In [1]:
```
import findspark
findspark.init()
```

In [2]:
```
import pyspark
from pyspark.sql import SparkSession
from pyspark.sql.types import *
from pyspark.sql.functions import *

# Configure spark session
spark = SparkSession\
    .builder\
    .master('local[2]')\
    .appName('AMAZON_BOOK')\
    .config('spark.jars.packages', 'org.mongodb.spark:mongo-spark-connector_2.12:2.4.1
    .config("spark.driver.memory", "5g")\
    .getOrCreate()
```

In [3]:
```
# MongoDB connection URI
mongo_uri = "mongodb://localhost:27017/AMAZON_BOOK.RATE"
# Read data from MongoDB collection into a DataFrame and limit to 1,000,000 rows
df_rate = spark.read.format("mongo").option("uri", mongo_uri).load().limit(1000000)
# Show the DataFrame
df_rate.show()
```

```
+----------+-----+--------------------+-------------+--------------------+-----+----
----------------+--------------------+
|        Id|Price|               Title|      User_id|                 _id|score|
summary|                text|
+----------+-----+--------------------+-------------+--------------------+-----+----
----------------+--------------------+
|1558746153| NULL|Chicken Soup for ...| AEKP4FJRWRGZT|{6570367e824b9730...|  5.0|
Helpful|Shows you what ot...|
|1882931173| NULL|Its Only Art If I...| AVCGYZL8FQQTD|{6570367e824b9730...|  4.0|Nice
collection o...|This is only for ...|
|1558746153| NULL|Chicken Soup for ...|         NULL|{6570367e824b9730...|  5.0|"Thi
s book hit th...|This book was ver...|
|0826414346| NULL|Dr. Seuss: Americ...|A30TK6U7DNS82R|{6570367e824b9730...|  5.0|   R
eally Enjoyed It|I don't care much...|
|0826414346| NULL|Dr. Seuss: Americ...|A3UH4UZ4RSVO82|{6570367e824b9730...|  5.0|Esse
ntial for eve...|"If people become...|
|1558746153| NULL|Chicken Soup for ...|         NULL|{6570367e824b9730...|  4.0| oh
this book was ok|well me and my fr...|
|1558746153| NULL|Chicken Soup for ...|         NULL|{6570367e824b9730...|  4.0| oh
this book was ok|well me and my fr...|
|0826414346| NULL|Dr. Seuss: Americ...|A2MVUWT453QH61|{6570367e824b9730...|  4.0|Phli
p Nel gives s...|Theodore Seuss Ge...|
|1558746153| NULL|Chicken Soup for ...|         NULL|{6570367e824b9730...|  5.0|Chic
ken Soup for ...|Chicken Soup for ...|
|0826414346| NULL|Dr. Seuss: Americ...|A22X4XUPKF66MR|{6570367e824b9730...|  4.0|Good
academic ove...|"Philip Nel - Dr....|
|1558746153| NULL|Chicken Soup for ...|A1M7N5V6W0Z0H7|{6570367e824b9730...|  5.0|More
stories abou...|Another great boo...|
|0826414346| NULL|Dr. Seuss: Americ...|A2F6NONFUDB6UK|{6570367e824b9730...|  4.0|One
of America's ...|"""Dr. Seuss: Ame...|
|1558746153| NULL|Chicken Soup for ...|         NULL|{6570367e824b9730...|  5.0|
Heart-Warmer|Chicken Soup is a...|
|0826414346| NULL|Dr. Seuss: Americ...|A14OJS0VWMOSWO|{6570367e824b9730...|  5.0|A me
morably excel...|Theodor Seuss Gie...|
|1558746153| NULL|Chicken Soup for ...|         NULL|{6570367e824b9730...|  5.0|
RACHEL'S REVIEW|One day my friend...|
|0826414346| NULL|Dr. Seuss: Americ...|A2RSSXTDZDUSH4|{6570367e824b9730...|  5.0|Acad
emia At It's ...|"When I recieved ...|
|1558746153| NULL|Chicken Soup for ...|         NULL|{6570367e824b9730...|  5.0|This
is one of th...|Chicken Soup for ...|
|0826414346| NULL|Dr. Seuss: Americ...|A25MD5I2GUIW6W|{6570367e824b9730...|  5.0|And
to think that...|"Trams (or any pu...|
|0826414346| NULL|Dr. Seuss: Americ...|A3VA4XFS5WNJO3|{6570367e824b9730...|  4.0|Fasc
inating accou...|As far as I am aw...|
|1558746153| NULL|Chicken Soup for ...|A1UMNA2JK9NELD|{6570367e824b9730...|  5.0|Klee
nex Needed to...|Warning: Do not r...|
+----------+-----+--------------------+-------------+--------------------+-----+----
----------------+--------------------+
only showing top 20 rows
```

