```
pip install pyspark
```

```
Collecting pyspark
  Downloading pyspark-3.5.2.tar.gz (317.3 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 317.3/317.3 MB 2.8 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: py4j==0.10.9.7 in /usr/local/lib/python3.10/dist-packages (from pyspark) (0.10.9.7)
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) ... done
  Created wheel for pyspark: filename=pyspark-3.5.2-py2.py3-none-any.whl size=317812365 sha256=cdf2c53550ca95e2271f3aae90cc0901fdc95
  Stored in directory: /root/.cache/pip/wheels/34/34/bd/03944534c44b677cd5859f248090daa9fb27b3c8f8e5f49574
Successfully built pyspark
Installing collected packages: pyspark
Successfully installed pyspark-3.5.2
```

```
import pandas as pd
from pyspark.sql import SparkSession
spark = SparkSession.builder.master("local").appName(
    "Movie_recommendation").config(
        "spark.some.config.option","some-value").getOrCreate()


rating=spark.read.csv(
    '/content/ratings_small.csv',header=True,inferSchema=True)
rating=rating.drop('timestamp')
movies=spark.read.csv(
    '/content/movies_metadata.csv',header=True,inferSchema=True)
#Rename the 'id' column to 'movieId' in the movies DataFrame.
movies = movies.withColumnRenamed('id','movieId')
movie_data_vishnav=rating.join(movies,on='movieId')
Columns=len(movie_data_vishnav.columns)
Rows=movie_data_vishnav.count()
print('Number of Columns: {}\nNumber of Rows: {}'.format(Columns,Rows))
movie_data_vishnav.columns
```

```
Number of Columns: 26
Number of Rows: 44925
['movieId',
 'userId',
 'rating',
 'adult',
 'belongs_to_collection',
 'budget',
 'genres',
 'homepage',
 'imdb_id',
 'original_language',
 'original_title',
 'overview',
 'popularity',
 'poster_path',
 'production_companies',
 'production_countries',
 'release_date',
 'revenue',
 'runtime',
 'spoken_languages',
 'status',
 'tagline',
 'title',
 'video',
 'vote_average',
 'vote_count']
```

Data Cleaning: Replace all the zeros in the above mentioned fields with NaN

```
import numpy as np
from pyspark.sql.functions import when
movie_data_vishnav=movie_data_vishnav.withColumn(
    "userId",when(movie_data_vishnav.userId==0,np.nan).otherwise(movie_data_vishnav.userId))
movie_data_vishnav=movie_data_vishnav.withColumn(
    "movieId",when(movie_data_vishnav.movieId==0,np.nan).otherwise(movie_data_vishnav.movieId))
movie_data_vishnav=movie_data_vishnav.withColumn(
    "rating",when(movie_data_vishnav.rating==0,np.nan).otherwise(movie_data_vishnav.rating))
movie_data_vishnav=movie_data_vishnav.withColumn(
    "title",when(movie_data_vishnav.title==0,np.nan).otherwise(movie_data_vishnav.title))
from pyspark.sql.types import IntegerType
movie_data_vishnav=movie_data_vishnav.withColumn(
    "budget", movie_data_vishnav["budget"].cast(IntegerType()))
movie_data_vishnav.show()
```

```
+-------+------+------+-----+--------------------+--------+--------------------+--------------------+---------+----------------+-
|movieId|userId|rating|adult|belongs_to_collection|  budget|              genres|            homepage|  imdb_id|original_language|
+-------+------+------+-----+--------------------+--------+--------------------+--------------------+---------+----------------+-
|  949.0| 647.0|   4.0|False|                NULL|60000000|[{'id': 28, 'name...|                NULL|tt0113277|              en|
|  949.0| 564.0|   3.0|False|                NULL|60000000|[{'id': 28, 'name...|                NULL|tt0113277|              en|
|  949.0| 558.0|   4.0|False|                NULL|60000000|[{'id': 28, 'name...|                NULL|tt0113277|              en|
|  949.0| 547.0|   4.0|False|                NULL|60000000|[{'id': 28, 'name...|                NULL|tt0113277|              en|
|  949.0| 537.0|   3.0|False|                NULL|60000000|[{'id': 28, 'name...|                NULL|tt0113277|              en|
|  949.0| 509.0|   2.0|False|                NULL|60000000|[{'id': 28, 'name...|                NULL|tt0113277|              en|
|  949.0| 505.0|   3.5|False|                NULL|60000000|[{'id': 28, 'name...|                NULL|tt0113277|              en|
|  949.0| 452.0|   4.5|False|                NULL|60000000|[{'id': 28, 'name...|                NULL|tt0113277|              en|
|  949.0| 387.0|   5.0|False|                NULL|60000000|[{'id': 28, 'name...|                NULL|tt0113277|              en|
|  949.0| 363.0|   4.0|False|                NULL|60000000|[{'id': 28, 'name...|                NULL|tt0113277|              en|
|  949.0| 311.0|   3.0|False|                NULL|60000000|[{'id': 28, 'name...|                NULL|tt0113277|              en|
|  949.0| 263.0|   3.0|False|                NULL|60000000|[{'id': 28, 'name...|                NULL|tt0113277|              en|
|  949.0| 242.0|   5.0|False|                NULL|60000000|[{'id': 28, 'name...|                NULL|tt0113277|              en|
|  949.0| 232.0|   2.0|False|                NULL|60000000|[{'id': 28, 'name...|                NULL|tt0113277|              en|
|  949.0| 102.0|   4.0|False|                NULL|60000000|[{'id': 28, 'name...|                NULL|tt0113277|              en|
|  949.0|  23.0|   3.5|False|                NULL|60000000|[{'id': 28, 'name...|                NULL|tt0113277|              en|
|  710.0| 390.0|   2.0|False| {'id': 645, 'name...|58000000|[{'id': 12, 'name...|http://www.mgm.co...|tt0113189|              en|
|  710.0| 358.0|   1.0|False| {'id': 645, 'name...|58000000|[{'id': 12, 'name...|http://www.mgm.co...|tt0113189|              en|
| 1408.0| 665.0|   3.0|False|                NULL|98000000|[{'id': 28, 'name...|                NULL|tt0112760|              en|Cu
| 1408.0| 658.0|   5.0|False|                NULL|98000000|[{'id': 28, 'name...|                NULL|tt0112760|              en|Cu
+-------+------+------+-----+--------------------+--------+--------------------+--------------------+---------+----------------+-
only showing top 20 rows
```

Building the Recommendatin model using ALS on the training data

```
(training,test)=movie_data.randomSplit([0.8,0.2])


from pyspark.ml.evaluation import RegressionEvaluator
from pyspark.ml.recommendation import ALS

# we set cold strategy to 'drop' to ensure we dont get NaN evaluation
als_vishnav=ALS(maxIter=5,regParam=0.09,rank=25,userCol='userId',itemCol='movieId',coldStartStrategy="drop",nonnegative=True)
model_vishnav=als_vishnav.fit(training)


evaluator=RegressionEvaluator(metricName="rmse",labelCol="rating",predictionCol="prediction")
predictions=model_vishnav.transform(test)
rmse=evaluator.evaluate(predictions)
print("RMSE="+str(rmse))
predictions.show()
```

```
RMSE=0.9195242773432019
+-------+------+------+--------------+--------------------+----------+--------------------+--------------------+----------
|movieId|userId|rating|         adult|belongs_to_collection|    budget|              genres|            homepage|      imdb
+-------+------+------+--------------+--------------------+----------+--------------------+--------------------+----------
|      1|    15|   2.0| But when Ahmed|  the perfect scap...| is charged| the gravitas of ...| marking the star...|/hPObJrJhUw92DU5z
|      1|    89|   5.0| But when Ahmed|  the perfect scap...| is charged| the gravitas of ...| marking the star...|/hPObJrJhUw92DU5z
|      1|    90|   4.0| But when Ahmed|  the perfect scap...| is charged| the gravitas of ...| marking the star...|/hPObJrJhUw92DU5z
|      1|    91|   5.0| But when Ahmed|  the perfect scap...| is charged| the gravitas of ...| marking the star...|/hPObJrJhUw92DU5z
|      1|   112|   5.0| But when Ahmed|  the perfect scap...| is charged| the gravitas of ...| marking the star...|/hPObJrJhUw92DU5z
|      1|   128|   5.0| But when Ahmed|  the perfect scap...| is charged| the gravitas of ...| marking the star...|/hPObJrJhUw92DU5z
|      1|   130|   3.0| But when Ahmed|  the perfect scap...| is charged| the gravitas of ...| marking the star...|/hPObJrJhUw92DU5z
|      1|   138|   2.0| But when Ahmed|  the perfect scap...| is charged| the gravitas of ...| marking the star...|/hPObJrJhUw92DU5z
|      1|   168|   5.0| But when Ahmed|  the perfect scap...| is charged| the gravitas of ...| marking the star...|/hPObJrJhUw92DU5z
|      1|   175|   4.0| But when Ahmed|  the perfect scap...| is charged| the gravitas of ...| marking the star...|/hPObJrJhUw92DU5z
|      1|   176|   2.0| But when Ahmed|  the perfect scap...| is charged| the gravitas of ...| marking the star...|/hPObJrJhUw92DU5z
|      1|   179|   5.0| But when Ahmed|  the perfect scap...| is charged| the gravitas of ...| marking the star...|/hPObJrJhUw92DU5z
|      1|   185|   5.0| But when Ahmed|  the perfect scap...| is charged| the gravitas of ...| marking the star...|/hPObJrJhUw92DU5z
|      1|   205|   4.5| But when Ahmed|  the perfect scap...| is charged| the gravitas of ...| marking the star...|/hPObJrJhUw92DU5z
|      1|   212|   3.0| But when Ahmed|  the perfect scap...| is charged| the gravitas of ...| marking the star...|/hPObJrJhUw92DU5z
|      1|   213|   3.0| But when Ahmed|  the perfect scap...| is charged| the gravitas of ...| marking the star...|/hPObJrJhUw92DU5z
|      1|   219|   5.0| But when Ahmed|  the perfect scap...| is charged| the gravitas of ...| marking the star...|/hPObJrJhUw92DU5z
|      1|   242|   5.0| But when Ahmed|  the perfect scap...| is charged| the gravitas of ...| marking the star...|/hPObJrJhUw92DU5z
|      1|   261|   1.5| But when Ahmed|  the perfect scap...| is charged| the gravitas of ...| marking the star...|/hPObJrJhUw92DU5z
|      1|   283|   3.0| But when Ahmed|  the perfect scap...| is charged| the gravitas of ...| marking the star...|/hPObJrJhUw92DU5z
+-------+------+------+--------------+--------------------+----------+--------------------+--------------------+----------
only showing top 20 rows
```

```
single_user=test.filter(test['userId']==29).select(['movieId','userId','title','genres'])
# User had 10 ratings in the test data set
# Realistically this should be some sort of hold out set!
single_user.show(truncate = False)
```

```
+-------+------+-------------------------------------+-----------------------------+
|movieId|userId|title                                |genres                       |
+-------+------+-------------------------------------+-----------------------------+
|2459   |29    |An Elephant Can Be Extremely Deceptive|[{'id': 35, 'name': 'Comedy'}]|
```

```
|58559  |29    |Confession of a Child of the Century  |[{'id': 18, 'name': 'Drama'}] |
+-------+------+--------------------------------------+------------------------------+
```

```
#Now we will use model.transform() function in order to generate recommended movies along with their predicted features.

recomendations = model_vishnav.transform(single_user)
recomendations.orderBy('prediction',ascending=False).show(truncate = False )
```

```
+-------+------+--------------------------------------+------------------------------+----------+
|movieId|userId|title                                 |genres                        |prediction|
+-------+------+--------------------------------------+------------------------------+----------+
|2459   |29    |An Elephant Can Be Extremely Deceptive|[{'id': 35, 'name': 'Comedy'}]|3.742302  |
|58559  |29    |Confession of a Child of the Century  |[{'id': 18, 'name': 'Drama'}] |3.169405  |
+-------+------+--------------------------------------+------------------------------+----------+
```