

Movie Recommendation System Using Different Models

Introduction

Recommendation system is one of the most widespread machine learning applications in the industry. It is not only built for movies and music streaming, but also for multiple products and services like e-commerce business, news and so on. Companies like Netflix, Youtube, and Amazon have leveraged recommendation systems to discover user preference and provide them with more relevant goods and services to enhance user experience and generate recurring revenue. No doubt, successful recommendation system is an indispensable part of their business success.

This assignment went through processes of building necessary elements for recommender engine, implementing an ensemble of machine learning algorithms and making use of AWS Sagemaker endpoint. The final pipeline is capable of offering specific movie suggestions for different users based on a combination result of three different models: ALS, AWS SageMaker and Scikit-Surprise.

Dataset

The dataset used in this assignment is the famous MovieLens dataset. There are various versions available online(<https://grouplens.org/datasets/movielens/> (<https://grouplens.org/datasets/movielens/>)), but for ease of use, 100k version is chosen to apply to those three models mentioned above. Features included in 100k version dataset are rating, genre, tag, movieid, userid, tagid and so on.

MODEL 1: ALS Estimator Using Spark

Step 1 : Set up the environment / necessary housekeeping

In [22]:

```
!pip install --upgrade pip
!pip install -q findspark
```

Requirement already up-to-date: pip in /home/ec2-user/anaconda3/envs/python3/lib/python3.6/site-packages (20.0.2)

In [23]:

```
#Python library PySpark is imported to interface with Spark
import pyspark
# get a spark context
sc = pyspark.SparkContext.getOrCreate()
print(sc)
# get a spark session
spark = pyspark.sql.Session.builder.getOrCreate()
print(spark)
spark.version
```

```
<SparkContext master=local[*] appName=pyspark-shell>
<pyspark.sql.session.Session object at 0x7f0b47595198>
```

Out[23]:

```
'2.3.4'
```

In [26]:

```
from pyspark.sql import SparkSession
from pyspark.sql import Row
from pyspark.sql.types import StructType
from pyspark.sql.types import StructField
from pyspark.sql.types import *
import pandas as pd
import numpy as np
```

Step 2 : Data preprocessing

In [40]:

```
#download data from grouplens
!wget http://files.grouplens.org/datasets/movielens/ml-100k.zip
!unzip -o ml-100k.zip
#shuffle the data
%cd ml-100k
!shuf ua.base -o ua.base.shuffled

--2020-04-28 14:06:19-- http://files.grouplens.org/datasets/movielens/ml-100k.zip
Resolving files.grouplens.org (files.grouplens.org)... 128.101.65.152
Connecting to files.grouplens.org (files.grouplens.org)|128.101.65.152|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4924029 (4.7M) [application/zip]
Saving to: 'ml-100k.zip.1'

ml-100k.zip.1      100%[=====>]   4.70M   8.23MB/s
in 0.6s

2020-04-28 14:06:20 (8.23 MB/s) - 'ml-100k.zip.1' saved [4924029/4924029]

Archive:  ml-100k.zip
  inflating: ml-100k/allbut.pl
  inflating: ml-100k/mku.sh
  inflating: ml-100k/README
  inflating: ml-100k/u.data
  inflating: ml-100k/u.genre
  inflating: ml-100k/u.info
  inflating: ml-100k/u.item
  inflating: ml-100k/u.occupation
  inflating: ml-100k/u.user
  inflating: ml-100k/u1.base
  inflating: ml-100k/u1.test
  inflating: ml-100k/u2.base
  inflating: ml-100k/u2.test
  inflating: ml-100k/u3.base
  inflating: ml-100k/u3.test
  inflating: ml-100k/u4.base
  inflating: ml-100k/u4.test
  inflating: ml-100k/u5.base
  inflating: ml-100k/u5.test
  inflating: ml-100k/ua.base
  inflating: ml-100k/ua.test
  inflating: ml-100k/ub.base
  inflating: ml-100k/ub.test
/home/ec2-user/SageMaker/ml-100k
```

In [41]:

```
#load training data
train_movie_ratings= pd.read_csv('ua.base.shuffled', sep='\t',
                                index_col=False, names=['user_id' , 'movie_id' , 'rating'])
#load test data
test_movie_ratings= pd.read_csv('ua.test', sep='\t',
                                index_col=False, names=['user_id' , 'movie_id' , 'rating'])
#train_movie_ratings.head()
#test_movie_ratings.head()
```

In [251]:

```
# create a SparkSession
spark = SparkSession.builder.getOrCreate()
# create the training and test dataframe
training=spark.createDataFrame(train_movie_ratings)
test=spark.createDataFrame(test_movie_ratings)
#show the column names of the dataset
training.printSchema()
# count length of the dataset
print(training.count())
```

```
root
 |-- user_id: long (nullable = true)
 |-- movie_id: long (nullable = true)
 |-- rating: long (nullable = true)
```

90570

Step 3 : Train ALS estimator and perform cross validation

In [253]:

```
from pyspark.ml.tuning import CrossValidator, ParamGridBuilder, TrainValidationSplit
from pyspark.ml.evaluation import RegressionEvaluator
from pyspark.ml.recommendation import ALS

# Build the recommendation model using ALS on the training data
als = ALS(maxIter=3, rank=10, regParam=0.1, userCol="user_id", itemCol="movie_id", ratingCol="rating", coldStartStrategy="drop")

#Set the parameter grid
paramGrid = ParamGridBuilder() \
    .addGrid(als.regParam, [0.03,0.1,0.3]) \
    .addGrid(als.rank, [3,10,30]).build()

regEval = RegressionEvaluator(metricName="rmse", labelCol="rating", predictionCol="prediction")

#Start the cross validation and fit on the training dataset
crossVal = CrossValidator(estimator=als, estimatorParamMaps=paramGrid, evaluator=regEval)
cvModel = crossVal.fit(training)
```

In [254]:

```
# Show the metrics form the CrossValidation
print(cvModel.avgMetrics)

# Gives us the parameter combinations
print(cvModel.getEstimatorParamMaps())
paramMap = list(zip(cvModel.getEstimatorParamMaps(), cvModel.avgMetrics))

# Print the parameter that gives us the smallest rmse
paramMin = min(paramMap, key=lambda x: x[1])
print(paramMin)
```

```
[0.9736478590531773, 1.0256734372601422, 1.0732557393493907, 0.94797
94111368625, 0.9499695766544693, 0.9504500279730561, 0.9713087649862
429, 1.0234699008431651, 1.0448014103739727]
[{'Param(parent='ALS_446f9135931edb765cfb', name='regParam', doc='reg
ularization parameter (>= 0).'): 0.03, Param(parent='ALS_446f9135931
edb765cfb', name='rank', doc='rank of the factorization'): 3}, {'Para
m(parent='ALS_446f9135931edb765cfb', name='regParam', doc='regulariz
ation parameter (>= 0).'): 0.03, Param(parent='ALS_446f9135931edb765
cfb', name='rank', doc='rank of the factorization'): 10}, {'Param(par
ent='ALS_446f9135931edb765cfb', name='regParam', doc='regularization
parameter (>= 0).'): 0.03, Param(parent='ALS_446f9135931edb765cfb',
name='rank', doc='rank of the factorization'): 30}, {'Param(parent='A
LS_446f9135931edb765cfb', name='regParam', doc='regularization param
eter (>= 0).'): 0.1, Param(parent='ALS_446f9135931edb765cfb', name
='rank', doc='rank of the factorization'): 3}, {'Param(parent='ALS_44
6f9135931edb765cfb', name='regParam', doc='regularization parameter
(>= 0).'): 0.1, Param(parent='ALS_446f9135931edb765cfb', name='ran
k', doc='rank of the factorization'): 10}, {'Param(parent='ALS_446f91
35931edb765cfb', name='regParam', doc='regularization parameter (>=
0).'): 0.1, Param(parent='ALS_446f9135931edb765cfb', name='rank', do
c='rank of the factorization'): 30}, {'Param(parent='ALS_446f9135931e
db765cfb', name='regParam', doc='regularization parameter (>= 0).'):
0.3, Param(parent='ALS_446f9135931edb765cfb', name='rank', doc='rank
of the factorization'): 3}, {'Param(parent='ALS_446f9135931edb765cf
b', name='regParam', doc='regularization parameter (>= 0).'): 0.3, P
aram(parent='ALS_446f9135931edb765cfb', name='rank', doc='rank of th
e factorization'): 10}, {'Param(parent='ALS_446f9135931edb765cfb', na
me='regParam', doc='regularization parameter (>= 0).'): 0.3, Param(p
arent='ALS_446f9135931edb765cfb', name='rank', doc='rank of the fact
orization'): 30}]
({'Param(parent='ALS_446f9135931edb765cfb', name='regParam', doc='reg
ularization parameter (>= 0).'): 0.1, Param(parent='ALS_446f9135931e
db765cfb', name='rank', doc='rank of the factorization'): 3}, 0.9479
794111368625)
```

In [256]:

```
#Implement the als model with the best parameter
als = ALS(maxIter=3, rank=3, regParam=0.1, userCol="user_id", itemCol="movie_id"
, ratingCol="rating",
        coldStartStrategy="drop")
model = als.fit(training)
```

In [257]:

```
# Evaluate the model by computing the RMSE on the test data
predictions = model.transform(test)
evaluator = RegressionEvaluator(metricName="rmse", labelCol="rating",
                                predictionCol="prediction")
rmse = evaluator.evaluate(predictions)
print("RMSE = " + str(rmse))
```

RMSE = 0.9568521793447027

To train an ALS estimator, initially we can simply use default parameters or set certain values, at the same time, specifying user, rating and item column respectively. For cold start strategy, users can choose between "drop" and "NaN". In order for us to derive a performance metric to evaluate the recommendation system, we set the coldStartStrategy parameter as "drop" to drop rows in the prediction dataframe which contains NaN values.

Further, cross validation with parameter tuning is applied to enhance ALS model. We set a parameter grid and apply the cross validation method, finally the best parameter with the smallest rmse is derived. When implementing the ALS model with the best parameter, the RMSE computed on the test data is 0.957.

The prediction result of the first model ALS estimator might not be satisfactory, but this is only from a single ALS model. We can expect further improvement later on when aggregating multiple effective models altogether.

In [258]:

```
#Get the prediction result
predictions.show()
predictions.createOrReplaceTempView('predictions')
```

user_id	movie_id	rating	prediction
251	148	2	3.18209
580	148	4	3.5047975
602	148	4	3.725699
372	148	5	4.218792
274	148	2	3.4845352
923	148	4	3.8626158
447	148	4	2.895263
586	148	3	3.4014761
761	148	5	3.065056
677	148	4	3.8198245
930	148	1	3.0983498
51	148	3	3.8849201
403	148	5	3.7467852
361	148	1	2.5771089
893	148	3	3.3657274
396	148	4	3.575336
203	148	3	3.2016485
186	148	4	3.5464697
6	463	4	3.6850455
23	463	4	3.6496463

only showing top 20 rows

Step 4 · Movie recommendation results for each user

In [259]:

```
# Get top 5 movie recommendations for each user
userRecs = model.recommendForAllUsers(5)
userRecs.show()
```

```
+-----+-----+
|user_id| recommendations|
+-----+-----+
|    471| [[1554, 7.181269]...|
|    463| [[1155, 4.350897]...|
|    833| [[1643, 5.1402636...|
|    496| [[1536, 4.9627852...|
|    148| [[1463, 5.825232]...|
|    540| [[814, 5.399788],...|
|    392| [[1463, 6.0887423...|
|    243| [[1536, 5.3301077...|
|    623| [[1463, 5.2869897...|
|    737| [[1558, 6.0107007...|
|    897| [[814, 5.7407727]...|
|    858| [[1536, 5.438197]...|
|     31| [[1463, 5.5697336...|
|    516| [[814, 5.6217227]...|
|    580| [[1155, 5.105934]...|
|    251| [[1463, 5.6510725...|
|    451| [[1554, 6.0504746...|
|     85| [[1463, 5.261311]...|
|    137| [[1554, 6.4420385...|
|    808| [[1463, 6.94685],...|
+-----+-----+
```

only showing top 20 rows

In [261]:

```
# Get top 5 movie recommendations for a specified set of users
users = test.select(als.getUserCol()).distinct()
userSubsetRecs = model.recommendForUserSubset(users, 5)
model.recommendProductsForUsers(3).take(2)
userSubsetRecs.show()
```

```
+-----+-----+
|user_id| recommendations|
+-----+-----+
|    463| [[1155, 4.350897]...|
|    496| [[1536, 4.9627852...|
|    833| [[1643, 5.1402636...|
|    471| [[1554, 7.181269]...|
|    148| [[1463, 5.825232]...|
|    243| [[1536, 5.3301077...|
|    858| [[1536, 5.438197]...|
|    737| [[1558, 6.0107007...|
|    897| [[814, 5.7407727]...|
|    623| [[1463, 5.2869897...|
|    392| [[1463, 6.0887423...|
|    540| [[814, 5.399788],...|
|     31| [[1463, 5.5697336...|
|    516| [[814, 5.6217227]...|
|     85| [[1463, 5.261311]...|
|    451| [[1554, 6.0504746...|
|    580| [[1155, 5.105934]...|
|    137| [[1554, 6.4420385...|
|    251| [[1463, 5.6510725...|
|    808| [[1463, 6.94685],...|
+-----+-----+
```

only showing top 20 rows

Step 5 : Adjust prediction result format

In [262]:

```
#convert spark dataframe to pandas dataframe for ease of use
pred_als=userSubsetRecs.toPandas()
pred_als.head()
```

Out[262]:

	user_id	recommendations
0	463	[(1155, 4.350896835327148), (1242, 4.192982673...
1	496	[(1536, 4.962785243988037), (1639, 4.897671222...
2	833	[(1643, 5.140263557434082), (1536, 5.043638706...
3	471	[(1554, 7.18126916885376), (1472, 6.2251939773...
4	148	[(1463, 5.825232028961182), (1536, 5.598376750...

In [270]:

```
# expand recommendation results into a separate dataframe
rmd=pred_als[ 'recommendations' ].apply(pd.Series)

# rename each feature and add userid
rmd=rmd.rename(columns = lambda x : 'recommendations_' + str(x))
rmd=pd.concat([pred_als["user_id"], rmd[:]], axis=1)
rmd.head()
```

Out[270]:

	user_id	recommendations_0	recommendations_1	recommendations_2	recommendations_3
0	463	(1155, 4.350896835327148)	(1242, 4.1929826736450195)	(1639, 4.190911293029785)	(1645, 4.1444854736328125)
1	496	(1536, 4.962785243988037)	(1639, 4.897671222686768)	(1643, 4.802825450897217)	(1558, 4.748034954071045)
2	833	(1643, 5.140263557434082)	(1536, 5.043638706207275)	(1463, 4.988624095916748)	(1558, 4.986846446990967)
3	471	(1554, 7.18126916885376)	(1472, 6.225193977355957)	(868, 5.669477462768555)	(74, 5.483709812164307)
4	148	(1463, 5.825232028961182)	(1536, 5.598376750946045)	(1643, 5.589471817016602)	(814, 5.552923202514648)

In [271]:

```
user_rmd=pd.DataFrame(columns=[ "userId", "recommendations" ])
# place different recommendations in different rows
for i in range(len(rmd)):
    user_rmd=user_rmd.append( {"userId":rmd.loc[i][ "user_id" ], "recommendations":rmd.loc[i][ "recommendations_0" ]}, ignore_index=True)
    user_rmd=user_rmd.append( {"userId":rmd.loc[i][ "user_id" ], "recommendations":rmd.loc[i][ "recommendations_1" ]}, ignore_index=True)
    user_rmd=user_rmd.append( {"userId":rmd.loc[i][ "user_id" ], "recommendations":rmd.loc[i][ "recommendations_2" ]}, ignore_index=True)
    user_rmd=user_rmd.append( {"userId":rmd.loc[i][ "user_id" ], "recommendations":rmd.loc[i][ "recommendations_3" ]}, ignore_index=True)
    user_rmd=user_rmd.append( {"userId":rmd.loc[i][ "user_id" ], "recommendations":rmd.loc[i][ "recommendations_4" ]}, ignore_index=True)

user_rmd.head()
```

Out[271]:

	userId	recommendations
0	463	(1155, 4.350896835327148)
1	463	(1242, 4.1929826736450195)
2	463	(1639, 4.190911293029785)
3	463	(1645, 4.1444854736328125)
4	463	(1651, 4.1444854736328125)

In [357]:

```
# expand recommendations column into its own dataframe
recommendations = user_rmd['recommendations'].apply(pd.Series)
# rename those two columns
recommendations.columns=["movieId", "predicted_rating"]
# Get the final format
als_recommendations=pd.concat([user_rmd["userId"], recommendations[:]], axis=1)
als_recommendations.head()
```

Out[357]:

	userId	movieId	predicted_rating
0	463	1155.0	4.350897
1	463	1242.0	4.192983
2	463	1639.0	4.190911
3	463	1645.0	4.144485
4	463	1651.0	4.144485

Step 6 : Recommendation results for given user list

In [272]:

```
# Specified user list
user_list=[198,11,314,184,163,710,881,504,267,653]
```

In [553]:

```
als_user=als_recommendations[als_recommendations.userId.isin(user_list)]  
als_user
```

Out[553]:

	userId	movieId	predicted_rating
1350	881	1463.0	5.182528
1351	881	814.0	5.130446
1352	881	1554.0	4.892687
1353	881	1536.0	4.831121
1354	881	1500.0	4.821637
1565	163	1554.0	4.327571
1566	163	814.0	4.225963
1567	163	1463.0	4.040522
1568	163	1500.0	4.017696
1569	163	1293.0	3.912068
1710	504	814.0	5.423500
1711	504	1463.0	5.170277
1712	504	1449.0	5.041507
1713	504	1500.0	5.026254
1714	504	1293.0	5.019629
2625	314	1554.0	5.709924
2626	314	1662.0	5.508576
2627	314	814.0	5.385200
2628	314	1472.0	5.310832
2629	314	867.0	5.249988
2975	267	1463.0	6.188421
2976	267	1536.0	6.150066
2977	267	1643.0	6.073094
2978	267	814.0	5.907568
2979	267	1449.0	5.808830
3345	653	814.0	4.083629
3346	653	1463.0	3.963510
3347	653	1554.0	3.833260
3348	653	1500.0	3.810387
3349	653	1449.0	3.769383
3350	710	1463.0	5.538241
3351	710	1643.0	5.523271
3352	710	1536.0	5.500473
3353	710	814.0	5.106331
3354	710	1449.0	5.064532
3910	11	814.0	5.106674

	userId	movieId	predicted_rating
3911	11	1463.0	5.019383
3912	11	1536.0	4.789463
3913	11	1449.0	4.766718
3914	11	1500.0	4.736952
4020	198	1463.0	5.172289
4021	198	1643.0	5.061422
4022	198	1536.0	5.019175
4023	198	814.0	4.800490
4024	198	1449.0	4.650020
4325	184	1536.0	5.393170
4326	184	1463.0	5.359420
4327	184	1643.0	5.285818
4328	184	814.0	5.155608
4329	184	1449.0	5.117677

In [360]:

```
#define a function to get the recommendation stored in a dictionary
def GetAlsRec(userId):
    df_als=als_user[als_user.userId==userId]
    rec=df_als.movieId.values.astype('int').tolist()
    rec_dic={'userId':userId, 'als_recommendations':rec}
    return rec_dic
GetAlsRec(184)
```

Out[360]:

```
{'userId': 184, 'als_recommendations': [1536, 1463, 1643, 814, 1449]}
```

In [361]:

```
als_user_rec=[GetAlsRec(i) for i in user_list]
als_user_result=pd.DataFrame(als_user_rec)
als_user_result
```

Out[361]:

	als_recommendations	userId
0	[1463, 1643, 1536, 814, 1449]	198
1	[814, 1463, 1536, 1449, 1500]	11
2	[1554, 1662, 814, 1472, 867]	314
3	[1536, 1463, 1643, 814, 1449]	184
4	[1554, 814, 1463, 1500, 1293]	163
5	[1463, 1643, 1536, 814, 1449]	710
6	[1463, 814, 1554, 1536, 1500]	881
7	[814, 1463, 1449, 1500, 1293]	504
8	[1463, 1536, 1643, 814, 1449]	267
9	[814, 1463, 1554, 1500, 1449]	653

Model 2 : Scikit Surprise

Surprise is a python scikit package used for recommendation system and it stands for Simple Python Recommendation System Engine. Users can use built-in datasets like movielens and custom dataset to build a recommendation system. There are various ready-to-use algorithms in Surprise package including collaborative filtering, matrix decomposition and so on. This package is a quite simple and easy to use. However, Surprise doesn't support content-based information and implicit ratings.

Example below demonstrates the process of building a simple recommendation system using Surprise. The RMSE on the testdata prediction is 0.675, which is much lower than ALS estimator. Then a function is defined to find the top movie recommendations for each user.

Step 1 : Install and import Surprise package

In [156]:

```
#relevant surprise imports
!pip install scikit-surprise
import surprise
from collections import defaultdict
from surprise import SVD
from surprise import Dataset
from surprise import accuracy
```

Requirement already satisfied: scikit-surprise in /home/ec2-user/anaconda3/envs/python3/lib/python3.6/site-packages (1.1.0)

Requirement already satisfied: scipy>=1.0.0 in /home/ec2-user/anaconda3/envs/python3/lib/python3.6/site-packages (from scikit-surprise) (1.1.0)

Requirement already satisfied: six>=1.10.0 in /home/ec2-user/anaconda3/envs/python3/lib/python3.6/site-packages (from scikit-surprise) (1.11.0)

Requirement already satisfied: numpy>=1.11.2 in /home/ec2-user/anaconda3/envs/python3/lib/python3.6/site-packages (from scikit-surprise) (1.18.3)

Requirement already satisfied: joblib>=0.11 in /home/ec2-user/anaconda3/envs/python3/lib/python3.6/site-packages (from scikit-surprise) (0.14.1)

Step 2 : Fit SVD model and compute accuracy

In [147]:

```
#load the dataset and separate into trainset and testset
#download the movielens-100k dataset if it has not already been downloaded
data = Dataset.load_builtin('ml-100k')
trainset = data.build_full_trainset()
testset = trainset.build_testset()

# train SVD algorithm on the movielens dataset.
alg_svd = SVD()
alg_svd.fit(trainset)
pred_svd = alg_svd.test(testset)

# Get the RMSE of the prediction result
accuracy.rmse(pred_svd, verbose=True)
```

RMSE: 0.6750

Out[147]:

0.6750463451886863

In [155]:

```
def get_top_n(predictions, n=5):  
  
    # map the predictions to each userid  
    top_n = defaultdict(list)  
    for uid, iid, true_r, est, _ in predictions:  
        top_n[uid].append((iid, est))  
  
    # order the predictions for each user and get the 5 highest ones.  
    for uid, user_ratings in top_n.items():  
        user_ratings.sort(key=lambda x: x[1], reverse=True)  
        top_n[uid] = user_ratings[:n]  
  
    return top_n
```

In [148]:

```
# retrieve top recommendations for all user and movie pairs (u, i) in the test set.  
top_n = get_top_n(pred_svd, n=5)
```


In [306]:

```
# Print the recommended movieid for each user
sp_recommendations=[]
for uid, user_ratings in top_n.items():
    sp_recommendations.append({'userId':uid, 'sp_recomendations':[iid for (iid,
_) in user_ratings]})
sp_recommendations
```

Out[306]:

```

[{'userId': '196', 'sp_recomendations': ['285', '663', '655', '100',
7', '8']},
 {'userId': '186', 'sp_recomendations': ['98', '71', '939', '79', '3',
00']},
 {'userId': '22', 'sp_recomendations': ['173', '50', '96', '144', '1',
74']},
 {'userId': '244', 'sp_recomendations': ['475', '9', '56', '208', '1',
69']},
 {'userId': '166', 'sp_recomendations': ['313', '300', '258', '347',
'322']},
 {'userId': '298', 'sp_recomendations': ['318', '174', '496', '483',
'132']},
 {'userId': '115', 'sp_recomendations': ['56', '127', '100', '511',
'137']},
 {'userId': '253', 'sp_recomendations': ['318', '496', '64', '427',
'483']},
 {'userId': '305', 'sp_recomendations': ['474', '462', '134', '709',
'483']},
 {'userId': '6', 'sp_recomendations': ['511', '474', '134', '135',
'483']},
 {'userId': '62', 'sp_recomendations': ['56', '12', '173', '50', '17',
2']},
 {'userId': '286', 'sp_recomendations': ['83', '588', '707', '315',
'133']},
 {'userId': '200', 'sp_recomendations': ['174', '50', '172', '483',
'169']},
 {'userId': '210', 'sp_recomendations': ['187', '127', '50', '357',
'197']},
 {'userId': '224', 'sp_recomendations': ['22', '318', '97', '402',
'313']},
 {'userId': '303', 'sp_recomendations': ['234', '483', '50', '64',
'100']},
 {'userId': '122', 'sp_recomendations': ['511', '127', '197', '191',
'180']},
 {'userId': '194', 'sp_recomendations': ['127', '64', '318', '168',
'12']},
 {'userId': '291', 'sp_recomendations': ['56', '12', '64', '48', '9',
8']},
 {'userId': '234', 'sp_recomendations': ['511', '654', '170', '199',
'603']},
 {'userId': '119', 'sp_recomendations': ['199', '272', '313', '275',
'511']},
 {'userId': '167', 'sp_recomendations': ['133', '435', '318', '478',
'615']},
 {'userId': '299', 'sp_recomendations': ['169', '89', '408', '474',
'168']},
 {'userId': '308', 'sp_recomendations': ['56', '169', '285', '12',
'187']},
 {'userId': '95', 'sp_recomendations': ['174', '496', '194', '423',
'204']},
 {'userId': '38', 'sp_recomendations': ['419', '69', '97', '313', '7',
1']},
 {'userId': '102', 'sp_recomendations': ['98', '272', '89', '183',
'174']},
 {'userId': '63', 'sp_recomendations': ['100', '408', '251', '1137',
'276']},
 {'userId': '160', 'sp_recomendations': ['127', '100', '187', '56',
'169']},
 {'userId': '50', 'sp_recomendations': ['1084', '253', '475', '100

```

```

8', '288']},
{'userId': '301', 'sp_recomendations': ['318', '64', '50', '79', '3
57']},
{'userId': '225', 'sp_recomendations': ['603', '480', '479', '482',
'478']},
{'userId': '290', 'sp_recomendations': ['174', '172', '50', '181',
'22']},
{'userId': '97', 'sp_recomendations': ['408', '169', '89', '663',
'661']},
{'userId': '157', 'sp_recomendations': ['127', '50', '515', '1', '1
00']},
{'userId': '181', 'sp_recomendations': ['274', '282', '25', '275',
'222']},
{'userId': '278', 'sp_recomendations': ['22', '313', '603', '173',
'515']},
{'userId': '276', 'sp_recomendations': ['56', '169', '50', '183',
'174']},
{'userId': '7', 'sp_recomendations': ['661', '127', '174', '190',
'135']},
{'userId': '10', 'sp_recomendations': ['483', '64', '191', '480',
'185']},
{'userId': '284', 'sp_recomendations': ['272', '315', '302', '269',
'303']},
{'userId': '201', 'sp_recomendations': ['56', '185', '318', '276',
'340']},
{'userId': '287', 'sp_recomendations': ['64', '1', '9', '28', '5
6']},
{'userId': '246', 'sp_recomendations': ['50', '100', '172', '12',
'11']},
{'userId': '242', 'sp_recomendations': ['1137', '306', '268', '27
5', '740']},
{'userId': '249', 'sp_recomendations': ['169', '100', '483', '603',
'408']},
{'userId': '99', 'sp_recomendations': ['64', '12', '172', '181', '1
74']},
{'userId': '178', 'sp_recomendations': ['173', '174', '64', '50',
'12']},
{'userId': '251', 'sp_recomendations': ['64', '50', '480', '520',
'313']},
{'userId': '81', 'sp_recomendations': ['98', '318', '210', '79', '1
69']},
{'userId': '260', 'sp_recomendations': ['313', '270', '1243', '32
6', '272']},
{'userId': '25', 'sp_recomendations': ['169', '408', '114', '189',
'98']},
{'userId': '59', 'sp_recomendations': ['12', '127', '100', '172',
'50']},
{'userId': '72', 'sp_recomendations': ['9', '191', '318', '56', '12
7']},
{'userId': '87', 'sp_recomendations': ['96', '496', '174', '216',
'204']},
{'userId': '42', 'sp_recomendations': ['496', '174', '318', '50',
'210']},
{'userId': '292', 'sp_recomendations': ['56', '64', '483', '223',
'169']},
{'userId': '20', 'sp_recomendations': ['496', '22', '87', '174', '9
8']},
{'userId': '13', 'sp_recomendations': ['493', '663', '484', '202',
'511']},
{'userId': '138', 'sp_recomendations': ['318', '483', '98', '100',
'513']},

```

```

{'userId': '60', 'sp_recomendations': ['474', '89', '641', '483',
'178']},
{'userId': '57', 'sp_recomendations': ['318', '50', '64', '181',
'1']},
{'userId': '223', 'sp_recomendations': ['313', '22', '69', '143',
'318']},
{'userId': '189', 'sp_recomendations': ['185', '480', '603', '134',
'318']},
{'userId': '243', 'sp_recomendations': ['285', '511', '632', '318',
'86']},
{'userId': '92', 'sp_recomendations': ['56', '180', '12', '169', '5
0']},
{'userId': '241', 'sp_recomendations': ['313', '302', '286', '268',
'887']},
{'userId': '254', 'sp_recomendations': ['174', '172', '50', '22',
'98']},
{'userId': '293', 'sp_recomendations': ['127', '474', '483', '134',
'100']},
{'userId': '127', 'sp_recomendations': ['288', '300', '230', '228',
'271']},
{'userId': '222', 'sp_recomendations': ['174', '22', '173', '172',
'144']},
{'userId': '267', 'sp_recomendations': ['114', '50', '474', '408',
'169']},
{'userId': '11', 'sp_recomendations': ['83', '318', '9', '286', '60
3']},
{'userId': '8', 'sp_recomendations': ['127', '174', '50', '183', '1
87']},
{'userId': '162', 'sp_recomendations': ['50', '174', '181', '28',
'79']},
{'userId': '279', 'sp_recomendations': ['173', '207', '189', '741',
'209']},
{'userId': '145', 'sp_recomendations': ['50', '172', '174', '181',
'96']},
{'userId': '28', 'sp_recomendations': ['96', '480', '12', '195', '9
8']},
{'userId': '135', 'sp_recomendations': ['98', '56', '603', '12', '1
85']},
{'userId': '32', 'sp_recomendations': ['50', '268', '408', '313',
'475']},
{'userId': '90', 'sp_recomendations': ['483', '515', '607', '505',
'479']},
{'userId': '216', 'sp_recomendations': ['64', '318', '100', '98',
'357']},
{'userId': '250', 'sp_recomendations': ['474', '98', '480', '135',
'50']},
{'userId': '271', 'sp_recomendations': ['98', '318', '357', '427',
'132']},
{'userId': '265', 'sp_recomendations': ['117', '742', '298', '300',
'1']},
{'userId': '198', 'sp_recomendations': ['50', '474', '127', '172',
'173']},
{'userId': '168', 'sp_recomendations': ['181', '282', '222', '258',
'313']},
{'userId': '110', 'sp_recomendations': ['64', '22', '272', '79', '6
9']},
{'userId': '58', 'sp_recomendations': ['173', '272', '156', '56',
'12']},
{'userId': '237', 'sp_recomendations': ['169', '483', '408', '474',
'64']},
{'userId': '94', 'sp_recomendations': ['180', '654', '357', '474',

```

```

'127']},
{'userId': '128', 'sp_recomendations': ['83', '143', '64', '496',
'483']},
{'userId': '44', 'sp_recomendations': ['228', '89', '64', '174', '1
73']},
{'userId': '264', 'sp_recomendations': ['12', '603', '175', '173',
'525']},
{'userId': '41', 'sp_recomendations': ['180', '168', '474', '50',
'318']},
{'userId': '82', 'sp_recomendations': ['50', '64', '483', '318', '5
23']},
{'userId': '262', 'sp_recomendations': ['98', '169', '318', '64',
'923']},
{'userId': '174', 'sp_recomendations': ['196', '272', '12', '87',
'98']},
{'userId': '43', 'sp_recomendations': ['313', '318', '169', '64',
'408']},
{'userId': '84', 'sp_recomendations': ['318', '64', '12', '408', '8
7']},
{'userId': '269', 'sp_recomendations': ['475', '285', '100', '175',
'124']},
{'userId': '259', 'sp_recomendations': ['12', '168', '357', '313',
'317']},
{'userId': '85', 'sp_recomendations': ['127', '511', '64', '705',
'483']},
{'userId': '213', 'sp_recomendations': ['318', '64', '172', '357',
'483']},
{'userId': '121', 'sp_recomendations': ['50', '12', '172', '137',
'181']},
{'userId': '49', 'sp_recomendations': ['56', '179', '175', '100',
'475']},
{'userId': '155', 'sp_recomendations': ['306', '286', '292', '872',
'319']},
{'userId': '68', 'sp_recomendations': ['178', '50', '276', '127',
'275']},
{'userId': '172', 'sp_recomendations': ['183', '483', '603', '657',
'485']},
{'userId': '19', 'sp_recomendations': ['435', '8', '211', '153', '8
87']},
{'userId': '268', 'sp_recomendations': ['169', '114', '179', '474',
'56']},
{'userId': '5', 'sp_recomendations': ['169', '408', '430', '50', '2
28']},
{'userId': '80', 'sp_recomendations': ['64', '483', '100', '205',
'79']},
{'userId': '66', 'sp_recomendations': ['50', '181', '300', '127',
'742']},
{'userId': '18', 'sp_recomendations': ['56', '9', '408', '515', '28
6']},
{'userId': '26', 'sp_recomendations': ['127', '302', '269', '313',
'100']},
{'userId': '130', 'sp_recomendations': ['234', '357', '183', '22',
'333']},
{'userId': '256', 'sp_recomendations': ['96', '98', '64', '1', '26
5']},
{'userId': '1', 'sp_recomendations': ['177', '100', '50', '127', '1
72']},
{'userId': '56', 'sp_recomendations': ['22', '172', '195', '181',
'64']},
{'userId': '15', 'sp_recomendations': ['269', '50', '936', '181',
'306']},

```

```

{'userId': '207', 'sp_recomendations': ['64', '515', '318', '316',
'483']},
{'userId': '232', 'sp_recomendations': ['474', '483', '48', '475',
'175']},
{'userId': '52', 'sp_recomendations': ['318', '427', '657', '191',
'531']},
{'userId': '161', 'sp_recomendations': ['272', '316', '318', '315',
'100']},
{'userId': '148', 'sp_recomendations': ['50', '408', '174', '181',
'114']},
{'userId': '125', 'sp_recomendations': ['174', '50', '195', '210',
'64']},
{'userId': '83', 'sp_recomendations': ['174', '204', '210', '64',
'79']},
{'userId': '272', 'sp_recomendations': ['474', '134', '178', '187',
'357']},
{'userId': '151', 'sp_recomendations': ['483', '657', '170', '408',
'529']},
{'userId': '54', 'sp_recomendations': ['276', '127', '100', '272',
'313']},
{'userId': '16', 'sp_recomendations': ['64', '705', '191', '22', '6
03']},
{'userId': '91', 'sp_recomendations': ['174', '50', '181', '318',
'22']},
{'userId': '294', 'sp_recomendations': ['50', '127', '181', '603',
'483']},
{'userId': '229', 'sp_recomendations': ['302', '344', '286', '311',
'272']},
{'userId': '36', 'sp_recomendations': ['882', '288', '1026', '333',
'883']},
{'userId': '70', 'sp_recomendations': ['483', '496', '174', '50',
'132']},
{'userId': '14', 'sp_recomendations': ['408', '175', '12', '302',
'430']},
{'userId': '295', 'sp_recomendations': ['483', '318', '98', '966',
'493']},
{'userId': '233', 'sp_recomendations': ['98', '313', '318', '64',
'483']},
{'userId': '214', 'sp_recomendations': ['185', '285', '56', '179',
'114']},
{'userId': '192', 'sp_recomendations': ['302', '258', '50', '100',
'127']},
{'userId': '100', 'sp_recomendations': ['313', '272', '315', '316',
'300']},
{'userId': '307', 'sp_recomendations': ['169', '408', '114', '483',
'258']},
{'userId': '297', 'sp_recomendations': ['286', '56', '12', '238',
'98']},
{'userId': '193', 'sp_recomendations': ['313', '286', '199', '485',
'174']},
{'userId': '113', 'sp_recomendations': ['50', '258', '126', '255',
'246']},
{'userId': '275', 'sp_recomendations': ['50', '174', '181', '1', '5
15']},
{'userId': '219', 'sp_recomendations': ['114', '179', '269', '433',
'223']},
{'userId': '218', 'sp_recomendations': ['657', '183', '12', '100',
'98']},
{'userId': '123', 'sp_recomendations': ['9', '511', '134', '127',
'285']},
{'userId': '158', 'sp_recomendations': ['285', '190', '127', '483',

```

```

'173' ]}],
{'userId': '302', 'sp_recomendations': ['271', '258', '301', '303',
'307' ]}],
{'userId': '23', 'sp_recomendations': ['408', '175', '173', '479',
'124' ]}],
{'userId': '296', 'sp_recomendations': ['56', '134', '357', '127',
'50' ]}],
{'userId': '33', 'sp_recomendations': ['313', '258', '292', '300',
'271' ]}],
{'userId': '154', 'sp_recomendations': ['474', '480', '641', '175',
'89' ]}],
{'userId': '77', 'sp_recomendations': ['168', '474', '483', '89',
'641' ]}],
{'userId': '270', 'sp_recomendations': ['98', '173', '275', '50',
'200' ]}],
{'userId': '187', 'sp_recomendations': ['64', '83', '173', '659',
'191' ]}],
{'userId': '170', 'sp_recomendations': ['300', '292', '304', '333',
'258' ]}],
{'userId': '101', 'sp_recomendations': ['50', '237', '181', '117',
'742' ]}],
{'userId': '184', 'sp_recomendations': ['488', '134', '285', '483',
'9' ]}],
{'userId': '112', 'sp_recomendations': ['272', '315', '313', '316',
'302' ]}],
{'userId': '133', 'sp_recomendations': ['272', '269', '316', '258',
'313' ]}],
{'userId': '215', 'sp_recomendations': ['98', '50', '480', '483',
'183' ]}],
{'userId': '69', 'sp_recomendations': ['50', '98', '174', '12', '10
0' ]}],
{'userId': '104', 'sp_recomendations': ['50', '181', '750', '150',
'272' ]}],
{'userId': '240', 'sp_recomendations': ['272', '313', '302', '242',
'286' ]}],
{'userId': '144', 'sp_recomendations': ['174', '313', '64', '50',
'318' ]}],
{'userId': '191', 'sp_recomendations': ['313', '316', '315', '272',
'86' ]}],
{'userId': '61', 'sp_recomendations': ['347', '258', '300', '333',
'310' ]}],
{'userId': '142', 'sp_recomendations': ['169', '514', '408', '268',
'134' ]}],
{'userId': '177', 'sp_recomendations': ['127', '56', '12', '179',
'50' ]}],
{'userId': '203', 'sp_recomendations': ['50', '283', '181', '150',
'258' ]}],
{'userId': '21', 'sp_recomendations': ['56', '9', '127', '408',
'7' ]}],
{'userId': '197', 'sp_recomendations': ['174', '22', '172', '50',
'181' ]}],
{'userId': '134', 'sp_recomendations': ['313', '15', '316', '1', '2
58' ]}],
{'userId': '180', 'sp_recomendations': ['98', '694', '56', '196',
'173' ]}],
{'userId': '236', 'sp_recomendations': ['286', '318', '64', '483',
'127' ]}],
{'userId': '263', 'sp_recomendations': ['318', '483', '50', '132',
'205' ]}],
{'userId': '109', 'sp_recomendations': ['174', '172', '96', '82',
'50' ]}],

```

```

{'userId': '64', 'sp_recomendations': ['50', '173', '127', '172',
'168']},
{'userId': '114', 'sp_recomendations': ['474', '183', '89', '172',
'179']},
{'userId': '239', 'sp_recomendations': ['513', '654', '134', '175',
'474']},
{'userId': '117', 'sp_recomendations': ['50', '168', '56', '172',
'268']},
{'userId': '65', 'sp_recomendations': ['318', '50', '64', '178', '5
11']},
{'userId': '137', 'sp_recomendations': ['144', '50', '174', '96',
'79']},
{'userId': '257', 'sp_recomendations': ['285', '50', '14', '100',
'531']},
{'userId': '111', 'sp_recomendations': ['269', '302', '315', '313',
'272']},
{'userId': '285', 'sp_recomendations': ['313', '198', '357', '205',
'302']},
{'userId': '96', 'sp_recomendations': ['185', '64', '318', '127',
'50']},
{'userId': '116', 'sp_recomendations': ['56', '127', '187', '180',
'603']},
{'userId': '73', 'sp_recomendations': ['135', '12', '474', '127',
'89']},
{'userId': '221', 'sp_recomendations': ['12', '56', '50', '172', '2
72']},
{'userId': '235', 'sp_recomendations': ['483', '474', '318', '511',
'100']},
{'userId': '164', 'sp_recomendations': ['313', '181', '222', '258',
'282']},
{'userId': '281', 'sp_recomendations': ['288', '304', '748', '300',
'271']},
{'userId': '182', 'sp_recomendations': ['50', '178', '172', '423',
'191']},
{'userId': '129', 'sp_recomendations': ['302', '269', '286', '272',
'242']},
{'userId': '45', 'sp_recomendations': ['50', '1', '127', '100', '18
1']},
{'userId': '131', 'sp_recomendations': ['251', '269', '100', '124',
'14']},
{'userId': '230', 'sp_recomendations': ['22', '496', '98', '64', '7
1']},
{'userId': '126', 'sp_recomendations': ['302', '884', '313', '316',
'315']},
{'userId': '231', 'sp_recomendations': ['50', '15', '181', '126',
'471']},
{'userId': '280', 'sp_recomendations': ['316', '483', '127', '486',
'9']},
{'userId': '288', 'sp_recomendations': ['357', '127', '272', '318',
'64']},
{'userId': '152', 'sp_recomendations': ['272', '204', '22', '692',
'143']},
{'userId': '217', 'sp_recomendations': ['222', '22', '56', '176',
'96']},
{'userId': '79', 'sp_recomendations': ['285', '515', '10', '258',
'813']},
{'userId': '75', 'sp_recomendations': ['496', '56', '151', '190',
'427']},
{'userId': '245', 'sp_recomendations': ['258', '300', '50', '1028',
'181']},
{'userId': '282', 'sp_recomendations': ['302', '269', '327', '262',

```



```

'258']},
{'userId': '78', 'sp_recomendations': ['237', '269', '298', '257',
'1160']},
{'userId': '118', 'sp_recomendations': ['654', '174', '185', '511',
'23']},
{'userId': '283', 'sp_recomendations': ['50', '56', '173', '709',
'208']},
{'userId': '171', 'sp_recomendations': ['272', '302', '315', '313',
'303']},
{'userId': '107', 'sp_recomendations': ['302', '340', '269', '902',
'268']},
{'userId': '226', 'sp_recomendations': ['408', '12', '169', '98',
'174']},
{'userId': '306', 'sp_recomendations': ['242', '14', '285', '269',
'306']},
{'userId': '173', 'sp_recomendations': ['302', '242', '306', '269',
'327']},
{'userId': '185', 'sp_recomendations': ['127', '318', '480', '114',
'197']},
{'userId': '150', 'sp_recomendations': ['50', '181', '246', '475',
'127']},
{'userId': '274', 'sp_recomendations': ['318', '100', '98', '50',
'208']},
{'userId': '188', 'sp_recomendations': ['64', '318', '483', '174',
'98']},
{'userId': '48', 'sp_recomendations': ['483', '174', '183', '98',
'511']},
{'userId': '311', 'sp_recomendations': ['174', '50', '172', '318',
'22']},
{'userId': '165', 'sp_recomendations': ['318', '169', '174', '181',
'432']},
{'userId': '208', 'sp_recomendations': ['88', '197', '194', '435',
'216']},
{'userId': '2', 'sp_recomendations': ['127', '100', '275', '251',
'302']},
{'userId': '205', 'sp_recomendations': ['242', '313', '333', '316',
'315']},
{'userId': '248', 'sp_recomendations': ['515', '114', '198', '100',
'50']},
{'userId': '93', 'sp_recomendations': ['1', '15', '275', '14', '28
3']},
{'userId': '159', 'sp_recomendations': ['333', '273', '15', '272',
'258']},
{'userId': '146', 'sp_recomendations': ['272', '315', '313', '302',
'286']},
{'userId': '29', 'sp_recomendations': ['12', '98', '302', '480', '6
57']},
{'userId': '156', 'sp_recomendations': ['197', '187', '480', '178',
'180']},
{'userId': '37', 'sp_recomendations': ['56', '174', '50', '172', '8
9']},
{'userId': '141', 'sp_recomendations': ['282', '125', '274', '126',
'258']},
{'userId': '195', 'sp_recomendations': ['234', '134', '181', '127',
'100']},
{'userId': '108', 'sp_recomendations': ['275', '127', '137', '515',
'124']},
{'userId': '47', 'sp_recomendations': ['302', '340', '258', '306',
'269']},
{'userId': '255', 'sp_recomendations': ['98', '185', '56', '219',
'249']},

```

```

{'userId': '89', 'sp_recomendations': ['127', '269', '173', '187',
'50']},
{'userId': '140', 'sp_recomendations': ['286', '302', '268', '303',
'258']},
{'userId': '190', 'sp_recomendations': ['313', '272', '302', '237',
'117']},
{'userId': '24', 'sp_recomendations': ['64', '178', '127', '357',
'318']},
{'userId': '17', 'sp_recomendations': ['100', '919', '1', '150', '1
37']},
{'userId': '313', 'sp_recomendations': ['50', '603', '479', '134',
'488']},
{'userId': '53', 'sp_recomendations': ['64', '174', '199', '15', '1
00']},
{'userId': '124', 'sp_recomendations': ['166', '11', '98', '168',
'154']},
{'userId': '149', 'sp_recomendations': ['313', '286', '269', '272',
'302']},
{'userId': '176', 'sp_recomendations': ['475', '340', '285', '272',
'50']},
{'userId': '106', 'sp_recomendations': ['318', '191', '313', '923',
'12']},
{'userId': '312', 'sp_recomendations': ['641', '480', '134', '483',
'488']},
{'userId': '175', 'sp_recomendations': ['64', '50', '98', '172', '4
83']},
{'userId': '153', 'sp_recomendations': ['64', '357', '56', '182',
'127']},
{'userId': '220', 'sp_recomendations': ['286', '269', '306', '288',
'305']},
{'userId': '143', 'sp_recomendations': ['313', '272', '333', '288',
'315']},
{'userId': '199', 'sp_recomendations': ['408', '116', '14', '285',
'268']},
{'userId': '202', 'sp_recomendations': ['484', '604', '516', '96',
'269']},
{'userId': '277', 'sp_recomendations': ['286', '285', '302', '100',
'258']},
{'userId': '206', 'sp_recomendations': ['288', '272', '315', '313',
'333']},
{'userId': '76', 'sp_recomendations': ['56', '474', '98', '135', '1
37']},
{'userId': '314', 'sp_recomendations': ['64', '121', '692', '282',
'143']},
{'userId': '136', 'sp_recomendations': ['318', '515', '127', '56',
'14']},
{'userId': '179', 'sp_recomendations': ['288', '272', '333', '316',
'313']},
{'userId': '4', 'sp_recomendations': ['50', '357', '258', '210', '3
01']},
{'userId': '304', 'sp_recomendations': ['313', '300', '275', '237',
'298']},
{'userId': '3', 'sp_recomendations': ['320', '318', '340', '346',
'347']},
{'userId': '227', 'sp_recomendations': ['100', '19', '124', '137',
'127']},
{'userId': '252', 'sp_recomendations': ['9', '100', '124', '7', '26
8']},
{'userId': '212', 'sp_recomendations': ['318', '427', '199', '317',
'527']},
{'userId': '310', 'sp_recomendations': ['251', '14', '1142', '116',

```

```

'50']},
{'userId': '35', 'sp_recomendations': ['300', '333', '332', '937',
'242']},
{'userId': '147', 'sp_recomendations': ['313', '302', '750', '340',
'305']},
{'userId': '105', 'sp_recomendations': ['302', '258', '272', '313',
'269']},
{'userId': '34', 'sp_recomendations': ['242', '292', '898', '332',
'1024']},
{'userId': '71', 'sp_recomendations': ['197', '357', '923', '64',
'475']},
{'userId': '51', 'sp_recomendations': ['50', '172', '181', '64', '1
73']},
{'userId': '204', 'sp_recomendations': ['318', '302', '170', '242',
'12']},
{'userId': '315', 'sp_recomendations': ['56', '285', '127', '100',
'238']},
{'userId': '31', 'sp_recomendations': ['136', '705', '484', '498',
'493']},
{'userId': '316', 'sp_recomendations': ['427', '213', '190', '98',
'923']},
{'userId': '103', 'sp_recomendations': ['50', '127', '181', '56',
'487']},
{'userId': '318', 'sp_recomendations': ['318', '174', '143', '735',
'64']},
{'userId': '30', 'sp_recomendations': ['174', '313', '1007', '242',
'286']},
{'userId': '120', 'sp_recomendations': ['515', '127', '9', '50', '2
86']},
{'userId': '46', 'sp_recomendations': ['127', '313', '100', '1062',
'305']},
{'userId': '289', 'sp_recomendations': ['117', '1016', '15', '24',
'742']},
{'userId': '209', 'sp_recomendations': ['50', '127', '285', '251',
'408']},
{'userId': '261', 'sp_recomendations': ['340', '300', '359', '117',
'410']},
{'userId': '88', 'sp_recomendations': ['286', '315', '904', '302',
'354']},
{'userId': '9', 'sp_recomendations': ['50', '483', '286', '479', '6
15']},
{'userId': '247', 'sp_recomendations': ['64', '50', '251', '258',
'70']},
{'userId': '321', 'sp_recomendations': ['483', '515', '604', '603',
'133']},
{'userId': '266', 'sp_recomendations': ['100', '272', '285', '313',
'124']},
{'userId': '74', 'sp_recomendations': ['313', '272', '315', '302',
'15']},
{'userId': '238', 'sp_recomendations': ['286', '298', '300', '471',
'538']},
{'userId': '319', 'sp_recomendations': ['313', '302', '269', '306',
'750']},
{'userId': '323', 'sp_recomendations': ['127', '50', '172', '64',
'180']},
{'userId': '67', 'sp_recomendations': ['64', '7', '117', '125', '40
5']},
{'userId': '211', 'sp_recomendations': ['205', '199', '423', '127',
'528']},
{'userId': '98', 'sp_recomendations': ['659', '517', '428', '194',
'514']},

```

```

{'userId': '12', 'sp_recomendations': ['318', '98', '97', '174', '2
16']},
{'userId': '40', 'sp_recomendations': ['302', '896', '316', '750',
'305']},
{'userId': '258', 'sp_recomendations': ['313', '272', '286', '310',
'328']},
{'userId': '228', 'sp_recomendations': ['272', '313', '427', '98',
'286']},
{'userId': '325', 'sp_recomendations': ['474', '408', '199', '647',
'179']},
{'userId': '320', 'sp_recomendations': ['174', '173', '56', '96',
'172']},
{'userId': '326', 'sp_recomendations': ['496', '526', '178', '50',
'434']},
{'userId': '327', 'sp_recomendations': ['127', '603', '132', '192',
'318']},
{'userId': '183', 'sp_recomendations': ['483', '177', '485', '216',
'203']},
{'userId': '328', 'sp_recomendations': ['318', '272', '520', '661',
'22']},
{'userId': '322', 'sp_recomendations': ['603', '185', '127', '479',
'197']},
{'userId': '330', 'sp_recomendations': ['8', '603', '485', '208',
'427']},
{'userId': '27', 'sp_recomendations': ['100', '515', '50', '121',
'9']},
{'userId': '331', 'sp_recomendations': ['64', '302', '958', '180',
'182']},
{'userId': '332', 'sp_recomendations': ['22', '174', '182', '333',
'210']},
{'userId': '329', 'sp_recomendations': ['272', '169', '302', '100',
'313']},
{'userId': '86', 'sp_recomendations': ['258', '269', '242', '1176',
'889']},
{'userId': '139', 'sp_recomendations': ['127', '475', '100', '297',
'302']},
{'userId': '300', 'sp_recomendations': ['257', '300', '288', '101
2', '881']},
{'userId': '163', 'sp_recomendations': ['316', '318', '272', '64',
'357']},
{'userId': '333', 'sp_recomendations': ['174', '483', '98', '315',
'316']},
{'userId': '334', 'sp_recomendations': ['134', '408', '100', '124',
'12']},
{'userId': '39', 'sp_recomendations': ['302', '288', '313', '258',
'315']},
{'userId': '324', 'sp_recomendations': ['50', '286', '14', '9', '28
5']},
{'userId': '132', 'sp_recomendations': ['12', '484', '251', '922',
'521']},
{'userId': '336', 'sp_recomendations': ['216', '173', '50', '186',
'56']},
{'userId': '335', 'sp_recomendations': ['340', '347', '313', '269',
'902']},
{'userId': '169', 'sp_recomendations': ['603', '50', '134', '174',
'127']},
{'userId': '338', 'sp_recomendations': ['479', '169', '408', '603',
'474']},
{'userId': '339', 'sp_recomendations': ['23', '192', '179', '127',
'134']},
{'userId': '309', 'sp_recomendations': ['242', '258', '331', '286',

```

```

'326' ]},
{'userId': '342', 'sp_recomendations': ['56', '179', '209', '23',
'488' ]},
{'userId': '340', 'sp_recomendations': ['174', '1', '173', '50', '4
80' ]},
{'userId': '317', 'sp_recomendations': ['313', '300', '328', '268',
'748' ]},
{'userId': '341', 'sp_recomendations': ['887', '880', '292', '330',
'872' ]},
{'userId': '343', 'sp_recomendations': ['483', '427', '357', '50',
'318' ]},
{'userId': '344', 'sp_recomendations': ['169', '124', '12', '127',
'191' ]},
{'userId': '345', 'sp_recomendations': ['12', '64', '56', '191', '3
18' ]},
{'userId': '346', 'sp_recomendations': ['50', '318', '22', '174',
'195' ]},
{'userId': '347', 'sp_recomendations': ['182', '172', '50', '156',
'22' ]},
{'userId': '273', 'sp_recomendations': ['315', '272', '316', '896',
'268' ]},
{'userId': '55', 'sp_recomendations': ['89', '22', '50', '174', '14
4' ]},
{'userId': '349', 'sp_recomendations': ['10', '285', '100', '9', '2
76' ]},
{'userId': '348', 'sp_recomendations': ['313', '121', '7', '126',
'111' ]},
{'userId': '354', 'sp_recomendations': ['513', '285', '57', '603',
'59' ]},
{'userId': '351', 'sp_recomendations': ['313', '258', '311', '102
4', '300' ]},
{'userId': '358', 'sp_recomendations': ['318', '114', '512', '152
4', '357' ]},
{'userId': '352', 'sp_recomendations': ['172', '302', '98', '183',
'176' ]},
{'userId': '360', 'sp_recomendations': ['127', '134', '205', '483',
'654' ]},
{'userId': '363', 'sp_recomendations': ['50', '172', '89', '408',
'137' ]},
{'userId': '355', 'sp_recomendations': ['242', '306', '1175', '31
9', '286' ]},
{'userId': '362', 'sp_recomendations': ['302', '333', '300', '347',
'313' ]},
{'userId': '357', 'sp_recomendations': ['258', '237', '1', '121',
'117' ]},
{'userId': '356', 'sp_recomendations': ['313', '272', '258', '316',
'300' ]},
{'userId': '361', 'sp_recomendations': ['98', '197', '603', '178',
'100' ]},
{'userId': '365', 'sp_recomendations': ['272', '269', '1137', '31
6', '275' ]},
{'userId': '350', 'sp_recomendations': ['127', '480', '187', '603',
'657' ]},
{'userId': '367', 'sp_recomendations': ['50', '98', '100', '302',
'7' ]},
{'userId': '368', 'sp_recomendations': ['98', '313', '320', '100',
'183' ]},
{'userId': '371', 'sp_recomendations': ['64', '97', '186', '22', '9
8' ]},
{'userId': '373', 'sp_recomendations': ['64', '169', '318', '114',
'83' ]},

```

```
{'userId': '370', 'sp_recomendations': ['302', '661', '603', '57',  
'480']},  
{'userId': '374', 'sp_recomendations': ['161', '282', '117', '568',  
'69']},  
{'userId': '372', 'sp_recomendations': ['12', '79', '98', '333', '1  
85']},  
{'userId': '337', 'sp_recomendations': ['50', '520', '392', '222',  
'135']},  
{'userId': '378', 'sp_recomendations': ['318', '191', '9', '64', '3  
02']},  
{'userId': '366', 'sp_recomendations': ['98', '853', '443', '200',  
'448']},  
{'userId': '377', 'sp_recomendations': ['272', '173', '313', '98',  
'168']},  
{'userId': '375', 'sp_recomendations': ['302', '603', '234', '183',  
'185']},  
{'userId': '359', 'sp_recomendations': ['50', '408', '313', '181',  
'7']},  
{'userId': '379', 'sp_recomendations': ['191', '64', '56', '197',  
'12']},  
{'userId': '380', 'sp_recomendations': ['483', '12', '474', '50',  
'357']},  
{'userId': '381', 'sp_recomendations': ['483', '134', '479', '191',  
'480']},  
{'userId': '385', 'sp_recomendations': ['134', '474', '654', '56',  
'169']},  
{'userId': '382', 'sp_recomendations': ['180', '481', '59', '23',  
'197']},  
{'userId': '387', 'sp_recomendations': ['7', '179', '89', '56', '16  
9']},  
{'userId': '364', 'sp_recomendations': ['302', '269', '286', '690',  
'294']},  
{'userId': '369', 'sp_recomendations': ['172', '50', '114', '268',  
'316']},  
{'userId': '388', 'sp_recomendations': ['313', '333', '300', '117',  
'121']},  
{'userId': '386', 'sp_recomendations': ['515', '50', '127', '117',  
'222']},  
{'userId': '389', 'sp_recomendations': ['127', '483', '134', '50',  
'187']},  
{'userId': '383', 'sp_recomendations': ['513', '182', '427', '474',  
'480']},  
{'userId': '390', 'sp_recomendations': ['100', '515', '1', '258',  
'302']},  
{'userId': '393', 'sp_recomendations': ['316', '121', '136', '613',  
'143']},  
{'userId': '392', 'sp_recomendations': ['50', '98', '513', '134',  
'199']},  
{'userId': '376', 'sp_recomendations': ['98', '357', '603', '198',  
'707']},  
{'userId': '394', 'sp_recomendations': ['89', '174', '98', '7', '19  
5']},  
{'userId': '391', 'sp_recomendations': ['127', '511', '474', '64',  
'197']},  
{'userId': '398', 'sp_recomendations': ['496', '64', '205', '50',  
'520']},  
{'userId': '397', 'sp_recomendations': ['100', '135', '127', '50',  
'134']},  
{'userId': '399', 'sp_recomendations': ['318', '143', '196', '15',  
'98']},  
{'userId': '396', 'sp_recomendations': ['121', '222', '237', '282',
```

```

'333' ]}],
{'userId': '401', 'sp_recomendations': ['216', '735', '315', '196',
'316' ]}],
{'userId': '402', 'sp_recomendations': ['127', '408', '511', '50',
'286' ]}],
{'userId': '384', 'sp_recomendations': ['272', '313', '302', '316',
'300' ]}],
{'userId': '395', 'sp_recomendations': ['174', '50', '172', '318',
'64' ]}],
{'userId': '353', 'sp_recomendations': ['272', '313', '316', '315',
'286' ]}],
{'userId': '403', 'sp_recomendations': ['121', '222', '50', '515',
'118' ]}],
{'userId': '405', 'sp_recomendations': ['174', '204', '210', '50',
'22' ]}],
{'userId': '400', 'sp_recomendations': ['313', '269', '306', '258',
'300' ]}],
{'userId': '406', 'sp_recomendations': ['50', '134', '647', '48',
'194' ]}],
{'userId': '407', 'sp_recomendations': ['173', '169', '134', '50',
'127' ]}],
{'userId': '409', 'sp_recomendations': ['89', '98', '60', '483', '1
80' ]}],
{'userId': '404', 'sp_recomendations': ['313', '22', '272', '310',
'690' ]}],
{'userId': '413', 'sp_recomendations': ['515', '14', '50', '276',
'297' ]}],
{'userId': '416', 'sp_recomendations': ['272', '64', '174', '22',
'357' ]}],
{'userId': '408', 'sp_recomendations': ['302', '315', '242', '270',
'313' ]}],
{'userId': '410', 'sp_recomendations': ['313', '272', '315', '269',
'316' ]}],
{'userId': '411', 'sp_recomendations': ['50', '172', '603', '181',
'318' ]}],
{'userId': '417', 'sp_recomendations': ['190', '286', '173', '83',
'191' ]}],
{'userId': '412', 'sp_recomendations': ['173', '318', '114', '408',
'174' ]}],
{'userId': '420', 'sp_recomendations': ['408', '285', '190', '14',
'251' ]}],
{'userId': '422', 'sp_recomendations': ['98', '50', '137', '127',
'475' ]}],
{'userId': '425', 'sp_recomendations': ['12', '56', '172', '50', '8
9' ]}],
{'userId': '419', 'sp_recomendations': ['174', '50', '705', '173',
'134' ]}],
{'userId': '415', 'sp_recomendations': ['174', '480', '483', '56',
'136' ]}],
{'userId': '423', 'sp_recomendations': ['127', '100', '272', '9',
'313' ]}],
{'userId': '429', 'sp_recomendations': ['42', '357', '318', '50',
'100' ]}],
{'userId': '428', 'sp_recomendations': ['272', '313', '315', '302',
'316' ]}],
{'userId': '427', 'sp_recomendations': ['302', '359', '304', '332',
'303' ]}],
{'userId': '418', 'sp_recomendations': ['288', '333', '258', '269',
'899' ]}],
{'userId': '424', 'sp_recomendations': ['275', '1084', '14', '15',
'9' ]}],

```

```
{'userId': '432', 'sp_recomendations': ['313', '50', '181', '300',  
'237']},  
{'userId': '421', 'sp_recomendations': ['174', '172', '50', '183',  
'96']},  
{'userId': '435', 'sp_recomendations': ['169', '12', '173', '176',  
'195']},  
{'userId': '433', 'sp_recomendations': ['12', '50', '302', '657',  
'269']},  
{'userId': '426', 'sp_recomendations': ['603', '483', '427', '488',  
'318']},  
{'userId': '436', 'sp_recomendations': ['96', '127', '628', '746',  
'313']},  
{'userId': '430', 'sp_recomendations': ['318', '64', '98', '100',  
'12']},  
{'userId': '434', 'sp_recomendations': ['237', '288', '546', '111',  
'125']},  
{'userId': '437', 'sp_recomendations': ['582', '197', '511', '654',  
'479']},  
{'userId': '438', 'sp_recomendations': ['50', '121', '237', '181',  
'257']},  
{'userId': '431', 'sp_recomendations': ['302', '303', '300', '269',  
'538']},  
{'userId': '442', 'sp_recomendations': ['64', '12', '172', '228',  
'318']},  
{'userId': '440', 'sp_recomendations': ['272', '57', '921', '242',  
'923']},  
{'userId': '445', 'sp_recomendations': ['56', '408', '340', '475',  
'79']},  
{'userId': '447', 'sp_recomendations': ['50', '174', '12', '181',  
'83']},  
{'userId': '449', 'sp_recomendations': ['276', '100', '515', '127',  
'475']},  
{'userId': '450', 'sp_recomendations': ['216', '496', '318', '22',  
'64']},  
{'userId': '446', 'sp_recomendations': ['302', '292', '270', '269',  
'245']},  
{'userId': '439', 'sp_recomendations': ['242', '285', '276', '14',  
'237']},  
{'userId': '451', 'sp_recomendations': ['333', '294', '328', '332',  
'300']},  
{'userId': '452', 'sp_recomendations': ['318', '22', '480', '69',  
'89']},  
{'userId': '454', 'sp_recomendations': ['318', '69', '22', '64', '3  
13']},  
{'userId': '453', 'sp_recomendations': ['50', '172', '56', '12', '3  
57']},  
{'userId': '414', 'sp_recomendations': ['302', '272', '258', '313',  
'100']},  
{'userId': '455', 'sp_recomendations': ['127', '50', '197', '191',  
'318']},  
{'userId': '444', 'sp_recomendations': ['50', '272', '313', '306',  
'251']},  
{'userId': '448', 'sp_recomendations': ['896', '302', '1062', '26  
9', '750']},  
{'userId': '457', 'sp_recomendations': ['192', '12', '127', '114',  
'64']},  
{'userId': '456', 'sp_recomendations': ['474', '475', '56', '14',  
'483']},  
{'userId': '458', 'sp_recomendations': ['127', '9', '14', '169', '4  
08']},  
{'userId': '462', 'sp_recomendations': ['22', '313', '181', '272',
```



```
'655']},
{'userId': '459', 'sp_recomendations': ['22', '294', '250', '8', '1
74']},
{'userId': '460', 'sp_recomendations': ['100', '285', '127', '515',
'313']},
{'userId': '461', 'sp_recomendations': ['9', '50', '313', '285', '3
02']},
{'userId': '467', 'sp_recomendations': ['127', '124', '276', '7',
'100']},
{'userId': '468', 'sp_recomendations': ['318', '178', '64', '427',
'56']},
{'userId': '466', 'sp_recomendations': ['302', '22', '50', '181',
'174']},
{'userId': '472', 'sp_recomendations': ['258', '250', '181', '195',
'318']},
{'userId': '465', 'sp_recomendations': ['408', '169', '318', '513',
'114']},
{'userId': '463', 'sp_recomendations': ['127', '124', '50', '100',
'116']},
{'userId': '471', 'sp_recomendations': ['8', '82', '50', '465', '14
0']},
{'userId': '474', 'sp_recomendations': ['50', '127', '56', '100',
'357']},
{'userId': '469', 'sp_recomendations': ['134', '483', '474', '603',
'923']},
{'userId': '464', 'sp_recomendations': ['12', '127', '603', '302',
'520']},
{'userId': '476', 'sp_recomendations': ['168', '173', '194', '186',
'208']},
{'userId': '478', 'sp_recomendations': ['64', '318', '178', '357',
'12']},
{'userId': '473', 'sp_recomendations': ['285', '127', '9', '1143',
'1142']},
{'userId': '470', 'sp_recomendations': ['50', '285', '283', '150',
'9']},
{'userId': '480', 'sp_recomendations': ['169', '12', '603', '272',
'114']},
{'userId': '441', 'sp_recomendations': ['1', '313', '15', '100',
'9']},
{'userId': '479', 'sp_recomendations': ['174', '169', '181', '408',
'50']},
{'userId': '484', 'sp_recomendations': ['50', '174', '172', '181',
'79']},
{'userId': '486', 'sp_recomendations': ['127', '50', '100', '1142',
'124']},
{'userId': '487', 'sp_recomendations': ['22', '64', '69', '174', '7
9']},
{'userId': '482', 'sp_recomendations': ['313', '50', '127', '269',
'315']},
{'userId': '481', 'sp_recomendations': ['173', '479', '505', '659',
'191']},
{'userId': '492', 'sp_recomendations': ['127', '64', '56', '474',
'318']},
{'userId': '493', 'sp_recomendations': ['181', '50', '172', '79',
'176']},
{'userId': '490', 'sp_recomendations': ['127', '50', '515', '302',
'124']},
{'userId': '489', 'sp_recomendations': ['300', '315', '272', '316',
'313']},
{'userId': '483', 'sp_recomendations': ['50', '173', '318', '515',
'228']}
```

```

{'userId': '496', 'sp_recomendations': ['56', '89', '173', '42', '2
68']},
{'userId': '494', 'sp_recomendations': ['50', '64', '183', '174',
'357']},
{'userId': '495', 'sp_recomendations': ['50', '282', '357', '64',
'523']},
{'userId': '477', 'sp_recomendations': ['275', '709', '111', '274',
'15']},
{'userId': '497', 'sp_recomendations': ['50', '408', '114', '22',
'187']},
{'userId': '488', 'sp_recomendations': ['174', '64', '22', '318',
'496']},
{'userId': '498', 'sp_recomendations': ['474', '179', '175', '480',
'89']},
{'userId': '499', 'sp_recomendations': ['318', '98', '165', '511',
'427']},
{'userId': '491', 'sp_recomendations': ['408', '513', '657', '12',
'285']},
{'userId': '500', 'sp_recomendations': ['475', '483', '100', '179',
'89']},
{'userId': '502', 'sp_recomendations': ['271', '313', '333', '288',
'307']},
{'userId': '503', 'sp_recomendations': ['318', '963', '132', '190',
'100']},
{'userId': '504', 'sp_recomendations': ['318', '127', '735', '258',
'98']},
{'userId': '505', 'sp_recomendations': ['496', '313', '133', '98',
'132']},
{'userId': '506', 'sp_recomendations': ['79', '174', '50', '183',
'195']},
{'userId': '443', 'sp_recomendations': ['12', '313', '294', '258',
'286']},
{'userId': '507', 'sp_recomendations': ['302', '316', '750', '315',
'257']},
{'userId': '514', 'sp_recomendations': ['12', '174', '173', '22',
'114']},
{'userId': '508', 'sp_recomendations': ['474', '357', '318', '132',
'185']},
{'userId': '511', 'sp_recomendations': ['313', '292', '346', '288',
'340']},
{'userId': '515', 'sp_recomendations': ['300', '313', '302', '315',
'288']},
{'userId': '512', 'sp_recomendations': ['318', '50', '527', '56',
'198']},
{'userId': '513', 'sp_recomendations': ['50', '181', '210', '265',
'435']},
{'userId': '475', 'sp_recomendations': ['50', '316', '306', '902',
'100']},
{'userId': '523', 'sp_recomendations': ['114', '50', '181', '408',
'169']},
{'userId': '518', 'sp_recomendations': ['222', '125', '117', '126',
'100']},
{'userId': '509', 'sp_recomendations': ['50', '302', '288', '603',
'181']},
{'userId': '516', 'sp_recomendations': ['169', '50', '191', '474',
'515']},
{'userId': '510', 'sp_recomendations': ['313', '300', '258', '292',
'326']},
{'userId': '524', 'sp_recomendations': ['14', '127', '286', '302',
'511']},
{'userId': '501', 'sp_recomendations': ['127', '475', '922', '150',

```

```

'1007']},
{'userId': '525', 'sp_recomendations': ['100', '181', '276', '25',
'1']},
{'userId': '521', 'sp_recomendations': ['12', '127', '42', '268',
'168']},
{'userId': '520', 'sp_recomendations': ['100', '269', '286', '242',
'302']},
{'userId': '519', 'sp_recomendations': ['313', '909', '1062', '129
3', '268']},
{'userId': '528', 'sp_recomendations': ['50', '181', '174', '210',
'173']},
{'userId': '532', 'sp_recomendations': ['216', '96', '357', '318',
'191']},
{'userId': '530', 'sp_recomendations': ['64', '178', '607', '191',
'357']},
{'userId': '531', 'sp_recomendations': ['313', '286', '302', '690',
'311']},
{'userId': '529', 'sp_recomendations': ['268', '333', '258', '310',
'270']},
{'userId': '517', 'sp_recomendations': ['50', '127', '300', '275',
'1']},
{'userId': '527', 'sp_recomendations': ['134', '187', '127', '12',
'285']},
{'userId': '485', 'sp_recomendations': ['302', '242', '303', '313',
'269']},
{'userId': '533', 'sp_recomendations': ['313', '496', '64', '174',
'318']},
{'userId': '535', 'sp_recomendations': ['173', '209', '133', '963',
'496']},
{'userId': '536', 'sp_recomendations': ['174', '50', '172', '318',
'408']},
{'userId': '526', 'sp_recomendations': ['50', '408', '100', '285',
'313']},
{'userId': '537', 'sp_recomendations': ['511', '127', '134', '56',
'14']},
{'userId': '534', 'sp_recomendations': ['1', '282', '237', '117',
'471']},
{'userId': '541', 'sp_recomendations': ['50', '172', '174', '659',
'181']},
{'userId': '538', 'sp_recomendations': ['318', '199', '98', '496',
'528']},
{'userId': '542', 'sp_recomendations': ['127', '56', '432', '427',
'23']},
{'userId': '545', 'sp_recomendations': ['173', '181', '50', '172',
'174']},
{'userId': '539', 'sp_recomendations': ['483', '603', '132', '170',
'19']},
{'userId': '547', 'sp_recomendations': ['313', '316', '302', '315',
'269']},
{'userId': '543', 'sp_recomendations': ['12', '603', '169', '134',
'187']},
{'userId': '548', 'sp_recomendations': ['50', '79', '181', '98', '1
2']},
{'userId': '546', 'sp_recomendations': ['50', '185', '7', '56', '18
1']},
{'userId': '522', 'sp_recomendations': ['318', '480', '134', '180',
'168']},
{'userId': '551', 'sp_recomendations': ['272', '64', '187', '188',
'333']},
{'userId': '544', 'sp_recomendations': ['313', '286', '302', '750',
'292']},

```

```

{'userId': '553', 'sp_recomendations': ['603', '178', '98', '657',
'480']},
{'userId': '552', 'sp_recomendations': ['515', '127', '50', '126',
'100']},
{'userId': '540', 'sp_recomendations': ['515', '50', '100', '9', '2
86']},
{'userId': '554', 'sp_recomendations': ['174', '318', '172', '98',
'69']},
{'userId': '550', 'sp_recomendations': ['50', '313', '181', '310',
'258']},
{'userId': '556', 'sp_recomendations': ['64', '187', '127', '603',
'178']},
{'userId': '559', 'sp_recomendations': ['318', '513', '520', '519',
'205']},
{'userId': '560', 'sp_recomendations': ['134', '483', '127', '302',
'654']},
{'userId': '561', 'sp_recomendations': ['474', '179', '12', '56',
'175']},
{'userId': '563', 'sp_recomendations': ['50', '172', '181', '210',
'257']},
{'userId': '566', 'sp_recomendations': ['127', '100', '170', '192',
'165']},
{'userId': '557', 'sp_recomendations': ['12', '180', '165', '127',
'50']},
{'userId': '558', 'sp_recomendations': ['285', '19', '100', '253',
'137']},
{'userId': '564', 'sp_recomendations': ['50', '127', '313', '272',
'302']},
{'userId': '565', 'sp_recomendations': ['515', '190', '83', '179',
'86']},
{'userId': '573', 'sp_recomendations': ['127', '178', '480', '50',
'478']},
{'userId': '549', 'sp_recomendations': ['50', '515', '181', '127',
'258']},
{'userId': '567', 'sp_recomendations': ['474', '179', '641', '647',
'654']},
{'userId': '569', 'sp_recomendations': ['50', '100', '151', '286',
'9']},
{'userId': '562', 'sp_recomendations': ['185', '485', '143', '127',
'191']},
{'userId': '576', 'sp_recomendations': ['50', '181', '210', '7', '7
0']},
{'userId': '577', 'sp_recomendations': ['97', '71', '318', '22',
'1']},
{'userId': '579', 'sp_recomendations': ['50', '173', '194', '603',
'83']},
{'userId': '574', 'sp_recomendations': ['272', '302', '316', '242',
'258']},
{'userId': '555', 'sp_recomendations': ['169', '480', '285', '489',
'318']},
{'userId': '572', 'sp_recomendations': ['9', '124', '100', '14', '1
137']},
{'userId': '575', 'sp_recomendations': ['318', '357', '168', '603',
'96']},
{'userId': '584', 'sp_recomendations': ['50', '114', '313', '172',
'228']},
{'userId': '588', 'sp_recomendations': ['485', '8', '12', '316', '3
01']},
{'userId': '587', 'sp_recomendations': ['272', '313', '340', '271',
'269']},
{'userId': '568', 'sp_recomendations': ['603', '134', '474', '127',

```

```

'479' ]}],
{'userId': '586', 'sp_recomendations': ['559', '50', '69', '33', '1
81' ]}],
{'userId': '585', 'sp_recomendations': ['198', '275', '1449', '70
7', '170' ]}],
{'userId': '582', 'sp_recomendations': ['100', '50', '269', '313',
'124' ]}],
{'userId': '591', 'sp_recomendations': ['285', '100', '191', '357',
'64' ]}],
{'userId': '581', 'sp_recomendations': ['127', '100', '9', '285',
'813' ]}],
{'userId': '592', 'sp_recomendations': ['124', '318', '187', '56',
'238' ]}],
{'userId': '580', 'sp_recomendations': ['50', '250', '181', '257',
'258' ]}],
{'userId': '590', 'sp_recomendations': ['100', '285', '137', '150',
'14' ]}],
{'userId': '593', 'sp_recomendations': ['318', '196', '98', '216',
'173' ]}],
{'userId': '583', 'sp_recomendations': ['12', '483', '513', '357',
'519' ]}],
{'userId': '596', 'sp_recomendations': ['50', '313', '181', '328',
'258' ]}],
{'userId': '570', 'sp_recomendations': ['303', '302', '305', '327',
'286' ]}],
{'userId': '599', 'sp_recomendations': ['274', '282', '237', '1',
'255' ]}],
{'userId': '589', 'sp_recomendations': ['313', '272', '288', '300',
'310' ]}],
{'userId': '594', 'sp_recomendations': ['357', '50', '15', '515',
'127' ]}],
{'userId': '597', 'sp_recomendations': ['127', '50', '242', '283',
'293' ]}],
{'userId': '578', 'sp_recomendations': ['313', '272', '222', '300',
'298' ]}],
{'userId': '601', 'sp_recomendations': ['135', '483', '179', '168',
'357' ]}],
{'userId': '602', 'sp_recomendations': ['50', '181', '127', '117',
'1' ]}],
{'userId': '600', 'sp_recomendations': ['127', '187', '511', '56',
'183' ]}],
{'userId': '605', 'sp_recomendations': ['127', '64', '100', '98',
'191' ]}],
{'userId': '603', 'sp_recomendations': ['313', '50', '12', '172',
'181' ]}],
{'userId': '595', 'sp_recomendations': ['127', '50', '475', '100',
'1142' ]}],
{'userId': '606', 'sp_recomendations': ['174', '83', '50', '172',
'313' ]}],
{'userId': '608', 'sp_recomendations': ['197', '98', '603', '514',
'318' ]}],
{'userId': '607', 'sp_recomendations': ['511', '528', '483', '474',
'180' ]}],
{'userId': '610', 'sp_recomendations': ['480', '427', '318', '98',
'12' ]}],
{'userId': '611', 'sp_recomendations': ['302', '272', '315', '313',
'306' ]}],
{'userId': '617', 'sp_recomendations': ['185', '320', '183', '179',
'480' ]}],
{'userId': '618', 'sp_recomendations': ['174', '172', '50', '318',
'22' ]}],

```

```

{'userId': '614', 'sp_recomendations': ['100', '1', '126', '294',
'147']},
{'userId': '609', 'sp_recomendations': ['285', '408', '313', '750',
'304']},
{'userId': '615', 'sp_recomendations': ['127', '187', '357', '427',
'194']},
{'userId': '616', 'sp_recomendations': ['302', '272', '316', '313',
'286']},
{'userId': '620', 'sp_recomendations': ['313', '588', '15', '174',
'71']},
{'userId': '571', 'sp_recomendations': ['174', '114', '357', '484',
'657']},
{'userId': '619', 'sp_recomendations': ['79', '144', '313', '195',
'174']},
{'userId': '613', 'sp_recomendations': ['64', '50', '318', '272',
'603']},
{'userId': '622', 'sp_recomendations': ['173', '195', '64', '144',
'12']},
{'userId': '621', 'sp_recomendations': ['50', '172', '181', '100',
'79']},
{'userId': '604', 'sp_recomendations': ['48', '100', '127', '7', '2
34']},
{'userId': '624', 'sp_recomendations': ['285', '127', '50', '100',
'272']},
{'userId': '612', 'sp_recomendations': ['275', '480', '604', '1',
'1063']},
{'userId': '627', 'sp_recomendations': ['89', '179', '183', '199',
'64']},
{'userId': '623', 'sp_recomendations': ['50', '181', '483', '79',
'275']},
{'userId': '628', 'sp_recomendations': ['302', '242', '305', '270',
'333']},
{'userId': '625', 'sp_recomendations': ['50', '169', '483', '408',
'172']},
{'userId': '629', 'sp_recomendations': ['64', '98', '174', '22', '2
76']},
{'userId': '633', 'sp_recomendations': ['318', '79', '50', '96', '9
8']},
{'userId': '632', 'sp_recomendations': ['50', '172', '174', '181',
'173']},
{'userId': '631', 'sp_recomendations': ['272', '313', '315', '346',
'310']},
{'userId': '634', 'sp_recomendations': ['313', '272', '315', '302',
'286']},
{'userId': '639', 'sp_recomendations': ['487', '483', '178', '83',
'707']},
{'userId': '630', 'sp_recomendations': ['96', '216', '742', '117',
'118']},
{'userId': '642', 'sp_recomendations': ['748', '148', '845', '133',
'143']},
{'userId': '637', 'sp_recomendations': ['408', '181', '50', '936',
'690']},
{'userId': '640', 'sp_recomendations': ['357', '12', '64', '318',
'313']},
{'userId': '626', 'sp_recomendations': ['286', '272', '313', '923',
'302']},
{'userId': '643', 'sp_recomendations': ['127', '483', '185', '169',
'603']},
{'userId': '598', 'sp_recomendations': ['22', '750', '313', '258',
'286']},
{'userId': '638', 'sp_recomendations': ['174', '50', '181', '144',

```

```

'22']},
{'userId': '635', 'sp_recomendations': ['268', '246', '302', '269',
'262']},
{'userId': '644', 'sp_recomendations': ['300', '127', '258', '121',
'298']},
{'userId': '636', 'sp_recomendations': ['10', '272', '313', '258',
'813']},
{'userId': '645', 'sp_recomendations': ['483', '134', '357', '474',
'654']},
{'userId': '648', 'sp_recomendations': ['174', '173', '195', '82',
'181']},
{'userId': '647', 'sp_recomendations': ['173', '174', '197', '357',
'22']},
{'userId': '650', 'sp_recomendations': ['173', '483', '50', '474',
'603']},
{'userId': '651', 'sp_recomendations': ['269', '302', '515', '127',
'242']},
{'userId': '654', 'sp_recomendations': ['300', '22', '318', '313',
'79']},
{'userId': '653', 'sp_recomendations': ['195', '22', '174', '50',
'746']},
{'userId': '655', 'sp_recomendations': ['480', '320', '357', '483',
'427']},
{'userId': '649', 'sp_recomendations': ['127', '50', '1', '15', '29
8']},
{'userId': '658', 'sp_recomendations': ['408', '56', '169', '515',
'318']},
{'userId': '656', 'sp_recomendations': ['316', '896', '272', '302',
'303']},
{'userId': '660', 'sp_recomendations': ['173', '168', '50', '483',
'313']},
{'userId': '659', 'sp_recomendations': ['56', '191', '127', '603',
'197']},
{'userId': '646', 'sp_recomendations': ['313', '272', '315', '258',
'1176']},
{'userId': '663', 'sp_recomendations': ['127', '187', '50', '357',
'182']},
{'userId': '664', 'sp_recomendations': ['134', '408', '285', '100',
'127']},
{'userId': '657', 'sp_recomendations': ['9', '340', '269', '1', '11
1']},
{'userId': '665', 'sp_recomendations': ['483', '357', '127', '56',
'134']},
{'userId': '666', 'sp_recomendations': ['127', '474', '134', '483',
'89']},
{'userId': '661', 'sp_recomendations': ['318', '169', '174', '408',
'199']},
{'userId': '662', 'sp_recomendations': ['100', '285', '515', '93',
'268']},
{'userId': '667', 'sp_recomendations': ['272', '318', '357', '285',
'487']},
{'userId': '641', 'sp_recomendations': ['134', '513', '511', '23',
'242']},
{'userId': '668', 'sp_recomendations': ['50', '283', '272', '269',
'367']},
{'userId': '673', 'sp_recomendations': ['272', '315', '528', '12',
'79']},
{'userId': '671', 'sp_recomendations': ['210', '174', '172', '96',
'204']},
{'userId': '669', 'sp_recomendations': ['603', '511', '127', '408',
'479']},

```

```

{'userId': '676', 'sp_recomendations': ['50', '174', '181', '172',
'64']},
{'userId': '674', 'sp_recomendations': ['127', '282', '313', '50',
'1']},
{'userId': '652', 'sp_recomendations': ['96', '275', '699', '300',
'333']},
{'userId': '677', 'sp_recomendations': ['129', '288', '300', '124
0', '237']},
{'userId': '682', 'sp_recomendations': ['64', '50', '168', '172',
'173']},
{'userId': '679', 'sp_recomendations': ['50', '357', '172', '483',
'181']},
{'userId': '684', 'sp_recomendations': ['83', '483', '408', '50',
'172']},
{'userId': '685', 'sp_recomendations': ['302', '269', '882', '324',
'288']},
{'userId': '683', 'sp_recomendations': ['56', '127', '187', '513',
'302']},
{'userId': '691', 'sp_recomendations': ['318', '64', '603', '178',
'205']},
{'userId': '672', 'sp_recomendations': ['515', '275', '25', '269',
'127']},
{'userId': '692', 'sp_recomendations': ['194', '100', '208', '238',
'285']},
{'userId': '690', 'sp_recomendations': ['64', '69', '174', '357',
'496']},
{'userId': '689', 'sp_recomendations': ['50', '222', '121', '300',
'257']},
{'userId': '686', 'sp_recomendations': ['474', '187', '134', '127',
'12']},
{'userId': '693', 'sp_recomendations': ['318', '12', '480', '127',
'178']},
{'userId': '688', 'sp_recomendations': ['304', '359', '302', '326',
'754']},
{'userId': '697', 'sp_recomendations': ['124', '50', '127', '237',
'129']},
{'userId': '698', 'sp_recomendations': ['50', '173', '172', '83',
'190']},
{'userId': '670', 'sp_recomendations': ['482', '483', '96', '480',
'611']},
{'userId': '694', 'sp_recomendations': ['172', '318', '50', '483',
'174']},
{'userId': '680', 'sp_recomendations': ['408', '318', '169', '50',
'285']},
{'userId': '705', 'sp_recomendations': ['300', '282', '121', '79',
'181']},
{'userId': '701', 'sp_recomendations': ['50', '127', '285', '19',
'750']},
{'userId': '699', 'sp_recomendations': ['98', '185', '10', '23', '1
9']},
{'userId': '704', 'sp_recomendations': ['318', '134', '488', '178',
'205']},
{'userId': '707', 'sp_recomendations': ['582', '286', '170', '285',
'9']},
{'userId': '700', 'sp_recomendations': ['50', '173', '318', '181',
'174']},
{'userId': '687', 'sp_recomendations': ['313', '268', '269', '300',
'895']},
{'userId': '695', 'sp_recomendations': ['302', '268', '346', '242',
'1024']},
{'userId': '675', 'sp_recomendations': ['427', '318', '306', '531',

```



```

'1101']},
{'userId': '708', 'sp_recomendations': ['313', '50', '181', '1', '1
50']},
{'userId': '709', 'sp_recomendations': ['22', '318', '64', '98', '5
6']},
{'userId': '711', 'sp_recomendations': ['275', '408', '98', '8', '1
69']},
{'userId': '710', 'sp_recomendations': ['127', '483', '134', '192',
'23']},
{'userId': '712', 'sp_recomendations': ['143', '71', '210', '174',
'451']},
{'userId': '715', 'sp_recomendations': ['64', '318', '56', '176',
'98']},
{'userId': '713', 'sp_recomendations': ['344', '269', '302', '272',
'315']},
{'userId': '716', 'sp_recomendations': ['427', '64', '496', '318',
'98']},
{'userId': '681', 'sp_recomendations': ['286', '750', '289', '292',
'1394']},
{'userId': '678', 'sp_recomendations': ['127', '515', '100', '50',
'1']},
{'userId': '719', 'sp_recomendations': ['318', '64', '98', '520',
'735']},
{'userId': '702', 'sp_recomendations': ['222', '313', '228', '258',
'230']},
{'userId': '721', 'sp_recomendations': ['174', '357', '50', '1', '1
73']},
{'userId': '714', 'sp_recomendations': ['50', '181', '117', '121',
'118']},
{'userId': '717', 'sp_recomendations': ['313', '285', '126', '50',
'258']},
{'userId': '718', 'sp_recomendations': ['300', '222', '471', '121',
'15']},
{'userId': '696', 'sp_recomendations': ['302', '427', '178', '520',
'285']},
{'userId': '722', 'sp_recomendations': ['471', '124', '333', '845',
'117']},
{'userId': '724', 'sp_recomendations': ['313', '272', '258', '333',
'310']},
{'userId': '727', 'sp_recomendations': ['83', '173', '195', '12',
'169']},
{'userId': '725', 'sp_recomendations': ['100', '181', '19', '286',
'333']},
{'userId': '706', 'sp_recomendations': ['50', '181', '125', '628',
'1']},
{'userId': '720', 'sp_recomendations': ['302', '272', '316', '258',
'315']},
{'userId': '729', 'sp_recomendations': ['313', '272', '333', '354',
'300']},
{'userId': '726', 'sp_recomendations': ['257', '1', '310', '274',
'294']},
{'userId': '728', 'sp_recomendations': ['285', '100', '116', '124',
'15']},
{'userId': '703', 'sp_recomendations': ['50', '181', '127', '121',
'222']},
{'userId': '738', 'sp_recomendations': ['318', '408', '174', '169',
'173']},
{'userId': '736', 'sp_recomendations': ['515', '127', '286', '246',
'253']},
{'userId': '734', 'sp_recomendations': ['318', '496', '483', '313',
'83']},

```

```

{'userId': '730', 'sp_recomendations': ['258', '100', '50', '268',
'246']},
{'userId': '743', 'sp_recomendations': ['100', '276', '408', '9',
'302']},
{'userId': '742', 'sp_recomendations': ['127', '100', '14', '50',
'475']},
{'userId': '737', 'sp_recomendations': ['127', '192', '474', '64',
'357']},
{'userId': '733', 'sp_recomendations': ['100', '276', '515', '124',
'1142']},
{'userId': '745', 'sp_recomendations': ['12', '169', '190', '64',
'98']},
{'userId': '740', 'sp_recomendations': ['302', '286', '269', '242',
'300']},
{'userId': '735', 'sp_recomendations': ['50', '515', '285', '242',
'1']},
{'userId': '747', 'sp_recomendations': ['480', '603', '357', '318',
'187']},
{'userId': '723', 'sp_recomendations': ['174', '169', '168', '50',
'178']},
{'userId': '739', 'sp_recomendations': ['318', '22', '195', '172',
'96']},
{'userId': '749', 'sp_recomendations': ['496', '173', '50', '174',
'136']},
{'userId': '748', 'sp_recomendations': ['50', '172', '408', '174',
'318']},
{'userId': '746', 'sp_recomendations': ['174', '83', '50', '64', '1
72']},
{'userId': '731', 'sp_recomendations': ['64', '496', '170', '427',
'357']},
{'userId': '750', 'sp_recomendations': ['269', '303', '288', '270',
'306']},
{'userId': '741', 'sp_recomendations': ['357', '178', '480', '22',
'98']},
{'userId': '751', 'sp_recomendations': ['50', '172', '178', '83',
'181']},
{'userId': '756', 'sp_recomendations': ['178', '603', '251', '919',
'100']},
{'userId': '757', 'sp_recomendations': ['64', '89', '174', '173',
'50']},
{'userId': '752', 'sp_recomendations': ['302', '306', '269', '288',
'327']},
{'userId': '758', 'sp_recomendations': ['483', '474', '657', '134',
'427']},
{'userId': '732', 'sp_recomendations': ['269', '690', '286', '300',
'882']},
{'userId': '762', 'sp_recomendations': ['515', '173', '302', '955',
'270']},
{'userId': '744', 'sp_recomendations': ['127', '302', '657', '603',
'483']},
{'userId': '754', 'sp_recomendations': ['127', '9', '15', '276', '9
22']},
{'userId': '753', 'sp_recomendations': ['313', '98', '483', '199',
'427']},
{'userId': '763', 'sp_recomendations': ['59', '12', '127', '483',
'60']},
{'userId': '764', 'sp_recomendations': ['64', '318', '22', '143',
'496']},
{'userId': '767', 'sp_recomendations': ['483', '98', '170', '478',
'187']},
{'userId': '769', 'sp_recomendations': ['222', '269', '111', '1',

```

```

'258' ]},
{'userId': '755', 'sp_recomendations': ['300', '269', '302', '311',
'258' ]},
{'userId': '771', 'sp_recomendations': ['408', '83', '251', '258',
'169' ]},
{'userId': '768', 'sp_recomendations': ['127', '173', '313', '272',
'100' ]},
{'userId': '773', 'sp_recomendations': ['169', '919', '408', '100',
'60' ]},
{'userId': '765', 'sp_recomendations': ['127', '170', '285', '14',
'242' ]},
{'userId': '772', 'sp_recomendations': ['313', '302', '272', '315',
'258' ]},
{'userId': '766', 'sp_recomendations': ['474', '318', '127', '135',
'205' ]},
{'userId': '774', 'sp_recomendations': ['179', '183', '199', '530',
'178' ]},
{'userId': '760', 'sp_recomendations': ['98', '204', '604', '216',
'195' ]},
{'userId': '761', 'sp_recomendations': ['50', '117', '181', '222',
'237' ]},
{'userId': '777', 'sp_recomendations': ['204', '357', '202', '168',
'196' ]},
{'userId': '759', 'sp_recomendations': ['50', '181', '1', '121', '1
17' ]},
{'userId': '776', 'sp_recomendations': ['50', '127', '174', '168',
'511' ]},
{'userId': '780', 'sp_recomendations': ['318', '174', '50', '313',
'172' ]},
{'userId': '779', 'sp_recomendations': ['50', '195', '181', '1', '2
58' ]},
{'userId': '778', 'sp_recomendations': ['168', '56', '11', '42',
'7' ]},
{'userId': '782', 'sp_recomendations': ['313', '272', '315', '316',
'748' ]},
{'userId': '786', 'sp_recomendations': ['318', '496', '174', '98',
'191' ]},
{'userId': '784', 'sp_recomendations': ['313', '302', '258', '272',
'750' ]},
{'userId': '770', 'sp_recomendations': ['275', '258', '813', '100',
'246' ]},
{'userId': '788', 'sp_recomendations': ['318', '483', '498', '661',
'98' ]},
{'userId': '789', 'sp_recomendations': ['50', '127', '9', '100', '1
50' ]},
{'userId': '790', 'sp_recomendations': ['22', '56', '174', '50', '1
72' ]},
{'userId': '787', 'sp_recomendations': ['313', '750', '690', '310',
'326' ]},
{'userId': '783', 'sp_recomendations': ['346', '258', '269', '887',
'750' ]},
{'userId': '785', 'sp_recomendations': ['318', '50', '174', '183',
'269' ]},
{'userId': '794', 'sp_recomendations': ['127', '50', '285', '187',
'515' ]},
{'userId': '781', 'sp_recomendations': ['50', '135', '483', '134',
'172' ]},
{'userId': '796', 'sp_recomendations': ['22', '69', '204', '79', '3
16' ]},
{'userId': '795', 'sp_recomendations': ['169', '209', '168', '173',
'89' ]},

```

```
{'userId': '793', 'sp_recomendations': ['50', '127', '1142', '100',  
'181']},  
{'userId': '798', 'sp_recomendations': ['748', '50', '121', '143',  
'181']},  
{'userId': '791', 'sp_recomendations': ['50', '306', '302', '275',  
'300']},  
{'userId': '802', 'sp_recomendations': ['258', '183', '326', '176',  
'98']},  
{'userId': '800', 'sp_recomendations': ['223', '127', '50', '275',  
'292']},  
{'userId': '804', 'sp_recomendations': ['50', '318', '174', '172',  
'64']},  
{'userId': '803', 'sp_recomendations': ['242', '286', '269', '340',  
'306']},  
{'userId': '775', 'sp_recomendations': ['313', '272', '315', '269',  
'750']},  
{'userId': '792', 'sp_recomendations': ['124', '100', '15', '1',  
'7']},  
{'userId': '799', 'sp_recomendations': ['173', '174', '654', '427',  
'479']},  
{'userId': '805', 'sp_recomendations': ['175', '475', '137', '238',  
'55']},  
{'userId': '806', 'sp_recomendations': ['187', '197', '654', '127',  
'318']},  
{'userId': '807', 'sp_recomendations': ['496', '22', '313', '172',  
'174']},  
{'userId': '797', 'sp_recomendations': ['50', '181', '257', '127',  
'781']},  
{'userId': '801', 'sp_recomendations': ['313', '333', '288', '302',  
'300']},  
{'userId': '809', 'sp_recomendations': ['302', '272', '315', '313',  
'307']},  
{'userId': '815', 'sp_recomendations': ['50', '318', '172', '174',  
'496']},  
{'userId': '817', 'sp_recomendations': ['273', '117', '7', '327',  
'288']},  
{'userId': '821', 'sp_recomendations': ['64', '318', '98', '427',  
'357']},  
{'userId': '818', 'sp_recomendations': ['302', '316', '303', '313',  
'288']},  
{'userId': '814', 'sp_recomendations': ['98', '7', '100', '185', '6  
56']},  
{'userId': '812', 'sp_recomendations': ['333', '294', '748', '288',  
'328']},  
{'userId': '823', 'sp_recomendations': ['191', '127', '64', '22',  
'318']},  
{'userId': '825', 'sp_recomendations': ['121', '222', '174', '12',  
'117']},  
{'userId': '827', 'sp_recomendations': ['269', '313', '272', '302',  
'316']},  
{'userId': '829', 'sp_recomendations': ['408', '192', '318', '190',  
'275']},  
{'userId': '811', 'sp_recomendations': ['258', '300', '315', '304',  
'286']},  
{'userId': '830', 'sp_recomendations': ['50', '98', '174', '313',  
'205']},  
{'userId': '826', 'sp_recomendations': ['22', '50', '174', '89', '1  
72']},  
{'userId': '831', 'sp_recomendations': ['50', '174', '64', '12', '2  
2']},  
{'userId': '819', 'sp_recomendations': ['315', '302', '286', '182',
```

```

'248' ]},
{'userId': '828', 'sp_recomendations': ['958', '316', '19', '512',
'652' ]},
{'userId': '808', 'sp_recomendations': ['313', '750', '302', '262',
'346' ]},
{'userId': '835', 'sp_recomendations': ['318', '98', '197', '488',
'313' ]},
{'userId': '833', 'sp_recomendations': ['179', '127', '185', '89',
'654' ]},
{'userId': '836', 'sp_recomendations': ['185', '318', '659', '12',
'174' ]},
{'userId': '816', 'sp_recomendations': ['313', '294', '331', '258',
'328' ]},
{'userId': '838', 'sp_recomendations': ['173', '50', '172', '318',
'169' ]},
{'userId': '839', 'sp_recomendations': ['127', '813', '285', '50',
'326' ]},
{'userId': '840', 'sp_recomendations': ['169', '98', '100', '483',
'14' ]},
{'userId': '832', 'sp_recomendations': ['313', '50', '258', '471',
'307' ]},
{'userId': '810', 'sp_recomendations': ['313', '269', '902', '333',
'289' ]},
{'userId': '844', 'sp_recomendations': ['318', '173', '50', '12',
'83' ]},
{'userId': '843', 'sp_recomendations': ['89', '183', '195', '205',
'135' ]},
{'userId': '834', 'sp_recomendations': ['50', '316', '127', '313',
'315' ]},
{'userId': '846', 'sp_recomendations': ['174', '50', '12', '302',
'191' ]},
{'userId': '837', 'sp_recomendations': ['151', '283', '740', '275',
'19' ]},
{'userId': '813', 'sp_recomendations': ['270', '300', '750', '289',
'307' ]},
{'userId': '842', 'sp_recomendations': ['302', '269', '272', '313',
'303' ]},
{'userId': '847', 'sp_recomendations': ['173', '50', '172', '181',
'174' ]},
{'userId': '848', 'sp_recomendations': ['603', '181', '50', '173',
'483' ]},
{'userId': '822', 'sp_recomendations': ['408', '169', '189', '1',
'272' ]},
{'userId': '852', 'sp_recomendations': ['50', '127', '408', '181',
'515' ]},
{'userId': '851', 'sp_recomendations': ['172', '64', '180', '174',
'50' ]},
{'userId': '849', 'sp_recomendations': ['143', '172', '427', '174',
'633' ]},
{'userId': '854', 'sp_recomendations': ['56', '100', '127', '12',
'357' ]},
{'userId': '850', 'sp_recomendations': ['173', '210', '174', '121',
'50' ]},
{'userId': '858', 'sp_recomendations': ['127', '515', '9', '269',
'100' ]},
{'userId': '853', 'sp_recomendations': ['302', '288', '300', '270',
'880' ]},
{'userId': '855', 'sp_recomendations': ['198', '166', '165', '512',
'59' ]},
{'userId': '824', 'sp_recomendations': ['288', '268', '292', '304',
'325' ]},

```

```
{'userId': '845', 'sp_recomendations': ['313', '286', '272', '690',  
'302']},  
{'userId': '841', 'sp_recomendations': ['302', '313', '272', '316',  
'315']},  
{'userId': '859', 'sp_recomendations': ['313', '15', '421', '955',  
'475']},  
{'userId': '862', 'sp_recomendations': ['50', '64', '22', '657', '4  
80']},  
{'userId': '856', 'sp_recomendations': ['316', '313', '272', '315',  
'750']},  
{'userId': '820', 'sp_recomendations': ['302', '316', '313', '288',  
'315']},  
{'userId': '863', 'sp_recomendations': ['272', '316', '313', '315',  
'268']},  
{'userId': '860', 'sp_recomendations': ['100', '514', '272', '313',  
'285']},  
{'userId': '857', 'sp_recomendations': ['116', '475', '19', '258',  
'14']},  
{'userId': '864', 'sp_recomendations': ['50', '173', '174', '318',  
'12']},  
{'userId': '865', 'sp_recomendations': ['169', '408', '100', '189',  
'919']},  
{'userId': '868', 'sp_recomendations': ['135', '50', '474', '100',  
'408']},  
{'userId': '867', 'sp_recomendations': ['511', '50', '134', '474',  
'483']},  
{'userId': '861', 'sp_recomendations': ['529', '275', '170', '83',  
'52']},  
{'userId': '870', 'sp_recomendations': ['127', '64', '657', '56',  
'318']},  
{'userId': '871', 'sp_recomendations': ['50', '172', '195', '96',  
'174']},  
{'userId': '875', 'sp_recomendations': ['514', '169', '134', '302',  
'603']},  
{'userId': '876', 'sp_recomendations': ['318', '48', '511', '178',  
'527']},  
{'userId': '872', 'sp_recomendations': ['313', '300', '272', '117',  
'282']},  
{'userId': '866', 'sp_recomendations': ['315', '303', '306', '272',  
'242']},  
{'userId': '877', 'sp_recomendations': ['173', '98', '515', '170',  
'59']},  
{'userId': '873', 'sp_recomendations': ['313', '258', '300', '292',  
'326']},  
{'userId': '880', 'sp_recomendations': ['100', '357', '318', '127',  
'98']},  
{'userId': '878', 'sp_recomendations': ['64', '427', '269', '474',  
'496']},  
{'userId': '869', 'sp_recomendations': ['100', '127', '515', '151',  
'116']},  
{'userId': '881', 'sp_recomendations': ['174', '22', '651', '121',  
'265']},  
{'userId': '879', 'sp_recomendations': ['127', '50', '237', '181',  
'15']},  
{'userId': '883', 'sp_recomendations': ['285', '408', '512', '513',  
'127']},  
{'userId': '882', 'sp_recomendations': ['515', '427', '174', '98',  
'50']},  
{'userId': '884', 'sp_recomendations': ['9', '921', '269', '100',  
'127']},  
{'userId': '886', 'sp_recomendations': ['50', '172', '174', '64',
```

```

'318']],
{'userId': '885', 'sp_recomendations': ['318', '169', '174', '189',
'300']],
{'userId': '889', 'sp_recomendations': ['178', '64', '513', '318',
'56']],
{'userId': '874', 'sp_recomendations': ['127', '302', '654', '285',
'357']],
{'userId': '892', 'sp_recomendations': ['127', '483', '187', '511',
'100']],
{'userId': '890', 'sp_recomendations': ['134', '127', '23', '480',
'483']],
{'userId': '893', 'sp_recomendations': ['50', '172', '69', '96', '5
6']],
{'userId': '887', 'sp_recomendations': ['210', '432', '763', '50',
'181']],
{'userId': '891', 'sp_recomendations': ['127', '313', '285', '100',
'740']],
{'userId': '894', 'sp_recomendations': ['318', '12', '190', '124',
'515']],
{'userId': '896', 'sp_recomendations': ['174', '22', '50', '318',
'173']],
{'userId': '897', 'sp_recomendations': ['659', '496', '143', '96',
'79']],
{'userId': '901', 'sp_recomendations': ['174', '22', '88', '12', '5
68']],
{'userId': '899', 'sp_recomendations': ['174', '50', '194', '496',
'318']],
{'userId': '903', 'sp_recomendations': ['127', '187', '98', '192',
'185']],
{'userId': '904', 'sp_recomendations': ['275', '237', '603', '255',
'216']],
{'userId': '907', 'sp_recomendations': ['282', '813', '198', '98',
'318']],
{'userId': '905', 'sp_recomendations': ['302', '313', '124', '100',
'150']],
{'userId': '902', 'sp_recomendations': ['318', '515', '480', '50',
'483']],
{'userId': '898', 'sp_recomendations': ['316', '272', '302', '313',
'315']],
{'userId': '895', 'sp_recomendations': ['50', '275', '181', '151',
'283']],
{'userId': '906', 'sp_recomendations': ['408', '285', '286', '100',
'276']],
{'userId': '900', 'sp_recomendations': ['318', '100', '480', '589',
'474']],
{'userId': '908', 'sp_recomendations': ['318', '427', '496', '483',
'603']],
{'userId': '916', 'sp_recomendations': ['100', '483', '98', '9', '1
56']],
{'userId': '911', 'sp_recomendations': ['173', '603', '709', '191',
'89']],
{'userId': '912', 'sp_recomendations': ['483', '318', '132', '64',
'357']],
{'userId': '914', 'sp_recomendations': ['197', '402', '313', '778',
'736']],
{'userId': '918', 'sp_recomendations': ['213', '709', '1137', '19
0', '507']],
{'userId': '919', 'sp_recomendations': ['318', '9', '272', '340',
'313']],
{'userId': '921', 'sp_recomendations': ['174', '173', '143', '181',
'121']]

```

```

{'userId': '910', 'sp_recomendations': ['98', '127', '357', '56',
'174']},
{'userId': '913', 'sp_recomendations': ['172', '50', '408', '64',
'318']},
{'userId': '915', 'sp_recomendations': ['313', '286', '268', '302',
'750']},
{'userId': '922', 'sp_recomendations': ['50', '172', '181', '174',
'173']},
{'userId': '923', 'sp_recomendations': ['174', '50', '168', '276',
'100']},
{'userId': '928', 'sp_recomendations': ['98', '187', '496', '165',
'269']},
{'userId': '927', 'sp_recomendations': ['121', '64', '210', '228',
'1']},
{'userId': '924', 'sp_recomendations': ['496', '178', '318', '173',
'50']},
{'userId': '929', 'sp_recomendations': ['127', '12', '185', '318',
'483']},
{'userId': '931', 'sp_recomendations': ['515', '316', '272', '286',
'275']},
{'userId': '917', 'sp_recomendations': ['9', '276', '100', '285',
'740']},
{'userId': '932', 'sp_recomendations': ['199', '479', '513', '511',
'169']},
{'userId': '909', 'sp_recomendations': ['170', '165', '166', '275',
'707']},
{'userId': '934', 'sp_recomendations': ['663', '174', '963', '488',
'56']},
{'userId': '933', 'sp_recomendations': ['127', '56', '100', '64',
'98']},
{'userId': '935', 'sp_recomendations': ['313', '237', '15', '117',
'300']},
{'userId': '938', 'sp_recomendations': ['50', '181', '222', '121',
'288']},
{'userId': '940', 'sp_recomendations': ['258', '482', '96', '12',
'313']},
{'userId': '888', 'sp_recomendations': ['191', '514', '269', '100',
'202']},
{'userId': '925', 'sp_recomendations': ['98', '185', '288', '333',
'447']},
{'userId': '942', 'sp_recomendations': ['318', '520', '174', '480',
'496']},
{'userId': '937', 'sp_recomendations': ['408', '9', '50', '285', '5
15']},
{'userId': '926', 'sp_recomendations': ['272', '269', '315', '302',
'313']},
{'userId': '943', 'sp_recomendations': ['64', '50', '127', '12', '9
8']},
{'userId': '939', 'sp_recomendations': ['127', '257', '121', '285',
'742']},
{'userId': '936', 'sp_recomendations': ['127', '272', '1007', '93',
'1129']},
{'userId': '930', 'sp_recomendations': ['165', '64', '113', '269',
'116']},
{'userId': '920', 'sp_recomendations': ['313', '302', '272', '258',
'347']},
{'userId': '941', 'sp_recomendations': ['408', '124', '1007', '1',
'919']}

```


In [307]:

```

user_list_sp=[]

for i in sp_recommendations:
    if int(i["userId"]) in user_list:
        user_list_sp.append(i)
user_list_sp

```

Out[307]:

```

[{'userId': '267', 'sp_recomendations': ['114', '50', '474', '408',
'169']},
{'userId': '11', 'sp_recomendations': ['83', '318', '9', '286', '60
3']},
{'userId': '198', 'sp_recomendations': ['50', '474', '127', '172',
'173']},
{'userId': '184', 'sp_recomendations': ['488', '134', '285', '483',
'9']},
{'userId': '314', 'sp_recomendations': ['64', '121', '692', '282',
'143']},
{'userId': '163', 'sp_recomendations': ['316', '318', '272', '64',
'357']},
{'userId': '504', 'sp_recomendations': ['318', '127', '735', '258',
'98']},
{'userId': '653', 'sp_recomendations': ['195', '22', '174', '50',
'746']},
{'userId': '710', 'sp_recomendations': ['127', '483', '134', '192',
'23']},
{'userId': '881', 'sp_recomendations': ['174', '22', '651', '121',
'265']}]]

```

In [308]:

```

sp_user_result=pd.DataFrame(user_list_sp)
sp_user_result

```

Out[308]:

	sp_recomendations	userId
0	[114, 50, 474, 408, 169]	267
1	[83, 318, 9, 286, 603]	11
2	[50, 474, 127, 172, 173]	198
3	[488, 134, 285, 483, 9]	184
4	[64, 121, 692, 282, 143]	314
5	[316, 318, 272, 64, 357]	163
6	[318, 127, 735, 258, 98]	504
7	[195, 22, 174, 50, 746]	653
8	[127, 483, 134, 192, 23]	710
9	[174, 22, 651, 121, 265]	881

In [380]:

```
# Get the test pandas dataframe
test_pd=test.toPandas()
test_pd.columns=["userId", "movieId", "rating"]
test_pd.head()
```

Out[380]:

	userId	movieId	rating
0	1	20	4
1	1	33	4
2	1	61	4
3	1	117	3
4	1	155	2

In [475]:

```
# rearrange the output format
df_sp=pd.DataFrame(top_n)
df_sp2 = pd.DataFrame(df_sp.values.T,index=df_sp.columns, columns=df_sp.index)
df_sp2.head()
```

Out[475]:

	0	1	2	3
196	(285, 4.282994600470375)	(663, 4.265198850470498)	(655, 4.1866216612752485)	(1007, 4.157957746470623)
186	(98, 4.98048269419873)	(71, 4.471490033280152)	(939, 4.383274280970822)	(79, 4.3635237627152055)
22	(173, 5)	(50, 4.8279578514880095)	(96, 4.642193943427386)	(144, 4.62467283054835)
244	(475, 5)	(9, 5)	(56, 5)	(208, 4.8161568425113295)
166	(313, 4.788111641230368)	(300, 4.621002939116737)	(258, 4.254044671164396)	(347, 4.21435984535225)

In [562]:

```
# adjust the output dataframe to the right format
rmd = df_sp2.rename(columns = lambda x : 'rmd_' + str(x))
rmd_list=["rmd_0","rmd_1","rmd_2","rmd_3","rmd_4"]

rmd_sp = rmd['rmd_0'].apply(pd.Series)
rmd_sp.columns=["movieId","score"]
rmd_sp["userId"]=df_sp2.index
for i in range(4):
    df_rmd = rmd[rmd_list[i+1]].apply(pd.Series)
    df_rmd.columns=["movieId","score"]
    df_rmd["userId"]=df_sp2.index
    rmd_sp=pd.concat([rmd_sp, df_rmd], ignore_index=True)

rmd_sp[['userId']]= rmd_sp[['userId']].astype(int)
rmd_sp[['movieId']]= rmd_sp[['movieId']].astype(int)
rmd_sp.head()
```

Out[562]:

	movieId	score	userId
0	285	4.282995	196
1	98	4.980483	186
2	173	5.000000	22
3	475	5.000000	244
4	313	4.788112	166

In [563]:

```
#merge with the real rating
sp_user_df=pd.merge(rmd_sp,test_pd,on=['userId','movieId'])
sp_user_df.head()
```

Out[563]:

	movieId	score	userId	rating
0	100	4.344550	63	5
1	1084	4.201571	50	5
2	127	5.000000	157	5
3	22	4.983510	278	5
4	661	5.000000	7	5

In [564]:

```

sp_user=rmd_sp[rmd_sp["userId"].isin(user_list)]
sp_user.head()

rmd_sp.columns=["movieId","predict_score","userId"]
rmd_sp_final = rmd_sp[['userId', 'movieId', 'predict_score']]
rmd_sp_final.head()

```

Out[564]:

	userId	movieId	predict_score
0	196	285	4.282995
1	186	98	4.980483
2	22	173	5.000000
3	244	475	5.000000
4	166	313	4.788112

In [540]:

```

# get the user prediction summary dataframe in the user list
sp_user_final=rmd_sp_final.merge(test_pd,on=['userId','movieId'])
sp_user_final=sp_user_final[sp_user_final.userId.isin(user_list)]
sp_user_final

```

Out[540]:

	userId	movieId	predict_score	rating
271	163	318	4.130945	4
445	314	692	4.953824	5
630	163	64	3.882521	4
680	653	50	4.214062	5

Model 3 : SageMaker's Factorization Machines

Another model built for this recommendation system is Factorization Machines(FM), the famous built-in algorithm of Amazon SageMaker, which is widely used for recommendation system.(eg, FM models are used by Netflix to recommend movies for users.) Factorization Machines as is one class of collaborative filtering algorithms. As the name suggests, FM uses matrix factorization to reduce problem dimensionality and thus, greatly boost computational efficiency on large sparse dataset.

In the movielens dataset (and also in real world practice !), the number of users and items are often large whereas users normally rate a small portion of all movies available. Therefore, the actual number of recommendations is quite small, resulting a large sparse dataset. The basic idea of factorization in FM model is that a sparse rating matrix can be decomposed into a dense user matrix and item matrix with lower dimension. Another benefit of using FM is that matrix factorization can also help us fill the blank values in the rating matrix, which means we can recommend new items to users.

Step 1 : Preprocessing and creating sparse matrix

In [39]:

```
#necessary imports
import sagemaker
import sagemaker.amazon.common as smac
from sagemaker import get_execution_role
from sagemaker.predictor import json_deserializer
from sagemaker.amazon.amazon_estimator import get_image_uri

from scipy.sparse import lil_matrix
import boto3, io, os
```

In [42]:

```
# get the number of unique users and movies
n_users= train_movie_ratings['user_id'].max()
n_movies=train_movie_ratings['movie_id'].max()
# In sparse matrix, features are one-hot encoded
#number of features should be the sum of number of movies and users
n_features=n_users+n_movies

#refence previously created training and test ddataframe in als model
n_test_ratings=len(test_movie_ratings.index)
n_train_ratings=len(train_movie_ratings.index)
```

In [43]:

```
# demonstrate the result
print (" number of users: ", n_users)
print (" number of movies: ", n_movies)
print (" Training Count: ", n_train_ratings)
print (" Test Count: ", n_test_ratings)
print (" Features (number of users + number of movies): ", n_features)
```

```
number of users: 943
number of movies: 1682
Training Count: 90570
Test Count: 9430
Features (number of users + number of movies): 2625
```

In [44]:

```

#define a function to convert dataframe into sparse matrix
def loadDataset(data, lines, columns):
    # Features are one-hot encoded in a sparse matrix
    X = lil_matrix((lines, columns)).astype('float32')
    # Labels are stored in a vector
    y = []
    line=0
    for index, row in data.iterrows():
        X[line,row['user_id']-1] = 1
        X[line, n_users+(row['movie_id']-1)] = 1
        if int(row['rating']) >= 4:
            y.append(1)
        else:
            y.append(0)
        line=line+1

    y=np.array(y).astype('float32')
    return X,y

#derive the sparse matrix for test and training dataset
X_train, y_train = loadDataset(train_movie_ratings, n_train_ratings, n_features)
X_test, y_test = loadDataset(test_movie_ratings, n_test_ratings, n_features)

```

In [316]:

```

#show the shape of sparse matrix
print(X_test.shape)
print(y_test.shape)
assert X_test.shape == (n_test_ratings, n_features)
assert y_test.shape == (n_test_ratings, )

#test labels are quite balanced
zero_labels = np.count_nonzero(y_test)
print("Test labels: %d zeros, %d ones" % (zero_labels, n_test_ratings-zero_labels))
print("Ratio of ones to zeros:", round((n_test_ratings-zero_labels)/zero_labels, 2) )

```

```

(9430, 2625)
(9430,)
Test labels: 5469 zeros, 3961 ones
Ratio of ones to zeros: 0.72

```

Step 2 : Store the data to S3 in protobuf format

In [46]:

```

#specify my personal bucket name
bucket = 'movierecom'
prefix = 'fm'

#write the key and prefix for train, test and output
train_key      = 'train.protobuf'
train_prefix    = '{}/{}'.format(prefix, 'train')

test_key        = 'test.protobuf'
test_prefix     = '{}/{}'.format(prefix, 'test')

output_prefix   = 's3://{}/{}'/output'.format(bucket, prefix)

```

In [47]:

```

# define a function to convert data to Protobuf format
def writeDatasetToProtobuf(X, bucket, prefix, key, d_type, y=None):
    Pbuf = io.BytesIO()
    if d_type == "sparse":
        smac.write_spmatrix_to_sparse_tensor(Pbuf, X, labels=y)
    else:
        smac.write_numpy_to_dense_tensor(Pbuf, X, labels=y)

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

# show the storage path in S3
fm_train_data_path = writeDatasetToProtobuf(X_train, bucket, train_prefix, train_key, "sparse", y_train)
fm_test_data_path  = writeDatasetToProtobuf(X_test, bucket, test_prefix, test_key, "sparse", y_test)

print ("Training data S3 path: ",fm_train_data_path)
print ("Test data S3 path: ",fm_test_data_path)
print ("FM model output S3 path: {}".format(output_prefix))

```

Training data S3 path: s3://movierecom/fm/train/train.protobuf

Test data S3 path: s3://movierecom/fm/test/test.protobuf

FM model output S3 path: s3://movierecom/fm/output

Step 3 : Fit SageMaker factorization machines

In [48]:

```

fm = sagemaker.estimator.Estimator(get_image_uri(boto3.Session().region_name, "factorization-machines"),
                                   get_execution_role(),
                                   train_instance_count=1,
                                   train_instance_type='ml.m5.large',
                                   output_path=output_prefix,
                                   sagemaker_session=sagemaker.Session())

```

In [49]:

```
# initial a set of hyperparameters first
fm.set_hyperparameters(feature_dim=n_features,
                        predictor_type='binary_classifier',
                        mini_batch_size=1000,
                        num_factors=64,
                        epochs=50)
```


In [50]:

```
#fit the fm model specifying storage path  
fm.fit({'train': fm_train_data_path, 'test': fm_test_data_path})
```

```

2020-04-28 14:06:55 Starting - Starting the training job...
2020-04-28 14:06:56 Starting - Launching requested ML instances...
2020-04-28 14:07:54 Starting - Preparing the instances for training.....
2020-04-28 14:08:40 Downloading - Downloading input data...
2020-04-28 14:09:18 Training - Downloading the training image..Docker
entrypoint called with argument(s): train
Running default environment configuration script
/opt/amazon/lib/python2.7/site-packages/pandas/util/nosetester.py:1
3: DeprecationWarning: Importing from numpy.testing.nosetester is de
precated, import from numpy.testing instead.
    from numpy.testing import nosetester
[04/28/2020 14:09:33 INFO 140478314145600] Reading default configura
tion from /opt/amazon/lib/python2.7/site-packages/algorithm/resource
s/default-conf.json: {u'factors_lr': u'0.0001', u'linear_init_sigm
a': u'0.01', u'epochs': 1, u'_wd': u'1.0', u'_num_kv_servers': u'aut
o', u'use_bias': u'true', u'factors_init_sigma': u'0.001', u'_log_le
vel': u'info', u'bias_init_method': u'normal', u'linear_init_metho
d': u'normal', u'linear_lr': u'0.001', u'factors_init_method': u'nor
mal', 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'1000',
  u'_use_full_symbolic': u'true', u'batch_metrics_publish_interval':
  u'500', 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'0.00
1', u'_kvstore': u'auto', u'_learning_rate': u'1.0', u'_optimizer':
  u'adam'}
[04/28/2020 14:09:33 INFO 140478314145600] Reading provided configur
ation from /opt/ml/input/config/hyperparameters.json: {u'epochs':
  u'50', u'feature_dim': u'2625', u'mini_batch_size': u'1000', u'pred
ictor_type': u'binary_classifier', u'num_factors': u'64'}
[04/28/2020 14:09:33 INFO 140478314145600] Final configuration: {u'f
actors_lr': u'0.0001', u'linear_init_sigma': u'0.01', u'epochs': u'5
0', u'feature_dim': u'2625', u'num_factors': u'64', u'_wd': u'1.0',
  u'_num_kv_servers': u'auto', u'use_bias': u'true', u'factors_init_s
igma': u'0.001', u'_log_level': u'info', u'bias_init_method': u'norm
al', u'linear_init_method': u'normal', u'linear_lr': u'0.001', u'fac
tors_init_method': u'normal', u'_tuning_objective_metric': u'', u'bi
as_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_metric
s_publish_interval': u'500', u'predictor_type': u'binary_classifie
r', u'bias_init_sigma': u'0.01', u'_num_gpus': u'auto', u'_data_form
at': 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'a
dam'}
[04/28/2020 14:09:33 WARNING 140478314145600] Loggers have already b
een setup.
Process 1 is a worker.
[04/28/2020 14:09:34 INFO 140478314145600] Using default worker.
[2020-04-28 14:09:34.006] [tensorio] [warning] TensorIO is already i
nitialized; ignoring the initialization routine.
[2020-04-28 14:09:34.017] [tensorio] [info] epoch_stats={"data_pipel
ine": "/opt/ml/input/data/train", "epoch": 0, "duration": 15, "num_e
xamples": 1, "num_bytes": 64000}
[04/28/2020 14:09:34 INFO 140478314145600] nvidia-smi took: 0.025141
0007477 secs to identify 0 gpus
[04/28/2020 14:09:34 INFO 140478314145600] Number of GPUs being use
d: 0
[04/28/2020 14:09:34 INFO 140478314145600] [Sparse network] Building
a sparse network.
[04/28/2020 14:09:34 INFO 140478314145600] Create Store: local
#metrics {"Metrics": {"initialize.time": {"count": 1, "max": 42.4320

```

```
6977844238, "sum": 42.43206977844238, "min": 42.43206977844238}}, "EndTime": 1588082974.057732, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588082974.000808}
```

```
#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": {"count": 1, "max": 0, "sum": 0.0, "min": 0}, "Total Batches Seen": {"count": 1, "max": 1, "sum": 1.0, "min": 1}, "Total Records Seen": {"count": 1, "max": 1000, "sum": 1000.0, "min": 1000}, "Max Records Seen Between Resets": {"count": 1, "max": 1000, "sum": 1000.0, "min": 1000}, "Reset Count": {"count": 1, "max": 1, "sum": 1.0, "min": 1}}, "EndTime": 1588082974.057982, "Dimensions": {"Host": "algo-1", "Meta": "init_train_data_iter", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588082974.057883}
```

```
[14:09:34] /opt/brazil-pkg-cache/packages/AIAlgorithmsMXNet/AIAlgorithmsMXNet-1.1.x.202841.0/AL2012/generic-flavor/src/src/kvstore/./kvstore_local.h:280: Warning: non-default weights detected during kvstore pull. This call has been ignored. Please make sure to use row_sparse_pull with row_ids.
```

```
[14:09:34] /opt/brazil-pkg-cache/packages/AIAlgorithmsMXNet/AIAlgorithmsMXNet-1.1.x.202841.0/AL2012/generic-flavor/src/src/kvstore/./kvstore_local.h:280: Warning: non-default weights detected during kvstore pull. This call has been ignored. Please make sure to use row_sparse_pull with row_ids.
```

```
[04/28/2020 14:09:34 INFO 140478314145600] #quality_metric: host=algo-1, epoch=0, batch=0 train binary_classification_accuracy <score>=0.549
```

```
[04/28/2020 14:09:34 INFO 140478314145600] #quality_metric: host=algo-1, epoch=0, batch=0 train binary_classification_cross_entropy <loss>=0.691680419922
```

```
[04/28/2020 14:09:34 INFO 140478314145600] #quality_metric: host=algo-1, epoch=0, batch=0 train binary_f_1.000 <score>=0.698730794923
```

```
[2020-04-28 14:09:35.122] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 2, "duration": 1039, "num_examples": 91, "num_bytes": 5796480}
```

```
[04/28/2020 14:09:35 INFO 140478314145600] #quality_metric: host=algo-1, epoch=0, train binary_classification_accuracy <score>=0.549615384615
```

```
[04/28/2020 14:09:35 INFO 140478314145600] #quality_metric: host=algo-1, epoch=0, train binary_classification_cross_entropy <loss>=0.683836189689
```

```
[04/28/2020 14:09:35 INFO 140478314145600] #quality_metric: host=algo-1, epoch=0, train binary_f_1.000 <score>=0.708654700551
```

```
#metrics {"Metrics": {"epochs": {"count": 1, "max": 50, "sum": 50.0, "min": 50}, "update.time": {"count": 1, "max": 1065.8340454101562, "sum": 1065.8340454101562, "min": 1065.8340454101562}}, "EndTime": 1588082975.124095, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588082974.057801}
```

```
[04/28/2020 14:09:35 INFO 140478314145600] #progress_metric: host=algo-1, completed 2 % of epochs
```

```
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 92, "s
```

```
um": 92.0, "min": 92}, "Total Records Seen": {"count": 1, "max": 91570, "sum": 91570.0, "min": 91570}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 2, "sum": 2.0, "min": 2}}, "EndTime": 1588082975.124633, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 0}, "StartTime": 1588082974.058225}
```

```
[04/28/2020 14:09:35 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=84916.1180436 records/second
```

```
[04/28/2020 14:09:35 INFO 140478314145600] #quality_metric: host=algo-1, epoch=1, batch=0 train binary_classification_accuracy <score>=0.714
```

```
[04/28/2020 14:09:35 INFO 140478314145600] #quality_metric: host=algo-1, epoch=1, batch=0 train binary_classification_cross_entropy <loss>=0.679529418945
```

```
[04/28/2020 14:09:35 INFO 140478314145600] #quality_metric: host=algo-1, epoch=1, batch=0 train binary_f1.000 <score>=0.774803149606
```

```
[2020-04-28 14:09:36.271] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 4, "duration": 1144, "num_examples": 91, "num_bytes": 5796480}
```

```
[04/28/2020 14:09:36 INFO 140478314145600] #quality_metric: host=algo-1, epoch=1, train binary_classification_accuracy <score>=0.56732967033
```

```
[04/28/2020 14:09:36 INFO 140478314145600] #quality_metric: host=algo-1, epoch=1, train binary_classification_cross_entropy <loss>=0.673812549633
```

```
[04/28/2020 14:09:36 INFO 140478314145600] #quality_metric: host=algo-1, epoch=1, train binary_f1.000 <score>=0.710805232579
```

```
#metrics {"Metrics": {"update.time": {"count": 1, "max": 1147.292137145996, "sum": 1147.292137145996, "min": 1147.292137145996}}, "EndTime": 1588082976.272978, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588082975.124228}
```

```
[04/28/2020 14:09:36 INFO 140478314145600] #progress_metric: host=algo-1, completed 4 % of epochs
```

```
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 183, "sum": 183.0, "min": 183}, "Total Records Seen": {"count": 1, "max": 182140, "sum": 182140.0, "min": 182140}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 3, "sum": 3.0, "min": 3}}, "EndTime": 1588082976.27371, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 1}, "StartTime": 1588082975.125653}
```

```
[04/28/2020 14:09:36 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=78858.6966038 records/second
```

```
[04/28/2020 14:09:36 INFO 140478314145600] #quality_metric: host=algo-1, epoch=2, batch=0 train binary_classification_accuracy <score>=0.723
```

```
[04/28/2020 14:09:36 INFO 140478314145600] #quality_metric: host=algo-1, epoch=2, batch=0 train binary_classification_cross_entropy <loss>=0.670300964355
```

```
[04/28/2020 14:09:36 INFO 140478314145600] #quality_metric: host=algo-1, epoch=2, batch=0 train binary_f1.000 <score>=0.752014324082
```

```
[2020-04-28 14:09:37.361] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 5, "duration": 1144, "num_examples": 91, "num_bytes": 5796480}
```

```

ine": "/opt/ml/input/data/train", "epoch": 6, "duration": 1084, "num
_examples": 91, "num_bytes": 5796480}
[04/28/2020 14:09:37 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=2, train binary_classification_accuracy <score>=0.5960109
89011
[04/28/2020 14:09:37 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=2, train binary_classification_cross_entropy <loss>=0.664
654704
[04/28/2020 14:09:37 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=2, train binary_f_1.000 <score>=0.720703199192
#metrics {"Metrics": {"update.time": {"count": 1, "max": 1087.226152
420044, "sum": 1087.226152420044, "min": 1087.226152420044}}, "EndTi
me": 1588082977.362158, "Dimensions": {"Host": "algo-1", "Operatio
n": "training", "Algorithm": "factorization-machines"}, "StartTime":
1588082976.273371}

```

```

[04/28/2020 14:09:37 INFO 140478314145600] #progress_metric: host=al
go-1, completed 6 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count":
1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Las
t Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number o
f Records Since Last Reset": {"count": 1, "max": 90570, "sum": 9057
0.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 274,
"sum": 274.0, "min": 274}, "Total Records Seen": {"count": 1, "ma
x": 272710, "sum": 272710.0, "min": 272710}, "Max Records Seen Betwe
en Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 9057
0}, "Reset Count": {"count": 1, "max": 4, "sum": 4.0, "min": 4}}, "E
ndTime": 1588082977.3624, "Dimensions": {"Host": "algo-1", "Meta":
"training_data_iter", "Operation": "training", "Algorithm": "factor
ization-machines", "epoch": 2}, "StartTime": 1588082976.274898}

```

```

[04/28/2020 14:09:37 INFO 140478314145600] #throughput_metric: host=
algo-1, train throughput=83272.199223 records/second

```

```

[04/28/2020 14:09:37 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=3, batch=0 train binary_classification_accuracy <score>=
0.724

```

```

[04/28/2020 14:09:37 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=3, batch=0 train binary_classification_cross_entropy <los
s>=0.660577514648

```

```

[04/28/2020 14:09:37 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=3, batch=0 train binary_f_1.000 <score>=0.752688172043

```

```

[2020-04-28 14:09:38.372] [tensorio] [info] epoch_stats={"data_pipel
ine": "/opt/ml/input/data/train", "epoch": 8, "duration": 1007, "num
_examples": 91, "num_bytes": 5796480}

```

```

[04/28/2020 14:09:38 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=3, train binary_classification_accuracy <score>=0.6285164
83516

```

```

[04/28/2020 14:09:38 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=3, train binary_classification_cross_entropy <loss>=0.655
966228779

```

```

[04/28/2020 14:09:38 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=3, train binary_f_1.000 <score>=0.734172636413

```

```

#metrics {"Metrics": {"update.time": {"count": 1, "max": 1009.929180
1452637, "sum": 1009.9291801452637, "min": 1009.9291801452637}}, "En
dTime": 1588082978.373122, "Dimensions": {"Host": "algo-1", "Operati
on": "training", "Algorithm": "factorization-machines"}, "StartTim
e": 1588082977.362233}

```

```

[04/28/2020 14:09:38 INFO 140478314145600] #progress_metric: host=al
go-1, completed 8 % of epochs

```

```

#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count":

```



```
1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 365, "sum": 365.0, "min": 365}, "Total Records Seen": {"count": 1, "max": 363280, "sum": 363280.0, "min": 363280}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 5, "sum": 5.0, "min": 5}}, "EndTime": 1588082978.373486, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 3}, "StartTime": 1588082977.363164}
```

```
[04/28/2020 14:09:38 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=89634.6530403 records/second
```

```
[04/28/2020 14:09:38 INFO 140478314145600] #quality_metric: host=algo-1, epoch=4, batch=0 train binary_classification_accuracy <score>=0.73
```

```
[04/28/2020 14:09:38 INFO 140478314145600] #quality_metric: host=algo-1, epoch=4, batch=0 train binary_classification_cross_entropy <loss>=0.651498046875
```

```
[04/28/2020 14:09:38 INFO 140478314145600] #quality_metric: host=algo-1, epoch=4, batch=0 train binary_f1.000 <score>=0.758064516129
```

```
[2020-04-28 14:09:39.262] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 10, "duration": 887, "num_examples": 91, "num_bytes": 5796480}
```

```
[04/28/2020 14:09:39 INFO 140478314145600] #quality_metric: host=algo-1, epoch=4, train binary_classification_accuracy <score>=0.652912087912
```

```
[04/28/2020 14:09:39 INFO 140478314145600] #quality_metric: host=algo-1, epoch=4, train binary_classification_cross_entropy <loss>=0.647807986081
```

```
[04/28/2020 14:09:39 INFO 140478314145600] #quality_metric: host=algo-1, epoch=4, train binary_f1.000 <score>=0.743696899369
```

```
#metrics {"Metrics": {"update.time": {"count": 1, "max": 889.1308307647705, "sum": 889.1308307647705, "min": 889.1308307647705}}, "EndTime": 1588082979.263372, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588082978.37319}
```

```
[04/28/2020 14:09:39 INFO 140478314145600] #progress_metric: host=algo-1, completed 10 % of epochs
```

```
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 456, "sum": 456.0, "min": 456}, "Total Records Seen": {"count": 1, "max": 453850, "sum": 453850.0, "min": 453850}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 6, "sum": 6.0, "min": 6}}, "EndTime": 1588082979.263683, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 4}, "StartTime": 1588082978.374184}
```

```
[04/28/2020 14:09:39 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=101808.542623 records/second
```

```
[04/28/2020 14:09:39 INFO 140478314145600] #quality_metric: host=algo-1, epoch=5, batch=0 train binary_classification_accuracy <score>=0.727
```

```
[04/28/2020 14:09:39 INFO 140478314145600] #quality_metric: host=algo-1, epoch=5, batch=0 train binary_classification_cross_entropy <loss>=
```

```
s>=0.643096679687
[04/28/2020 14:09:39 INFO 140478314145600] #quality_metric: host=algo-1, epoch=5, batch=0 train binary_f_1.000 <score>=0.752941176471
[2020-04-28 14:09:40.242] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 12, "duration": 976, "num_examples": 91, "num_bytes": 5796480}
[04/28/2020 14:09:40 INFO 140478314145600] #quality_metric: host=algo-1, epoch=5, train binary_classification_accuracy <score>=0.66989010989
[04/28/2020 14:09:40 INFO 140478314145600] #quality_metric: host=algo-1, epoch=5, train binary_classification_cross_entropy <loss>=0.640211695409
[04/28/2020 14:09:40 INFO 140478314145600] #quality_metric: host=algo-1, epoch=5, train binary_f_1.000 <score>=0.749870938733
#metrics {"Metrics": {"update.time": {"count": 1, "max": 978.5070419311523, "sum": 978.5070419311523, "min": 978.5070419311523}}, "EndTime": 1588082980.242867, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588082979.263441}

[04/28/2020 14:09:40 INFO 140478314145600] #progress_metric: host=algo-1, completed 12 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 547, "sum": 547.0, "min": 547}, "Total Records Seen": {"count": 1, "max": 544420, "sum": 544420.0, "min": 544420}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 7, "sum": 7.0, "min": 7}}, "EndTime": 1588082980.243201, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 5}, "StartTime": 1588082979.26433}

[04/28/2020 14:09:40 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=92509.9208007 records/second
[04/28/2020 14:09:40 INFO 140478314145600] #quality_metric: host=algo-1, epoch=6, batch=0 train binary_classification_accuracy <score>=0.731
[04/28/2020 14:09:40 INFO 140478314145600] #quality_metric: host=algo-1, epoch=6, batch=0 train binary_classification_cross_entropy <loss>=0.63535144043
[04/28/2020 14:09:40 INFO 140478314145600] #quality_metric: host=algo-1, epoch=6, batch=0 train binary_f_1.000 <score>=0.755676657584
[2020-04-28 14:09:41.174] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 14, "duration": 928, "num_examples": 91, "num_bytes": 5796480}
[04/28/2020 14:09:41 INFO 140478314145600] #quality_metric: host=algo-1, epoch=6, train binary_classification_accuracy <score>=0.682494505495
[04/28/2020 14:09:41 INFO 140478314145600] #quality_metric: host=algo-1, epoch=6, train binary_classification_cross_entropy <loss>=0.633175523024
[04/28/2020 14:09:41 INFO 140478314145600] #quality_metric: host=algo-1, epoch=6, train binary_f_1.000 <score>=0.75442607624
#metrics {"Metrics": {"update.time": {"count": 1, "max": 930.992841720581, "sum": 930.992841720581, "min": 930.992841720581}}, "EndTime": 1588082981.174922, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588082980.242978}
```

```
[04/28/2020 14:09:41 INFO 140478314145600] #progress_metric: host=algo-1, completed 14 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 638, "sum": 638.0, "min": 638}, "Total Records Seen": {"count": 1, "max": 634990, "sum": 634990.0, "min": 634990}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 8, "sum": 8.0, "min": 8}}, "EndTime": 1588082981.175222, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 6}, "StartTime": 1588082980.243897}
```

```
[04/28/2020 14:09:41 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=97232.6921544 records/second
```

```
[04/28/2020 14:09:41 INFO 140478314145600] #quality_metric: host=algo-1, epoch=7, batch=0 train binary_classification_accuracy <score>=0.734
```

```
[04/28/2020 14:09:41 INFO 140478314145600] #quality_metric: host=algo-1, epoch=7, batch=0 train binary_classification_cross_entropy <loss>=0.628227783203
```

```
[04/28/2020 14:09:41 INFO 140478314145600] #quality_metric: host=algo-1, epoch=7, batch=0 train binary_f1.000 <score>=0.75641025641
```

```
[2020-04-28 14:09:42.027] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 16, "duration": 850, "num_examples": 91, "num_bytes": 5796480}
```

```
[04/28/2020 14:09:42 INFO 140478314145600] #quality_metric: host=algo-1, epoch=7, train binary_classification_accuracy <score>=0.692010989011
```

```
[04/28/2020 14:09:42 INFO 140478314145600] #quality_metric: host=algo-1, epoch=7, train binary_classification_cross_entropy <loss>=0.626676928174
```

```
[04/28/2020 14:09:42 INFO 140478314145600] #quality_metric: host=algo-1, epoch=7, train binary_f1.000 <score>=0.757667199862
```

```
#metrics {"Metrics": {"update.time": {"count": 1, "max": 852.5691032409668, "sum": 852.5691032409668, "min": 852.5691032409668}}, "EndTime": 1588082982.02851, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588082981.175018}
```

```
[04/28/2020 14:09:42 INFO 140478314145600] #progress_metric: host=algo-1, completed 16 % of epochs
```

```
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 729, "sum": 729.0, "min": 729}, "Total Records Seen": {"count": 1, "max": 725560, "sum": 725560.0, "min": 725560}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 9, "sum": 9.0, "min": 9}}, "EndTime": 1588082982.028765, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 7}, "StartTime": 1588082981.17591}
```

```
[04/28/2020 14:09:42 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=106179.68776 records/second
```

```
[04/28/2020 14:09:42 INFO 140478314145600] #quality_metric: host=algo-1, epoch=7, batch=0 train binary_classification_accuracy <score>=0.734
```



```

o-1, epoch=8, batch=0 train binary_classification_accuracy <score>=
0.737
[04/28/2020 14:09:42 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=8, batch=0 train binary_classification_cross_entropy <los
s>=0.621684814453
[04/28/2020 14:09:42 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=8, batch=0 train binary_f_1.000 <score>=0.758494031221
[2020-04-28 14:09:43.001] [tensorio] [info] epoch_stats={"data_pipel
ine": "/opt/ml/input/data/train", "epoch": 18, "duration": 970, "num
_examples": 91, "num_bytes": 5796480}
[04/28/2020 14:09:43 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=8, train binary_classification_accuracy <score>=0.6994505
49451
[04/28/2020 14:09:43 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=8, train binary_classification_cross_entropy <loss>=0.620
682972667
[04/28/2020 14:09:43 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=8, train binary_f_1.000 <score>=0.760407176396
#metrics {"Metrics": {"update.time": {"count": 1, "max": 972.9969501
495361, "sum": 972.9969501495361, "min": 972.9969501495361}}, "EndTi
me": 1588082983.002448, "Dimensions": {"Host": "algo-1", "Operatio
n": "training", "Algorithm": "factorization-machines"}, "StartTime":
1588082982.028594}

[04/28/2020 14:09:43 INFO 140478314145600] #progress_metric: host=al
go-1, completed 18 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count":
1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Las
t Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number o
f Records Since Last Reset": {"count": 1, "max": 90570, "sum": 9057
0.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 820,
"sum": 820.0, "min": 820}, "Total Records Seen": {"count": 1, "ma
x": 816130, "sum": 816130.0, "min": 816130}, "Max Records Seen Betwe
en Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 9057
0}, "Reset Count": {"count": 1, "max": 10, "sum": 10.0, "min": 10}},
"EndTime": 1588082983.002804, "Dimensions": {"Host": "algo-1", "Met
a": "training_data_iter", "Operation": "training", "Algorithm": "fac
torization-machines", "epoch": 8}, "StartTime": 1588082982.02942}

[04/28/2020 14:09:43 INFO 140478314145600] #throughput_metric: host=
algo-1, train throughput=93033.7290038 records/second
[04/28/2020 14:09:43 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=9, batch=0 train binary_classification_accuracy <score>=
0.736
[04/28/2020 14:09:43 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=9, batch=0 train binary_classification_cross_entropy <los
s>=0.615678344727
[04/28/2020 14:09:43 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=9, batch=0 train binary_f_1.000 <score>=0.756906077348
[2020-04-28 14:09:43.867] [tensorio] [info] epoch_stats={"data_pipel
ine": "/opt/ml/input/data/train", "epoch": 20, "duration": 862, "num
_examples": 91, "num_bytes": 5796480}
[04/28/2020 14:09:43 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=9, train binary_classification_accuracy <score>=0.7054175
82418
[04/28/2020 14:09:43 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=9, train binary_classification_cross_entropy <loss>=0.615
156304999
[04/28/2020 14:09:43 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=9, train binary_f_1.000 <score>=0.762246011122
#metrics {"Metrics": {"update.time": {"count": 1, "max": 864.1879558

```

```
563232, "sum": 864.1879558563232, "min": 864.1879558563232}}, "EndTime": 1588082983.867716, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588082983.002529}
```

```
[04/28/2020 14:09:43 INFO 140478314145600] #progress_metric: host=algo-1, completed 20 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 911, "sum": 911.0, "min": 911}, "Total Records Seen": {"count": 1, "max": 906700, "sum": 906700.0, "min": 906700}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 11, "sum": 11.0, "min": 11}}, "EndTime": 1588082983.867974, "Dimensions": {"Host": "algo-1", "Metric": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 9}, "StartTime": 1588082983.003498}
```

```
[04/28/2020 14:09:43 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=104752.928034 records/second
[04/28/2020 14:09:43 INFO 140478314145600] #quality_metric: host=algo-1, epoch=10, batch=0 train binary_classification_accuracy <score>=0.733
[04/28/2020 14:09:43 INFO 140478314145600] #quality_metric: host=algo-1, epoch=10, batch=0 train binary_classification_cross_entropy <loss>=0.610163635254
[04/28/2020 14:09:43 INFO 140478314145600] #quality_metric: host=algo-1, epoch=10, batch=0 train binary_f_1.000 <score>=0.753006475486
```

```
2020-04-28 14:09:32 Training - Training image download completed. Training in progress.[2020-04-28 14:09:44.764] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 22, "duration": 893, "num_examples": 91, "num_bytes": 5796480}
[04/28/2020 14:09:44 INFO 140478314145600] #quality_metric: host=algo-1, epoch=10, train binary_classification_accuracy <score>=0.709252747253
[04/28/2020 14:09:44 INFO 140478314145600] #quality_metric: host=algo-1, epoch=10, train binary_classification_cross_entropy <loss>=0.610058519301
[04/28/2020 14:09:44 INFO 140478314145600] #quality_metric: host=algo-1, epoch=10, train binary_f_1.000 <score>=0.763163077143
#metrics {"Metrics": {"update.time": {"count": 1, "max": 896.2259292602539, "sum": 896.2259292602539, "min": 896.2259292602539}}, "EndTime": 1588082984.764859, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588082983.867796}
```

```
[04/28/2020 14:09:44 INFO 140478314145600] #progress_metric: host=algo-1, completed 22 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 1002, "sum": 1002.0, "min": 1002}, "Total Records Seen": {"count": 1, "max": 997270, "sum": 997270.0, "min": 997270}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 12, "sum": 12.0, "min": 12}}, "EndTime": 1588082984.765207, "Dimensions": {"Host": "algo-1", "Metric": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 10}, "StartTime": 1588082983.003498}
```

```
a": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 10, "StartTime": 1588082983.868602}
```

```
[04/28/2020 14:09:44 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=100998.400597 records/second
```

```
[04/28/2020 14:09:44 INFO 140478314145600] #quality_metric: host=algo-1, epoch=11, batch=0 train binary_classification_accuracy <score>=0.732
```

```
[04/28/2020 14:09:44 INFO 140478314145600] #quality_metric: host=algo-1, epoch=11, batch=0 train binary_classification_cross_entropy <loss>=0.605097167969
```

```
[04/28/2020 14:09:44 INFO 140478314145600] #quality_metric: host=algo-1, epoch=11, batch=0 train binary_f_1.000 <score>=0.751851851852
```

```
[2020-04-28 14:09:45.729] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 24, "duration": 961, "num_examples": 91, "num_bytes": 5796480}
```

```
[04/28/2020 14:09:45 INFO 140478314145600] #quality_metric: host=algo-1, epoch=11, train binary_classification_accuracy <score>=0.712296703297
```

```
[04/28/2020 14:09:45 INFO 140478314145600] #quality_metric: host=algo-1, epoch=11, train binary_classification_cross_entropy <loss>=0.605352084988
```

```
[04/28/2020 14:09:45 INFO 140478314145600] #quality_metric: host=algo-1, epoch=11, train binary_f_1.000 <score>=0.763758425598
```

```
#metrics {"Metrics": {"update.time": {"count": 1, "max": 963.8688564300537, "sum": 963.8688564300537, "min": 963.8688564300537}}, "EndTime": 1588082985.729806, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588082984.764962}
```

```
[04/28/2020 14:09:45 INFO 140478314145600] #progress_metric: host=algo-1, completed 24 % of epochs
```

```
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 1093, "sum": 1093.0, "min": 1093}, "Total Records Seen": {"count": 1, "max": 1087840, "sum": 1087840.0, "min": 1087840}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 13, "sum": 13.0, "min": 13}}, "EndTime": 1588082985.730118, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 11}, "StartTime": 1588082984.765906}
```

```
[04/28/2020 14:09:45 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=93916.8147435 records/second
```

```
[04/28/2020 14:09:45 INFO 140478314145600] #quality_metric: host=algo-1, epoch=12, batch=0 train binary_classification_accuracy <score>=0.73
```

```
[04/28/2020 14:09:45 INFO 140478314145600] #quality_metric: host=algo-1, epoch=12, batch=0 train binary_classification_cross_entropy <loss>=0.60043737793
```

```
[04/28/2020 14:09:45 INFO 140478314145600] #quality_metric: host=algo-1, epoch=12, batch=0 train binary_f_1.000 <score>=0.749536178108
```

```
[2020-04-28 14:09:46.621] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 26, "duration": 889, "num_examples": 91, "num_bytes": 5796480}
```

```
[04/28/2020 14:09:46 INFO 140478314145600] #quality_metric: host=algo-1, epoch=12, train binary_classification_accuracy <score>=0.714879
```

120879

```
[04/28/2020 14:09:46 INFO 140478314145600] #quality_metric: host=algo-1, epoch=12, train binary_classification_cross_entropy <loss>=0.601001334054
```

```
[04/28/2020 14:09:46 INFO 140478314145600] #quality_metric: host=algo-1, epoch=12, train binary_f_1.000 <score>=0.76428156116
#metrics {"Metrics": {"update.time": {"count": 1, "max": 891.2620544433594, "sum": 891.2620544433594, "min": 891.2620544433594}}, "EndTime": 1588082986.622149, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588082985.729897}
```

```
[04/28/2020 14:09:46 INFO 140478314145600] #progress_metric: host=algo-1, completed 26 % of epochs
```

```
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 1184, "sum": 1184.0, "min": 1184}, "Total Records Seen": {"count": 1, "max": 1178410, "sum": 1178410.0, "min": 1178410}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 14, "sum": 14.0, "min": 14}}, "EndTime": 1588082986.622471, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 12}, "StartTime": 1588082985.730855}
```

```
[04/28/2020 14:09:46 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=101563.106775 records/second
```

```
[04/28/2020 14:09:46 INFO 140478314145600] #quality_metric: host=algo-1, epoch=13, batch=0 train binary_classification_accuracy <score>=0.732
```

```
[04/28/2020 14:09:46 INFO 140478314145600] #quality_metric: host=algo-1, epoch=13, batch=0 train binary_classification_cross_entropy <loss>=0.596145629883
```

```
[04/28/2020 14:09:46 INFO 140478314145600] #quality_metric: host=algo-1, epoch=13, batch=0 train binary_f_1.000 <score>=0.75092936803
```

```
[2020-04-28 14:09:47.604] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 28, "duration": 979, "num_examples": 91, "num_bytes": 5796480}
```

```
[04/28/2020 14:09:47 INFO 140478314145600] #quality_metric: host=algo-1, epoch=13, train binary_classification_accuracy <score>=0.717241758242
```

```
[04/28/2020 14:09:47 INFO 140478314145600] #quality_metric: host=algo-1, epoch=13, train binary_classification_cross_entropy <loss>=0.596973035205
```

```
[04/28/2020 14:09:47 INFO 140478314145600] #quality_metric: host=algo-1, epoch=13, train binary_f_1.000 <score>=0.764912793619
```

```
#metrics {"Metrics": {"update.time": {"count": 1, "max": 981.7090034484863, "sum": 981.7090034484863, "min": 981.7090034484863}}, "EndTime": 1588082987.604905, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588082986.622246}
```

```
[04/28/2020 14:09:47 INFO 140478314145600] #progress_metric: host=algo-1, completed 28 % of epochs
```

```
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 1184, "sum": 1184.0, "min": 1184}, "Total Records Seen": {"count": 1, "max": 1178410, "sum": 1178410.0, "min": 1178410}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 14, "sum": 14.0, "min": 14}}, "EndTime": 1588082987.604905, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 13}, "StartTime": 1588082986.622246}
```



```
0.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 1275,
"sum": 1275.0, "min": 1275}, "Total Records Seen": {"count": 1, "ma
x": 1268980, "sum": 1268980.0, "min": 1268980}, "Max Records Seen Be
tween Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 905
70}, "Reset Count": {"count": 1, "max": 15, "sum": 15.0, "min": 1
5}}, "EndTime": 1588082987.605245, "Dimensions": {"Host": "algo-1",
"Meta": "training_data_iter", "Operation": "training", "Algorithm":
"factorization-machines", "epoch": 13}, "StartTime": 1588082986.6231
64}
```

```
[04/28/2020 14:09:47 INFO 140478314145600] #throughput_metric: host=
algo-1, train throughput=92209.669849 records/second
```

```
[04/28/2020 14:09:47 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=14, batch=0 train binary_classification_accuracy <score>=
0.729
```

```
[04/28/2020 14:09:47 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=14, batch=0 train binary_classification_cross_entropy <lo
ss>=0.592186157227
```

```
[04/28/2020 14:09:47 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=14, batch=0 train binary_f_1.000 <score>=0.747437092265
```

```
[2020-04-28 14:09:48.557] [tensorio] [info] epoch_stats={"data_pipel
ine": "/opt/ml/input/data/train", "epoch": 30, "duration": 949, "num
_examples": 91, "num_bytes": 5796480}
```

```
[04/28/2020 14:09:48 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=14, train binary_classification_accuracy <score>=0.723758
241758
```

```
[04/28/2020 14:09:48 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=14, train binary_classification_cross_entropy <loss>=0.59
3236636654
```

```
[04/28/2020 14:09:48 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=14, train binary_f_1.000 <score>=0.768279193245
```

```
#metrics {"Metrics": {"update.time": {"count": 1, "max": 951.9059658
050537, "sum": 951.9059658050537, "min": 951.9059658050537}}, "EndTi
me": 1588082988.557912, "Dimensions": {"Host": "algo-1", "Operatio
n": "training", "Algorithm": "factorization-machines"}, "StartTime":
1588082987.605}
```

```
[04/28/2020 14:09:48 INFO 140478314145600] #progress_metric: host=al
go-1, completed 30 % of epochs
```

```
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count":
1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Las
t Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number o
f Records Since Last Reset": {"count": 1, "max": 90570, "sum": 9057
0.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 1366,
"sum": 1366.0, "min": 1366}, "Total Records Seen": {"count": 1, "ma
x": 1359550, "sum": 1359550.0, "min": 1359550}, "Max Records Seen Be
tween Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 905
70}, "Reset Count": {"count": 1, "max": 16, "sum": 16.0, "min": 1
6}}, "EndTime": 1588082988.558285, "Dimensions": {"Host": "algo-1",
"Meta": "training_data_iter", "Operation": "training", "Algorithm":
"factorization-machines", "epoch": 14}, "StartTime": 1588082987.6059
39}
```

```
[04/28/2020 14:09:48 INFO 140478314145600] #throughput_metric: host=
algo-1, train throughput=95090.0073593 records/second
```

```
[04/28/2020 14:09:48 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=15, batch=0 train binary_classification_accuracy <score>=
0.733
```

```
[04/28/2020 14:09:48 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=15, batch=0 train binary_classification_cross_entropy <lo
ss>=0.588526367187
```

```
[04/28/2020 14:09:48 INFO 140478314145600] #quality_metric: host=algo-1, epoch=15, batch=0 train binary_f_1.000 <score>=0.751164958062
[2020-04-28 14:09:49.405] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 32, "duration": 845, "num_examples": 91, "num_bytes": 5796480}
[04/28/2020 14:09:49 INFO 140478314145600] #quality_metric: host=algo-1, epoch=15, train binary_classification_accuracy <score>=0.725835164835
[04/28/2020 14:09:49 INFO 140478314145600] #quality_metric: host=algo-1, epoch=15, train binary_classification_cross_entropy <loss>=0.589764260093
[04/28/2020 14:09:49 INFO 140478314145600] #quality_metric: host=algo-1, epoch=15, train binary_f_1.000 <score>=0.769052754353
#metrics {"Metrics": {"update.time": {"count": 1, "max": 847.3050594329834, "sum": 847.3050594329834, "min": 847.3050594329834}}, "EndTime": 1588082989.40634, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588082988.557991}

[04/28/2020 14:09:49 INFO 140478314145600] #progress_metric: host=algo-1, completed 32 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 1457, "sum": 1457.0, "min": 1457}, "Total Records Seen": {"count": 1, "max": 1450120, "sum": 1450120.0, "min": 1450120}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 17, "sum": 17.0, "min": 17}}, "EndTime": 1588082989.406649, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 15}, "StartTime": 1588082988.559004}

[04/28/2020 14:09:49 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=106829.709223 records/second
[04/28/2020 14:09:49 INFO 140478314145600] #quality_metric: host=algo-1, epoch=16, batch=0 train binary_classification_accuracy <score>=0.734
[04/28/2020 14:09:49 INFO 140478314145600] #quality_metric: host=algo-1, epoch=16, batch=0 train binary_classification_cross_entropy <loss>=0.58513659668
[04/28/2020 14:09:49 INFO 140478314145600] #quality_metric: host=algo-1, epoch=16, batch=0 train binary_f_1.000 <score>=0.751865671642
[2020-04-28 14:09:50.267] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 34, "duration": 858, "num_examples": 91, "num_bytes": 5796480}
[04/28/2020 14:09:50 INFO 140478314145600] #quality_metric: host=algo-1, epoch=16, train binary_classification_accuracy <score>=0.727131868132
[04/28/2020 14:09:50 INFO 140478314145600] #quality_metric: host=algo-1, epoch=16, train binary_classification_cross_entropy <loss>=0.586530614832
[04/28/2020 14:09:50 INFO 140478314145600] #quality_metric: host=algo-1, epoch=16, train binary_f_1.000 <score>=0.769230769231
#metrics {"Metrics": {"update.time": {"count": 1, "max": 860.5990409851074, "sum": 860.5990409851074, "min": 860.5990409851074}}, "EndTime": 1588082990.268006, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588082989.406428}
```

```
[04/28/2020 14:09:50 INFO 140478314145600] #progress_metric: host=algo-1, completed 34 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 1548, "sum": 1548.0, "min": 1548}, "Total Records Seen": {"count": 1, "max": 1540690, "sum": 1540690.0, "min": 1540690}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 18, "sum": 18.0, "min": 18}}, "EndTime": 1588082990.268295, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 16}, "StartTime": 1588082989.407359}
```

```
[04/28/2020 14:09:50 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=105179.543236 records/second
```

```
[04/28/2020 14:09:50 INFO 140478314145600] #quality_metric: host=algo-1, epoch=17, batch=0 train binary_classification_accuracy <score>=0.733
```

```
[04/28/2020 14:09:50 INFO 140478314145600] #quality_metric: host=algo-1, epoch=17, batch=0 train binary_classification_cross_entropy <loss>=0.58199005127
```

```
[04/28/2020 14:09:50 INFO 140478314145600] #quality_metric: host=algo-1, epoch=17, batch=0 train binary_f_1.000 <score>=0.751164958062
```

```
[2020-04-28 14:09:51.127] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 36, "duration": 856, "num_examples": 91, "num_bytes": 5796480}
```

```
[04/28/2020 14:09:51 INFO 140478314145600] #quality_metric: host=algo-1, epoch=17, train binary_classification_accuracy <score>=0.728637362637
```

```
[04/28/2020 14:09:51 INFO 140478314145600] #quality_metric: host=algo-1, epoch=17, train binary_classification_cross_entropy <loss>=0.583512887137
```

```
[04/28/2020 14:09:51 INFO 140478314145600] #quality_metric: host=algo-1, epoch=17, train binary_f_1.000 <score>=0.769667008675
```

```
#metrics {"Metrics": {"update.time": {"count": 1, "max": 858.8268756866455, "sum": 858.8268756866455, "min": 858.8268756866455}}, "EndTime": 1588082991.127842, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588082990.268071}
```

```
[04/28/2020 14:09:51 INFO 140478314145600] #progress_metric: host=algo-1, completed 36 % of epochs
```

```
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 1639, "sum": 1639.0, "min": 1639}, "Total Records Seen": {"count": 1, "max": 1631260, "sum": 1631260.0, "min": 1631260}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 19, "sum": 19.0, "min": 19}}, "EndTime": 1588082991.128099, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 17}, "StartTime": 1588082990.268983}
```

```
[04/28/2020 14:09:51 INFO 140478314145600] #throughput_metric: host=
```

```
algo-1, train throughput=105406.482992 records/second
[04/28/2020 14:09:51 INFO 140478314145600] #quality_metric: host=algo-1, epoch=18, batch=0 train binary_classification_accuracy <score>=0.733
[04/28/2020 14:09:51 INFO 140478314145600] #quality_metric: host=algo-1, epoch=18, batch=0 train binary_classification_cross_entropy <loss>=0.579062805176
[04/28/2020 14:09:51 INFO 140478314145600] #quality_metric: host=algo-1, epoch=18, batch=0 train binary_f_1.000 <score>=0.751627906977
[2020-04-28 14:09:51.991] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 38, "duration": 861, "num_examples": 91, "num_bytes": 5796480}
[04/28/2020 14:09:51 INFO 140478314145600] #quality_metric: host=algo-1, epoch=18, train binary_classification_accuracy <score>=0.729494505495
[04/28/2020 14:09:51 INFO 140478314145600] #quality_metric: host=algo-1, epoch=18, train binary_classification_cross_entropy <loss>=0.580690551758
[04/28/2020 14:09:51 INFO 140478314145600] #quality_metric: host=algo-1, epoch=18, train binary_f_1.000 <score>=0.769620963968
#metrics {"Metrics": {"update.time": {"count": 1, "max": 863.9659881591797, "sum": 863.9659881591797, "min": 863.9659881591797}}, "EndTime": 1588082991.992745, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588082991.127927}

[04/28/2020 14:09:51 INFO 140478314145600] #progress_metric: host=algo-1, completed 38 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 1730, "sum": 1730.0, "min": 1730}, "Total Records Seen": {"count": 1, "max": 1721830, "sum": 1721830.0, "min": 1721830}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 20, "sum": 20.0, "min": 20}}, "EndTime": 1588082991.993072, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 18}, "StartTime": 1588082991.128744}

[04/28/2020 14:09:51 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=104772.776506 records/second
[04/28/2020 14:09:52 INFO 140478314145600] #quality_metric: host=algo-1, epoch=19, batch=0 train binary_classification_accuracy <score>=0.735
[04/28/2020 14:09:52 INFO 140478314145600] #quality_metric: host=algo-1, epoch=19, batch=0 train binary_classification_cross_entropy <loss>=0.576333129883
[04/28/2020 14:09:52 INFO 140478314145600] #quality_metric: host=algo-1, epoch=19, batch=0 train binary_f_1.000 <score>=0.753488372093
[2020-04-28 14:09:52.844] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 40, "duration": 849, "num_examples": 91, "num_bytes": 5796480}
[04/28/2020 14:09:52 INFO 140478314145600] #quality_metric: host=algo-1, epoch=19, train binary_classification_accuracy <score>=0.730714285714
[04/28/2020 14:09:52 INFO 140478314145600] #quality_metric: host=algo-1, epoch=19, train binary_classification_cross_entropy <loss>=0.578045166686
```



```
[04/28/2020 14:09:52 INFO 140478314145600] #quality_metric: host=algo-1, epoch=19, train binary_f_1.000 <score>=0.770046450523
#metrics {"Metrics": {"update.time": {"count": 1, "max": 851.5119552612305, "sum": 851.5119552612305, "min": 851.5119552612305}}, "EndTime": 1588082992.84532, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588082991.992811}
```

```
[04/28/2020 14:09:52 INFO 140478314145600] #progress_metric: host=algo-1, completed 40 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 1821, "sum": 1821.0, "min": 1821}, "Total Records Seen": {"count": 1, "max": 1812400, "sum": 1812400.0, "min": 1812400}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 21, "sum": 21.0, "min": 21}}, "EndTime": 1588082992.845682, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 19}, "StartTime": 1588082991.993778}
```

```
[04/28/2020 14:09:52 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=106301.329803 records/second
```

```
[04/28/2020 14:09:52 INFO 140478314145600] #quality_metric: host=algo-1, epoch=20, batch=0 train binary_classification_accuracy <score>=0.734
```

```
[04/28/2020 14:09:52 INFO 140478314145600] #quality_metric: host=algo-1, epoch=20, batch=0 train binary_classification_cross_entropy <loss>=0.573781677246
```

```
[04/28/2020 14:09:52 INFO 140478314145600] #quality_metric: host=algo-1, epoch=20, batch=0 train binary_f_1.000 <score>=0.752327746741
```

```
[2020-04-28 14:09:53.776] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 42, "duration": 928, "num_examples": 91, "num_bytes": 5796480}
```

```
[04/28/2020 14:09:53 INFO 140478314145600] #quality_metric: host=algo-1, epoch=20, train binary_classification_accuracy <score>=0.731505494505
```

```
[04/28/2020 14:09:53 INFO 140478314145600] #quality_metric: host=algo-1, epoch=20, train binary_classification_cross_entropy <loss>=0.575560181335
```

```
[04/28/2020 14:09:53 INFO 140478314145600] #quality_metric: host=algo-1, epoch=20, train binary_f_1.000 <score>=0.770178623499
```

```
#metrics {"Metrics": {"update.time": {"count": 1, "max": 930.8340549468994, "sum": 930.8340549468994, "min": 930.8340549468994}}, "EndTime": 1588082993.77727, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588082992.845382}
```

```
[04/28/2020 14:09:53 INFO 140478314145600] #progress_metric: host=algo-1, completed 42 % of epochs
```

```
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 1912, "sum": 1912.0, "min": 1912}, "Total Records Seen": {"count": 1, "max": 1902970, "sum": 1902970.0, "min": 1902970}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 905
```

```
70}, "Reset Count": {"count": 1, "max": 22, "sum": 22.0, "min": 2
2}}, "EndTime": 1588082993.777648, "Dimensions": {"Host": "algo-1",
"Meta": "training_data_iter", "Operation": "training", "Algorithm":
"factorization-machines", "epoch": 20}, "StartTime": 1588082992.8463
87}
```

```
[04/28/2020 14:09:53 INFO 140478314145600] #throughput_metric: host=
algo-1, train throughput=97237.0476527 records/second
```

```
[04/28/2020 14:09:53 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=21, batch=0 train binary_classification_accuracy <score>=
0.738
```

```
[04/28/2020 14:09:53 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=21, batch=0 train binary_classification_cross_entropy <lo
ss>=0.571391174316
```

```
[04/28/2020 14:09:53 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=21, batch=0 train binary_f_1.000 <score>=0.755597014925
```

```
[2020-04-28 14:09:54.673] [tensorio] [info] epoch_stats={"data_pipel
ine": "/opt/ml/input/data/train", "epoch": 44, "duration": 893, "num
_examples": 91, "num_bytes": 5796480}
```

```
[04/28/2020 14:09:54 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=21, train binary_classification_accuracy <score>=0.732340
659341
```

```
[04/28/2020 14:09:54 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=21, train binary_classification_cross_entropy <loss>=0.57
3220805745
```

```
[04/28/2020 14:09:54 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=21, train binary_f_1.000 <score>=0.770401093463
```

```
#metrics {"Metrics": {"update.time": {"count": 1, "max": 895.8101272
583008, "sum": 895.8101272583008, "min": 895.8101272583008}}, "EndTi
me": 1588082994.674255, "Dimensions": {"Host": "algo-1", "Operatio
n": "training", "Algorithm": "factorization-machines"}, "StartTime":
1588082993.777332}
```

```
[04/28/2020 14:09:54 INFO 140478314145600] #progress_metric: host=al
go-1, completed 44 % of epochs
```

```
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count":
1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Las
t Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number o
f Records Since Last Reset": {"count": 1, "max": 90570, "sum": 9057
0.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 2003,
"sum": 2003.0, "min": 2003}, "Total Records Seen": {"count": 1, "ma
x": 1993540, "sum": 1993540.0, "min": 1993540}, "Max Records Seen Be
tween Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 905
70}, "Reset Count": {"count": 1, "max": 23, "sum": 23.0, "min": 2
3}}, "EndTime": 1588082994.674616, "Dimensions": {"Host": "algo-1",
"Meta": "training_data_iter", "Operation": "training", "Algorithm":
"factorization-machines", "epoch": 21}, "StartTime": 1588082993.7784
15}
```

```
[04/28/2020 14:09:54 INFO 140478314145600] #throughput_metric: host=
algo-1, train throughput=101046.946387 records/second
```

```
[04/28/2020 14:09:54 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=22, batch=0 train binary_classification_accuracy <score>=
0.738
```

```
[04/28/2020 14:09:54 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=22, batch=0 train binary_classification_cross_entropy <lo
ss>=0.569146118164
```

```
[04/28/2020 14:09:54 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=22, batch=0 train binary_f_1.000 <score>=0.756052141527
```

```
[2020-04-28 14:09:55.551] [tensorio] [info] epoch_stats={"data_pipel
ine": "/opt/ml/input/data/train", "epoch": 46, "duration": 874, "num
```

```
_examples": 91, "num_bytes": 5796480}
[04/28/2020 14:09:55 INFO 140478314145600] #quality_metric: host=algo-1, epoch=22, train binary_classification_accuracy <score>=0.73278021978
[04/28/2020 14:09:55 INFO 140478314145600] #quality_metric: host=algo-1, epoch=22, train binary_classification_cross_entropy <loss>=0.571013777177
[04/28/2020 14:09:55 INFO 140478314145600] #quality_metric: host=algo-1, epoch=22, train binary_f_1.000 <score>=0.770362535767
#metrics {"Metrics": {"update.time": {"count": 1, "max": 876.5559196472168, "sum": 876.5559196472168, "min": 876.5559196472168}}, "EndTime": 1588082995.551857, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588082994.674371}
```

```
[04/28/2020 14:09:55 INFO 140478314145600] #progress_metric: host=algo-1, completed 46 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 2094, "sum": 2094.0, "min": 2094}, "Total Records Seen": {"count": 1, "max": 2084110, "sum": 2084110.0, "min": 2084110}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 24, "sum": 24.0, "min": 24}}, "EndTime": 1588082995.552092, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 22}, "StartTime": 1588082994.675272}
```

```
[04/28/2020 14:09:55 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=103278.068167 records/second
[04/28/2020 14:09:55 INFO 140478314145600] #quality_metric: host=algo-1, epoch=23, batch=0 train binary_classification_accuracy <score>=0.736
[04/28/2020 14:09:55 INFO 140478314145600] #quality_metric: host=algo-1, epoch=23, batch=0 train binary_classification_cross_entropy <loss>=0.567032775879
[04/28/2020 14:09:55 INFO 140478314145600] #quality_metric: host=algo-1, epoch=23, batch=0 train binary_f_1.000 <score>=0.754189944134
[2020-04-28 14:09:56.437] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 48, "duration": 883, "num_examples": 91, "num_bytes": 5796480}
[04/28/2020 14:09:56 INFO 140478314145600] #quality_metric: host=algo-1, epoch=23, train binary_classification_accuracy <score>=0.733538461538
[04/28/2020 14:09:56 INFO 140478314145600] #quality_metric: host=algo-1, epoch=23, train binary_classification_cross_entropy <loss>=0.568927229997
[04/28/2020 14:09:56 INFO 140478314145600] #quality_metric: host=algo-1, epoch=23, train binary_f_1.000 <score>=0.770609047736
#metrics {"Metrics": {"update.time": {"count": 1, "max": 885.6208324432373, "sum": 885.6208324432373, "min": 885.6208324432373}}, "EndTime": 1588082996.438415, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588082995.551928}
```

```
[04/28/2020 14:09:56 INFO 140478314145600] #progress_metric: host=algo-1, completed 48 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count":
```

```
1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 2185, "sum": 2185.0, "min": 2185}, "Total Records Seen": {"count": 1, "max": 2174680, "sum": 2174680.0, "min": 2174680}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 25, "sum": 25.0, "min": 25}}, "EndTime": 1588082996.438648, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 23}, "StartTime": 1588082995.552765}
```

```
[04/28/2020 14:09:56 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=102222.330443 records/second
```

```
[04/28/2020 14:09:56 INFO 140478314145600] #quality_metric: host=algo-1, epoch=24, batch=0 train binary_classification_accuracy <score>=0.736
```

```
[04/28/2020 14:09:56 INFO 140478314145600] #quality_metric: host=algo-1, epoch=24, batch=0 train binary_classification_cross_entropy <loss>=0.565038635254
```

```
[04/28/2020 14:09:56 INFO 140478314145600] #quality_metric: host=algo-1, epoch=24, batch=0 train binary_f_1.000 <score>=0.754646840149
```

```
[2020-04-28 14:09:57.309] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 50, "duration": 869, "num_examples": 91, "num_bytes": 5796480}
```

```
[04/28/2020 14:09:57 INFO 140478314145600] #quality_metric: host=algo-1, epoch=24, train binary_classification_accuracy <score>=0.734274725275
```

```
[04/28/2020 14:09:57 INFO 140478314145600] #quality_metric: host=algo-1, epoch=24, train binary_classification_cross_entropy <loss>=0.566950572255
```

```
[04/28/2020 14:09:57 INFO 140478314145600] #quality_metric: host=algo-1, epoch=24, train binary_f_1.000 <score>=0.770885248387
```

```
#metrics {"Metrics": {"update.time": {"count": 1, "max": 871.225118637085, "sum": 871.225118637085, "min": 871.225118637085}}, "EndTime": 1588082997.310549, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588082996.438486}
```

```
[04/28/2020 14:09:57 INFO 140478314145600] #progress_metric: host=algo-1, completed 50 % of epochs
```

```
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 2276, "sum": 2276.0, "min": 2276}, "Total Records Seen": {"count": 1, "max": 2265250, "sum": 2265250.0, "min": 2265250}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 26, "sum": 26.0, "min": 26}}, "EndTime": 1588082997.310864, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 24}, "StartTime": 1588082996.439292}
```

```
[04/28/2020 14:09:57 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=103898.514726 records/second
```

```
[04/28/2020 14:09:57 INFO 140478314145600] #quality_metric: host=algo-1, epoch=25, batch=0 train binary_classification_accuracy <score>=0.735
```



```
[04/28/2020 14:09:57 INFO 140478314145600] #quality_metric: host=algo-1, epoch=25, batch=0 train binary_classification_cross_entropy <loss>=0.563152893066
[04/28/2020 14:09:57 INFO 140478314145600] #quality_metric: host=algo-1, epoch=25, batch=0 train binary_f_1.000 <score>=0.753488372093
[2020-04-28 14:09:58.222] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 52, "duration": 909, "num_examples": 91, "num_bytes": 5796480}
[04/28/2020 14:09:58 INFO 140478314145600] #quality_metric: host=algo-1, epoch=25, train binary_classification_accuracy <score>=0.734802197802
[04/28/2020 14:09:58 INFO 140478314145600] #quality_metric: host=algo-1, epoch=25, train binary_classification_cross_entropy <loss>=0.565074333442
[04/28/2020 14:09:58 INFO 140478314145600] #quality_metric: host=algo-1, epoch=25, train binary_f_1.000 <score>=0.770997219665
#metrics {"Metrics": {"update.time": {"count": 1, "max": 911.0691547393799, "sum": 911.0691547393799, "min": 911.0691547393799}}, "EndTime": 1588082998.222652, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588082997.310638}

[04/28/2020 14:09:58 INFO 140478314145600] #progress_metric: host=algo-1, completed 52 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 2367, "sum": 2367.0, "min": 2367}, "Total Records Seen": {"count": 1, "max": 2355820, "sum": 2355820.0, "min": 2355820}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 27, "sum": 27.0, "min": 27}}, "EndTime": 1588082998.2229, "Dimensions": {"Host": "algo-1", "Operation": "training_data_iter", "Algorithm": "factorization-machines", "epoch": 25}, "StartTime": 1588082997.311549}

[04/28/2020 14:09:58 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=99366.2878236 records/second
[04/28/2020 14:09:58 INFO 140478314145600] #quality_metric: host=algo-1, epoch=26, batch=0 train binary_classification_accuracy <score>=0.733
[04/28/2020 14:09:58 INFO 140478314145600] #quality_metric: host=algo-1, epoch=26, batch=0 train binary_classification_cross_entropy <loss>=0.561365600586
[04/28/2020 14:09:58 INFO 140478314145600] #quality_metric: host=algo-1, epoch=26, batch=0 train binary_f_1.000 <score>=0.75208913649
[2020-04-28 14:09:59.139] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 54, "duration": 913, "num_examples": 91, "num_bytes": 5796480}
[04/28/2020 14:09:59 INFO 140478314145600] #quality_metric: host=algo-1, epoch=26, train binary_classification_accuracy <score>=0.735296703297
[04/28/2020 14:09:59 INFO 140478314145600] #quality_metric: host=algo-1, epoch=26, train binary_classification_cross_entropy <loss>=0.563290040404
[04/28/2020 14:09:59 INFO 140478314145600] #quality_metric: host=algo-1, epoch=26, train binary_f_1.000 <score>=0.771126693651
#metrics {"Metrics": {"update.time": {"count": 1, "max": 917.025089263916, "sum": 917.025089263916, "min": 917.025089263916}}, "EndTime": 1588082998.2229, "Dimensions": {"Host": "algo-1", "Operation": "training_data_iter", "Algorithm": "factorization-machines", "epoch": 26}, "StartTime": 1588082997.311549}}
```

```
e": 1588082999.140657, "Dimensions": {"Host": "algo-1", "Operation":
"training", "Algorithm": "factorization-machines"}, "StartTime": 158
8082998.222728}
```

```
[04/28/2020 14:09:59 INFO 140478314145600] #progress_metric: host=al
go-1, completed 54 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count":
1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Las
t Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number o
f Records Since Last Reset": {"count": 1, "max": 90570, "sum": 9057
0.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 2458,
"sum": 2458.0, "min": 2458}, "Total Records Seen": {"count": 1, "ma
x": 2446390, "sum": 2446390.0, "min": 2446390}, "Max Records Seen Be
tween Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 905
70}, "Reset Count": {"count": 1, "max": 28, "sum": 28.0, "min": 2
8}}, "EndTime": 1588082999.141046, "Dimensions": {"Host": "algo-1",
"Meta": "training_data_iter", "Operation": "training", "Algorithm":
"factorization-machines", "epoch": 26}, "StartTime": 1588082998.2235
98}
```

```
[04/28/2020 14:09:59 INFO 140478314145600] #throughput_metric: host=
algo-1, train throughput=98704.7376795 records/second
```

```
[04/28/2020 14:09:59 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=27, batch=0 train binary_classification_accuracy <score>=
0.734
```

```
[04/28/2020 14:09:59 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=27, batch=0 train binary_classification_cross_entropy <lo
ss>=0.559668212891
```

```
[04/28/2020 14:09:59 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=27, batch=0 train binary_f_1.000 <score>=0.752788104089
```

```
[2020-04-28 14:09:59.999] [tensorio] [info] epoch_stats={"data_pipel
ine": "/opt/ml/input/data/train", "epoch": 56, "duration": 855, "num
_examples": 91, "num_bytes": 5796480}
```

```
[04/28/2020 14:09:59 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=27, train binary_classification_accuracy <score>=0.735912
087912
```

```
[04/28/2020 14:09:59 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=27, train binary_classification_cross_entropy <loss>=0.56
1590129475
```

```
[04/28/2020 14:09:59 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=27, train binary_f_1.000 <score>=0.771424223402
```

```
#metrics {"Metrics": {"update.time": {"count": 1, "max": 858.2260608
673096, "sum": 858.2260608673096, "min": 858.2260608673096}}, "EndTi
me": 1588083000.000024, "Dimensions": {"Host": "algo-1", "Operatio
n": "training", "Algorithm": "factorization-machines"}, "StartTime":
1588082999.140724}
```

```
[04/28/2020 14:10:00 INFO 140478314145600] #progress_metric: host=al
go-1, completed 56 % of epochs
```

```
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count":
1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Las
t Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number o
f Records Since Last Reset": {"count": 1, "max": 90570, "sum": 9057
0.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 2549,
"sum": 2549.0, "min": 2549}, "Total Records Seen": {"count": 1, "ma
x": 2536960, "sum": 2536960.0, "min": 2536960}, "Max Records Seen Be
tween Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 905
70}, "Reset Count": {"count": 1, "max": 29, "sum": 29.0, "min": 2
9}}, "EndTime": 1588083000.000428, "Dimensions": {"Host": "algo-1",
"Meta": "training_data_iter", "Operation": "training", "Algorithm":
"factorization-machines", "epoch": 27}, "StartTime": 1588082999.1417
```

72}

```
[04/28/2020 14:10:00 INFO 140478314145600] #throughput_metric: host=
algo-1, train throughput=105465.420613 records/second
[04/28/2020 14:10:00 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=28, batch=0 train binary_classification_accuracy <score>=
0.736
[04/28/2020 14:10:00 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=28, batch=0 train binary_classification_cross_entropy <lo
ss>=0.558052734375
[04/28/2020 14:10:00 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=28, batch=0 train binary_f_1.000 <score>=0.755102040816
[2020-04-28 14:10:00.872] [tensorio] [info] epoch_stats={"data_pipel
ine": "/opt/ml/input/data/train", "epoch": 58, "duration": 869, "num
_examples": 91, "num_bytes": 5796480}
[04/28/2020 14:10:00 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=28, train binary_classification_accuracy <score>=0.736175
824176
[04/28/2020 14:10:00 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=28, train binary_classification_cross_entropy <loss>=0.55
99678224
[04/28/2020 14:10:00 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=28, train binary_f_1.000 <score>=0.771313177497
#metrics {"Metrics": {"update.time": {"count": 1, "max": 871.8528747
558594, "sum": 871.8528747558594, "min": 871.8528747558594}}, "EndTi
me": 1588083000.873024, "Dimensions": {"Host": "algo-1", "Operatio
n": "training", "Algorithm": "factorization-machines"}, "StartTime":
1588083000.000084}

[04/28/2020 14:10:00 INFO 140478314145600] #progress_metric: host=al
go-1, completed 58 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count":
1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Las
t Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number o
f Records Since Last Reset": {"count": 1, "max": 90570, "sum": 9057
0.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 2640,
"sum": 2640.0, "min": 2640}, "Total Records Seen": {"count": 1, "ma
x": 2627530, "sum": 2627530.0, "min": 2627530}, "Max Records Seen Be
tween Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 905
70}, "Reset Count": {"count": 1, "max": 30, "sum": 30.0, "min": 3
0}}, "EndTime": 1588083000.873437, "Dimensions": {"Host": "algo-1",
"Meta": "training_data_iter", "Operation": "training", "Algorithm":
"factorization-machines", "epoch": 28}, "StartTime": 1588083000.0011
2}

[04/28/2020 14:10:00 INFO 140478314145600] #throughput_metric: host=
algo-1, train throughput=103813.249258 records/second
[04/28/2020 14:10:00 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=29, batch=0 train binary_classification_accuracy <score>=
0.738
[04/28/2020 14:10:00 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=29, batch=0 train binary_classification_cross_entropy <lo
ss>=0.556512573242
[04/28/2020 14:10:00 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=29, batch=0 train binary_f_1.000 <score>=0.756957328386
[2020-04-28 14:10:01.783] [tensorio] [info] epoch_stats={"data_pipel
ine": "/opt/ml/input/data/train", "epoch": 60, "duration": 908, "num
_examples": 91, "num_bytes": 5796480}
[04/28/2020 14:10:01 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=29, train binary_classification_accuracy <score>=0.736758
241758
```

```
[04/28/2020 14:10:01 INFO 140478314145600] #quality_metric: host=algo-1, epoch=29, train binary_classification_cross_entropy <loss>=0.558417090783
[04/28/2020 14:10:01 INFO 140478314145600] #quality_metric: host=algo-1, epoch=29, train binary_f_1.000 <score>=0.771567795398
#metrics {"Metrics": {"update.time": {"count": 1, "max": 910.4888439178467, "sum": 910.4888439178467, "min": 910.4888439178467}}, "EndTime": 1588083001.784664, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588083000.873133}

[04/28/2020 14:10:01 INFO 140478314145600] #progress_metric: host=algo-1, completed 60 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 2731, "sum": 2731.0, "min": 2731}, "Total Records Seen": {"count": 1, "max": 2718100, "sum": 2718100.0, "min": 2718100}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 31, "sum": 31.0, "min": 31}}, "EndTime": 1588083001.784983, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 29}, "StartTime": 1588083000.874143}

[04/28/2020 14:10:01 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=99417.8820337 records/second
[04/28/2020 14:10:01 INFO 140478314145600] #quality_metric: host=algo-1, epoch=30, batch=0 train binary_classification_accuracy <score>=0.74
[04/28/2020 14:10:01 INFO 140478314145600] #quality_metric: host=algo-1, epoch=30, batch=0 train binary_classification_cross_entropy <loss>=0.555041259766
[04/28/2020 14:10:01 INFO 140478314145600] #quality_metric: host=algo-1, epoch=30, batch=0 train binary_f_1.000 <score>=0.758812615955
[2020-04-28 14:10:02.651] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 62, "duration": 864, "num_examples": 91, "num_bytes": 5796480}
[04/28/2020 14:10:02 INFO 140478314145600] #quality_metric: host=algo-1, epoch=30, train binary_classification_accuracy <score>=0.737395604396
[04/28/2020 14:10:02 INFO 140478314145600] #quality_metric: host=algo-1, epoch=30, train binary_classification_cross_entropy <loss>=0.556932489751
[04/28/2020 14:10:02 INFO 140478314145600] #quality_metric: host=algo-1, epoch=30, train binary_f_1.000 <score>=0.771885947747
#metrics {"Metrics": {"update.time": {"count": 1, "max": 866.4181232452393, "sum": 866.4181232452393, "min": 866.4181232452393}}, "EndTime": 1588083002.652161, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588083001.784758}

[04/28/2020 14:10:02 INFO 140478314145600] #progress_metric: host=algo-1, completed 62 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 2822,
```



```
"sum": 2822.0, "min": 2822}, "Total Records Seen": {"count": 1, "max": 2808670, "sum": 2808670.0, "min": 2808670}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 32, "sum": 32.0, "min": 32}}, "EndTime": 1588083002.652464, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 30}, "StartTime": 1588083001.785712}
```

```
[04/28/2020 14:10:02 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=104476.555304 records/second
[04/28/2020 14:10:02 INFO 140478314145600] #quality_metric: host=algo-1, epoch=31, batch=0 train binary_classification_accuracy <score>=0.741
[04/28/2020 14:10:02 INFO 140478314145600] #quality_metric: host=algo-1, epoch=31, batch=0 train binary_classification_cross_entropy <loss>=0.553633422852
[04/28/2020 14:10:02 INFO 140478314145600] #quality_metric: host=algo-1, epoch=31, batch=0 train binary_f1.000 <score>=0.759962928638
[2020-04-28 14:10:03.553] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 64, "duration": 899, "num_examples": 91, "num_bytes": 5796480}
[04/28/2020 14:10:03 INFO 140478314145600] #quality_metric: host=algo-1, epoch=31, train binary_classification_accuracy <score>=0.737813186813
[04/28/2020 14:10:03 INFO 140478314145600] #quality_metric: host=algo-1, epoch=31, train binary_classification_cross_entropy <loss>=0.555509192163
[04/28/2020 14:10:03 INFO 140478314145600] #quality_metric: host=algo-1, epoch=31, train binary_f1.000 <score>=0.77202235918
#metrics {"Metrics": {"update.time": {"count": 1, "max": 901.2179374694824, "sum": 901.2179374694824, "min": 901.2179374694824}}, "EndTime": 1588083003.554481, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588083002.652227}
```

```
[04/28/2020 14:10:03 INFO 140478314145600] #progress_metric: host=algo-1, completed 64 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 2913, "sum": 2913.0, "min": 2913}, "Total Records Seen": {"count": 1, "max": 2899240, "sum": 2899240.0, "min": 2899240}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 33, "sum": 33.0, "min": 33}}, "EndTime": 1588083003.554809, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 31}, "StartTime": 1588083002.653232}
```

```
[04/28/2020 14:10:03 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=100441.373464 records/second
[04/28/2020 14:10:03 INFO 140478314145600] #quality_metric: host=algo-1, epoch=32, batch=0 train binary_classification_accuracy <score>=0.742
[04/28/2020 14:10:03 INFO 140478314145600] #quality_metric: host=algo-1, epoch=32, batch=0 train binary_classification_cross_entropy <loss>=0.552284240723
[04/28/2020 14:10:03 INFO 140478314145600] #quality_metric: host=algo-
```

```
o-1, epoch=32, batch=0 train binary_f_1.000 <score>=0.760667903525
[2020-04-28 14:10:04.443] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 66, "duration": 886, "num_examples": 91, "num_bytes": 5796480}
[04/28/2020 14:10:04 INFO 140478314145600] #quality_metric: host=algo-1, epoch=32, train binary_classification_accuracy <score>=0.73821978022
[04/28/2020 14:10:04 INFO 140478314145600] #quality_metric: host=algo-1, epoch=32, train binary_classification_cross_entropy <loss>=0.554142837021
[04/28/2020 14:10:04 INFO 140478314145600] #quality_metric: host=algo-1, epoch=32, train binary_f_1.000 <score>=0.77219959072
#metrics {"Metrics": {"update.time": {"count": 1, "max": 888.3960247039795, "sum": 888.3960247039795, "min": 888.3960247039795}}, "EndTime": 1588083004.443954, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588083003.554575}

[04/28/2020 14:10:04 INFO 140478314145600] #progress_metric: host=algo-1, completed 66 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 3004, "sum": 3004.0, "min": 3004}, "Total Records Seen": {"count": 1, "max": 2989810, "sum": 2989810.0, "min": 2989810}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 34, "sum": 34.0, "min": 34}}, "EndTime": 1588083004.444232, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 32}, "StartTime": 1588083003.555496}

[04/28/2020 14:10:04 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=101891.939312 records/second
[04/28/2020 14:10:04 INFO 140478314145600] #quality_metric: host=algo-1, epoch=33, batch=0 train binary_classification_accuracy <score>=0.743
[04/28/2020 14:10:04 INFO 140478314145600] #quality_metric: host=algo-1, epoch=33, batch=0 train binary_classification_cross_entropy <loss>=0.550989318848
[04/28/2020 14:10:04 INFO 140478314145600] #quality_metric: host=algo-1, epoch=33, batch=0 train binary_f_1.000 <score>=0.761374187558
[2020-04-28 14:10:05.352] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 68, "duration": 906, "num_examples": 91, "num_bytes": 5796480}
[04/28/2020 14:10:05 INFO 140478314145600] #quality_metric: host=algo-1, epoch=33, train binary_classification_accuracy <score>=0.738483516484
[04/28/2020 14:10:05 INFO 140478314145600] #quality_metric: host=algo-1, epoch=33, train binary_classification_cross_entropy <loss>=0.552829526126
[04/28/2020 14:10:05 INFO 140478314145600] #quality_metric: host=algo-1, epoch=33, train binary_f_1.000 <score>=0.772267942584
#metrics {"Metrics": {"update.time": {"count": 1, "max": 908.7190628051758, "sum": 908.7190628051758, "min": 908.7190628051758}}, "EndTime": 1588083005.353694, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588083004.444027}
```

```
[04/28/2020 14:10:05 INFO 140478314145600] #progress_metric: host=algo-1, completed 68 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 3095, "sum": 3095.0, "min": 3095}, "Total Records Seen": {"count": 1, "max": 3080380, "sum": 3080380.0, "min": 3080380}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 35, "sum": 35.0, "min": 35}}, "EndTime": 1588083005.354015, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 33}, "StartTime": 1588083004.444942}
```

```
[04/28/2020 14:10:05 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=99611.655597 records/second
```

```
[04/28/2020 14:10:05 INFO 140478314145600] #quality_metric: host=algo-1, epoch=34, batch=0 train binary_classification_accuracy <score>=0.742
```

```
[04/28/2020 14:10:05 INFO 140478314145600] #quality_metric: host=algo-1, epoch=34, batch=0 train binary_classification_cross_entropy <loss>=0.549744750977
```

```
[04/28/2020 14:10:05 INFO 140478314145600] #quality_metric: host=algo-1, epoch=34, batch=0 train binary_f_1.000 <score>=0.760223048327
```

```
[2020-04-28 14:10:06.234] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 70, "duration": 878, "num_examples": 91, "num_bytes": 5796480}
```

```
[04/28/2020 14:10:06 INFO 140478314145600] #quality_metric: host=algo-1, epoch=34, train binary_classification_accuracy <score>=0.738846153846
```

```
[04/28/2020 14:10:06 INFO 140478314145600] #quality_metric: host=algo-1, epoch=34, train binary_classification_cross_entropy <loss>=0.551565726815
```

```
[04/28/2020 14:10:06 INFO 140478314145600] #quality_metric: host=algo-1, epoch=34, train binary_f_1.000 <score>=0.772437830954
```

```
#metrics {"Metrics": {"update.time": {"count": 1, "max": 880.2001476287842, "sum": 880.2001476287842, "min": 880.2001476287842}}, "EndTime": 1588083006.23494, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588083005.353787}
```

```
[04/28/2020 14:10:06 INFO 140478314145600] #progress_metric: host=algo-1, completed 70 % of epochs
```

```
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 3186, "sum": 3186.0, "min": 3186}, "Total Records Seen": {"count": 1, "max": 3170950, "sum": 3170950.0, "min": 3170950}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 36, "sum": 36.0, "min": 36}}, "EndTime": 1588083006.235279, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 34}, "StartTime": 1588083005.354708}
```

```
[04/28/2020 14:10:06 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=102837.830714 records/second
```

```
[04/28/2020 14:10:06 INFO 140478314145600] #quality_metric: host=algo-1, epoch=35, batch=0 train binary_classification_accuracy <score>=0.747
[04/28/2020 14:10:06 INFO 140478314145600] #quality_metric: host=algo-1, epoch=35, batch=0 train binary_classification_cross_entropy <loss>=0.548547119141
[04/28/2020 14:10:06 INFO 140478314145600] #quality_metric: host=algo-1, epoch=35, batch=0 train binary_f_1.000 <score>=0.765523632994
[2020-04-28 14:10:07.108] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 72, "duration": 870, "num_examples": 91, "num_bytes": 5796480}
[04/28/2020 14:10:07 INFO 140478314145600] #quality_metric: host=algo-1, epoch=35, train binary_classification_accuracy <score>=0.739175824176
[04/28/2020 14:10:07 INFO 140478314145600] #quality_metric: host=algo-1, epoch=35, train binary_classification_cross_entropy <loss>=0.550348285382
[04/28/2020 14:10:07 INFO 140478314145600] #quality_metric: host=algo-1, epoch=35, train binary_f_1.000 <score>=0.772563937945
#metrics {"Metrics": {"update.time": {"count": 1, "max": 873.013973236084, "sum": 873.013973236084, "min": 873.013973236084}}, "EndTime": 1588083007.109072, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588083006.235029}

[04/28/2020 14:10:07 INFO 140478314145600] #progress_metric: host=algo-1, completed 72 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 3277, "sum": 3277.0, "min": 3277}, "Total Records Seen": {"count": 1, "max": 3261520, "sum": 3261520.0, "min": 3261520}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 37, "sum": 37.0, "min": 37}}, "EndTime": 1588083007.109325, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 35}, "StartTime": 1588083006.236004}

[04/28/2020 14:10:07 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=103692.250006 records/second
[04/28/2020 14:10:07 INFO 140478314145600] #quality_metric: host=algo-1, epoch=36, batch=0 train binary_classification_accuracy <score>=0.748
[04/28/2020 14:10:07 INFO 140478314145600] #quality_metric: host=algo-1, epoch=36, batch=0 train binary_classification_cross_entropy <loss>=0.547393249512
[04/28/2020 14:10:07 INFO 140478314145600] #quality_metric: host=algo-1, epoch=36, batch=0 train binary_f_1.000 <score>=0.766233766234
[2020-04-28 14:10:07.988] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 74, "duration": 877, "num_examples": 91, "num_bytes": 5796480}
[04/28/2020 14:10:07 INFO 140478314145600] #quality_metric: host=algo-1, epoch=36, train binary_classification_accuracy <score>=0.739516483516
[04/28/2020 14:10:07 INFO 140478314145600] #quality_metric: host=algo-1, epoch=36, train binary_classification_cross_entropy <loss>=0.549174335186
[04/28/2020 14:10:07 INFO 140478314145600] #quality_metric: host=algo-1, epoch=36, batch=0 train binary_classification_accuracy <score>=0.748
```



```
o-1, epoch=36, train binary_f_1.000 <score>=0.772710710519
#metrics {"Metrics": {"update.time": {"count": 1, "max": 879.5039653
778076, "sum": 879.5039653778076, "min": 879.5039653778076}}, "EndTi
me": 1588083007.989727, "Dimensions": {"Host": "algo-1", "Operatio
n": "training", "Algorithm": "factorization-machines"}, "StartTime":
1588083007.109145}
```

```
[04/28/2020 14:10:07 INFO 140478314145600] #progress_metric: host=al
go-1, completed 74 % of epochs
```

```
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count":
1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Las
t Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number o
f Records Since Last Reset": {"count": 1, "max": 90570, "sum": 9057
0.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 3368,
"sum": 3368.0, "min": 3368}, "Total Records Seen": {"count": 1, "ma
x": 3352090, "sum": 3352090.0, "min": 3352090}, "Max Records Seen Be
tween Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 905
70}, "Reset Count": {"count": 1, "max": 38, "sum": 38.0, "min": 3
8}}, "EndTime": 1588083007.990106, "Dimensions": {"Host": "algo-1",
"Meta": "training_data_iter", "Operation": "training", "Algorithm":
"factorization-machines", "epoch": 36}, "StartTime": 1588083007.1101
49}
```

```
[04/28/2020 14:10:07 INFO 140478314145600] #throughput_metric: host=
algo-1, train throughput=102910.515324 records/second
```

```
[04/28/2020 14:10:08 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=37, batch=0 train binary_classification_accuracy <score>=
0.746
```

```
[04/28/2020 14:10:08 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=37, batch=0 train binary_classification_cross_entropy <lo
ss>=0.546280395508
```

```
[04/28/2020 14:10:08 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=37, batch=0 train binary_f_1.000 <score>=0.764378478664
```

```
[2020-04-28 14:10:08.996] [tensorio] [info] epoch_stats={"data_pipel
ine": "/opt/ml/input/data/train", "epoch": 76, "duration": 1004, "nu
m_examples": 91, "num_bytes": 5796480}
```

```
[04/28/2020 14:10:08 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=37, train binary_classification_accuracy <score>=0.739714
285714
```

```
[04/28/2020 14:10:08 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=37, train binary_classification_cross_entropy <loss>=0.54
8041288606
```

```
[04/28/2020 14:10:08 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=37, train binary_f_1.000 <score>=0.772695865802
```

```
#metrics {"Metrics": {"update.time": {"count": 1, "max": 1006.686925
8880615, "sum": 1006.6869258880615, "min": 1006.6869258880615}}, "En
dTime": 1588083008.997547, "Dimensions": {"Host": "algo-1", "Operati
on": "training", "Algorithm": "factorization-machines"}, "StartTim
e": 1588083007.989795}
```

```
[04/28/2020 14:10:08 INFO 140478314145600] #progress_metric: host=al
go-1, completed 76 % of epochs
```

```
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count":
1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Las
t Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number o
f Records Since Last Reset": {"count": 1, "max": 90570, "sum": 9057
0.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 3459,
"sum": 3459.0, "min": 3459}, "Total Records Seen": {"count": 1, "ma
x": 3442660, "sum": 3442660.0, "min": 3442660}, "Max Records Seen Be
tween Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 905
70}, "Reset Count": {"count": 1, "max": 39, "sum": 39.0, "min": 3
```

```

    }}, "EndTime": 1588083008.997793, "Dimensions": {"Host": "algo-1",
    "Meta": "training_data_iter", "Operation": "training", "Algorithm":
    "factorization-machines", "epoch": 37}, "StartTime": 1588083007.9908
    25}

```

```

[04/28/2020 14:10:08 INFO 140478314145600] #throughput_metric: host=
algo-1, train throughput=89930.4125485 records/second
[04/28/2020 14:10:09 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=38, batch=0 train binary_classification_accuracy <score>=
0.746
[04/28/2020 14:10:09 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=38, batch=0 train binary_classification_cross_entropy <lo
ss>=0.545205993652
[04/28/2020 14:10:09 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=38, batch=0 train binary_f_1.000 <score>=0.764378478664

```

```

2020-04-28 14:10:24 Uploading - Uploading generated training model[2
020-04-28 14:10:09.896] [tensorio] [info] epoch_stats={"data_pipel
ine": "/opt/ml/input/data/train", "epoch": 78, "duration": 896, "num_e
xamples": 91, "num_bytes": 5796480}
[04/28/2020 14:10:09 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=38, train binary_classification_accuracy <score>=0.739956
043956
[04/28/2020 14:10:09 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=38, train binary_classification_cross_entropy <loss>=0.54
694677399
[04/28/2020 14:10:09 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=38, train binary_f_1.000 <score>=0.772776156092
#metrics {"Metrics": {"update.time": {"count": 1, "max": 899.2660045
623779, "sum": 899.2660045623779, "min": 899.2660045623779}}, "EndTi
me": 1588083009.897833, "Dimensions": {"Host": "algo-1", "Operatio
n": "training", "Algorithm": "factorization-machines"}, "StartTime":
1588083008.997622}

```

```

[04/28/2020 14:10:09 INFO 140478314145600] #progress_metric: host=al
go-1, completed 78 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count":
1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Las
t Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number o
f Records Since Last Reset": {"count": 1, "max": 90570, "sum": 9057
0.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 3550,
"sum": 3550.0, "min": 3550}, "Total Records Seen": {"count": 1, "ma
x": 3533230, "sum": 3533230.0, "min": 3533230}, "Max Records Seen Be
tween Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 905
70}, "Reset Count": {"count": 1, "max": 40, "sum": 40.0, "min": 4
0}}, "EndTime": 1588083009.898139, "Dimensions": {"Host": "algo-1",
"Meta": "training_data_iter", "Operation": "training", "Algorithm":
"factorization-machines", "epoch": 38}, "StartTime": 1588083008.9985
02}

```

```

[04/28/2020 14:10:09 INFO 140478314145600] #throughput_metric: host=
algo-1, train throughput=100653.694058 records/second
[04/28/2020 14:10:09 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=39, batch=0 train binary_classification_accuracy <score>=
0.747
[04/28/2020 14:10:09 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=39, batch=0 train binary_classification_cross_entropy <lo
ss>=0.544167907715
[04/28/2020 14:10:09 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=39, batch=0 train binary_f_1.000 <score>=0.765523632994
[2020-04-28 14:10:10.830] [tensorio] [info] epoch_stats={"data_pipel

```

```
ine": "/opt/ml/input/data/train", "epoch": 80, "duration": 930, "num_examples": 91, "num_bytes": 5796480}
[04/28/2020 14:10:10 INFO 140478314145600] #quality_metric: host=algo-1, epoch=39, train binary_classification_accuracy <score>=0.740417582418
[04/28/2020 14:10:10 INFO 140478314145600] #quality_metric: host=algo-1, epoch=39, train binary_classification_cross_entropy <loss>=0.545888668521
[04/28/2020 14:10:10 INFO 140478314145600] #quality_metric: host=algo-1, epoch=39, train binary_f_1.000 <score>=0.772974531475
#metrics {"Metrics": {"update.time": {"count": 1, "max": 932.08909034729, "sum": 932.08909034729, "min": 932.08909034729}}, "EndTime": 1588083010.831007, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588083009.897923}
```

```
[04/28/2020 14:10:10 INFO 140478314145600] #progress_metric: host=algo-1, completed 80 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 3641, "sum": 3641.0, "min": 3641}, "Total Records Seen": {"count": 1, "max": 3623800, "sum": 3623800.0, "min": 3623800}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 41, "sum": 41.0, "min": 41}}, "EndTime": 1588083010.831251, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 39}, "StartTime": 1588083009.898886}
```

```
[04/28/2020 14:10:10 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=97126.8367733 records/second
[04/28/2020 14:10:10 INFO 140478314145600] #quality_metric: host=algo-1, epoch=40, batch=0 train binary_classification_accuracy <score>=0.747
[04/28/2020 14:10:10 INFO 140478314145600] #quality_metric: host=algo-1, epoch=40, batch=0 train binary_classification_cross_entropy <loss>=0.54316394043
[04/28/2020 14:10:10 INFO 140478314145600] #quality_metric: host=algo-1, epoch=40, batch=0 train binary_f_1.000 <score>=0.765523632994
[2020-04-28 14:10:11.694] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 82, "duration": 861, "num_examples": 91, "num_bytes": 5796480}
[04/28/2020 14:10:11 INFO 140478314145600] #quality_metric: host=algo-1, epoch=40, train binary_classification_accuracy <score>=0.740802197802
[04/28/2020 14:10:11 INFO 140478314145600] #quality_metric: host=algo-1, epoch=40, train binary_classification_cross_entropy <loss>=0.54486500499
[04/28/2020 14:10:11 INFO 140478314145600] #quality_metric: host=algo-1, epoch=40, train binary_f_1.000 <score>=0.773204103807
#metrics {"Metrics": {"update.time": {"count": 1, "max": 863.5900020599365, "sum": 863.5900020599365, "min": 863.5900020599365}}, "EndTime": 1588083011.695508, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588083010.831082}
```

```
[04/28/2020 14:10:11 INFO 140478314145600] #progress_metric: host=algo-1, completed 82 % of epochs
```

```
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 3732, "sum": 3732.0, "min": 3732}, "Total Records Seen": {"count": 1, "max": 3714370, "sum": 3714370.0, "min": 3714370}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 42, "sum": 42.0, "min": 42}}, "EndTime": 1588083011.695834, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 40}, "StartTime": 1588083010.831887}
```

```
[04/28/2020 14:10:11 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=104816.920804 records/second
```

```
[04/28/2020 14:10:11 INFO 140478314145600] #quality_metric: host=algo-1, epoch=41, batch=0 train binary_classification_accuracy <score>=0.747
```

```
[04/28/2020 14:10:11 INFO 140478314145600] #quality_metric: host=algo-1, epoch=41, batch=0 train binary_classification_cross_entropy <loss>=0.542192321777
```

```
[04/28/2020 14:10:11 INFO 140478314145600] #quality_metric: host=algo-1, epoch=41, batch=0 train binary_f_1.000 <score>=0.765523632994
```

```
[2020-04-28 14:10:12.556] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 84, "duration": 858, "num_examples": 91, "num_bytes": 5796480}
```

```
[04/28/2020 14:10:12 INFO 140478314145600] #quality_metric: host=algo-1, epoch=41, train binary_classification_accuracy <score>=0.741197802198
```

```
[04/28/2020 14:10:12 INFO 140478314145600] #quality_metric: host=algo-1, epoch=41, train binary_classification_cross_entropy <loss>=0.543873982524
```

```
[04/28/2020 14:10:12 INFO 140478314145600] #quality_metric: host=algo-1, epoch=41, train binary_f_1.000 <score>=0.773423895789
```

```
#metrics {"Metrics": {"update.time": {"count": 1, "max": 860.8810901641846, "sum": 860.8810901641846, "min": 860.8810901641846}}, "EndTime": 1588083012.557382, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588083011.695581}
```

```
[04/28/2020 14:10:12 INFO 140478314145600] #progress_metric: host=algo-1, completed 84 % of epochs
```

```
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 3823, "sum": 3823.0, "min": 3823}, "Total Records Seen": {"count": 1, "max": 3804940, "sum": 3804940.0, "min": 3804940}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 43, "sum": 43.0, "min": 43}}, "EndTime": 1588083012.557717, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 41}, "StartTime": 1588083011.69647}
```

```
[04/28/2020 14:10:12 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=105145.132524 records/second
```

```
[04/28/2020 14:10:12 INFO 140478314145600] #quality_metric: host=algo-1, epoch=42, batch=0 train binary_classification_accuracy <score>=
```


0.748

```
[04/28/2020 14:10:12 INFO 140478314145600] #quality_metric: host=algo-1, epoch=42, batch=0 train binary_classification_cross_entropy <loss>=0.541251342773
```

```
[04/28/2020 14:10:12 INFO 140478314145600] #quality_metric: host=algo-1, epoch=42, batch=0 train binary_f_1.000 <score>=0.766233766234
```

```
[2020-04-28 14:10:13.411] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 86, "duration": 852, "num_examples": 91, "num_bytes": 5796480}
```

```
[04/28/2020 14:10:13 INFO 140478314145600] #quality_metric: host=algo-1, epoch=42, train binary_classification_accuracy <score>=0.741450549451
```

```
[04/28/2020 14:10:13 INFO 140478314145600] #quality_metric: host=algo-1, epoch=42, train binary_classification_cross_entropy <loss>=0.542913947808
```

```
[04/28/2020 14:10:13 INFO 140478314145600] #quality_metric: host=algo-1, epoch=42, train binary_f_1.000 <score>=0.773564567975
```

```
#metrics {"Metrics": {"update.time": {"count": 1, "max": 854.2070388793945, "sum": 854.2070388793945, "min": 854.2070388793945}}, "EndTime": 1588083013.412668, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588083012.557461}
```

```
[04/28/2020 14:10:13 INFO 140478314145600] #progress_metric: host=algo-1, completed 86 % of epochs
```

```
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 3914, "sum": 3914.0, "min": 3914}, "Total Records Seen": {"count": 1, "max": 3895510, "sum": 3895510.0, "min": 3895510}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 44, "sum": 44.0, "min": 44}}, "EndTime": 1588083013.413075, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 42}, "StartTime": 1588083012.558409}
```

```
[04/28/2020 14:10:13 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=105957.329922 records/second
```

```
[04/28/2020 14:10:13 INFO 140478314145600] #quality_metric: host=algo-1, epoch=43, batch=0 train binary_classification_accuracy <score>=0.746
```

```
[04/28/2020 14:10:13 INFO 140478314145600] #quality_metric: host=algo-1, epoch=43, batch=0 train binary_classification_cross_entropy <loss>=0.540339477539
```

```
[04/28/2020 14:10:13 INFO 140478314145600] #quality_metric: host=algo-1, epoch=43, batch=0 train binary_f_1.000 <score>=0.763940520446
```

```
[2020-04-28 14:10:14.337] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 88, "duration": 922, "num_examples": 91, "num_bytes": 5796480}
```

```
[04/28/2020 14:10:14 INFO 140478314145600] #quality_metric: host=algo-1, epoch=43, train binary_classification_accuracy <score>=0.741527472527
```

```
[04/28/2020 14:10:14 INFO 140478314145600] #quality_metric: host=algo-1, epoch=43, train binary_classification_cross_entropy <loss>=0.54198339173
```

```
[04/28/2020 14:10:14 INFO 140478314145600] #quality_metric: host=algo-1, epoch=43, train binary_f_1.000 <score>=0.773507689049
```

```
#metrics {"Metrics": {"update.time": {"count": 1, "max": 924.2420196
```

```
533203, "sum": 924.2420196533203, "min": 924.2420196533203}}, "EndTime": 1588083014.338072, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588083013.412777}
```

```
[04/28/2020 14:10:14 INFO 140478314145600] #progress_metric: host=algo-1, completed 88 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 4005, "sum": 4005.0, "min": 4005}, "Total Records Seen": {"count": 1, "max": 3986080, "sum": 3986080.0, "min": 3986080}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 45, "sum": 45.0, "min": 45}}, "EndTime": 1588083014.338509, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 43}, "StartTime": 1588083013.413799}
```

```
[04/28/2020 14:10:14 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=97931.2130301 records/second
```

```
[04/28/2020 14:10:14 INFO 140478314145600] #quality_metric: host=algo-1, epoch=44, batch=0 train binary_classification_accuracy <score>=0.747
```

```
[04/28/2020 14:10:14 INFO 140478314145600] #quality_metric: host=algo-1, epoch=44, batch=0 train binary_classification_cross_entropy <loss>=0.539455200195
```

```
[04/28/2020 14:10:14 INFO 140478314145600] #quality_metric: host=algo-1, epoch=44, batch=0 train binary_f_1.000 <score>=0.764651162791
```

```
[2020-04-28 14:10:15.220] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 90, "duration": 879, "num_examples": 91, "num_bytes": 5796480}
```

```
[04/28/2020 14:10:15 INFO 140478314145600] #quality_metric: host=algo-1, epoch=44, train binary_classification_accuracy <score>=0.741736263736
```

```
[04/28/2020 14:10:15 INFO 140478314145600] #quality_metric: host=algo-1, epoch=44, train binary_classification_cross_entropy <loss>=0.541080893716
```

```
[04/28/2020 14:10:15 INFO 140478314145600] #quality_metric: host=algo-1, epoch=44, train binary_f_1.000 <score>=0.773553273081
```

```
#metrics {"Metrics": {"update.time": {"count": 1, "max": 881.7050457000732, "sum": 881.7050457000732, "min": 881.7050457000732}}, "EndTime": 1588083015.220959, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588083014.338202}
```

```
[04/28/2020 14:10:15 INFO 140478314145600] #progress_metric: host=algo-1, completed 90 % of epochs
```

```
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 4096, "sum": 4096.0, "min": 4096}, "Total Records Seen": {"count": 1, "max": 4076650, "sum": 4076650.0, "min": 4076650}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 46, "sum": 46.0, "min": 46}}, "EndTime": 1588083015.221405, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm":
```

```
"factorization-machines", "epoch": 44}, "StartTime": 1588083014.339204}
```

```
[04/28/2020 14:10:15 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=102650.894776 records/second
```

```
[04/28/2020 14:10:15 INFO 140478314145600] #quality_metric: host=algo-1, epoch=45, batch=0 train binary_classification_accuracy <score>=0.748
```

```
[04/28/2020 14:10:15 INFO 140478314145600] #quality_metric: host=algo-1, epoch=45, batch=0 train binary_classification_cross_entropy <loss>=0.538597229004
```

```
[04/28/2020 14:10:15 INFO 140478314145600] #quality_metric: host=algo-1, epoch=45, batch=0 train binary_f_1.000 <score>=0.765799256506
```

```
[2020-04-28 14:10:16.088] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 92, "duration": 865, "num_examples": 91, "num_bytes": 5796480}
```

```
[04/28/2020 14:10:16 INFO 140478314145600] #quality_metric: host=algo-1, epoch=45, train binary_classification_accuracy <score>=0.741912087912
```

```
[04/28/2020 14:10:16 INFO 140478314145600] #quality_metric: host=algo-1, epoch=45, train binary_classification_cross_entropy <loss>=0.540205143855
```

```
[04/28/2020 14:10:16 INFO 140478314145600] #quality_metric: host=algo-1, epoch=45, train binary_f_1.000 <score>=0.773637642886
```

```
#metrics {"Metrics": {"update.time": {"count": 1, "max": 867.6221370697021, "sum": 867.6221370697021, "min": 867.6221370697021}}, "EndTime": 1588083016.089756, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588083015.221086}
```

```
[04/28/2020 14:10:16 INFO 140478314145600] #progress_metric: host=algo-1, completed 92 % of epochs
```

```
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 4187, "sum": 4187.0, "min": 4187}, "Total Records Seen": {"count": 1, "max": 4167220, "sum": 4167220.0, "min": 4167220}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 47, "sum": 47.0, "min": 47}}, "EndTime": 1588083016.090014, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 45}, "StartTime": 1588083015.222102}
```

```
[04/28/2020 14:10:16 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=104337.066671 records/second
```

```
[04/28/2020 14:10:16 INFO 140478314145600] #quality_metric: host=algo-1, epoch=46, batch=0 train binary_classification_accuracy <score>=0.748
```

```
[04/28/2020 14:10:16 INFO 140478314145600] #quality_metric: host=algo-1, epoch=46, batch=0 train binary_classification_cross_entropy <loss>=0.537764465332
```

```
[04/28/2020 14:10:16 INFO 140478314145600] #quality_metric: host=algo-1, epoch=46, batch=0 train binary_f_1.000 <score>=0.765799256506
```

```
[2020-04-28 14:10:16.962] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 94, "duration": 869, "num_examples": 91, "num_bytes": 5796480}
```

```
[04/28/2020 14:10:16 INFO 140478314145600] #quality_metric: host=algo-1, epoch=46, train binary_classification_accuracy <score>=0.742241
```

758242

```
[04/28/2020 14:10:16 INFO 140478314145600] #quality_metric: host=algo-1, epoch=46, train binary_classification_cross_entropy <loss>=0.539354917421
```

```
[04/28/2020 14:10:16 INFO 140478314145600] #quality_metric: host=algo-1, epoch=46, train binary_f_1.000 <score>=0.773896279159
#metrics {"Metrics": {"update.time": {"count": 1, "max": 872.2488880157471, "sum": 872.2488880157471, "min": 872.2488880157471}}, "EndTime": 1588083016.962958, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588083016.089838}
```

```
[04/28/2020 14:10:16 INFO 140478314145600] #progress_metric: host=algo-1, completed 94 % of epochs
```

```
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 4278, "sum": 4278.0, "min": 4278}, "Total Records Seen": {"count": 1, "max": 4257790, "sum": 4257790.0, "min": 4257790}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 48, "sum": 48.0, "min": 48}}, "EndTime": 1588083016.96333, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 46}, "StartTime": 1588083016.090681}
```

```
[04/28/2020 14:10:16 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=103774.226635 records/second
```

```
[04/28/2020 14:10:16 INFO 140478314145600] #quality_metric: host=algo-1, epoch=47, batch=0 train binary_classification_accuracy <score>=0.747
```

```
[04/28/2020 14:10:16 INFO 140478314145600] #quality_metric: host=algo-1, epoch=47, batch=0 train binary_classification_cross_entropy <loss>=0.536955688477
```

```
[04/28/2020 14:10:16 INFO 140478314145600] #quality_metric: host=algo-1, epoch=47, batch=0 train binary_f_1.000 <score>=0.765088207985
```

```
[2020-04-28 14:10:17.933] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 96, "duration": 967, "num_examples": 91, "num_bytes": 5796480}
```

```
[04/28/2020 14:10:17 INFO 140478314145600] #quality_metric: host=algo-1, epoch=47, train binary_classification_accuracy <score>=0.742505494505
```

```
[04/28/2020 14:10:17 INFO 140478314145600] #quality_metric: host=algo-1, epoch=47, train binary_classification_cross_entropy <loss>=0.538529082246
```

```
[04/28/2020 14:10:17 INFO 140478314145600] #quality_metric: host=algo-1, epoch=47, train binary_f_1.000 <score>=0.774066646097
```

```
#metrics {"Metrics": {"update.time": {"count": 1, "max": 970.0658321380615, "sum": 970.0658321380615, "min": 970.0658321380615}}, "EndTime": 1588083017.93413, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588083016.963019}
```

```
[04/28/2020 14:10:17 INFO 140478314145600] #progress_metric: host=algo-1, completed 96 % of epochs
```

```
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 4278, "sum": 4278.0, "min": 4278}, "Total Records Seen": {"count": 1, "max": 4257790, "sum": 4257790.0, "min": 4257790}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 48, "sum": 48.0, "min": 48}}, "EndTime": 1588083017.93413, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 47}, "StartTime": 1588083016.090681}
```



```
0.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 4369,
"sum": 4369.0, "min": 4369}, "Total Records Seen": {"count": 1, "ma
x": 4348360, "sum": 4348360.0, "min": 4348360}, "Max Records Seen Be
tween Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 905
70}, "Reset Count": {"count": 1, "max": 49, "sum": 49.0, "min": 4
9}}, "EndTime": 1588083017.934472, "Dimensions": {"Host": "algo-1",
"Meta": "training_data_iter", "Operation": "training", "Algorithm":
"factorization-machines", "epoch": 47}, "StartTime": 1588083016.9640
33}
```

```
[04/28/2020 14:10:17 INFO 140478314145600] #throughput_metric: host=
algo-1, train throughput=93315.992266 records/second
```

```
[04/28/2020 14:10:17 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=48, batch=0 train binary_classification_accuracy <score>=
0.745
```

```
[04/28/2020 14:10:17 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=48, batch=0 train binary_classification_cross_entropy <lo
ss>=0.536169921875
```

```
[04/28/2020 14:10:17 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=48, batch=0 train binary_f_1.000 <score>=0.763231197772
```

```
[2020-04-28 14:10:18.867] [tensorio] [info] epoch_stats={"data_pipel
ine": "/opt/ml/input/data/train", "epoch": 98, "duration": 931, "num
_examples": 91, "num_bytes": 5796480}
```

```
[04/28/2020 14:10:18 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=48, train binary_classification_accuracy <score>=0.742769
230769
```

```
[04/28/2020 14:10:18 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=48, train binary_classification_cross_entropy <loss>=0.53
772656686
```

```
[04/28/2020 14:10:18 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=48, train binary_f_1.000 <score>=0.774228395062
```

```
#metrics {"Metrics": {"update.time": {"count": 1, "max": 933.2239627
838135, "sum": 933.2239627838135, "min": 933.2239627838135}}, "EndTi
me": 1588083018.868475, "Dimensions": {"Host": "algo-1", "Operatio
n": "training", "Algorithm": "factorization-machines"}, "StartTime":
1588083017.934221}
```

```
[04/28/2020 14:10:18 INFO 140478314145600] #progress_metric: host=al
go-1, completed 98 % of epochs
```

```
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count":
1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Las
t Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number o
f Records Since Last Reset": {"count": 1, "max": 90570, "sum": 9057
0.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 4460,
"sum": 4460.0, "min": 4460}, "Total Records Seen": {"count": 1, "ma
x": 4438930, "sum": 4438930.0, "min": 4438930}, "Max Records Seen Be
tween Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 905
70}, "Reset Count": {"count": 1, "max": 50, "sum": 50.0, "min": 5
0}}, "EndTime": 1588083018.868777, "Dimensions": {"Host": "algo-1",
"Meta": "training_data_iter", "Operation": "training", "Algorithm":
"factorization-machines", "epoch": 48}, "StartTime": 1588083017.9351
51}
```

```
[04/28/2020 14:10:18 INFO 140478314145600] #throughput_metric: host=
algo-1, train throughput=96991.8322708 records/second
```

```
[04/28/2020 14:10:18 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=49, batch=0 train binary_classification_accuracy <score>=
0.745
```

```
[04/28/2020 14:10:18 INFO 140478314145600] #quality_metric: host=alg
o-1, epoch=49, batch=0 train binary_classification_cross_entropy <lo
ss>=0.53540612793
```

```
[04/28/2020 14:10:18 INFO 140478314145600] #quality_metric: host=algo-1, epoch=49, batch=0 train binary_f_1.000 <score>=0.763231197772
[2020-04-28 14:10:19.748] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 100, "duration": 877, "num_examples": 91, "num_bytes": 5796480}
[04/28/2020 14:10:19 INFO 140478314145600] #quality_metric: host=algo-1, epoch=49, train binary_classification_accuracy <score>=0.743021978022
[04/28/2020 14:10:19 INFO 140478314145600] #quality_metric: host=algo-1, epoch=49, train binary_classification_cross_entropy <loss>=0.536946357476
[04/28/2020 14:10:19 INFO 140478314145600] #quality_metric: host=algo-1, epoch=49, train binary_f_1.000 <score>=0.774352294109
[04/28/2020 14:10:19 INFO 140478314145600] #quality_metric: host=algo-1, train binary_classification_accuracy <score>=0.743021978022
[04/28/2020 14:10:19 INFO 140478314145600] #quality_metric: host=algo-1, train binary_classification_cross_entropy <loss>=0.536946357476
[04/28/2020 14:10:19 INFO 140478314145600] #quality_metric: host=algo-1, train binary_f_1.000 <score>=0.774352294109
#metrics {"Metrics": {"update.time": {"count": 1, "max": 879.9691200256348, "sum": 879.9691200256348, "min": 879.9691200256348}}, "EndTime": 1588083019.749571, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588083018.868544}

[04/28/2020 14:10:19 INFO 140478314145600] #progress_metric: host=algo-1, completed 100 % of epochs
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Batches Since Last Reset": {"count": 1, "max": 91, "sum": 91.0, "min": 91}, "Number of Records Since Last Reset": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Total Batches Seen": {"count": 1, "max": 4551, "sum": 4551.0, "min": 4551}, "Total Records Seen": {"count": 1, "max": 4529500, "sum": 4529500.0, "min": 4529500}, "Max Records Seen Between Resets": {"count": 1, "max": 90570, "sum": 90570.0, "min": 90570}, "Reset Count": {"count": 1, "max": 51, "sum": 51.0, "min": 51}}, "EndTime": 1588083019.749873, "Dimensions": {"Host": "algo-1", "Meta": "training_data_iter", "Operation": "training", "Algorithm": "factorization-machines", "epoch": 49}, "StartTime": 1588083018.869571}

[04/28/2020 14:10:19 INFO 140478314145600] #throughput_metric: host=algo-1, train throughput=102867.265496 records/second
[04/28/2020 14:10:19 WARNING 140478314145600] wait_for_all_workers will not sync workers since the kv store is not running distributed
[04/28/2020 14:10:19 INFO 140478314145600] Pulling entire model from kvstore to finalize
#metrics {"Metrics": {"finalize.time": {"count": 1, "max": 2.0270347595214844, "sum": 2.0270347595214844, "min": 2.0270347595214844}}, "EndTime": 1588083019.752371, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588083019.749658}

[04/28/2020 14:10:19 INFO 140478314145600] Saved checkpoint to "/tmp/tmpPfMR_a/state-0001.params"
[2020-04-28 14:10:19.759] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/test", "epoch": 0, "duration": 45753, "num_examples": 1, "num_bytes": 64000}
[2020-04-28 14:10:19.809] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/test", "epoch": 1, "duration": 49, "num_examples": 10, "num_bytes": 603520}
```

```
#metrics {"Metrics": {"Max Batches Seen Between Resets": {"count": 1, "max": 10, "sum": 10.0, "min": 10}, "Number of Batches Since Last Reset": {"count": 1, "max": 10, "sum": 10.0, "min": 10}, "Number of Records Since Last Reset": {"count": 1, "max": 9430, "sum": 9430.0, "min": 9430}, "Total Batches Seen": {"count": 1, "max": 10, "sum": 10.0, "min": 10}, "Total Records Seen": {"count": 1, "max": 9430, "sum": 9430.0, "min": 9430}, "Max Records Seen Between Resets": {"count": 1, "max": 9430, "sum": 9430.0, "min": 9430}, "Reset Count": {"count": 1, "max": 1, "sum": 1.0, "min": 1}}, "EndTime": 1588083019.809238, "Dimensions": {"Host": "algo-1", "Meta": "test_data_iter", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588083019.759544}
```

```
[04/28/2020 14:10:19 INFO 140478314145600] #test_score (algo-1) : ('binary_classification_accuracy', 0.684729586426299)
[04/28/2020 14:10:19 INFO 140478314145600] #test_score (algo-1) : ('binary_classification_cross_entropy', 0.5942866174691808)
[04/28/2020 14:10:19 INFO 140478314145600] #test_score (algo-1) : ('binary_f_1.000', 0.711443268950791)
[04/28/2020 14:10:19 INFO 140478314145600] #quality_metric: host=algo-1, test binary_classification_accuracy <score>=0.684729586426
[04/28/2020 14:10:19 INFO 140478314145600] #quality_metric: host=algo-1, test binary_classification_cross_entropy <loss>=0.594286617469
[04/28/2020 14:10:19 INFO 140478314145600] #quality_metric: host=algo-1, test binary_f_1.000 <score>=0.711443268951
#metrics {"Metrics": {"totaltime": {"count": 1, "max": 45855.4208278656, "sum": 45855.4208278656, "min": 45855.4208278656}, "setuptime": {"count": 1, "max": 41.6719913482666, "sum": 41.6719913482666, "min": 41.6719913482666}}, "EndTime": 1588083019.810621, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "factorization-machines"}, "StartTime": 1588083019.752441}
```

2020-04-28 14:10:31 Completed - Training job completed
 Training seconds: 111
 Billable seconds: 111

Step 4 : Create hyperparameter tuning jobs in SageMaker

In [51]:

```
# Use the best hyperparameters that give us the highest binary classification accuracy
fm.set_hyperparameters(feature_dim=n_features,
                        predictor_type='binary_classifier',
                        mini_batch_size=359,
                        num_factors=64,
                        epochs=403)
```

Step 5 : Deploy fm model and predict

In []:

```
#deploy fm model and get the prediction
fm_predictor = fm.deploy(initial_instance_count=1,
                          instance_type='ml.t2.medium')
```

-----!

In [171]:

```
#serializing data into JSON
import json
from sagemaker.predictor import json_deserializer

def fm_serializer(data):
    js = {'instances': []}
    for row in data:
        js['instances'].append({'features': row.tolist()})
    return json.dumps(js)

fm_predictor.content_type = 'application/json'
fm_predictor.serializer = fm_serializer
fm_predictor.deserializer = json_deserializer
```

In [172]:

```
#show prediction for a single record using endpoint
prediction = X_test[1000].toarray()
result = fm_predictor.predict(prediction)

print(y_test[1000])
print(result)
```

0.0

```
{'predictions': [{'score': 0.5855739712715149, 'predicted_label': 1.0}]}
```

In [356]:

```
#show predictions for a set of records(100 records)
predictions = []
for array in np.array_split(X_test[1000:1100].toarray(), 1):
    result = fm_predictor.predict(array)
    predictions += [r['predicted_label'] for r in result['predictions']]

predictions = np.array(predictions)
#predictions.shape
```


In [174]:

```
#show confusion matrix
pd.crosstab(y_test, predictions, rownames=['actuals'], colnames=['predictions'])
```

Out[174]:

	0.0	1.0
actuals		
0.0	49	12
1.0	22	17

Step 6 : Get the recommendation result for each user in user list

In [468]:

```
# get the corresponding recommendation movie index/location (not movieid)
def GetRecIndex(user_id):
    #get each user's sparse matrix(specify the range)
    loc_low=(user_id-1)*10
    loc_high=loc_low+10

    #store prediction pair result for each user into a list
    predictions = []
    for array in np.array_split(X_test[loc_low:loc_high].toarray(), 1):
        result = fm_predictor.predict(array)
        predictions += [r['predicted_label'] for r in result['predictions']]
    predictions = np.array(predictions)

    #add the movie index/location for finding the movieid later
    pred=result["predictions"]
    for i in pred:
        i["index"]=pred.index(i)
    #print(pred)

    # filter records with predicted_label being 1(recommend)
    pred_ft= filter(lambda dic: dic["predicted_label"]==1, pred)
    pred_filter=list(pred_ft)

    #get the top 5 movie recommendaions index for each user
    pred_filter.sort(key=lambda x: x['score'], reverse=True)
    #sort the score and get the top 5 records
    top5_index=pred_filter[:5]
    #agg_index=[i["index"] for i in top5_index]
    #agg_score=[i["score"] for i in top5_index]
    return top5_index
```

In [469]:

```
# write function to map recommendation movie id according to movie index/location
def GetRecMovieID(user_id):
    #Get the dataframe for each userid
    df_user=test_movie_ratings[test_movie_ratings.user_id==user_id]
    #Get the recommendation movie index for each user
    top5_index=GetRecIndex(user_id)
    loc=[i["index"] for i in top5_index]
    #map the index to the movieid
    recommendation=df_user.reset_index().movie_id[loc].values.tolist()
    return {'userId':user_id,'fm_recommendations': recommendation }
```

In [466]:

```
def UserFmDf(userId):
    agg_score=[i["score"] for i in GetRecIndex(userId)]
    a=pd.DataFrame(agg_score,columns=["predicted_score"])
    b=pd.DataFrame(GetRecMovieID(userId))
    c=pd.concat([a,b],axis=1)
    c.rename(columns={'fm_recommendations':'movieId'}, inplace=True)
    fm_result=pd.merge(c,test_pd,on=['userId','movieId'])
    fm_result=fm_result[["userId","movieId","predicted_score","rating"]]
    return fm_result
```

In [443]:

```
#try to get recommendations movieid for userid 198
GetRecMovieID(198)
```

Out[443]:

```
{'userId': 198, 'fm_recommendations': [100, 179, 498, 135, 7]}
```

In [467]:

```
#get the user recommendation summary dataframe
UserFmDf(198)
```

Out[467]:

	userId	movieId	predicted_score	rating
0	198	100	0.766967	1
1	198	179	0.720465	4
2	198	498	0.717518	3
3	198	135	0.700930	5
4	198	7	0.506485	4

In [228]:

```
user_list=[198,11,314,184,163,710,881,504,267,653]
```

In [444]:

```
# Show the recommendation result for each user in the list
fm_user_list=[GetRecMovieID(i) for i in user_list]
fm_user_list
```

Out[444]:

```
{'userId': 198, 'fm_recommendations': [100, 179, 498, 135, 7]},
{'userId': 11, 'fm_recommendations': [425, 558]},
{'userId': 314, 'fm_recommendations': [28, 95, 692, 417, 1518]},
{'userId': 184, 'fm_recommendations': [98, 191, 187, 153, 602]},
{'userId': 163, 'fm_recommendations': [98, 318, 64]},
{'userId': 710, 'fm_recommendations': [50, 197, 22, 116, 200]},
{'userId': 881, 'fm_recommendations': [180, 423, 663, 133]},
{'userId': 504, 'fm_recommendations': [66, 163, 581, 72, 179]},
{'userId': 267, 'fm_recommendations': [423, 238, 518, 980, 403]},
{'userId': 653, 'fm_recommendations': [50, 272]}
```

In [445]:

```
fm_user_result=pd.DataFrame(fm_user_list)
fm_user_result
```

Out[445]:

	fm_recommendations	userId
0	[100, 179, 498, 135, 7]	198
1	[425, 558]	11
2	[28, 95, 692, 417, 1518]	314
3	[98, 191, 187, 153, 602]	184
4	[98, 318, 64]	163
5	[50, 197, 22, 116, 200]	710
6	[180, 423, 663, 133]	881
7	[66, 163, 581, 72, 179]	504
8	[423, 238, 518, 980, 403]	267
9	[50, 272]	653

In [557]:

```

final_fm_user = []
for i in user_list:
    data = UserFmDf(i)
    # store DataFrame in list
    final_fm_user.append(data)
# see pd.concat documentation for more info
final_fm_user = pd.concat(final_fm_user)
final_fm_user.head(10)

```

Out[557]:

	userId	movieId	predicted_score	rating
0	198	100	0.766967	1
1	198	179	0.720465	4
2	198	498	0.717518	3
3	198	135	0.700930	5
4	198	7	0.506485	4
0	11	425	0.545703	4
1	11	558	0.517921	3
0	314	28	0.776762	5
1	314	95	0.749725	5
2	314	692	0.741075	5

In []:

```

#import sagemaker
#sagemaker.Session().delete_endpoint(fm_predictor.endpoint)

```

Model Combination : Vote Strategy

The scoring strategy shown below considers the effect of different rmse of different models and the predicted scores of each model. To make sure those the predicted score of each model is comparable, standardization transform process is implemented since we only care about the relative distance of each predicted score in a model. Next, a weight metric is constructed as using predicted score(after scaling) divided by the rmse of each model. The greater the weight is, the more accurate the recommendation result will be. Then, we add the weight of each pair(userId, movieid) generated by each model. By sorting the weight of different pairs, we can easily get the final recommendation result.

In [354]:

```
# print the prediction result of three models
print(als_user_result)
print(sp_user_result)
print(fm_user_result)
```

```
      als_recommendations  userId
0  [1463, 1643, 1536, 814, 1449]    198
1  [814, 1463, 1536, 1449, 1500]     11
2  [1554, 1662, 814, 1472, 867]    314
3  [1536, 1463, 1643, 814, 1449]    184
4  [1554, 814, 1463, 1500, 1293]    163
5  [1463, 1643, 1536, 814, 1449]    710
6  [1463, 814, 1554, 1536, 1500]    881
7  [814, 1463, 1449, 1500, 1293]    504
8  [1463, 1536, 1643, 814, 1449]    267
9  [814, 1463, 1554, 1500, 1449]    653
```

```
      sp_recomendations  userId
0  [114, 50, 474, 408, 169]    267
1   [83, 318, 9, 286, 603]     11
2  [50, 474, 127, 172, 173]    198
3  [488, 134, 285, 483, 9]    184
4  [64, 121, 692, 282, 143]    314
5  [316, 318, 272, 64, 357]    163
6  [318, 127, 735, 258, 98]    504
7  [195, 22, 174, 50, 746]    653
8  [127, 483, 134, 192, 23]    710
9  [174, 22, 651, 121, 265]    881
```

```
      fm_recommendations  userId
0  [100, 179, 498, 135, 7]    198
1           [425, 558]        11
2  [28, 95, 692, 417, 1518]   314
3  [98, 191, 187, 153, 602]   184
4           [98, 318, 64]    163
5  [50, 197, 22, 116, 200]    710
6  [180, 423, 663, 133]    881
7  [66, 163, 581, 72, 179]    504
8  [423, 238, 518, 980, 403]   267
9           [50, 272]        653
```

In [558]:

```
final_fm_user.head()
```

Out[558]:

	userId	movieId	predicted_score	rating
0	198	100	0.766967	1
1	198	179	0.720465	4
2	198	498	0.717518	3
3	198	135	0.700930	5
4	198	7	0.506485	4

In [550]:

```
als_user.head()
```

Out[550]:

	userId	movieId	predicted_rating
1350	881	1463.0	5.182528
1351	881	814.0	5.130446
1352	881	1554.0	4.892687
1353	881	1536.0	4.831121
1354	881	1500.0	4.821637

In [565]:

```
sp_user.head()
```

Out[565]:

	movieId	score	userId
71	114	5.000000	267
72	83	4.558399	11
85	50	4.537311	198
162	488	4.746187	184
264	64	5.000000	314

In [554]:

```
import sklearn
from sklearn import decomposition
# perform standardization transformation of the predicted rating
x = als_user.loc[:, ["predicted_rating"]].values
x = sklearn.preprocessing.StandardScaler().fit_transform(x)
als_user.drop(['predicted_rating'], axis=1)
als_user["predicted_rating"] = x
# recall the rmse of als estimator on test data is around 0.957
als_user["weight"] = als_user["predicted_rating"] * (1/0.957)
als_user.head()
```

```
/home/ec2-user/anaconda3/envs/python3/lib/python3.6/site-packages/ip
ykernel/__main__.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: http://pandas.pydata.org/panda
s-docs/stable/indexing.html#indexing-view-versus-copy
/home/ec2-user/anaconda3/envs/python3/lib/python3.6/site-packages/ip
ykernel/__main__.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: http://pandas.pydata.org/panda
s-docs/stable/indexing.html#indexing-view-versus-copy
```

Out[554]:

	userId	movieId	predicted_rating	weight
1350	881	1463.0	0.298999	0.312434
1351	881	814.0	0.213841	0.223449
1352	881	1554.0	-0.174917	-0.182777
1353	881	1536.0	-0.275582	-0.287964
1354	881	1500.0	-0.291090	-0.304169

In [525]:

```
# calculate the rmse for the fm prediction result
from sklearn.metrics import mean_squared_error
from math import sqrt

x = final_fm_user.loc[:, ["predicted_score", "rating"]].values
x = sklearn.preprocessing.StandardScaler().fit_transform(x)
RMSE_table = pd.DataFrame(x, columns=["predicted_score", "rating"])
RMSE_table.head()

rmse = sqrt(mean_squared_error(RMSE_table["predicted_score"], RMSE_table["rating"]))
print('Root mean squared error of the test_data for fm result: %.4f' % rmse)
```

```
Root mean squared error of the test_data for fm result: 1.3624
```

In [559]:

```
#perform the same transformation on fm recommendation result

x = final_fm_user.loc[:, ["predicted_score"]].values
x = sklearn.preprocessing.StandardScaler().fit_transform(x)
final_fm_user.drop(['predicted_score'], axis=1)
final_fm_user["predicted_score"] = x
final_fm_user["weight"] = final_fm_user["predicted_score"] * (1/1.36)
final_fm_user.head()
```

Out[559]:

	userId	movieId	predicted_score	rating	weight
0	198	100	0.758106	1	0.557431
1	198	179	0.377026	4	0.277225
2	198	498	0.352875	3	0.259467
3	198	135	0.216937	5	0.159513
4	198	7	-1.376539	4	-1.012161

In [566]:

```
# perform the same transformation on surprise result

x = sp_user.loc[:, ["score"]].values
x = sklearn.preprocessing.StandardScaler().fit_transform(x)
sp_user.drop(['score'], axis=1)
sp_user["score"] = x
sp_user["score"] = sp_user["score"] * 0.675
sp_user.rename(columns={'score': 'predicted_score'}, inplace=True)
sp_user["weight"] = sp_user["predicted_score"] * (1/0.675)
sp_user.head()
```

```
/home/ec2-user/anaconda3/envs/python3/lib/python3.6/site-packages/ip
ykernel/__main__.py:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: http://pandas.pydata.org/panda
s-docs/stable/indexing.html#indexing-view-versus-copy
/home/ec2-user/anaconda3/envs/python3/lib/python3.6/site-packages/ip
ykernel/__main__.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: http://pandas.pydata.org/panda
s-docs/stable/indexing.html#indexing-view-versus-copy
/home/ec2-user/anaconda3/envs/python3/lib/python3.6/site-packages/pa
ndas/core/frame.py:4025: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

```
See the caveats in the documentation: http://pandas.pydata.org/panda
s-docs/stable/indexing.html#indexing-view-versus-copy
return super(DataFrame, self).rename(**kwargs)
/home/ec2-user/anaconda3/envs/python3/lib/python3.6/site-packages/ip
ykernel/__main__.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: http://pandas.pydata.org/panda
s-docs/stable/indexing.html#indexing-view-versus-copy
```

Out[566]:

	movielfld	predicted_score	userId	weight
71	114	1.062711	267	1.574387
72	83	0.075229	11	0.111451
85	50	0.028073	198	0.041590
162	488	0.495149	184	0.733555
264	64	1.062711	314	1.574387

In [573]:

```
#concat the weight into one big dataframe
final_recommdend=pd.concat([als_user, final_fm_user], ignore_index=True)
final_recom=pd.concat([final_recommdend, sp_user], ignore_index=True)
final_recom
```

```
/home/ec2-user/anaconda3/envs/python3/lib/python3.6/site-packages/ipykernel/__main__.py:2: FutureWarning: Sorting because non-concatenation axis is not aligned. A future version of pandas will change to not sort by default.
```

To accept the future behavior, pass 'sort=False'.

To retain the current behavior and silence the warning, pass 'sort=True'.

```
from ipykernel import kernelapp as app
/home/ec2-user/anaconda3/envs/python3/lib/python3.6/site-packages/ipykernel/__main__.py:3: FutureWarning: Sorting because non-concatenation axis is not aligned. A future version of pandas will change to not sort by default.
```

To accept the future behavior, pass 'sort=False'.

To retain the current behavior and silence the warning, pass 'sort=True'.

```
app.launch_new_instance()
```

Out[573]:

	movielid	predicted_rating	predicted_score	rating	userId	weight
0	1463.0	0.298999	NaN	NaN	881	0.312434
1	814.0	0.213841	NaN	NaN	881	0.223449
2	1554.0	-0.174917	NaN	NaN	881	-0.182777
3	1536.0	-0.275582	NaN	NaN	881	-0.287964
4	1500.0	-0.291090	NaN	NaN	881	-0.304169
5	1554.0	-1.098930	NaN	NaN	163	-1.148308
6	814.0	-1.265069	NaN	NaN	163	-1.321911
7	1463.0	-1.568281	NaN	NaN	163	-1.638747
8	1500.0	-1.605603	NaN	NaN	163	-1.677746
9	1293.0	-1.778316	NaN	NaN	163	-1.858219
10	814.0	0.693009	NaN	NaN	504	0.724147
11	1463.0	0.278967	NaN	NaN	504	0.291502
12	1449.0	0.068417	NaN	NaN	504	0.071491
13	1500.0	0.043477	NaN	NaN	504	0.045431
14	1293.0	0.032644	NaN	NaN	504	0.034110
15	1554.0	1.161338	NaN	NaN	314	1.213519
16	1662.0	0.832116	NaN	NaN	314	0.869505
17	814.0	0.630385	NaN	NaN	314	0.658710
18	1472.0	0.508787	NaN	NaN	314	0.531648
19	867.0	0.409301	NaN	NaN	314	0.427692
20	1463.0	1.943722	NaN	NaN	267	2.031058
21	1536.0	1.881008	NaN	NaN	267	1.965525
22	1643.0	1.755153	NaN	NaN	267	1.834015
23	814.0	1.484503	NaN	NaN	267	1.551205
24	1449.0	1.323057	NaN	NaN	267	1.382505
25	814.0	-1.497798	NaN	NaN	653	-1.565097
26	1463.0	-1.694203	NaN	NaN	653	-1.770327
27	1554.0	-1.907174	NaN	NaN	653	-1.992867
28	1500.0	-1.944572	NaN	NaN	653	-2.031946
29	1449.0	-2.011618	NaN	NaN	653	-2.102004
...
111	474.0	NaN	1.062711	NaN	267	1.574387
112	9.0	NaN	-0.360072	NaN	11	-0.533440
113	127.0	NaN	-0.513063	NaN	198	-0.760094
114	285.0	NaN	0.357769	NaN	184	0.530028
115	692.0	NaN	0.959456	NaN	314	1.421416

	movieId	predicted_rating	predicted_score	rating	userId	weight
116	272.0	NaN	-1.358415	NaN	163	-2.012467
117	735.0	NaN	0.166878	NaN	504	0.247227
118	174.0	NaN	-0.599498	NaN	653	-0.888145
119	134.0	NaN	-0.117847	NaN	710	-0.174588
120	651.0	NaN	-0.042283	NaN	881	-0.062641
121	408.0	NaN	1.062711	NaN	267	1.574387
122	286.0	NaN	-0.387458	NaN	11	-0.574012
123	172.0	NaN	-0.649062	NaN	198	-0.961573
124	483.0	NaN	0.291518	NaN	184	0.431879
125	282.0	NaN	0.799044	NaN	314	1.183768
126	64.0	NaN	-1.436129	NaN	163	-2.127599
127	258.0	NaN	-0.040159	NaN	504	-0.059494
128	50.0	NaN	-0.694756	NaN	653	-1.029267
129	192.0	NaN	-0.118151	NaN	710	-0.175039
130	121.0	NaN	-0.048298	NaN	881	-0.071552
131	169.0	NaN	1.062711	NaN	267	1.574387
132	603.0	NaN	-0.424560	NaN	11	-0.628978
133	173.0	NaN	-0.654858	NaN	198	-0.970160
134	9.0	NaN	0.271087	NaN	184	0.401610
135	143.0	NaN	0.721880	NaN	314	1.069451
136	357.0	NaN	-1.589402	NaN	163	-2.354669
137	98.0	NaN	-0.281048	NaN	504	-0.416367
138	746.0	NaN	-1.017984	NaN	653	-1.508124
139	23.0	NaN	-0.276943	NaN	710	-0.410285
140	265.0	NaN	-0.061811	NaN	881	-0.091572

141 rows × 6 columns

In [574]:

```
#drop irrelevant columns and only retain the weight for pair comparision
final_recom['movieId']=final_recom['movieId'].astype(int)
final_recom=final_recom.drop(['predicted_score',"rating","predicted_rating"], axis=1)
final_recom.head()
```

Out[574]:

	movieId	userId	weight
0	1463	881	0.312434
1	814	881	0.223449
2	1554	881	-0.182777
3	1536	881	-0.287964
4	1500	881	-0.304169

In [578]:

```
summary=final_recom.sort_values(["userId","weight"],ascending=[True,False])
summary.head(10)
```

Out[578]:

	movieId	userId	weight
35	814	11	0.182833
92	83	11	0.111451
36	1463	11	0.033692
102	318	11	-0.086218
37	1536	11	-0.359141
38	1449	11	-0.398001
39	1500	11	-0.448859
112	9	11	-0.533440
122	286	11	-0.574012
132	603	11	-0.628978

In [583]:

```
#get the recommendations for each user in the list
final_df_198=summary[summary.userId==198].head(5)
final_df_11=summary[summary.userId==11].head(5)
final_df_314=summary[summary.userId==314].head(5)
final_df_184=summary[summary.userId==184].head(5)
final_df_163=summary[summary.userId==163].head(5)
final_df_710=summary[summary.userId==710].head(5)
final_df_881=summary[summary.userId==881].head(5)
final_df_504=summary[summary.userId==504].head(5)
final_df_267=summary[summary.userId==267].head(5)
final_df_653=summary[summary.userId==653].head(5)
```

In [589]:

```
#get the final result
df_result=pd.concat([final_df_198,final_df_11,final_df_314,final_df_184,
                    final_df_163,final_df_710,final_df_881,final_df_504,final_df_267,final_df_653])
df_result
```


Out[589]:

	movieId	userId	weight
50	100	198	0.557431
40	1463	198	0.294939
51	179	198	0.277225
52	498	198	0.259467
53	135	198	0.159513
35	814	11	0.182833
92	83	11	0.111451
36	1463	11	0.033692
102	318	11	-0.086218
37	1536	11	-0.359141
95	64	314	1.574387
105	121	314	1.426293
115	692	314	1.421416
15	1554	314	1.213519
125	282	314	1.183768
62	98	184	1.507711
63	191	184	1.124895
64	187	184	1.108101
94	488	184	0.733555
45	1536	184	0.672327
67	98	163	-0.759232
68	318	163	-0.848456
69	64	163	-0.998578
5	1554	163	-1.148308
96	316	163	-1.278732
70	50	710	1.552879
71	197	710	1.044064
30	1463	710	0.920188
31	1643	710	0.894612
32	1536	710	0.855661
0	1463	881	0.312434
1	814	881	0.223449
100	174	881	0.146528
110	22	881	0.009157
120	651	881	-0.062641
10	814	504	0.724147

	movieId	userId	weight
97	318	504	0.697395
107	127	504	0.325998
11	1463	504	0.291502
117	735	504	0.247227
20	1463	267	2.031058
21	1536	267	1.965525
22	1643	267	1.834015
91	114	267	1.574387
101	50	267	1.574387
98	195	653	0.028195
89	50	653	0.008720
108	22	653	-0.281046
90	272	653	-0.606939
118	174	653	-0.888145

In [590]:

```
df_result.to_csv("recommendation_for_user_list.csv")
```

Github

<https://github.com/Lyhq1996/LHQ> (<https://github.com/Lyhq1996/LHQ>)

In []: