ALS

In [26]:

```
import os
import time

# spark imports
from pyspark.sql import SparkSession
from pyspark.sql.functions import UserDefinedFunction, explode, desc
from pyspark.sql.types import StringType, ArrayType
from pyspark.mllib.recommendation import ALS

# data science imports
import math
import numpy as np
import pandas as pd

# visualization imports
import seaborn as sns
import matplotlib.pyplot as plt

%matplotlib inline
```

In [28]:

```
# spark config
spark = SparkSession \
    .builder \
    .appName("movie recommendation") \
    .config("spark.driver.maxResultSize", "96g") \
    .config("spark.driver.memory", "96g") \
    .config("spark.executor.memory", "8g") \
    .config("spark.master", "local[12]") \
    .getOrCreate()
# get spark context
sc = spark.sparkContext
```

In [29]:

```
data_path = "ml-20m"
```

In [30]:

```
movies = spark.read.load(os.path.join(data_path, 'movies.csv'), format='csv', header=True, inferSchema=True)
ratings = spark.read.load(os.path.join(data_path, 'ratings.csv'), format='csv', header=True, inferSchema=True)
links = spark.read.load(os.path.join(data_path, 'links.csv'), format='csv', header=True, inferSchema=True)
tags = spark.read.load(os.path.join(data_path, 'tags.csv'), format='csv', header=True, inferSchema=True)
```

In [31]:

```
# load data
movie_rating = sc.textFile(os.path.join(data_path, 'ratings.csv'))
# preprocess data -- only need ["userId", "movieId", "rating"]
header = movie_rating.take(1)[0]
rating_data = movie_rating \
    .filter(lambda line: line!=header) \
    .map(lambda line: line.split(",")) \
    .map(lambda tokens: (int(tokens[0]), int(tokens[1]), float(tokens[2]))) \
    .cache()
# check three rows
rating_data.take(3)
```

Out[31]:

```
[(1, 2, 3.5), (1, 29, 3.5), (1, 32, 3.5)]
```

In [32]:

```
small_data, drop_data = rating_data.randomSplit([3,7], seed=99)
small_data.cache()
train, validation, test = small_data.randomSplit([6, 2, 2], seed=99)
# cache data
train.cache()
validation.cache()
test.cache()
```

Out[32]:

```
PythonRDD[44] at RDD at PythonRDD.scala:53
```

In [36]:

```
def train_ALS(train_data, validation_data, num_iters, reg_param, ranks):
    """
    Grid Search Function to select the best model based on RMSE of hold-out data
    """
    # initial
    min_error = float('inf')
    best_rank = -1
    best_regularization = 0
    best_model = None
    for iters in num_iters:
        for rank in ranks:
            for reg in reg_param:
                # train ALS model
                model = ALS.train(
                    ratings=train_data,    # (userID, productID, rating) tuple
                    iterations=iters,
                    rank=rank,
                    lambda_=reg,           # regularization param
                    seed=99)
                # make prediction
                valid_data = validation_data.map(lambda p: (p[0], p[1]))
                predictions = model.predictAll(valid_data).map(lambda r: ((r[0],
r[1]), r[2]))
                # get the rating result
                ratesAndPreds = validation_data.map(lambda r: ((r[0], r[1]), r[2
])).join(predictions)
                # get the RMSE
                MSE = ratesAndPreds.map(lambda r: (r[1][0] - r[1][1])**2).mean()
                error = math.sqrt(MSE)
                print('{} latent factors, {} iterations and regularization = {}:
validation RMSE is {}'.format(rank,iters, reg, error))
                if error < min_error:
                    min_error = error
                    best_rank = rank
                    best_regularization = reg
                    best_model = model
                    best_iterations=iters
    print('\nThe best model has {} latent factors,{} iterations and regularizati
on = {}'.format(best_rank,best_iterations, best_regularization))
    return best_model
```

In [37]:

```
# hyper-param config
num_iterations = [5,10,15,20]
ranks = [8, 10, 12, 14, 16, 18, 20]
reg_params = [0.001, 0.01, 0.05, 0.1, 0.2]

# grid search and select best model
start_time = time.time()
final_model = train_ALS(train, validation, num_iterations, reg_params, ranks)

print ('Total Runtime: {:.2f} seconds'.format(time.time() - start_time))
```

```
8 latent factors, 5 iterations and regularization = 0.001: validatio
n RMSE is 1.0978310055945717
8 latent factors, 5 iterations and regularization = 0.01: validation
RMSE is 0.9512585699486441
8 latent factors, 5 iterations and regularization = 0.05: validation
RMSE is 0.9029716434021641
8 latent factors, 5 iterations and regularization = 0.1: validation
RMSE is 0.8653390023472605
8 latent factors, 5 iterations and regularization = 0.2: validation
RMSE is 0.8662126445515408
10 latent factors, 5 iterations and regularization = 0.001: validati
on RMSE is 1.1313538637183422
10 latent factors, 5 iterations and regularization = 0.01: validatio
n RMSE is 0.9682460863797264
10 latent factors, 5 iterations and regularization = 0.05: validatio
n RMSE is 0.9038032451482121
10 latent factors, 5 iterations and regularization = 0.1: validation
RMSE is 0.8605333006687613
10 latent factors, 5 iterations and regularization = 0.2: validation
RMSE is 0.8642836044308183
12 latent factors, 5 iterations and regularization = 0.001: validati
on RMSE is 1.1770767772963115
12 latent factors, 5 iterations and regularization = 0.01: validatio
n RMSE is 0.9827213765926149
12 latent factors, 5 iterations and regularization = 0.05: validatio
n RMSE is 0.9120792844855544
12 latent factors, 5 iterations and regularization = 0.1: validation
RMSE is 0.8618355140199258
12 latent factors, 5 iterations and regularization = 0.2: validation
RMSE is 0.8632999078204758
14 latent factors, 5 iterations and regularization = 0.001: validati
on RMSE is 1.2258499391386017
14 latent factors, 5 iterations and regularization = 0.01: validatio
n RMSE is 1.0028279085655245
14 latent factors, 5 iterations and regularization = 0.05: validatio
n RMSE is 0.9210242806970018
14 latent factors, 5 iterations and regularization = 0.1: validation
RMSE is 0.8650480416547903
14 latent factors, 5 iterations and regularization = 0.2: validation
RMSE is 0.8646173146775767
```

```
16 latent factors, 5 iterations and regularization = 0.001: validati
on RMSE is 1.2490077613155428
16 latent factors, 5 iterations and regularization = 0.01: validatio
n RMSE is 1.0128096451638764
16 latent factors, 5 iterations and regularization = 0.05: validatio
n RMSE is 0.9298406576807708
16 latent factors, 5 iterations and regularization = 0.1: validation
RMSE is 0.8683283329529465
16 latent factors, 5 iterations and regularization = 0.2: validation
RMSE is 0.8649730153143963
18 latent factors, 5 iterations and regularization = 0.001: validati
on RMSE is 1.2932179011382785
18 latent factors, 5 iterations and regularization = 0.01: validatio
n RMSE is 1.0253272867599996
18 latent factors, 5 iterations and regularization = 0.05: validatio
n RMSE is 0.9401995541798129
18 latent factors, 5 iterations and regularization = 0.1: validation
RMSE is 0.8754341411789612
18 latent factors, 5 iterations and regularization = 0.2: validation
RMSE is 0.8673439170139036
20 latent factors, 5 iterations and regularization = 0.001: validati
on RMSE is 1.3322257673208489
20 latent factors, 5 iterations and regularization = 0.01: validatio
n RMSE is 1.036870360517641
20 latent factors, 5 iterations and regularization = 0.05: validatio
n RMSE is 0.9386947550839789
20 latent factors, 5 iterations and regularization = 0.1: validation
RMSE is 0.8702032340322843
20 latent factors, 5 iterations and regularization = 0.2: validation
RMSE is 0.865110779329479
8 latent factors, 10 iterations and regularization = 0.001: validati
on RMSE is 1.0668352357022106
8 latent factors, 10 iterations and regularization = 0.01: validatio
n RMSE is 0.9797928545199768
8 latent factors, 10 iterations and regularization = 0.05: validatio
n RMSE is 0.8878754890133085
8 latent factors, 10 iterations and regularization = 0.1: validation
RMSE is 0.8543423428154189
8 latent factors, 10 iterations and regularization = 0.2: validation
RMSE is 0.8700759892641091
10 latent factors, 10 iterations and regularization = 0.001: validat
ion RMSE is 1.0970635899065795
10 latent factors, 10 iterations and regularization = 0.01: validati
on RMSE is 0.9994364855465895
10 latent factors, 10 iterations and regularization = 0.05: validati
on RMSE is 0.8915847131747107
10 latent factors, 10 iterations and regularization = 0.1: validatio
n RMSE is 0.8525377629672534
10 latent factors, 10 iterations and regularization = 0.2: validatio
n RMSE is 0.8689926178741824
12 latent factors, 10 iterations and regularization = 0.001: validat
ion RMSE is 1.1389480646145866
12 latent factors, 10 iterations and regularization = 0.01: validati
```

```
on RMSE is 1.0232154522553347
12 latent factors, 10 iterations and regularization = 0.05: validati
on RMSE is 0.8972473210199161
12 latent factors, 10 iterations and regularization = 0.1: validatio
n RMSE is 0.8530382679253228
12 latent factors, 10 iterations and regularization = 0.2: validatio
n RMSE is 0.8687290498917214
14 latent factors, 10 iterations and regularization = 0.001: validat
ion RMSE is 1.172881546924749
14 latent factors, 10 iterations and regularization = 0.01: validati
on RMSE is 1.0394405322468783
14 latent factors, 10 iterations and regularization = 0.05: validati
on RMSE is 0.9023847265042546
14 latent factors, 10 iterations and regularization = 0.1: validatio
n RMSE is 0.853817087280848
14 latent factors, 10 iterations and regularization = 0.2: validatio
n RMSE is 0.8691475912011367
16 latent factors, 10 iterations and regularization = 0.001: validat
ion RMSE is 1.2055981178992488
16 latent factors, 10 iterations and regularization = 0.01: validati
on RMSE is 1.054084161409032
16 latent factors, 10 iterations and regularization = 0.05: validati
on RMSE is 0.9094244673529962
16 latent factors, 10 iterations and regularization = 0.1: validatio
n RMSE is 0.8562320138375279
16 latent factors, 10 iterations and regularization = 0.2: validatio
n RMSE is 0.869421319752378
18 latent factors, 10 iterations and regularization = 0.001: validat
ion RMSE is 1.2347057383132805
18 latent factors, 10 iterations and regularization = 0.01: validati
on RMSE is 1.069838483595508
18 latent factors, 10 iterations and regularization = 0.05: validati
on RMSE is 0.9148743287786926
18 latent factors, 10 iterations and regularization = 0.1: validatio
n RMSE is 0.8586895697783362
18 latent factors, 10 iterations and regularization = 0.2: validatio
n RMSE is 0.8703617748411678
20 latent factors, 10 iterations and regularization = 0.001: validat
ion RMSE is 1.2606334401183292
20 latent factors, 10 iterations and regularization = 0.01: validati
on RMSE is 1.0801918150466085
20 latent factors, 10 iterations and regularization = 0.05: validati
on RMSE is 0.9145875562221466
20 latent factors, 10 iterations and regularization = 0.1: validatio
n RMSE is 0.8562461999308856
20 latent factors, 10 iterations and regularization = 0.2: validatio
n RMSE is 0.8694655057243695
8 latent factors, 15 iterations and regularization = 0.001: validati
on RMSE is 1.0929999853959134
8 latent factors, 15 iterations and regularization = 0.01: validatio
n RMSE is 0.9872516844354393
8 latent factors, 15 iterations and regularization = 0.05: validatio
n RMSE is 0.8791049972751807
```

```
8 latent factors, 15 iterations and regularization = 0.1: validation
RMSE is 0.8503109468116004
8 latent factors, 15 iterations and regularization = 0.2: validation
RMSE is 0.8716816705579618
10 latent factors, 15 iterations and regularization = 0.001: validat
ion RMSE is 1.1231376022716968
10 latent factors, 15 iterations and regularization = 0.01: validati
on RMSE is 1.0104484938252845
10 latent factors, 15 iterations and regularization = 0.05: validati
on RMSE is 0.8839657532676076
10 latent factors, 15 iterations and regularization = 0.1: validatio
n RMSE is 0.8496756626386992
10 latent factors, 15 iterations and regularization = 0.2: validatio
n RMSE is 0.8713625201460728
12 latent factors, 15 iterations and regularization = 0.001: validat
ion RMSE is 1.1674502609245536
12 latent factors, 15 iterations and regularization = 0.01: validati
on RMSE is 1.0364739464658714
12 latent factors, 15 iterations and regularization = 0.05: validati
on RMSE is 0.8884948276741487
12 latent factors, 15 iterations and regularization = 0.1: validatio
n RMSE is 0.8499184239259753
12 latent factors, 15 iterations and regularization = 0.2: validatio
n RMSE is 0.8711872231208159
14 latent factors, 15 iterations and regularization = 0.001: validat
ion RMSE is 1.1967255530585565
14 latent factors, 15 iterations and regularization = 0.01: validati
on RMSE is 1.0522152608336772
14 latent factors, 15 iterations and regularization = 0.05: validati
on RMSE is 0.8914124928567952
14 latent factors, 15 iterations and regularization = 0.1: validatio
n RMSE is 0.8498169879648363
14 latent factors, 15 iterations and regularization = 0.2: validatio
n RMSE is 0.8714190725514253
16 latent factors, 15 iterations and regularization = 0.001: validat
ion RMSE is 1.2313659620883093
16 latent factors, 15 iterations and regularization = 0.01: validati
on RMSE is 1.0682795212777194
16 latent factors, 15 iterations and regularization = 0.05: validati
on RMSE is 0.8969806464111626
16 latent factors, 15 iterations and regularization = 0.1: validatio
n RMSE is 0.8513657974062605
16 latent factors, 15 iterations and regularization = 0.2: validatio
n RMSE is 0.8714444324055209
18 latent factors, 15 iterations and regularization = 0.001: validat
ion RMSE is 1.2549533290352928
18 latent factors, 15 iterations and regularization = 0.01: validati
on RMSE is 1.083749585620484
18 latent factors, 15 iterations and regularization = 0.05: validati
on RMSE is 0.9003124275349262
18 latent factors, 15 iterations and regularization = 0.1: validatio
n RMSE is 0.8524647613360298
18 latent factors, 15 iterations and regularization = 0.2: validatio
```

```
n RMSE is 0.8717314501921022
20 latent factors, 15 iterations and regularization = 0.001: validat
ion RMSE is 1.280119518233305
20 latent factors, 15 iterations and regularization = 0.01: validati
on RMSE is 1.0928701547581283
20 latent factors, 15 iterations and regularization = 0.05: validati
on RMSE is 0.9002512540831895
20 latent factors, 15 iterations and regularization = 0.1: validatio
n RMSE is 0.8510715495854642
20 latent factors, 15 iterations and regularization = 0.2: validatio
n RMSE is 0.8714230583680392
8 latent factors, 20 iterations and regularization = 0.001: validati
on RMSE is 1.1157639272981585
8 latent factors, 20 iterations and regularization = 0.01: validatio
n RMSE is 0.9863878807108679
8 latent factors, 20 iterations and regularization = 0.05: validatio
n RMSE is 0.8747722942706798
8 latent factors, 20 iterations and regularization = 0.1: validation
RMSE is 0.8486195550302413
8 latent factors, 20 iterations and regularization = 0.2: validation
RMSE is 0.8721162477087824
10 latent factors, 20 iterations and regularization = 0.001: validat
ion RMSE is 1.1498307467702726
10 latent factors, 20 iterations and regularization = 0.01: validati
on RMSE is 1.0121976436909395
10 latent factors, 20 iterations and regularization = 0.05: validati
on RMSE is 0.8798114478825502
10 latent factors, 20 iterations and regularization = 0.1: validatio
n RMSE is 0.8483177489822764
10 latent factors, 20 iterations and regularization = 0.2: validatio
n RMSE is 0.8720052009746228
12 latent factors, 20 iterations and regularization = 0.001: validat
ion RMSE is 1.1955482009526481
12 latent factors, 20 iterations and regularization = 0.01: validati
on RMSE is 1.0381484142263435
12 latent factors, 20 iterations and regularization = 0.05: validati
on RMSE is 0.8839520736018891
12 latent factors, 20 iterations and regularization = 0.1: validatio
n RMSE is 0.8484779732888454
12 latent factors, 20 iterations and regularization = 0.2: validatio
n RMSE is 0.8719710649029855
14 latent factors, 20 iterations and regularization = 0.001: validat
ion RMSE is 1.225536440318509
14 latent factors, 20 iterations and regularization = 0.01: validati
on RMSE is 1.05320786354733
14 latent factors, 20 iterations and regularization = 0.05: validati
on RMSE is 0.8856895862590016
14 latent factors, 20 iterations and regularization = 0.1: validatio
n RMSE is 0.8481593799931745
14 latent factors, 20 iterations and regularization = 0.2: validatio
n RMSE is 0.8720399738112924
16 latent factors, 20 iterations and regularization = 0.001: validat
ion RMSE is 1.2574541409567934
```

```
16 latent factors, 20 iterations and regularization = 0.01: validati
on RMSE is 1.0702835909123953
16 latent factors, 20 iterations and regularization = 0.05: validati
on RMSE is 0.8897832570518707
16 latent factors, 20 iterations and regularization = 0.1: validatio
n RMSE is 0.8488731207788847
16 latent factors, 20 iterations and regularization = 0.2: validatio
n RMSE is 0.872058061768827
18 latent factors, 20 iterations and regularization = 0.001: validat
ion RMSE is 1.2791840383848743
18 latent factors, 20 iterations and regularization = 0.01: validati
on RMSE is 1.084405995485733
18 latent factors, 20 iterations and regularization = 0.05: validati
on RMSE is 0.8925269177193539
18 latent factors, 20 iterations and regularization = 0.1: validatio
n RMSE is 0.8495973398823542
18 latent factors, 20 iterations and regularization = 0.2: validatio
n RMSE is 0.8721151083857303
20 latent factors, 20 iterations and regularization = 0.001: validat
ion RMSE is 1.3056908869609647
20 latent factors, 20 iterations and regularization = 0.01: validati
on RMSE is 1.0933570274513387
20 latent factors, 20 iterations and regularization = 0.05: validati
on RMSE is 0.8923672098871738
20 latent factors, 20 iterations and regularization = 0.1: validatio
n RMSE is 0.8487349964131898
20 latent factors, 20 iterations and regularization = 0.2: validatio
n RMSE is 0.8720189893076983

The best model has 14 latent factors,20 iterations and regularizatio
n = 0.1
Total Runtime: 15790.24 seconds
```

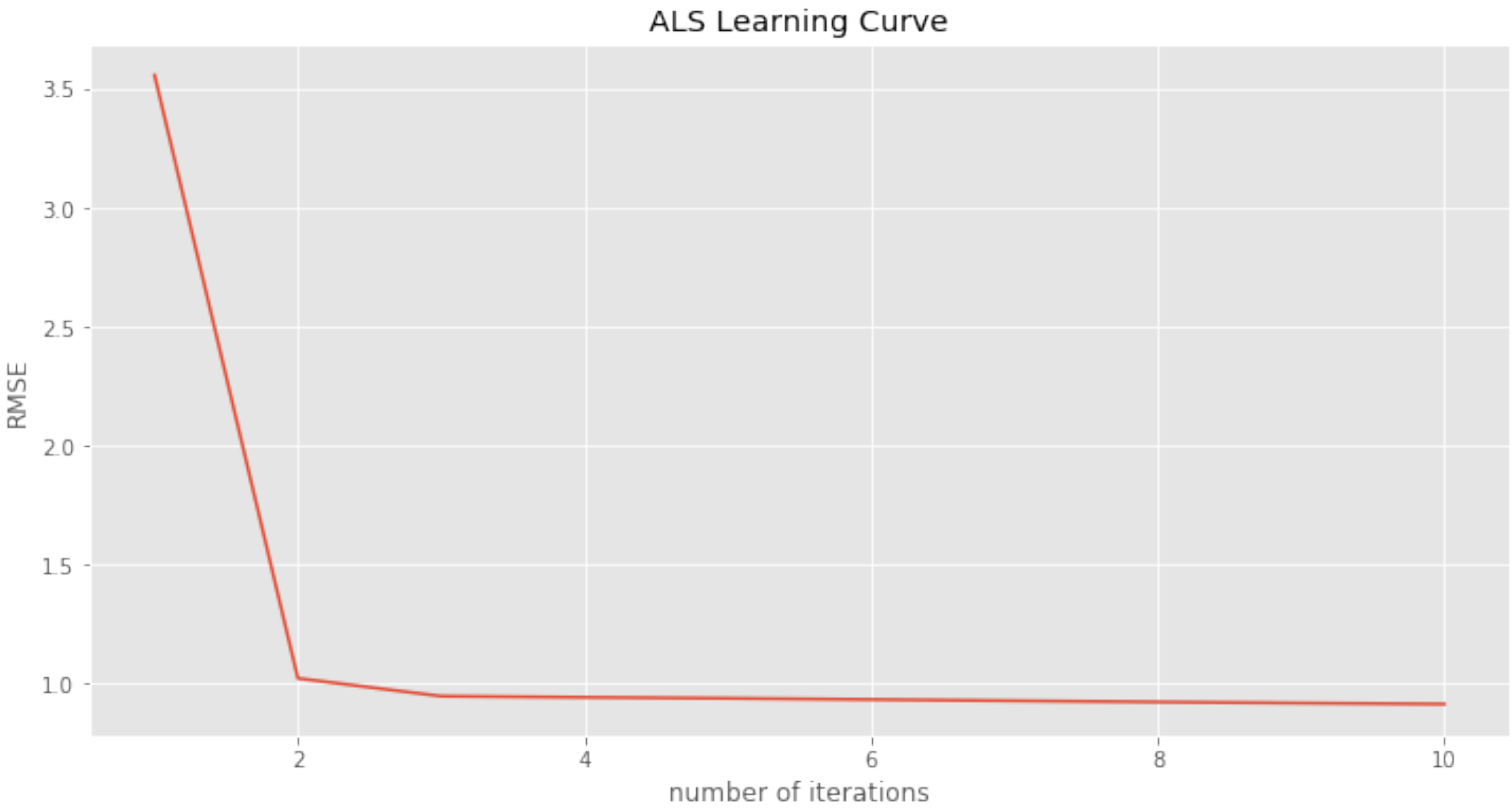
In [38]:

```
def plot_learning_curve(arr_iters, train_data, validation_data, reg, rank):
    """
    Plot function to show learning curve of ALS
    """
    errors = []
    for num_iters in arr_iters:
        # train ALS model
        model = ALS.train(
            ratings=train_data,    # (userID, productID, rating) tuple
            iterations=num_iters,
            rank=rank,
            lambda_=reg,           # regularization param
            seed=99)
        # make prediction
        valid_data = validation_data.map(lambda p: (p[0], p[1]))
        predictions = model.predictAll(valid_data).map(lambda r: ((r[0], r[1]),
r[2]))
        # get the rating result
        ratesAndPreds = validation_data.map(lambda r: ((r[0], r[1]), r[2])).join
(predictions)
        # get the RMSE
        MSE = ratesAndPreds.map(lambda r: (r[1][0] - r[1][1])**2).mean()
        error = math.sqrt(MSE)
        # add to errors
        errors.append(error)

    # plot
    plt.figure(figsize=(12, 6))
    plt.plot(arr_iters, errors)
    plt.xlabel('number of iterations')
    plt.ylabel('RMSE')
    plt.title('ALS Learning Curve')
    plt.grid(True)
    plt.show()
```

In [39]:

```
# create an array of num_iters
iter_array = list(range(1, 11))
# create learning curve plot
plot_learning_curve(iter_array, train, validation, 0.05, 20)
```



In [40]:

```
# make prediction using test data
test_data = test.map(lambda p: (p[0], p[1]))
predictions = final_model.predictAll(test_data).map(lambda r: ((r[0], r[1]), r[2
]))
# get the rating result
ratesAndPreds = test.map(lambda r: ((r[0], r[1]), r[2])).join(predictions)
# get the RMSE
MSE = ratesAndPreds.map(lambda r: (r[1][0] - r[1][1])**2).mean()
error = math.sqrt(MSE)
print('The out-of-sample RMSE of rating predictions is', round(error, 4))
```

```
The out-of-sample RMSE of rating predictions is 0.8492
```

In [57]:

```
def recommend_to_user_top_n(u,n):
    for i in final_model.recommendProducts(u,n):
        movie_name=df_movies.loc[df_movies['movieId']==i[1],'title'].iloc[0]
        print(movie_name)
```

In [58]:

```
recommend_to_user_top_n(196,10)
```

```
Batman & Mr. Freeze: Subzero (1998)
Brother Minister: The Assassination of Malcolm X (1994)
Long Night's Journey Into Day (2000)
Dishonored (1931)
Little Women (1949)
Cat Came Back, The (1988)
Dylan Moran: Yeah, Yeah (2011)
Geri's Game (1997)
Mulan (2009)
For the Birds (2000)
```

In [ ]:

```

```