940796}}, "EndTime": 1605111184.
333378, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605111182.309079}

[11/11/2020 16:13:04 INFO 139996470646592] #progress_metric: host=algo-1, com
pleted 76 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 319, "sum": 319.0, "min": 319}, "Number of Batches Since Last Reset": {"coun
t": 1, "max": 319, "sum": 319.0, "min": 319}, "Number of Records Since Last R
eset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "Total Ba
tches Seen": {"count": 1, "max": 6062, "sum": 6062.0, "min": 6062}, "Total Re
cords Seen": {"count": 1, "max": 3098728, "sum": 3098728.0, "min": 3098728},
 "Max Records Seen Between Resets": {"count": 1, "max": 163064, "sum": 16306
4.0, "min": 163064}, "Reset Count": {"count": 1, "max": 20, "sum": 20.0, "mi
n": 20}}, "EndTime": 1605111184.333705, "Dimensions": {"Host": "algo-1", "Met
a": "training_data_iter", "Operation": "training", "Algorithm": "factorizatio
n-machines", "epoch": 18}, "StartTime": 1605111182.30957}

[11/11/2020 16:13:04 INFO 139996470646592] #throughput_metric: host=algo-1, t
rain throughput=80554.5925336 records/second
[11/11/2020 16:13:04 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=19, batch=0 train binary_classification_accuracy <score>=0.7890625
[11/11/2020 16:13:04 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=19, batch=0 train binary_classification_cross_entropy <loss>=0.609174728394
[11/11/2020 16:13:04 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=19, batch=0 train binary_f_1.000 <score>=0.843930635838
[2020-11-11 16:13:06.337] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 40, "duration": 2001, "num_examples": 319,
 "num_bytes": 10436096}
[11/11/2020 16:13:06 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=19, train binary_classification_accuracy <score>=0.752247012147
[11/11/2020 16:13:06 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=19, train binary_classification_cross_entropy <loss>=0.603493140596
[11/11/2020 16:13:06 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=19, train binary_f_1.000 <score>=0.744576229462
#metrics {"Metrics": {"update.time": {"count": 1, "max": 2003.9710998535156,
 "sum": 2003.9710998535156, "min": 2003.9710998535156}}, "EndTime": 160511118
```

```
6.337957, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605111184.333511}

[11/11/2020 16:13:06 INFO 139996470646592] #progress_metric: host=algo-1, com
pleted 80 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 319, "sum": 319.0, "min": 319}, "Number of Batches Since Last Reset": {"coun
t": 1, "max": 319, "sum": 319.0, "min": 319}, "Number of Records Since Last R
eset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "Total Ba
tches Seen": {"count": 1, "max": 6381, "sum": 6381.0, "min": 6381}, "Total Re
cords Seen": {"count": 1, "max": 3261792, "sum": 3261792.0, "min": 3261792},
 "Max Records Seen Between Resets": {"count": 1, "max": 163064, "sum": 16306
4.0, "min": 163064}, "Reset Count": {"count": 1, "max": 21, "sum": 21.0, "mi
n": 21}}, "EndTime": 1605111186.338184, "Dimensions": {"Host": "algo-1", "Met
a": "training_data_iter", "Operation": "training", "Algorithm": "factorizatio
n-machines", "epoch": 19}, "StartTime": 1605111184.333952}

[11/11/2020 16:13:06 INFO 139996470646592] #throughput_metric: host=algo-1, t
rain throughput=81354.8903999 records/second
[11/11/2020 16:13:06 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=20, batch=0 train binary_classification_accuracy <score>=0.79296875
[11/11/2020 16:13:06 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=20, batch=0 train binary_classification_cross_entropy <loss>=0.605180442333
[11/11/2020 16:13:06 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=20, batch=0 train binary_f_1.000 <score>=0.847701149425
[2020-11-11 16:13:08.459] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 42, "duration": 2119, "num_examples": 319,
 "num_bytes": 10436096}
[11/11/2020 16:13:08 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=20, train binary_classification_accuracy <score>=0.755424666928
[11/11/2020 16:13:08 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=20, train binary_classification_cross_entropy <loss>=0.600856415717
[11/11/2020 16:13:08 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=20, train binary_f_1.000 <score>=0.747866592608
#metrics {"Metrics": {"update.time": {"count": 1, "max": 2121.9470500946045,
 "sum": 2121.9470500946045, "min": 2121.9470500946045}}, "EndTime": 160511118
8.460413, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605111186.338031}

[11/11/2020 16:13:08 INFO 139996470646592] #progress_metric: host=algo-1, com
pleted 84 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 319, "sum": 319.0, "min": 319}, "Number of Batches Since Last Reset": {"coun
t": 1, "max": 319, "sum": 319.0, "min": 319}, "Number of Records Since Last R
eset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "Total Ba
tches Seen": {"count": 1, "max": 6700, "sum": 6700.0, "min": 6700}, "Total Re
cords Seen": {"count": 1, "max": 3424856, "sum": 3424856.0, "min": 3424856},
 "Max Records Seen Between Resets": {"count": 1, "max": 163064, "sum": 16306
4.0, "min": 163064}, "Reset Count": {"count": 1, "max": 22, "sum": 22.0, "mi
n": 22}}, "EndTime": 1605111188.460749, "Dimensions": {"Host": "algo-1", "Met
a": "training_data_iter", "Operation": "training", "Algorithm": "factorizatio
n-machines", "epoch": 20}, "StartTime": 1605111186.338426}

[11/11/2020 16:13:08 INFO 139996470646592] #throughput_metric: host=algo-1, t
rain throughput=76827.9480854 records/second
[11/11/2020 16:13:08 INFO 139996470646592] #quality_metric: host=algo-1, epoc
```

```
h=21, batch=0 train binary_classification_accuracy <score>=0.796875
[11/11/2020 16:13:08 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=21, batch=0 train binary_classification_cross_entropy <loss>=0.601283311844
[11/11/2020 16:13:08 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=21, batch=0 train binary_f_1.000 <score>=0.850574712644
[2020-11-11 16:13:10.462] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 44, "duration": 1999, "num_examples": 319,
 "num_bytes": 10436096}
[11/11/2020 16:13:10 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=21, train binary_classification_accuracy <score>=0.758449255486
[11/11/2020 16:13:10 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=21, train binary_classification_cross_entropy <loss>=0.598285121978
[11/11/2020 16:13:10 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=21, train binary_f_1.000 <score>=0.751072636414
#metrics {"Metrics": {"update.time": {"count": 1, "max": 2001.9700527191162,
 "sum": 2001.9700527191162, "min": 2001.9700527191162}}, "EndTime": 160511119
0.463048, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605111188.460501}

[11/11/2020 16:13:10 INFO 139996470646592] #progress_metric: host=algo-1, com
pleted 88 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 319, "sum": 319.0, "min": 319}, "Number of Batches Since Last Reset": {"coun
t": 1, "max": 319, "sum": 319.0, "min": 319}, "Number of Records Since Last R
eset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "Total Ba
tches Seen": {"count": 1, "max": 7019, "sum": 7019.0, "min": 7019}, "Total Re
cords Seen": {"count": 1, "max": 3587920, "sum": 3587920.0, "min": 3587920},
 "Max Records Seen Between Resets": {"count": 1, "max": 163064, "sum": 16306
4.0, "min": 163064}, "Reset Count": {"count": 1, "max": 23, "sum": 23.0, "mi
n": 23}}, "EndTime": 1605111190.463315, "Dimensions": {"Host": "algo-1", "Met
a": "training_data_iter", "Operation": "training", "Algorithm": "factorizatio
n-machines", "epoch": 21}, "StartTime": 1605111188.461039}

[11/11/2020 16:13:10 INFO 139996470646592] #throughput_metric: host=algo-1, t
rain throughput=81433.3027681 records/second
[11/11/2020 16:13:10 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=22, batch=0 train binary_classification_accuracy <score>=0.798828125
[11/11/2020 16:13:10 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=22, batch=0 train binary_classification_cross_entropy <loss>=0.597476959229
[11/11/2020 16:13:10 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=22, batch=0 train binary_f_1.000 <score>=0.852646638054
[2020-11-11 16:13:12.476] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 46, "duration": 2011, "num_examples": 319,
 "num_bytes": 10436096}
[11/11/2020 16:13:12 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=22, train binary_classification_accuracy <score>=0.761522825235
[11/11/2020 16:13:12 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=22, train binary_classification_cross_entropy <loss>=0.595772825625
[11/11/2020 16:13:12 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=22, train binary_f_1.000 <score>=0.754295879488
#metrics {"Metrics": {"update.time": {"count": 1, "max": 2013.239860534668,
 "sum": 2013.239860534668, "min": 2013.239860534668}}, "EndTime": 1605111192.
476918, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605111190.463124}

[11/11/2020 16:13:12 INFO 139996470646592] #progress_metric: host=algo-1, com
```

```
pleted 92 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 319, "sum": 319.0, "min": 319}, "Number of Batches Since Last Reset": {"coun
t": 1, "max": 319, "sum": 319.0, "min": 319}, "Number of Records Since Last R
eset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "Total Ba
tches Seen": {"count": 1, "max": 7338, "sum": 7338.0, "min": 7338}, "Total Re
cords Seen": {"count": 1, "max": 3750984, "sum": 3750984.0, "min": 3750984},
 "Max Records Seen Between Resets": {"count": 1, "max": 163064, "sum": 16306
4.0, "min": 163064}, "Reset Count": {"count": 1, "max": 24, "sum": 24.0, "mi
n": 24}}, "EndTime": 1605111192.477198, "Dimensions": {"Host": "algo-1", "Met
a": "training_data_iter", "Operation": "training", "Algorithm": "factorizatio
n-machines", "epoch": 22}, "StartTime": 1605111190.46364}

[11/11/2020 16:13:12 INFO 139996470646592] #throughput_metric: host=algo-1, t
rain throughput=80976.221739 records/second
[11/11/2020 16:13:12 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=23, batch=0 train binary_classification_accuracy <score>=0.798828125
[11/11/2020 16:13:12 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=23, batch=0 train binary_classification_cross_entropy <loss>=0.59375590086
[11/11/2020 16:13:12 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=23, batch=0 train binary_f_1.000 <score>=0.852646638054
[2020-11-11 16:13:14.499] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 48, "duration": 2020, "num_examples": 319,
 "num_bytes": 10436096}
[11/11/2020 16:13:14 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=23, train binary_classification_accuracy <score>=0.764461696708
[11/11/2020 16:13:14 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=23, train binary_classification_cross_entropy <loss>=0.593313920647
[11/11/2020 16:13:14 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=23, train binary_f_1.000 <score>=0.757409509396
#metrics {"Metrics": {"update.time": {"count": 1, "max": 2022.8431224822998,
 "sum": 2022.8431224822998, "min": 2022.8431224822998}}, "EndTime": 160511119
4.500398, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605111192.477004}

[11/11/2020 16:13:14 INFO 139996470646592] #progress_metric: host=algo-1, com
pleted 96 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 319, "sum": 319.0, "min": 319}, "Number of Batches Since Last Reset": {"coun
t": 1, "max": 319, "sum": 319.0, "min": 319}, "Number of Records Since Last R
eset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "Total Ba
tches Seen": {"count": 1, "max": 7657, "sum": 7657.0, "min": 7657}, "Total Re
cords Seen": {"count": 1, "max": 3914048, "sum": 3914048.0, "min": 3914048},
 "Max Records Seen Between Resets": {"count": 1, "max": 163064, "sum": 16306
4.0, "min": 163064}, "Reset Count": {"count": 1, "max": 25, "sum": 25.0, "mi
n": 25}}, "EndTime": 1605111194.50067, "Dimensions": {"Host": "algo-1", "Met
a": "training_data_iter", "Operation": "training", "Algorithm": "factorizatio
n-machines", "epoch": 23}, "StartTime": 1605111192.477515}

[11/11/2020 16:13:14 INFO 139996470646592] #throughput_metric: host=algo-1, t
rain throughput=80593.007714 records/second
[11/11/2020 16:13:14 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=24, batch=0 train binary_classification_accuracy <score>=0.798828125
[11/11/2020 16:13:14 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=24, batch=0 train binary_classification_cross_entropy <loss>=0.590115129948
[11/11/2020 16:13:14 INFO 139996470646592] #quality_metric: host=algo-1, epoc
```

```
h=24, batch=0 train binary_f_1.000 <score>=0.852646638054

2020-11-11 16:13:25 Uploading - Uploading generated training model
2020-11-11 16:13:25 Completed - Training job completed
[2020-11-11 16:13:16.513] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 50, "duration": 2010, "num_examples": 319,
 "num_bytes": 10436096}
[11/11/2020 16:13:16 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=24, train binary_classification_accuracy <score>=0.767461794671
[11/11/2020 16:13:16 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=24, train binary_classification_cross_entropy <loss>=0.590903534971
[11/11/2020 16:13:16 INFO 139996470646592] #quality_metric: host=algo-1, epoc
h=24, train binary_f_1.000 <score>=0.760580959946
[11/11/2020 16:13:16 INFO 139996470646592] #quality_metric: host=algo-1, trai
n binary_classification_accuracy <score>=0.767461794671
[11/11/2020 16:13:16 INFO 139996470646592] #quality_metric: host=algo-1, trai
n binary_classification_cross_entropy <loss>=0.590903534971
[11/11/2020 16:13:16 INFO 139996470646592] #quality_metric: host=algo-1, trai
n binary_f_1.000 <score>=0.760580959946
#metrics {"Metrics": {"update.time": {"count": 1, "max": 2013.6010646820068,
 "sum": 2013.6010646820068, "min": 2013.6010646820068}}, "EndTime": 160511119
6.514611, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorith
m": "factorization-machines"}, "StartTime": 1605111194.500493}

[11/11/2020 16:13:16 INFO 139996470646592] #progress_metric: host=algo-1, com
pleted 100 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 319, "sum": 319.0, "min": 319}, "Number of Batches Since Last Reset": {"coun
t": 1, "max": 319, "sum": 319.0, "min": 319}, "Number of Records Since Last R
eset": {"count": 1, "max": 163064, "sum": 163064.0, "min": 163064}, "Total Ba
tches Seen": {"count": 1, "max": 7976, "sum": 7976.0, "min": 7976}, "Total Re
cords Seen": {"count": 1, "max": 4077112, "sum": 4077112.0, "min": 4077112},
 "Max Records Seen Between Resets": {"count": 1, "max": 163064, "sum": 16306
4.0, "min": 163064}, "Reset Count": {"count": 1, "max": 26, "sum": 26.0, "mi
n": 26}}, "EndTime": 1605111196.514874, "Dimensions": {"Host": "algo-1", "Met
a": "training_data_iter", "Operation": "training", "Algorithm": "factorizatio
n-machines", "epoch": 24}, "StartTime": 1605111194.500971}

[11/11/2020 16:13:16 INFO 139996470646592] #throughput_metric: host=algo-1, t
rain throughput=80963.9997566 records/second
[11/11/2020 16:13:16 WARNING 139996470646592] wait_for_all_workers will not s
ync workers since the kv store is not running distributed
[11/11/2020 16:13:16 INFO 139996470646592] Pulling entire model from kvstore
 to finalize
#metrics {"Metrics": {"finalize.time": {"count": 1, "max": 4.62794303894043,
 "sum": 4.62794303894043, "min": 4.62794303894043}}, "EndTime": 1605111196.51
9839, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm":
 "factorization-machines"}, "StartTime": 1605111196.514693}

[11/11/2020 16:13:16 INFO 139996470646592] Saved checkpoint to "/tmp/tmpCV9_K
q/state-0001.params"
[2020-11-11 16:13:16.566] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/test", "epoch": 0, "duration": 51889, "num_examples": 1, "nu
m_bytes": 32768}
[2020-11-11 16:13:16.784] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/test", "epoch": 1, "duration": 217, "num_examples": 21, "num
```

```
_bytes": 667904}
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 21, "sum": 21.0, "min": 21}, "Number of Batches Since Last Reset": {"count":
1, "max": 21, "sum": 21.0, "min": 21}, "Number of Records Since Last Reset":
 {"count": 1, "max": 10436, "sum": 10436.0, "min": 10436}, "Total Batches See
n": {"count": 1, "max": 21, "sum": 21.0, "min": 21}, "Total Records Seen":
 {"count": 1, "max": 10436, "sum": 10436.0, "min": 10436}, "Max Records Seen
 Between Resets": {"count": 1, "max": 10436, "sum": 10436.0, "min": 10436},
 "Reset Count": {"count": 1, "max": 1, "sum": 1.0, "min": 1}}, "EndTime": 160
5111196.784648, "Dimensions": {"Host": "algo-1", "Meta": "test_data_iter", "O
peration": "training", "Algorithm": "factorization-machines"}, "StartTime": 1
605111196.566549}

[11/11/2020 16:13:16 INFO 139996470646592] #test_score (algo-1) : ('binary_cl
assification_accuracy', 0.6946147949405903)
[11/11/2020 16:13:16 INFO 139996470646592] #test_score (algo-1) : ('binary_cl
assification_cross_entropy', 0.6316653317809333)
[11/11/2020 16:13:16 INFO 139996470646592] #test_score (algo-1) : ('binary_f_
1.000', 0.6769386720729853)
[11/11/2020 16:13:16 INFO 139996470646592] #quality_metric: host=algo-1, test
binary_classification_accuracy <score>=0.694614794941
[11/11/2020 16:13:16 INFO 139996470646592] #quality_metric: host=algo-1, test
binary_classification_cross_entropy <loss>=0.631665331781
[11/11/2020 16:13:16 INFO 139996470646592] #quality_metric: host=algo-1, test
binary_f_1.000 <score>=0.676938672073
#metrics {"Metrics": {"totaltime": {"count": 1, "max": 52173.99883270264, "su
m": 52173.99883270264, "min": 52173.99883270264}, "setuptime": {"count": 1,
 "max": 54.83198165893555, "sum": 54.83198165893555, "min": 54.8319816589355
5}}, "EndTime": 1605111196.785652, "Dimensions": {"Host": "algo-1", "Operatio
n": "training", "Algorithm": "factorization-machines"}, "StartTime": 16051111
96.519935}

Training seconds: 120
Billable seconds: 120
```

Evaluate the performance of this new model. Deploy the model, determine a serializer, and then pass the test data.

In [54]:
```
from sagemaker import deserializers
fm_predictor = fm.deploy(initial_instance_count=1,
                         instance_type='ml.m4.xlarge',
                         serializer=fm_serializer,
                         deserializer=JSONDeserializer())
```

```
-------------!
```

fm_predictor.content_type = 'application/json' fm_predictor.serializer = fm_serializer fm_predictor.deserializer = deserializers.JSONDeserializer()

In [55]:
```
# Pass the testing data to the classifier and get all the predictions
Y_pred = []
for i in range(0, X_test_class.shape[0], 5):
    preds = fm_predictor.predict(X_test_class[i:i+5].toarray())['predictions']
    p = [Y_pred.append(x['score']) for x in preds]
```

Inspect the results

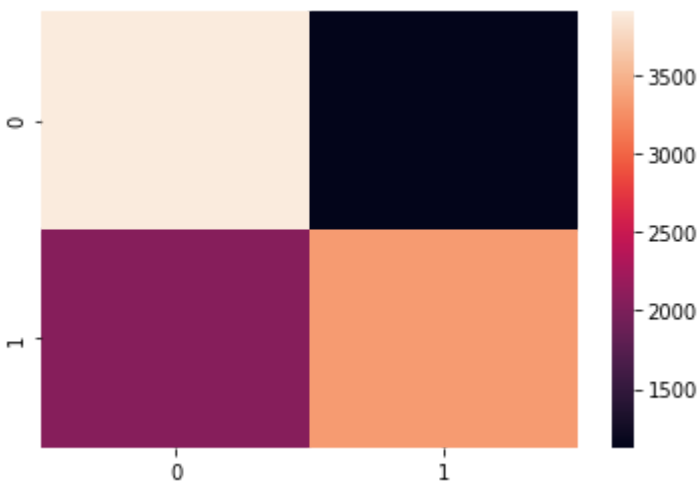
To inspect how well the classifier is doing, calculate and plot a confusion matrix. Use the implementation from **Scikit-Learn**.

In [56]:
```
from sklearn.metrics import confusion_matrix
```

In [57]:
```
true = Y_test_class.astype(int)
predicted = [1 if value > 0.5 else 0 for value in Y_pred]
conf_matrix = confusion_matrix(true, predicted)
print(conf_matrix)
sns.heatmap(conf_matrix)
```

```
[[3910 1125]
 [2062 3339]]
```

Out[57]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa83db84198>
```



In [58]:
```
# tn, fp, fn, tp
conf_matrix.ravel()
```

Out[58]:
```
array([3910, 1125, 2062, 3339])
```

Question: What is the accuracy of your model?

Hint:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

In [59]:
```
# Accuracy
(conf_matrix[0, 0] + conf_matrix[1, 1]) / conf_matrix.sum()
```

Out[59]:
```
0.6946147949405903
```

Question: How did your model do compared to a naive baseline model of predicting everything as 1?

In [60]:
```
(reduced_df.star_rating > 4).value_counts() / reduced_df.shape[0] * 100
```

Out[60]:
```
False    50.317003
True     49.682997
Name: star_rating, dtype: float64
```

Answer: This model is doing significantly better than the naive model, which randomly generates an output (~50% accuracy). If you look closely to the confusion matrix, the model is being conservative and preferring not to recommend things to users. That means most of the mistakes the model is making come from not recommending movies to users.

In [61]:
```
# Delete inference endpoint
sagemaker.Session().delete_endpoint(fm_predictor.endpoint_name)
```

Combining powers with KNN

You saw that your classifier model is doing a better job than the regressor model. Now, see if you can repackage it to fit a k-nearest neighbor (KNN) model to predict the *k closest* items to the one a customer likes and then recommend those, instead of predicting the ratings (regressor) or whether a user would like a movie or not (binary classification).

Start by downloading the model from Amazon S3. Then, repackage it to fit a KNN model.

Note: Make sure the kernel you are using is `conda_mxnet_p36` so you can run the next cells.

Download model data

In [62]:
```
import mxnet as mx
model_file_name = 'model.tar.gz'
model_full_path = f'{fm.output_path}/{fm.latest_training_job.job_name}/output/
{model_file_name}'
print(f'Model Path: {model_full_path}')

# Download FM model
os.system('aws s3 cp ' + model_full_path + ' .')

# Extract model file for loading to MXNet
os.system('tar xzvf ' + model_file_name)
os.system('unzip -o model_algo-1')
os.system('mv symbol.json model-symbol.json')
os.system('mv params model-0000.params')
```

```
Model Path: s3://qls-163754-04de04c0c698e64b-labbucket-q4zl2sa0cq9l/sagemaker
-fm/model/factorization-machines-2020-11-11-16-08-57-509/output/model.tar.gz
```

Out[62]: 0

Out[62]: 0

Out[62]: 0

Out[62]: 0

Out[62]: 0

Extract model data to create item and user latent matrixes

Now you are going to extract the values that represent each user and item after training a factorization machine. The result of the training is two matrices that, when multiplied together, will represent the target values (zero or one) as closely as possible.

In more mathematical terms, factorization machines model output consists of three N-dimensional arrays (ndarrays):

```
V - a (N x k) matrix, where:
    k is the dimension of the latent space
    N is the total count of users and items
w - an N-dimensional vector
b - a single number: the bias term
```

To extract these values, which you will use as features, you need to first load the model. Then, extract the values of each of the three matrices and build the `knn_item_matrix` and t `knn_user_matrix` matrices.

In [63]:
```
# Extract model data
m = mx.module.Module.load('./model', 0, False, label_names=['out_label'])
V = m._arg_params['v'].asnumpy()
w = m._arg_params['w1_weight'].asnumpy()
b = m._arg_params['w0_weight'].asnumpy()

nb_users = customers.shape[0]
nb_item = products.shape[0]

# Item latent matrix - concat(V[i], w[i]).
knn_item_matrix = np.concatenate((V[nb_users:], w[nb_users:]), axis=1)
knn_train_label = np.arange(1,nb_item+1)

# User latent matrix - concat (V[u], 1)
ones = np.ones(nb_users).reshape((nb_users, 1))
knn_user_matrix = np.concatenate((V[:nb_users], ones), axis=1)
```

Building KNN model

Now that you have the training data, you can now feed it to a KNN model. As you did before, you need to save the protobuf IO formatted data to Amazon S3, instantiate the model, and set the hyperparameters.

Start by setting up the path and the estimator.

In [64]:
```
print('KNN train features shape = ', knn_item_matrix.shape)
knn_prefix = 'knn'
train_key = 'train_knn'
knn_output_prefix  = f's3://{bucket}/{knn_prefix}/output'
knn_train_data_path = writeDatasetToProtobuf(knn_item_matrix, bucket,
                                             knn_prefix, train_key,
                                             "dense",
                                             knn_train_label)
print(f'Uploaded KNN train data: {knn_train_data_path}')

nb_recommendations = 100

# Set up the estimator
knn = sagemaker.estimator.Estimator(
    image_uris.retrieve("knn",boto3.Session().region_name),
    get_execution_role(),
    instance_count=1,
    instance_type=instance_type,
    output_path=knn_output_prefix,
    sagemaker_session=sagemaker.Session()
)
```

```
KNN train features shape =  (6659, 129)
Uploaded KNN train data: s3://qls-163754-04de04c0c698e64b-labbucket-q4zl2sa0c
q9l/knn/train_knn
```

Now, you will set the hyperparameters. Note that this approach uses the default `index_type` parameter for KNN. It is precise but can be slow for large datasets. In such cases, you may want to use a different `index_type` parameter leading to an approximate, yet faster answer.

For more information about index types, see [k-NN Hyperparameters (https://docs.aws.amazon.com/sagemaker/latest/dg/kNN_hyperparameters.html)](https://docs.aws.amazon.com/sagemaker/latest/dg/kNN_hyperparameters.html).

In [65]:
```
knn.set_hyperparameters(feature_dim=knn_item_matrix.shape[1],
                        k=nb_recommendations,
                        index_metric="INNER_PRODUCT",
                        predictor_type='classifier',
                        sample_size=200000)


knn.fit({'train': knn_train_data_path})
```

```
2020-11-11 16:22:09 Starting - Starting the training job......
2020-11-11 16:22:42 Starting - Launching requested ML instances.........
2020-11-11 16:24:11 Starting - Preparing the instances for training...
2020-11-11 16:25:05 Downloading - Downloading input data...
2020-11-11 16:25:17 Training - Downloading the training image.....Docker entr
ypoint called with argument(s): train
Running default environment configuration script
[11/11/2020 16:26:22 INFO 139877526685504] Reading default configuration from
/opt/amazon/lib/python2.7/site-packages/algorithm/resources/default-conf.jso
n: {u'index_metric': u'L2', u'_tuning_objective_metric': u'', u'_num_gpus':
 u'auto', u'_log_level': u'info', u'feature_dim': u'auto', u'faiss_index_ivf_
nlists': u'auto', u'epochs': u'1', u'index_type': u'faiss.Flat', u'_faiss_ind
ex_nprobe': u'5', u'_kvstore': u'dist_async', u'_num_kv_servers': u'1', u'min
i_batch_size': u'5000'}
[11/11/2020 16:26:22 INFO 139877526685504] Merging with provided configuratio
n from /opt/ml/input/config/hyperparameters.json: {u'sample_size': u'200000',
u'feature_dim': u'129', u'index_metric': u'INNER_PRODUCT', u'predictor_type':
u'classifier', u'k': u'100'}
[11/11/2020 16:26:22 INFO 139877526685504] Final configuration: {u'index_metr
ic': u'INNER_PRODUCT', u'predictor_type': u'classifier', u'_tuning_objective_
metric': u'', u'_num_gpus': u'auto', u'_log_level': u'info', u'feature_dim':
 u'129', u'faiss_index_ivf_nlists': u'auto', u'sample_size': u'200000', u'epo
chs': u'1', u'index_type': u'faiss.Flat', u'_faiss_index_nprobe': u'5', u'_kv
store': u'dist_async', u'_num_kv_servers': u'1', u'mini_batch_size': u'5000',
u'k': u'100'}
[11/11/2020 16:26:22 WARNING 139877526685504] Loggers have already been setu
p.
[11/11/2020 16:26:22 INFO 139877526685504] Launching parameter server for rol
e scheduler
[11/11/2020 16:26:22 INFO 139877526685504] {'ECS_CONTAINER_METADATA_URI': 'ht
tp://169.254.170.2/v3/3068af3f-835d-4da5-b8fa-c724a6515321', 'ECS_CONTAINER_M
ETADATA_URI_V4': 'http://169.254.170.2/v4/3068af3f-835d-4da5-b8fa-c724a651532
1', 'PROTOCOL_BUFFERS_PYTHON_IMPLEMENTATION_VERSION': '2', 'PATH': '/opt/amaz
on/bin:/usr/local/nvidia/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bi
n:/sbin:/bin:/opt/amazon/bin:/opt/amazon/bin', 'SAGEMAKER_HTTP_PORT': '8080',
'HOME': '/root', 'PYTHONUNBUFFERED': 'TRUE', 'CANONICAL_ENVROOT': '/opt/amazo
n', 'LD_LIBRARY_PATH': '/opt/amazon/lib/python2.7/site-packages/cv
2/../../../../lib:/usr/local/nvidia/lib64:/opt/amazon/lib', 'LANG': 'en_US.ut
f8', 'DMLC_INTERFACE': 'eth0', 'SHLVL': '1', 'AWS_REGION': 'us-west-2', 'SAGE
MAKER_METRICS_DIRECTORY': '/opt/ml/output/metrics/sagemaker', 'NVIDIA_VISIBLE
_DEVICES': 'void', 'TRAINING_JOB_NAME': 'knn-2020-11-11-16-22-09-550', 'PROTO
COL_BUFFERS_PYTHON_IMPLEMENTATION': 'cpp', 'ENVROOT': '/opt/amazon', 'SAGEMAK
ER_DATA_PATH': '/opt/ml', 'NVIDIA_DRIVER_CAPABILITIES': 'compute,utility', 'N
VIDIA_REQUIRE_CUDA': 'cuda>=9.0', 'OMP_NUM_THREADS': '2', 'HOSTNAME': 'ip-10-
0-176-243.us-west-2.compute.internal', 'AWS_CONTAINER_CREDENTIALS_RELATIVE_UR
I': '/v2/credentials/186b4e70-c194-41eb-9bac-459b1ba95a88', 'PWD': '/', 'TRAI
NING_JOB_ARN': 'arn:aws:sagemaker:us-west-2:260402929810:training-job/knn-202
0-11-11-16-22-09-550', 'AWS_EXECUTION_ENV': 'AWS_ECS_EC2'}
[11/11/2020 16:26:22 INFO 139877526685504] envs={'ECS_CONTAINER_METADATA_UR
I': 'http://169.254.170.2/v3/3068af3f-835d-4da5-b8fa-c724a6515321', 'ECS_CONT
AINER_METADATA_URI_V4': 'http://169.254.170.2/v4/3068af3f-835d-4da5-b8fa-c724
a6515321', 'PROTOCOL_BUFFERS_PYTHON_IMPLEMENTATION_VERSION': '2', 'DMLC_NUM_W
ORKER': '1', 'DMLC_PS_ROOT_PORT': '9000', 'PATH': '/opt/amazon/bin:/usr/loca
l/nvidia/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/op
t/amazon/bin:/opt/amazon/bin', 'SAGEMAKER_HTTP_PORT': '8080', 'HOME': '/roo
```

```
t', 'PYTHONUNBUFFERED': 'TRUE', 'CANONICAL_ENVROOT': '/opt/amazon', 'LD_LIBRA
RY_PATH': '/opt/amazon/lib/python2.7/site-packages/cv2/../../../../lib:/usr/l
ocal/nvidia/lib64:/opt/amazon/lib', 'LANG': 'en_US.utf8', 'DMLC_INTERFACE':
 'eth0', 'SHLVL': '1', 'DMLC_PS_ROOT_URI': '10.0.176.243', 'AWS_REGION': 'us-
west-2', 'SAGEMAKER_METRICS_DIRECTORY': '/opt/ml/output/metrics/sagemaker',
 'NVIDIA_VISIBLE_DEVICES': 'void', 'TRAINING_JOB_NAME': 'knn-2020-11-11-16-22
-09-550', 'PROTOCOL_BUFFERS_PYTHON_IMPLEMENTATION': 'cpp', 'ENVROOT': '/opt/a
mazon', 'SAGEMAKER_DATA_PATH': '/opt/ml', 'NVIDIA_DRIVER_CAPABILITIES': 'comp
ute,utility', 'NVIDIA_REQUIRE_CUDA': 'cuda>=9.0', 'OMP_NUM_THREADS': '2', 'HO
STNAME': 'ip-10-0-176-243.us-west-2.compute.internal', 'AWS_CONTAINER_CREDENT
IALS_RELATIVE_URI': '/v2/credentials/186b4e70-c194-41eb-9bac-459b1ba95a88',
 'DMLC_ROLE': 'scheduler', 'PWD': '/', 'DMLC_NUM_SERVER': '1', 'TRAINING_JOB_
ARN': 'arn:aws:sagemaker:us-west-2:260402929810:training-job/knn-2020-11-11-1
6-22-09-550', 'AWS_EXECUTION_ENV': 'AWS_ECS_EC2'}
[11/11/2020 16:26:22 INFO 139877526685504] Launching parameter server for rol
e server
[11/11/2020 16:26:22 INFO 139877526685504] {'ECS_CONTAINER_METADATA_URI': 'ht
tp://169.254.170.2/v3/3068af3f-835d-4da5-b8fa-c724a6515321', 'ECS_CONTAINER_M
ETADATA_URI_V4': 'http://169.254.170.2/v4/3068af3f-835d-4da5-b8fa-c724a651532
1', 'PROTOCOL_BUFFERS_PYTHON_IMPLEMENTATION_VERSION': '2', 'PATH': '/opt/amaz
on/bin:/usr/local/nvidia/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bi
n:/sbin:/bin:/opt/amazon/bin:/opt/amazon/bin', 'SAGEMAKER_HTTP_PORT': '8080',
'HOME': '/root', 'PYTHONUNBUFFERED': 'TRUE', 'CANONICAL_ENVROOT': '/opt/amazo
n', 'LD_LIBRARY_PATH': '/opt/amazon/lib/python2.7/site-packages/cv
2/../../../../lib:/usr/local/nvidia/lib64:/opt/amazon/lib', 'LANG': 'en_US.ut
f8', 'DMLC_INTERFACE': 'eth0', 'SHLVL': '1', 'AWS_REGION': 'us-west-2', 'SAGE
MAKER_METRICS_DIRECTORY': '/opt/ml/output/metrics/sagemaker', 'NVIDIA_VISIBLE
_DEVICES': 'void', 'TRAINING_JOB_NAME': 'knn-2020-11-11-16-22-09-550', 'PROTO
COL_BUFFERS_PYTHON_IMPLEMENTATION': 'cpp', 'ENVROOT': '/opt/amazon', 'SAGEMAK
ER_DATA_PATH': '/opt/ml', 'NVIDIA_DRIVER_CAPABILITIES': 'compute,utility', 'N
VIDIA_REQUIRE_CUDA': 'cuda>=9.0', 'OMP_NUM_THREADS': '2', 'HOSTNAME': 'ip-10-
0-176-243.us-west-2.compute.internal', 'AWS_CONTAINER_CREDENTIALS_RELATIVE_UR
I': '/v2/credentials/186b4e70-c194-41eb-9bac-459b1ba95a88', 'PWD': '/', 'TRAI
NING_JOB_ARN': 'arn:aws:sagemaker:us-west-2:260402929810:training-job/knn-202
0-11-11-16-22-09-550', 'AWS_EXECUTION_ENV': 'AWS_ECS_EC2'}
[11/11/2020 16:26:22 INFO 139877526685504] envs={'ECS_CONTAINER_METADATA_UR
I': 'http://169.254.170.2/v3/3068af3f-835d-4da5-b8fa-c724a6515321', 'ECS_CONT
AINER_METADATA_URI_V4': 'http://169.254.170.2/v4/3068af3f-835d-4da5-b8fa-c724
a6515321', 'PROTOCOL_BUFFERS_PYTHON_IMPLEMENTATION_VERSION': '2', 'DMLC_NUM_W
ORKER': '1', 'DMLC_PS_ROOT_PORT': '9000', 'PATH': '/opt/amazon/bin:/usr/loca
l/nvidia/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/op
t/amazon/bin:/opt/amazon/bin', 'SAGEMAKER_HTTP_PORT': '8080', 'HOME': '/roo
t', 'PYTHONUNBUFFERED': 'TRUE', 'CANONICAL_ENVROOT': '/opt/amazon', 'LD_LIBRA
RY_PATH': '/opt/amazon/lib/python2.7/site-packages/cv2/../../../../lib:/usr/l
ocal/nvidia/lib64:/opt/amazon/lib', 'LANG': 'en_US.utf8', 'DMLC_INTERFACE':
 'eth0', 'SHLVL': '1', 'DMLC_PS_ROOT_URI': '10.0.176.243', 'AWS_REGION': 'us-
west-2', 'SAGEMAKER_METRICS_DIRECTORY': '/opt/ml/output/metrics/sagemaker',
 'NVIDIA_VISIBLE_DEVICES': 'void', 'TRAINING_JOB_NAME': 'knn-2020-11-11-16-22
-09-550', 'PROTOCOL_BUFFERS_PYTHON_IMPLEMENTATION': 'cpp', 'ENVROOT': '/opt/a
mazon', 'SAGEMAKER_DATA_PATH': '/opt/ml', 'NVIDIA_DRIVER_CAPABILITIES': 'comp
ute,utility', 'NVIDIA_REQUIRE_CUDA': 'cuda>=9.0', 'OMP_NUM_THREADS': '2', 'HO
STNAME': 'ip-10-0-176-243.us-west-2.compute.internal', 'AWS_CONTAINER_CREDENT
IALS_RELATIVE_URI': '/v2/credentials/186b4e70-c194-41eb-9bac-459b1ba95a88',
 'DMLC_ROLE': 'server', 'PWD': '/', 'DMLC_NUM_SERVER': '1', 'TRAINING_JOB_AR
N': 'arn:aws:sagemaker:us-west-2:260402929810:training-job/knn-2020-11-11-16-
22-09-550', 'AWS_EXECUTION_ENV': 'AWS_ECS_EC2'}
```

```
[11/11/2020 16:26:22 INFO 139877526685504] Environment: {'ECS_CONTAINER_METAD
ATA_URI': 'http://169.254.170.2/v3/3068af3f-835d-4da5-b8fa-c724a6515321', 'EC
S_CONTAINER_METADATA_URI_V4': 'http://169.254.170.2/v4/3068af3f-835d-4da5-b8f
a-c724a6515321', 'PROTOCOL_BUFFERS_PYTHON_IMPLEMENTATION_VERSION': '2', 'DMLC
_PS_ROOT_PORT': '9000', 'DMLC_NUM_WORKER': '1', 'SAGEMAKER_HTTP_PORT': '808
0', 'PATH': '/opt/amazon/bin:/usr/local/nvidia/bin:/usr/local/sbin:/usr/loca
l/bin:/usr/sbin:/usr/bin:/sbin:/bin:/opt/amazon/bin:/opt/amazon/bin', 'PYTHON
UNBUFFERED': 'TRUE', 'CANONICAL_ENVROOT': '/opt/amazon', 'LD_LIBRARY_PATH':
 '/opt/amazon/lib/python2.7/site-packages/cv2/../../../../lib:/usr/local/nvid
ia/lib64:/opt/amazon/lib', 'LANG': 'en_US.utf8', 'DMLC_INTERFACE': 'eth0', 'S
HLVL': '1', 'DMLC_PS_ROOT_URI': '10.0.176.243', 'AWS_REGION': 'us-west-2', 'S
AGEMAKER_METRICS_DIRECTORY': '/opt/ml/output/metrics/sagemaker', 'NVIDIA_VISI
BLE_DEVICES': 'void', 'TRAINING_JOB_NAME': 'knn-2020-11-11-16-22-09-550', 'HO
ME': '/root', 'PROTOCOL_BUFFERS_PYTHON_IMPLEMENTATION': 'cpp', 'ENVROOT': '/o
pt/amazon', 'SAGEMAKER_DATA_PATH': '/opt/ml', 'NVIDIA_DRIVER_CAPABILITIES':
 'compute,utility', 'NVIDIA_REQUIRE_CUDA': 'cuda>=9.0', 'OMP_NUM_THREADS':
 '2', 'HOSTNAME': 'ip-10-0-176-243.us-west-2.compute.internal', 'AWS_CONTAINE
R_CREDENTIALS_RELATIVE_URI': '/v2/credentials/186b4e70-c194-41eb-9bac-459b1ba
95a88', 'DMLC_ROLE': 'worker', 'PWD': '/', 'DMLC_NUM_SERVER': '1', 'TRAINING_
JOB_ARN': 'arn:aws:sagemaker:us-west-2:260402929810:training-job/knn-2020-11-
11-16-22-09-550', 'AWS_EXECUTION_ENV': 'AWS_ECS_EC2'}
Process 61 is a shell:scheduler.
Process 70 is a shell:server.
Process 1 is a worker.
[11/11/2020 16:26:22 INFO 139877526685504] Using default worker.
[11/11/2020 16:26:23 INFO 139877526685504] Checkpoint loading and saving are
 disabled.
[11/11/2020 16:26:23 INFO 139877526685504] nvidia-smi took: 0.0252211093903 s
ecs to identify 0 gpus
[11/11/2020 16:26:23 INFO 139877526685504] Create Store: dist_async
[11/11/2020 16:26:24 ERROR 139877526685504] nvidia-smi: failed to run (127):
 /bin/sh: nvidia-smi: command not found
[11/11/2020 16:26:24 INFO 139877526685504] Using per-worker sample size = 200
000 (Available virtual memory = 15147425792 bytes, GPU free memory = 0 bytes,
number of workers = 1). If an out-of-memory error occurs, choose a larger ins
tance type, use dimension reduction, decrease sample_size, and/or decrease mi
ni_batch_size.
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 0, "sum": 0.0, "min": 0}, "Number of Batches Since Last Reset": {"count": 1,
"max": 0, "sum": 0.0, "min": 0}, "Number of Records Since Last Reset": {"coun
t": 1, "max": 0, "sum": 0.0, "min": 0}, "Total Batches Seen": {"count": 1, "m
ax": 0, "sum": 0.0, "min": 0}, "Total Records Seen": {"count": 1, "max": 0,
 "sum": 0.0, "min": 0}, "Max Records Seen Between Resets": {"count": 1, "ma
x": 0, "sum": 0.0, "min": 0}, "Reset Count": {"count": 1, "max": 0, "sum": 0.
0, "min": 0}}, "EndTime": 1605111984.62992, "Dimensions": {"Host": "algo-1",
 "Meta": "init_train_data_iter", "Operation": "training", "Algorithm": "AWS/K
NN"}, "StartTime": 1605111984.629871}

[2020-11-11 16:26:24.630] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 0, "duration": 1152, "num_examples": 1, "nu
m_bytes": 2820000}
[2020-11-11 16:26:24.677] [tensorio] [info] epoch_stats={"data_pipeline": "/o
pt/ml/input/data/train", "epoch": 1, "duration": 46, "num_examples": 2, "num_
bytes": 3755676}
[11/11/2020 16:26:24 INFO 139877526685504] push reservoir to kv... 1 num_work
ers 0 rank
```

```
[11/11/2020 16:26:24 INFO 139877526685504] ...done (6659)
[11/11/2020 16:26:24 INFO 139877526685504] #progress_metric: host=algo-1, com
pleted 100 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max":
 2, "sum": 2.0, "min": 2}, "Number of Batches Since Last Reset": {"count": 1,
"max": 2, "sum": 2.0, "min": 2}, "Number of Records Since Last Reset": {"coun
t": 1, "max": 6659, "sum": 6659.0, "min": 6659}, "Total Batches Seen": {"coun
t": 1, "max": 2, "sum": 2.0, "min": 2}, "Total Records Seen": {"count": 1, "m
ax": 6659, "sum": 6659.0, "min": 6659}, "Max Records Seen Between Resets":
 {"count": 1, "max": 6659, "sum": 6659.0, "min": 6659}, "Reset Count": {"coun
t": 1, "max": 1, "sum": 1.0, "min": 1}}, "EndTime": 1605111984.731164, "Dimen
sions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "traini
ng", "Algorithm": "AWS/KNN", "epoch": 0}, "StartTime": 1605111984.630535}

[11/11/2020 16:26:24 INFO 139877526685504] #throughput_metric: host=algo-1, t
rain throughput=66060.2334376 records/second
[11/11/2020 16:26:24 INFO 139877526685504] pulled row count... worker 0 rows
 6659
[11/11/2020 16:26:24 INFO 139877526685504] pulled... worker 0 data (6659, 12
9) labels (6659,) nans 0
[11/11/2020 16:26:24 INFO 139877526685504] calling index.train...
[11/11/2020 16:26:24 INFO 139877526685504] ...done calling index.train
[11/11/2020 16:26:24 INFO 139877526685504] calling index.add...
[11/11/2020 16:26:24 INFO 139877526685504] ...done calling index.add
#metrics {"Metrics": {"epochs": {"count": 1, "max": 1, "sum": 1.0, "min": 1},
"model.serialize.time": {"count": 1, "max": 3.8039684295654297, "sum": 3.8039
684295654297, "min": 3.8039684295654297}, "finalize.time": {"count": 1, "ma
x": 273.8909721374512, "sum": 273.8909721374512, "min": 273.8909721374512},
 "initialize.time": {"count": 1, "max": 1123.816967010498, "sum": 1123.816967
010498, "min": 1123.816967010498}, "update.time": {"count": 1, "max": 100.208
99772644043, "sum": 100.20899772644043, "min": 100.20899772644043}}, "EndTim
e": 1605111985.009435, "Dimensions": {"Host": "algo-1", "Operation": "trainin
g", "Algorithm": "AWS/KNN"}, "StartTime": 1605111983.476691}

[11/11/2020 16:26:25 INFO 139877526685504] Test data is not provided.
#metrics {"Metrics": {"totaltime": {"count": 1, "max": 2218.2719707489014, "s
um": 2218.2719707489014, "min": 2218.2719707489014}, "setuptime": {"count":
 1, "max": 33.33282470703125, "sum": 33.33282470703125, "min": 33.33282470703
125}}, "EndTime": 1605111985.010918, "Dimensions": {"Host": "algo-1", "Operat
ion": "training", "Algorithm": "AWS/KNN"}, "StartTime": 1605111985.009534}


2020-11-11 16:26:36 Uploading - Uploading generated training model
2020-11-11 16:26:36 Completed - Training job completed
Training seconds: 91
Billable seconds: 91
```

Now that you have a trained model, save it so you can reference it for batch inference.

In [66]:
```
knn_model_name =  knn.latest_training_job.job_name
print("created model: ", knn_model_name)

# Save the model so that you can reference it in the next step during batch in
ference
sm = boto3.client(service_name='sagemaker')
primary_container = {
    'Image': knn.image_uri,
    'ModelDataUrl': knn.model_data,
}

knn_model = sm.create_model(
        ModelName = knn.latest_training_job.job_name,
        ExecutionRoleArn = knn.role,
        PrimaryContainer = primary_container)
print("saved the model")
```

```
created model:  knn-2020-11-11-16-22-09-550
saved the model
```

Batch transform

To see the predictions your model made, you would have to create inferences and see if they make sense. You could repeat the process as last time and check one user at a time with all possible combinations of items. However, Amazon SageMaker provides a batch transform job that you can use to do inference over the entire dataset. For more information, see [Get Inferences for an Entire Dataset with Batch Transform (https://docs.aws.amazon.com/sagemaker/latest/dg/how-it-works-batch.html)](https://docs.aws.amazon.com/sagemaker/latest/dg/how-it-works-batch.html).

In this section, you will use a batch transform to predict the top 100 recommendations for all the users.

In [67]:
```
# Upload inference data to S3
knn_batch_data_path = writeDatasetToProtobuf(knn_user_matrix,
                                             bucket,
                                             knn_prefix,
                                             train_key,
                                             "dense")
print ("Batch inference data path: ",knn_batch_data_path)

# Initialize the transformer object
transformer =sagemaker.transformer.Transformer(
    base_transform_job_name="knn",
    model_name=knn_model_name,
    instance_count=1,
    instance_type=instance_type,
    output_path=knn_output_prefix,
    accept="application/jsonlines; verbose=true",

)

# Start a transform job
transformer.transform(knn_batch_data_path,
                      content_type='application/x-recordio-protobuf',
                      split_type='RecordIO')
transformer.wait()
```

```
Batch inference data path:  s3://qls-163754-04de04c0c698e64b-labbucket-q4zl2s
a0cq9l/knn/train_knn
.................................Docker entrypoint called with argument(s):
serve
Docker entrypoint called with argument(s): serve
Running default environment configuration script
Running default environment configuration script
[11/11/2020 16:32:31 INFO 140633781204800] loaded entry point class algorith
m.serve.server_config:config_api
[11/11/2020 16:32:31 INFO 140633781204800] loading entry points
[11/11/2020 16:32:31 INFO 140633781204800] loaded entry point class algorith
m.serve.server_config:config_api
[11/11/2020 16:32:31 INFO 140633781204800] loading entry points
[11/11/2020 16:32:31 INFO 140633781204800] loaded request iterator text/csv
[11/11/2020 16:32:31 INFO 140633781204800] loaded request iterator applicatio
n/x-recordio-protobuf
[11/11/2020 16:32:31 INFO 140633781204800] loaded request iterator applicatio
n/json
[11/11/2020 16:32:31 INFO 140633781204800] loaded request iterator applicatio
n/jsonlines
[11/11/2020 16:32:31 INFO 140633781204800] loaded response encoder applicatio
n/x-recordio-protobuf
[11/11/2020 16:32:31 INFO 140633781204800] loaded response encoder applicatio
n/json
[11/11/2020 16:32:31 INFO 140633781204800] loaded response encoder applicatio
n/jsonlines
[11/11/2020 16:32:31 INFO 140633781204800] loaded entry point class algorith
m:model
[11/11/2020 16:32:31 INFO 140633781204800] Number of server workers: 1
[2020-11-11 16:32:31 +0000] [1] [INFO] Starting gunicorn 19.7.1
[2020-11-11 16:32:31 +0000] [1] [INFO] Listening at: http://0.0.0.0:8080 (1)
[2020-11-11 16:32:31 +0000] [1] [INFO] Using worker: sync
[2020-11-11 16:32:31 +0000] [78] [INFO] Booting worker with pid: 78
[11/11/2020 16:32:31 INFO 140633781204800] loading model...
[11/11/2020 16:32:31 INFO 140633781204800] nvidia-smi took: 0.0252480506897 s
ecs to identify 0 gpus
[11/11/2020 16:32:31 INFO 140633781204800] ...model loaded.
#metrics {"Metrics": {"execution_parameters.count": {"count": 1, "max": 1, "s
um": 1.0, "min": 1}}, "EndTime": 1605112351.795282, "Dimensions": {"Host": "U
NKNOWN", "Operation": "scoring", "Algorithm": "KNNModel"}, "StartTime": 16051
12351.790014}

[11/11/2020 16:32:31 INFO 140633781204800] loaded request iterator text/csv
[11/11/2020 16:32:31 INFO 140633781204800] loaded request iterator applicatio
n/x-recordio-protobuf
[11/11/2020 16:32:31 INFO 140633781204800] loaded request iterator applicatio
n/json
[11/11/2020 16:32:31 INFO 140633781204800] loaded request iterator applicatio
n/jsonlines
[11/11/2020 16:32:31 INFO 140633781204800] loaded response encoder applicatio
n/x-recordio-protobuf
[11/11/2020 16:32:31 INFO 140633781204800] loaded response encoder applicatio
n/json
[11/11/2020 16:32:31 INFO 140633781204800] loaded response encoder applicatio
n/jsonlines
```

```
[11/11/2020 16:32:31 INFO 140633781204800] loaded entry point class algorith
m:model
[11/11/2020 16:32:31 INFO 140633781204800] Number of server workers: 1
[2020-11-11 16:32:31 +0000] [1] [INFO] Starting gunicorn 19.7.1
[2020-11-11 16:32:31 +0000] [1] [INFO] Listening at: http://0.0.0.0:8080 (1)
[2020-11-11 16:32:31 +0000] [1] [INFO] Using worker: sync
[2020-11-11 16:32:31 +0000] [78] [INFO] Booting worker with pid: 78
[11/11/2020 16:32:31 INFO 140633781204800] loading model...
[11/11/2020 16:32:31 INFO 140633781204800] nvidia-smi took: 0.0252480506897 s
ecs to identify 0 gpus
[11/11/2020 16:32:31 INFO 140633781204800] ...model loaded.
#metrics {"Metrics": {"execution_parameters.count": {"count": 1, "max": 1, "s
um": 1.0, "min": 1}}, "EndTime": 1605112351.795282, "Dimensions": {"Host": "U
NKNOWN", "Operation": "scoring", "Algorithm": "KNNModel"}, "StartTime": 16051
12351.790014}

[2020-11-11 16:32:32.603] [tensorio] [info] epoch_stats={"data_pipeline": "/t
mp/tmpntUQpm/tmpxnftje", "epoch": 0, "duration": 630, "num_examples": 1, "num
_bytes": 6291100}
[2020-11-11 16:32:32.603] [tensorio] [info] epoch_stats={"data_pipeline": "/t
mp/tmpntUQpm/tmpxnftje", "epoch": 0, "duration": 630, "num_examples": 1, "num
_bytes": 6291100}
#metrics {"Metrics": {"model.evaluate.time": {"count": 1, "max": 616.14799499
51172, "sum": 616.1479949951172, "min": 616.1479949951172}, "invocations.coun
t": {"count": 1, "max": 1, "sum": 1.0, "min": 1}}, "EndTime": 1605112353.9605
11, "Dimensions": {"Host": "UNKNOWN", "Operation": "scoring", "Algorithm": "K
NNModel"}, "StartTime": 1605112351.795391}

[2020-11-11 16:32:34.165] [tensorio] [warning] TensorIO is already initialize
d; ignoring the initialization routine.
[2020-11-11 16:32:34.620] [tensorio] [info] epoch_stats={"data_pipeline": "/t
mp/tmpD2N3dO/tmp17uD7V", "epoch": 0, "duration": 454, "num_examples": 1, "num
_bytes": 4771060}
#metrics {"Metrics": {"model.evaluate.time": {"count": 1, "max": 616.14799499
51172, "sum": 616.1479949951172, "min": 616.1479949951172}, "invocations.coun
t": {"count": 1, "max": 1, "sum": 1.0, "min": 1}}, "EndTime": 1605112353.9605
11, "Dimensions": {"Host": "UNKNOWN", "Operation": "scoring", "Algorithm": "K
NNModel"}, "StartTime": 1605112351.795391}

[2020-11-11 16:32:34.165] [tensorio] [warning] TensorIO is already initialize
d; ignoring the initialization routine.
[2020-11-11 16:32:34.620] [tensorio] [info] epoch_stats={"data_pipeline": "/t
mp/tmpD2N3dO/tmp17uD7V", "epoch": 0, "duration": 454, "num_examples": 1, "num
_bytes": 4771060}
#metrics {"Metrics": {"model.evaluate.time": {"count": 1, "max": 444.51999664
30664, "sum": 444.5199966430664, "min": 444.5199966430664}, "invocations.coun
t": {"count": 1, "max": 1, "sum": 1.0, "min": 1}}, "EndTime": 1605112355.6456
52, "Dimensions": {"Host": "UNKNOWN", "Operation": "scoring", "Algorithm": "K
NNModel"}, "StartTime": 1605112353.9606}

#metrics {"Metrics": {"model.evaluate.time": {"count": 1, "max": 444.51999664
30664, "sum": 444.5199966430664, "min": 444.5199966430664}, "invocations.coun
t": {"count": 1, "max": 1, "sum": 1.0, "min": 1}}, "EndTime": 1605112355.6456
52, "Dimensions": {"Host": "UNKNOWN", "Operation": "scoring", "Algorithm": "K
NNModel"}, "StartTime": 1605112353.9606}
```

```
2020-11-11T16:32:31.801:[sagemaker logs]: MaxConcurrentTransforms=1, MaxPaylo
adInMB=6, BatchStrategy=MULTI_RECORD

Docker entrypoint called with argument(s): serve
Docker entrypoint called with argument(s): serve
Running default environment configuration script
Running default environment configuration script
[11/11/2020 16:32:31 INFO 140633781204800] loaded entry point class algorith
m.serve.server_config:config_api
[11/11/2020 16:32:31 INFO 140633781204800] loading entry points
[11/11/2020 16:32:31 INFO 140633781204800] loaded entry point class algorith
m.serve.server_config:config_api
[11/11/2020 16:32:31 INFO 140633781204800] loading entry points
[11/11/2020 16:32:31 INFO 140633781204800] loaded request iterator text/csv
[11/11/2020 16:32:31 INFO 140633781204800] loaded request iterator applicatio
n/x-recordio-protobuf
[11/11/2020 16:32:31 INFO 140633781204800] loaded request iterator applicatio
n/json
[11/11/2020 16:32:31 INFO 140633781204800] loaded request iterator applicatio
n/jsonlines
[11/11/2020 16:32:31 INFO 140633781204800] loaded response encoder applicatio
n/x-recordio-protobuf
[11/11/2020 16:32:31 INFO 140633781204800] loaded response encoder applicatio
n/json
[11/11/2020 16:32:31 INFO 140633781204800] loaded response encoder applicatio
n/jsonlines
[11/11/2020 16:32:31 INFO 140633781204800] loaded entry point class algorith
m:model
[11/11/2020 16:32:31 INFO 140633781204800] Number of server workers: 1
[2020-11-11 16:32:31 +0000] [1] [INFO] Starting gunicorn 19.7.1
[2020-11-11 16:32:31 +0000] [1] [INFO] Listening at: http://0.0.0.0:8080 (1)
[2020-11-11 16:32:31 +0000] [1] [INFO] Using worker: sync
[2020-11-11 16:32:31 +0000] [78] [INFO] Booting worker with pid: 78
[11/11/2020 16:32:31 INFO 140633781204800] loading model...
[11/11/2020 16:32:31 INFO 140633781204800] nvidia-smi took: 0.0252480506897 s
ecs to identify 0 gpus
[11/11/2020 16:32:31 INFO 140633781204800] ...model loaded.
#metrics {"Metrics": {"execution_parameters.count": {"count": 1, "max": 1, "s
um": 1.0, "min": 1}}, "EndTime": 1605112351.795282, "Dimensions": {"Host": "U
NKNOWN", "Operation": "scoring", "Algorithm": "KNNModel"}, "StartTime": 16051
12351.790014}

[11/11/2020 16:32:31 INFO 140633781204800] loaded request iterator text/csv
[11/11/2020 16:32:31 INFO 140633781204800] loaded request iterator applicatio
n/x-recordio-protobuf
[11/11/2020 16:32:31 INFO 140633781204800] loaded request iterator applicatio
n/json
[11/11/2020 16:32:31 INFO 140633781204800] loaded request iterator applicatio
n/jsonlines
[11/11/2020 16:32:31 INFO 140633781204800] loaded response encoder applicatio
n/x-recordio-protobuf
[11/11/2020 16:32:31 INFO 140633781204800] loaded response encoder applicatio
n/json
[11/11/2020 16:32:31 INFO 140633781204800] loaded response encoder applicatio
n/jsonlines
[11/11/2020 16:32:31 INFO 140633781204800] loaded entry point class algorith
```

```
m:model
[11/11/2020 16:32:31 INFO 140633781204800] Number of server workers: 1
[2020-11-11 16:32:31 +0000] [1] [INFO] Starting gunicorn 19.7.1
[2020-11-11 16:32:31 +0000] [1] [INFO] Listening at: http://0.0.0.0:8080 (1)
[2020-11-11 16:32:31 +0000] [1] [INFO] Using worker: sync
[2020-11-11 16:32:31 +0000] [78] [INFO] Booting worker with pid: 78
[11/11/2020 16:32:31 INFO 140633781204800] loading model...
[11/11/2020 16:32:31 INFO 140633781204800] nvidia-smi took: 0.0252480506897 s
ecs to identify 0 gpus
[11/11/2020 16:32:31 INFO 140633781204800] ...model loaded.
#metrics {"Metrics": {"execution_parameters.count": {"count": 1, "max": 1, "s
um": 1.0, "min": 1}}, "EndTime": 1605112351.795282, "Dimensions": {"Host": "U
NKNOWN", "Operation": "scoring", "Algorithm": "KNNModel"}, "StartTime": 16051
12351.790014}

[2020-11-11 16:32:32.603] [tensorio] [info] epoch_stats={"data_pipeline": "/t
mp/tmpntUQpm/tmpxnftje", "epoch": 0, "duration": 630, "num_examples": 1, "num
_bytes": 6291100}
[2020-11-11 16:32:32.603] [tensorio] [info] epoch_stats={"data_pipeline": "/t
mp/tmpntUQpm/tmpxnftje", "epoch": 0, "duration": 630, "num_examples": 1, "num
_bytes": 6291100}
#metrics {"Metrics": {"model.evaluate.time": {"count": 1, "max": 616.14799499
51172, "sum": 616.1479949951172, "min": 616.1479949951172}, "invocations.coun
t": {"count": 1, "max": 1, "sum": 1.0, "min": 1}}, "EndTime": 1605112353.9605
11, "Dimensions": {"Host": "UNKNOWN", "Operation": "scoring", "Algorithm": "K
NNModel"}, "StartTime": 1605112351.795391}

[2020-11-11 16:32:34.165] [tensorio] [warning] TensorIO is already initialize
d; ignoring the initialization routine.
[2020-11-11 16:32:34.620] [tensorio] [info] epoch_stats={"data_pipeline": "/t
mp/tmpD2N3dO/tmp17uD7V", "epoch": 0, "duration": 454, "num_examples": 1, "num
_bytes": 4771060}
#metrics {"Metrics": {"model.evaluate.time": {"count": 1, "max": 616.14799499
51172, "sum": 616.1479949951172, "min": 616.1479949951172}, "invocations.coun
t": {"count": 1, "max": 1, "sum": 1.0, "min": 1}}, "EndTime": 1605112353.9605
11, "Dimensions": {"Host": "UNKNOWN", "Operation": "scoring", "Algorithm": "K
NNModel"}, "StartTime": 1605112351.795391}

[2020-11-11 16:32:34.165] [tensorio] [warning] TensorIO is already initialize
d; ignoring the initialization routine.
[2020-11-11 16:32:34.620] [tensorio] [info] epoch_stats={"data_pipeline": "/t
mp/tmpD2N3dO/tmp17uD7V", "epoch": 0, "duration": 454, "num_examples": 1, "num
_bytes": 4771060}
#metrics {"Metrics": {"model.evaluate.time": {"count": 1, "max": 444.51999664
30664, "sum": 444.5199966430664, "min": 444.5199966430664}, "invocations.coun
t": {"count": 1, "max": 1, "sum": 1.0, "min": 1}}, "EndTime": 1605112355.6456
52, "Dimensions": {"Host": "UNKNOWN", "Operation": "scoring", "Algorithm": "K
NNModel"}, "StartTime": 1605112353.9606}

#metrics {"Metrics": {"model.evaluate.time": {"count": 1, "max": 444.51999664
30664, "sum": 444.5199966430664, "min": 444.5199966430664}, "invocations.coun
t": {"count": 1, "max": 1, "sum": 1.0, "min": 1}}, "EndTime": 1605112355.6456
52, "Dimensions": {"Host": "UNKNOWN", "Operation": "scoring", "Algorithm": "K
NNModel"}, "StartTime": 1605112353.9606}
```

```
2020-11-11T16:32:31.801:[sagemaker logs]: MaxConcurrentTransforms=1, MaxPaylo
adInMB=6, BatchStrategy=MULTI_RECORD
```

You are now free to examine the predictions. Download them first.

In [68]:
```
# Download predictions
results_file_name = "inference_output"
inference_output_file = "knn/output/train_knn.out"
s3_client = boto3.client('s3')
s3_client.download_file(bucket, inference_output_file, results_file_name)
```

In [69]:
```
# Open file and load it to memory
with open(results_file_name) as f:
    results = f.readlines()
```

The results contain the 100 nearest neighbor movie IDs with their corresponding distances. See how it looks for user number 200.

In [70]:
```
test_user_idx = 200
u_one_json = json.loads(results[test_user_idx])
recommended_movies = [int(movie_id) for movie_id in u_one_json['labels']]
distances = [round(distance, 4) for distance in u_one_json['distances']]

print(f'Recommended movie Ids for user #{test_user_idx} : {recommended_movies}
')

print(f'Movie distances for user #{test_user_idx} : {distances}')
```

```
Recommended movie Ids for user #200 : [1001, 4108, 4780, 262, 249, 151, 1910,
5929, 6156, 94, 1463, 4632, 1043, 1821, 188, 139, 2827, 210, 4665, 1178, 169,
911, 287, 296, 1360, 1102, 708, 4728, 7, 23, 132, 5, 3039, 335, 34, 214, 119
7, 3165, 345, 376, 4282, 3595, 202, 1766, 8, 11, 5812, 3106, 3327, 6474, 796,
939, 5802, 62, 441, 26, 571, 327, 364, 2647, 618, 498, 1853, 5982, 361, 1987,
4464, 405, 2698, 141, 887, 6, 187, 627, 4659, 2172, 3193, 1798, 72, 1954, 248
2, 419, 28, 397, 64, 761, 203, 1249, 115, 1968, 396, 1222, 86, 2847, 6356, 46
67, 473, 37, 80, 2025]
Movie distances for user #200 : [0.1822, 0.1822, 0.1825, 0.1827, 0.1829, 0.18
32, 0.1835, 0.1835, 0.1861, 0.1873, 0.1877, 0.1881, 0.1882, 0.1882, 0.1887,
0.19, 0.1903, 0.1907, 0.1907, 0.1909, 0.1916, 0.1918, 0.1928, 0.193, 0.1933,
0.1937, 0.1937, 0.1939, 0.1957, 0.1963, 0.1965, 0.1975, 0.1977, 0.1987, 0.199
6, 0.2, 0.2002, 0.2026, 0.2026, 0.2027, 0.203, 0.2038, 0.2068, 0.2073, 0.207
6, 0.2079, 0.2084, 0.209, 0.2105, 0.2105, 0.2107, 0.211, 0.2112, 0.2121, 0.21
46, 0.2148, 0.2172, 0.2184, 0.2198, 0.2201, 0.2213, 0.2215, 0.2226, 0.2227,
0.2235, 0.2239, 0.2245, 0.2248, 0.2261, 0.2273, 0.2277, 0.2319, 0.233, 0.233
1, 0.2333, 0.2337, 0.2341, 0.2343, 0.2347, 0.2367, 0.2372, 0.2374, 0.2378, 0.
2383, 0.2388, 0.2401, 0.2412, 0.2418, 0.2421, 0.2421, 0.2427, 0.2471, 0.2476,
0.2523, 0.2531, 0.2547, 0.2643, 0.2681, 0.2855, 0.2893]
```

You got the movies closest to user 200's tastes. Now, you can see the titles.

In [71]:
```
titles_200 = reduced_df[reduced_df.item.isin(recommended_movies)].product_titl
e.unique()
titles_200
```

Out[71]:
```
array(['Downton Abbey Season 1', 'Grimm Season 1', 'Interstellar',
       'Justified Season 6', 'No Good Deed', 'Sneaky Pete - Season 1',
       'Dear Frankie', 'Grimm Season 3', 'Downton Abbey Season 3',
       'Justified Season 4', 'Amnesia', 'Sons of Anarchy Season 1',
       'Emperor', 'The Closer: The Complete Second Season',
       'Mountain Men Season 1', 'The Captive', 'American Sniper',
       'Godzilla (2014)', "Lee Daniels' The Butler", 'Red',
       'Transparent Season 1', 'The 100: The Complete Second Season',
       'Gone Girl', 'The Americans Season 1', 'Extant, Season 1',
       'Vikings Season 1', 'Lark Rise to Candleford Season 1',
       'Masterpiece: Inspector Lewis Season 6', 'Sherlock, Season 3',
       'Game of Thrones Season 1', 'Fargo Season 1',
       'American Horror Story: Coven', 'Gravity', 'The Last Legion',
       'The Wire Season 1', 'Breaking Bad Season 5', 'Brave',
       'Captain America: The Winter Soldier (Theatrical)',
       'An Unfinished Life', 'The Lone Ranger (2013)', 'The Judge (2014)',
       'All Is Lost', 'Elysium', 'Chappie',
       'Fringe: The Complete Second Season', 'Fury',
       'The Magic Of Belle Isle', 'Gran Torino', '24 Season 8',
       'The Closer: The Complete First Season',
       'Rise Of The Planet of the Apes', 'Escape Plan',
       '95ers: Time Runners', 'The November Man',
       'The Hunger Games: Catching Fire', 'Beetlejuice',
       'The Campaign (2012)', 'Joyful Noise', 'The Homesman', 'Divergent',
       'The Dark Knight Rises', 'Lucy', 'Wallander Season 2',
       'Kick-Ass 2', 'Campion Season 2',
       'The Dawn of the Planet of the Apes',
       'Cloud Atlas +Bonus Features', "Blue's Clues Season 2", 'Parkland',
       'Big Eyes', 'Into the Wild',
       "It's Always Sunny in Philadelphia Season 6", 'White House Down',
       'Hansel And Gretel: Witch Hunters', 'Taking Chance',
       'Saints and Soldiers', 'Dallas Buyers Club', 'R.I.P.D.',
       'Star Trek II: The Wrath of Khan', 'The Master',
       'Sherlock Holmes - Double Feature', 'Houseboat',
       'The Spectacular Now', 'Jiro Dreams of Sushi',
       'The Big Bang Theory: The Complete First Season',
       'Disneynature Chimpanzee', 'A Good Year', 'Catch Me If You Can',
       'Underworld Awakening', 'Don Jon', 'Transcendent Man',
       "World's Greatest Dad", 'RahXephon - Season 01',
       "Let's Kill Ward's Wife",
       'The Twilight Saga: Breaking Dawn Part 2', '180 South'],
      dtype=object)
```

Compare them with the favorite movies for user 200.

In [72]:
```
reduced_df.query('user==200 & star_rating == 5')
```

Out[72]:

	customer_id	product_id	star_rating	product_title	user	item
6753	43080952	B00IIVJU3Q	5	Philomena	200	180
14249	43080952	B00VFTA70I	5	The Imitation Game	200	88
28407	43080952	B00KG2NGPS	5	The Monuments Men	200	149
60486	43080952	B00DTOYJJ2	5	Breaking Bad The Final Season	200	79
71978	43080952	B00TPJHW7Q	5	The Theory of Everything	200	157
84419	43080952	B0047WJ12K	5	Inception	200	467
118790	43080952	B001P4NZ9I	5	Louis CK: Chewed Up	200	3840
119440	43080952	B0059R5Q08	5	Jane Eyre	200	2617
136597	43080952	B00TF87WQI	5	Rosewater	200	3606
161384	43080952	B00DGNO3WE	5	Quartet	200	1094
167594	43080952	B001VLKWUA	5	Out of Africa	200	3119

Question: Do you think these recommendations make sense? Explain why or why not.

Answer: From a subjective point of view, the recommendations make sense. If you compare them with the top 100 suggestions from the first regression model, you'll see they only have three elements in common.

In [73]:
```
np.isin(titles_200, titles.tail(100).product_title.unique()).sum()
```

Out[73]: 0

Super bonus question: Recover the predictions for user 201, and see how they compare with user 200. Are they still correlated? Do you think this approach was an improvement over the first regressor?

In [74]:
```
# Recover the predictions for user 201

test_user_idx = 201
u_one_json = json.loads(results[test_user_idx])
recommended_movies_201 = [int(movie_id) for movie_id in u_one_json['labels']]
```

In [75]:
```
# Print out recommendations

titles_201 = reduced_df[reduced_df.item.isin(recommended_movies_201)].product_
title.unique()
titles_201
```

Out[75]:
```
array(['Downton Abbey Season 1', 'Justified Season 6', 'No Good Deed',
       'Sneaky Pete - Season 1', 'Dear Frankie', 'Grimm Season 3',
       'Downton Abbey Season 3', 'Justified Season 4', 'Amnesia',
       'Sons of Anarchy Season 1', 'Emperor',
       'The Closer: The Complete Second Season', 'Mountain Men Season 1',
       'The Captive', 'American Sniper', 'Godzilla (2014)', 'Red',
       'Transparent Season 1', 'The 100: The Complete Second Season',
       'Gone Girl', 'The Americans Season 1', 'Extant, Season 1',
       'Vikings Season 1', 'Lark Rise to Candleford Season 1',
       'Masterpiece: Inspector Lewis Season 6', 'Sherlock, Season 3',
       'Game of Thrones Season 1', 'Fargo Season 1',
       'American Horror Story: Coven', 'Gravity', 'The Wire Season 1',
       'Breaking Bad Season 5', '3 Days to Kill',
       'Transformers: Age of Extinction', 'Brave', 'An Unfinished Life',
       'The Lone Ranger (2013)', 'All Is Lost',
       'Confessions of a Shopaholic', 'Elysium',
       'Fringe: The Complete Second Season', 'Fury',
       'The Magic Of Belle Isle', 'Suits Season 1', 'Gran Torino',
       '24 Season 8', 'The Professional',
       'The Closer: The Complete First Season',
       'Rise Of The Planet of the Apes', 'Escape Plan',
       '95ers: Time Runners', 'The November Man',
       'The Hunger Games: Catching Fire', 'Beetlejuice',
       'The Campaign (2012)', 'The Homesman', 'Divergent',
       'Sons of Anarchy Season 2', 'The Dark Knight Rises', 'Lucy',
       'Wallander Season 2', 'Kick-Ass 2', 'Campion Season 2',
       'The New Yorker Presents - Season 1',
       'The Dawn of the Planet of the Apes',
       'Cloud Atlas +Bonus Features', "Blue's Clues Season 2",
       'The River King', 'I Am Number Four', 'Parkland', 'Into the Wild',
       'Taking Chance', "Hachi: A Dog's Tale", 'Saints and Soldiers',
       'Dallas Buyers Club', "Moms' Night Out", 'R.I.P.D.',
       'The Penguins of Madagascar Season 1',
       'Star Trek II: The Wrath of Khan',
       'Percy Jackson: Sea Of Monsters',
       'Sherlock Holmes - Double Feature', 'Dogville',
       'The Spectacular Now', 'Jiro Dreams of Sushi', 'The Last Legion',
       'The Big Bang Theory: The Complete First Season',
       'Disneynature Chimpanzee', 'A Good Year', 'Catch Me If You Can',
       'Underworld Awakening', 'Don Jon', '20 Feet From Stardom',
       'Conan The Barbarian (2011)', 'RahXephon - Season 01',
       "Let's Kill Ward's Wife",
       'The Twilight Saga: Breaking Dawn Part 2', '180 South'],
      dtype=object)
```

In [76]:
```
# Compare the two predictions

overlap = np.isin(titles_200, titles_201).sum()
print(f'The recommendations for "user 201" that are present in "user 200" are:
{overlap} out of: {len(titles_200)}')
```

```
The recommendations for "user 201" that are present in "user 200" are: 81 out
of: 96
```

In [77]:
```
# Compare with user 201 likes

reduced_df.query('user==201 & star_rating == 5')
```

Out[77]:

	customer_id	product_id	star_rating	product_title	user	item
6616	51970325	B00ESNE696	5	The Good Wife, Season 5	201	340
9313	51970325	B0064MGU98	5	The Good Wife, Season 1	201	661
9349	51970325	B005HFB85Y	5	Philly Season 1	201	4156
21229	51970325	B00CW8DRU8	5	The Newsroom: Season 1	201	216
25844	51970325	B00COGXGNQ	5	Safe Haven	201	585
47019	51970325	B004BGSGLO	5	As Time Goes By Season 3	201	2273
47048	51970325	B004BGQA8A	5	As Time Goes By Season 1	201	1480
66576	51970325	B004BGS3C6	5	As Time Goes By Season 2	201	2136
66608	51970325	B004BTUON4	5	As Time Goes By Season 5	201	2798
78467	51970325	B0031KE9C6	5	Veronica Mars Season One	201	2130
89545	51970325	B00IUBXPEY	5	The Hunger Games: Catching Fire	201	28
96626	51970325	B00DQIT734	5	Suits Season 3	201	606
104031	51970325	B008BQ8VOC	5	Suits Season 2	201	1605
110522	51970325	B007IVN8QW	5	Dance Academy Season 1	201	4238
112770	51970325	B005544V6A	5	Suits Season 1	201	2036
131848	51970325	B00A954YS0	5	Dance Academy Season 2	201	6239
148312	51970325	B009JZWO3I	5	The Good Wife, Season 4	201	1259
151569	51970325	B007O62JG6	5	Tangled Ever After (Short)	201	1869
170853	51970325	B00196YNWU	5	Annie	201	6028

In [78]:
```
test_user_idx = 900
u_one_json = json.loads(results[test_user_idx])
recommended_movies_900 = [int(movie_id) for movie_id in u_one_json['labels']]
titles_900 = reduced_df[reduced_df.item.isin(recommended_movies_201)].product_
title.unique()
overlap_900 = np.isin(titles_200, titles_900).sum()
print(f'The recommendations for "user 900" that are present in "user 200" are:
{overlap} out of: {len(titles_200)}')
reduced_df.query('user==900 & star_rating == 5')
```

```
The recommendations for "user 900" that are present in "user 200" are: 81 out
of: 96
```

Out[78]:

	customer_id	product_id	star_rating	product_title	user	item
4745	35806677	B00ECU34ZM	5	Oblivion	900	21
52574	35806677	B0046B8GEQ	5	Despicable Me	900	237
61982	35806677	B000UU2YKE	5	Battlestar Galactica Season 1	900	5837
62258	35806677	B00HUNOS2G	5	Riddick	900	289
70465	35806677	B00IMYZL2S	5	Thor: The Dark World	900	136
75063	35806677	B00JJHDX9A	5	The Hobbit: The Desolation of Smaug	900	325
75865	35806677	B001GZPZXU	5	Cinderella Man	900	4117
83735	35806677	B00HB8UWCU	5	Escape Plan	900	125
98489	35806677	B005NFVZTM	5	X-men: First Class	900	928
110584	35806677	B00BUCW54Q	5	Jack Reacher	900	230
124240	35806677	B001MXKLNA	5	The Pink Panther Show Season 1	900	1438
139899	35806677	B00E8B1358	5	G.I. Joe: Retaliation	900	993
147561	35806677	B00AYB0TOW	5	Dredd	900	194
167251	35806677	B000VK99Z6	5	Hot Fuzz	900	1914

Answer: This final model is again recommending the same films to different users. That means there is a big cluster of movies that the KNN dims as close neighbors, and this cluster is dominating the rest. The thing about recommendation is that it is a subjective matter. You can get a sense of the taste of one person. but at the same time that person doesn't only like one genre. Most of the users have at least one comedy or animation movie in combination with their usual type of preference.

There are a number of things you can do to improve these models, such as adding features besides rating, trying different feature selection, hyperparameter tuning, and changing the models. The most sophisticated recommendation algorithms are based on deep learning. This can also be explored.

That is it! You now have a working recommender system that can tell you the top 100 movies for a user. Feel free to optimize and play with the hyperparameters and data to see if you can create an even better recommender system.

Final thoughts

In this notebook, you used different techniques to create a recommendations system using only Amazon SageMaker built-in algorithms. You learned how to prepare data in different formats and do feature engineering. You were able to identify problems with your trained models and reframe the problem in different ways to achieve an end result.

As you can see now, training a model requires a lot of steps, preparation, and validation. It is not a streamlined process but an iterative one. You can think of this as a virtuous cycle that usually has the following steps:

- Define the (business) problem.
- Frame the problem as a machine learning problem.
- Prepare data and perform freature engineerin.
- Train and evaluate the model.
- Deploy the model (inference).
- Monitor and evaluate.

Every step has its own challenges, and each of the steps feeds each other. So it is important to pay attention to the entire pipeline, not only the model training.