In [4]:
```
df_rate = df_rate.drop('summary')
df_rate
```

Out[4]:
```
DataFrame[Id: string, Price: string, Title: string, User_id: string, _id: struct<oid:
string>, score: string, text: string]
```

In [5]:
```
df_rate.show()
```

```
+----------+-----+--------------------+-------------+--------------------+-----+----
----------------+
|        Id|Price|               Title|      User_id|                 _id|score|
text|
+----------+-----+--------------------+-------------+--------------------+-----+----
----------------+
|1558746153| NULL|Chicken Soup for ...| AEKP4FJRWRGZT|{6570367e824b9730...|  5.0|Show
s you what ot...|
|1882931173| NULL|Its Only Art If I...| AVCGYZL8FQQTD|{6570367e824b9730...|  4.0|This
is only for ...|
|1558746153| NULL|Chicken Soup for ...|         NULL|{6570367e824b9730...|  5.0|This
book was ver...|
|0826414346| NULL|Dr. Seuss: Americ...|A30TK6U7DNS82R|{6570367e824b9730...|  5.0|I do
n't care much...|
|0826414346| NULL|Dr. Seuss: Americ...|A3UH4UZ4RSVO82|{6570367e824b9730...|  5.0|"If
people become...|
|1558746153| NULL|Chicken Soup for ...|         NULL|{6570367e824b9730...|  4.0|well
me and my fr...|
|1558746153| NULL|Chicken Soup for ...|         NULL|{6570367e824b9730...|  4.0|well
me and my fr...|
|0826414346| NULL|Dr. Seuss: Americ...|A2MVUWT453QH61|{6570367e824b9730...|  4.0|Theo
dore Seuss Ge...|
|1558746153| NULL|Chicken Soup for ...|         NULL|{6570367e824b9730...|  5.0|Chic
ken Soup for ...|
|0826414346| NULL|Dr. Seuss: Americ...|A22X4XUPKF66MR|{6570367e824b9730...|  4.0|"Phi
lip Nel - Dr....|
|1558746153| NULL|Chicken Soup for ...|A1M7N5V6W0Z0H7|{6570367e824b9730...|  5.0|Anot
her great boo...|
|0826414346| NULL|Dr. Seuss: Americ...|A2F6NONFUDB6UK|{6570367e824b9730...|  4.0|"""D
r. Seuss: Ame...|
|1558746153| NULL|Chicken Soup for ...|         NULL|{6570367e824b9730...|  5.0|Chic
ken Soup is a...|
|0826414346| NULL|Dr. Seuss: Americ...|A14OJS0VWMOSWO|{6570367e824b9730...|  5.0|Theo
dor Seuss Gie...|
|1558746153| NULL|Chicken Soup for ...|         NULL|{6570367e824b9730...|  5.0|One
day my friend...|
|0826414346| NULL|Dr. Seuss: Americ...|A2RSSXTDZDUSH4|{6570367e824b9730...|  5.0|"Whe
n I recieved ...|
|1558746153| NULL|Chicken Soup for ...|         NULL|{6570367e824b9730...|  5.0|Chic
ken Soup for ...|
|0826414346| NULL|Dr. Seuss: Americ...|A25MD5I2GUIW6W|{6570367e824b9730...|  5.0|"Tra
ms (or any pu...|
|0826414346| NULL|Dr. Seuss: Americ...|A3VA4XFS5WNJO3|{6570367e824b9730...|  4.0|As f
ar as I am aw...|
|1558746153| NULL|Chicken Soup for ...|A1UMNA2JK9NELD|{6570367e824b9730...|  5.0|Warn
ing: Do not r...|
+----------+-----+--------------------+-------------+--------------------+-----+----
----------------+
only showing top 20 rows
```

In [6]:
```
df_rate = df_rate.drop('_id')
df_rate = df_rate.drop('Price')
df_rate
```

Out[6]:
```
DataFrame[Id: string, Title: string, User_id: string, score: string, text: string]
```

In [7]:
```
# Convert the "score" column to float
df_rate = df_rate.withColumn("score", col("score").cast("float"))
```

```
df_rate
```

Out[7]:
```
DataFrame[Id: string, Title: string, User_id: string, score: float, text: string]
```

In [8]:
```
df_rate = df_rate.drop('Text')
df_rate
```

Out[8]:
```
DataFrame[Id: string, Title: string, User_id: string, score: float]
```

In [9]:
```
# Check for null or NaN values in the "score" column
df_rate.select([count(when(isnan('score') | col('score').isNull(), 'score'))]).show()
```

```
+-------------------------------------------------------------------+
|count(CASE WHEN (isnan(score) OR (score IS NULL)) THEN score END)|
+-------------------------------------------------------------------+
|                                                               5729|
+-------------------------------------------------------------------+
```

In [10]:
```
df_rate = df_rate.na.drop(subset=["score"])
```

In [11]:
```
# Check for null or NaN values in the "score" column
df_rate.select([count(when(isnan('score') | col('score').isNull(), 'score'))]).show()
```

```
+-------------------------------------------------------------------+
|count(CASE WHEN (isnan(score) OR (score IS NULL)) THEN score END)|
+-------------------------------------------------------------------+
|                                                                  0|
+-------------------------------------------------------------------+
```

In [12]:
```
df_rate.select('score').show()
```

```
+-----+
|score|
+-----+
|  5.0|
|  4.0|
|  5.0|
|  5.0|
|  5.0|
|  4.0|
|  4.0|
|  4.0|
|  5.0|
|  4.0|
|  5.0|
|  4.0|
|  5.0|
|  5.0|
|  5.0|
|  5.0|
|  5.0|
|  5.0|
|  4.0|
|  5.0|
+-----+
only showing top 20 rows
```

In [13]:
```
from pyspark.sql.functions import col
from pyspark.ml.feature import StringIndexer
from pyspark.ml import Pipeline
from pyspark.ml.recommendation import ALS
from pyspark.ml.evaluation import RegressionEvaluator


# Convert string columns to numerical indices
indexers = [StringIndexer(inputCol=column, outputCol=column+"_index", handleInvalid="s
pipeline = Pipeline(stages=indexers)
df_rate_indexed = pipeline.fit(df_rate).transform(df_rate)
```

In [14]:
```
df_rate_indexed.printSchema()
```

```
root
 |-- Id: string (nullable = true)
 |-- Title: string (nullable = true)
 |-- User_id: string (nullable = true)
 |-- score: float (nullable = true)
 |-- Id_index: double (nullable = false)
 |-- Title_index: double (nullable = false)
 |-- User_id_index: double (nullable = false)
```

In [15]:
```
# Create ALS model
als = ALS(
    userCol="User_id_index",
    itemCol="Title_index",
    ratingCol="score",
    rank=15,      # Adjust this parameter
    maxIter=15,
    regParam=0.1, # Adjust this parameter
    coldStartStrategy="drop",
)


# Split the data into training and test sets
(training, test) = df_rate_indexed.randomSplit([0.8, 0.2], seed=1234)

# Fit the ALS model to the training data
model = als.fit(training)

# Make predictions on the test data
predictions = model.transform(test)

# Evaluate the model
evaluator = RegressionEvaluator(metricName="rmse", labelCol="score", predictionCol="pr
rmse = evaluator.evaluate(predictions)
print(f"Root Mean Squared Error (RMSE) = {rmse}")
```

```
Root Mean Squared Error (RMSE) = 1.6910231008300247
```

In [18]:
```
# Create ALS model
als = ALS(
    userCol="User_id_index",
    itemCol="Title_index",
    ratingCol="score",
    rank=50,      # Adjust this parameter
    maxIter=20,
```

```
    regParam=0.1, # Adjust this parameter
    coldStartStrategy="drop",
)


# Split the data into training and test sets
(training, test) = df_rate_indexed.randomSplit([0.8, 0.2], seed=1234)

# Fit the ALS model to the training data
model = als.fit(training)

# Make predictions on the test data
predictions = model.transform(test)

# Evaluate the model
evaluator = RegressionEvaluator(metricName="rmse", labelCol="score", predictionCol="pr
rmse = evaluator.evaluate(predictions)
print(f"Root Mean Squared Error (RMSE) = {rmse}")
```

```
Root Mean Squared Error (RMSE) = 1.5040903560449084
```

In [17]:

```
# Generate top N book recommendations for all users
userRecs = model.recommendForAllUsers(5)  # You can change 5 to the desired number of

# Show the recommendations
userRecs.show(truncate=False)
```

```
+-------------+--------------------------------------------------------------------------------
------------------------------+
|User_id_index|recommendations
|
+-------------+--------------------------------------------------------------------------------
------------------------------+
|1            |[{9994, 5.9698405}, {10882, 5.7770405}, {11277, 5.7585497}, {11979, 5.
7438855}, {3456, 5.743495}]   |
|3            |[{56495, 6.4719734}, {66459, 6.072416}, {33491, 5.9220114}, {11175, 5.
875997}, {6849, 5.8391876}]   |
|6            |[{15651, 6.038165}, {22066, 5.840315}, {7832, 5.798223}, {22632, 5.783
9284}, {60924, 5.6988163}]    |
|12           |[{56495, 5.767653}, {14754, 5.479958}, {9087, 5.339184}, {7412, 5.2468
88}, {6184, 5.242036}]        |
|13           |[{52411, 6.301175}, {11175, 6.296687}, {64065, 6.206318}, {70918, 6.19
13047}, {56495, 6.1319957}]   |
|16           |[{52411, 6.165396}, {13267, 5.9807725}, {49132, 5.9774656}, {23930, 5.
743216}, {2966, 5.7290916}]   |
|22           |[{20689, 6.1716948}, {26287, 6.116484}, {11175, 6.083063}, {24517, 5.9
10527}, {56495, 5.8912096}]   |
|26           |[{33419, 5.6094565}, {26287, 5.5716634}, {18314, 5.5291524}, {25773,
5.517114}, {57577, 5.344919}]  |
|27           |[{56495, 6.4586134}, {26287, 5.947452}, {13577, 5.9130263}, {11175, 5.
8273687}, {35978, 5.7331333}] |
|28           |[{9318, 5.158664}, {19648, 4.8949895}, {49249, 4.881256}, {7379, 4.863
6456}, {5978, 4.8336973}]     |
|31           |[{52411, 6.6452184}, {64065, 5.872023}, {20222, 5.820219}, {11175, 5.7
95535}, {30857, 5.788367}]    |
|34           |[{56495, 5.3766413}, {26287, 5.234441}, {62038, 5.222541}, {29185, 5.1
64635}, {70918, 5.156376}]    |
|44           |[{14761, 5.789416}, {56495, 5.776514}, {15651, 5.702624}, {11175, 5.67
4769}, {26287, 5.6715593}]    |
|47           |[{20379, 5.81998}, {14267, 5.795021}, {16799, 5.583908}, {11175, 5.539
77}, {22632, 5.5367618}]      |
|52           |[{52411, 6.490432}, {11175, 6.251845}, {64065, 6.2091403}, {11193, 6.1
655293}, {17810, 6.114803}]   |
|53           |[{52411, 6.487915}, {11175, 6.0680885}, {17810, 6.0606337}, {13134, 6.
0310097}, {60924, 6.0236216}] |
|65           |[{52411, 6.3525763}, {27777, 6.08239}, {56495, 5.8802085}, {20222, 5.7
59116}, {50755, 5.738173}]    |
|76           |[{56495, 5.9441895}, {62038, 5.827719}, {64065, 5.7497287}, {60924, 5.
6861286}, {14267, 5.656944}]  |
|78           |[{11175, 6.3881626}, {52411, 6.2077203}, {56495, 6.0605154}, {64065,
6.0191984}, {66459, 6.0055165}]|
|81           |[{56495, 5.9728065}, {15762, 5.5394187}, {6232, 5.294282}, {3981, 5.28
06115}, {13257, 5.2355795}]   |
+-------------+--------------------------------------------------------------------------------
------------------------------+
only showing top 20 rows
```

In [ ]: