

▼ Big Data Analysis Lab-04

Name: Gurvinder Kaur Matharu

PRN: 20190802077

```
!pip install pyspark
```

```
Requirement already satisfied: pyspark in /usr/local/lib/python3.7/dist-packages (3.2.0)
Requirement already satisfied: py4j==0.10.9.3 in /usr/local/lib/python3.7/dist-packages (0.10.9.3)
```

```
from pyspark import SparkContext, SparkConf
sc = SparkContext.getOrCreate()
```

```
data = sc.textFile('books.csv')
```

```
type(data)
```

```
pyspark.rdd.RDD
```

```
data.top(2)
```

```
['id,book_id,best_book_id,work_id,books_count,isbn,isbn13,authors,original_publication_year,
'9999,8565083,8565083,13433613,7,61711527,9.78006171153e+12,Peggy Orenstein,2011.0,(
```

```
data.collect()
```

```
'292,140225,140225,3349054,299,006112527X,9.78006112527e+12,"C.S. Lewis, Pauline
'293,295,295,3077988,2573,753453800,9.7807534538e+12,Robert Louis Stevenson,1882.
'294,9520360,9520360,14406312,69,1423140591,9.7814231406e+12,Rick Riordan,2011.0,
'295,10644930,10644930,15553789,145,1451627289,9.78145162728e+12,Stephen King,201
'296,4948,4948,3144982,162,241003008,9.78024100301e+12,Eric Carle,1969.0,The Very
"297,135479,135479,1621115,155,140285601,9.7801402856e+12,Kurt Vonnegut Jr.,1963.
'298,2493,2493,3234863,1248,451528557,9.78045152855e+12,"H.G. Wells, Greg Bear, C
'299,7933292,7933292,11283577,69,849946158,9.78084994616e+12,"Todd Burpo, Lynn Vi
'300,39999,39999,1148702,178,385751060,9.78038575106e+12,John Boyne,2006.0,The Bc
'301,4900,4900,2877220,1369,1892295490,9.78189229549e+12,Joseph Conrad,1899.0,Hea
'302,7812659,7812659,10829530,117,044654759X,9.7804465476e+12,Nicholas Sparks,201
'303,52529,52529,2001660,113,1582701709,9.78158270171e+12,Rhonda Byrne,2006.0,The
'304,227711,227711,1191096,52,60987561,9.78006098756e+12,Wally Lamb,1998.0,I Know
'305,10583,33124137,150017,198,1416524347,9.78141652434e+12,Stephen King,1983.0,P
"306,5203,5203,1003370,61,671021001,9.78067102101e+12,Wally Lamb,1992.0,She's Cor
'307,1215032,1215032,2502882,102,756404738,9.78075640473e+12,Patrick Rothfuss,201
'308,33724,33724,2888997,94,440241901,9.7804402419e+12,Sophie Kinsella,2003.0,Can
'309,6752378,6752378,6948844,96,1442403543,9.78144240354e+12,Cassandra Clare,2011
'310,14748,14748,2796838,71,743418174,9.78074341817e+12,Jennifer Weiner,2001.0,Gc
'311,3412,3412,816449,201,380018179,9.78038001818e+12,Colleen McCullough,1977.0,T
'312,2213661,2213661,2219449,129,60530928,9.78006053092e+12,"Neil Gaiman, Dave Mc
'313,13158800,13158800,18337340,113,1451681739,9.78145168173e+12,M.L. Stedman,201
```

```
'314,28194,28194,2628323,134,439709105,9.7804397091e+12,"Cornelia Funke, Anthea B
'315,4894,4894,3332594,128,91883768,9.78009188377e+12,"Spencer Johnson, Kenneth H
'316,9791,9791,613469,95,307279464,9.78030727946e+12,Bill Bryson,1997.0,A Walk in
'317,13214,13214,1413589,107,553279378,9.78055327938e+12,Maya Angelou,1969.0,I Kn
'318,15818107,15818107,21545713,65,61950726,9.78006195073e+12,Christina Baker Kli
'319,8755776,8755776,13629058,101,1442416866,9.78144241686e+12,Cassandra Clare,20
'320,13526165,13526165,17626728,78,316204277,9.78031620428e+12,Maria Semple,2012.
'321,10365,10365,115,69,375806814,9.78037580681e+12,Wilson Rawls,1961.0,Where the
'322,14497,14497,16534,152,60557818,9.78006055781e+12,Neil Gaiman,1996.0,Neverwhe
'323,9717,9717,4489585,274,571224385,9.78057122439e+12,"Milan Kundera, Michael He
'324,16068905,16068905,21861351,80,,,"Rainbow Rowell,2013.0,,Fangirl,eng,4.12,3403
'325,4473,4473,1734019,138,552135399,9.7805521354e+12,John Irving,1989.0,A Prayer
'326,32234,32234,1223333,96,316182540,9.78031618255e+12,Janet Fitch,1999.0,White
'327,9275658,9275658,14157512,91,039925675X,9.78039925675e+12,Marie Lu,2011.0,Leg
'328,14866,14866,3375915,84,743496728,9.78074349673e+12,Jodi Picoult,2007.0,Ninet
'329,2318271,2318271,3364076,111,1401323251,9.78140132326e+12,"Randy Pausch, Jeff
'330,228665,228665,2008238,94,812511816,9.78081251182e+12,Robert Jordan,1990.0,Th
'331,10916,10916,3349846,76,61150142,9.78006115014e+12,Jodi Picoult,1998.0,The Pa
'332,65605,65605,1031537,312,60764902,9.78006076491e+12,C.S. Lewis,1953.0,The mag
'333,2153405,2153405,2158906,107,595440096,9.78059544009e+12,Lisa Genova,2007.0,S
'334,13596809,13596809,19186128,81,425263916,9.78042526391e+12,Sylvia Day,2012.0,
'335,6689,6689,2379261,182,375814248,9.78037581424e+12,"Roald Dahl, Quentin Blake
'336,13137,13137,1711194,107,446696617,9.78044669662e+12,James Patterson,2001.0,1
"337,13,13,135328,32,345453743,9.78034545375e+12,Douglas Adams,1996.0,The Ultimat
'338,6280118,6280118,6463667,129,340896965,9.78034089697e+12,David Nicholls,2009.
'339,32929,32929,1086867,98,60775858,9.78006077586e+12,"Margaret Wise Brown, Clem
'340,7747374,7747374,10576999,38,61969559,9.78006196955e+12,Pittacus Lore,2010.0,
'341,1371,1371,3293141,1726,140275363,9.78014027536e+12,"Homer, Robert Fagles, Fr
'342,13497818,13497818,19926990,139,316228532,9.78031622853e+12,J.K. Rowling,2012
'343,24192,24192,1022176,120,385339690,9.7803853397e+12,John Grisham,1996.0,The R
'344,4138,4138,2086690,54,316777730,9.78031677774e+12,David Sedaris,1997.0,Naked,
'345,45978,45978,2035753,163,375840400,9.7803758404e+12,Christopher Paolini,2005.
'346,236093,236093,1993810,1474,140621679,9.78014062168e+12,"L. Frank Baum, W.W.
'347,121749,121749,3348636,348,000720230X,9.7800072023e+12,C.S. Lewis,1951.0,Prin
'348,16143347,16143347,21975829,77,,,"E. Lockhart,2014.0,We Were Liars,We Were Lia
"349,11500,11500,2018007,211,150001000,9.78015000100e+12,"Shirley Jackson,1955.0,C
"350,11500,11500,2018007,211,150001000,9.78015000100e+12,"Shirley Jackson,1955.0,C
```

```
data.take(2)
```

```
['id,book_id,best_book_id,work_id,books_count,isbn,isbn13,authors,original_publication_
1,2767052,2767052,2792775,272,439023483,9.78043902348e+12,Suzanne Collins,2008.0,The
```

```
for line in data.take(5):
    print(line)
```

```
id,book_id,best_book_id,work_id,books_count,isbn,isbn13,authors,original_publication_
1,2767052,2767052,2792775,272,439023483,9.78043902348e+12,Suzanne Collins,2008.0,The
2,3,3,4640799,491,439554934,9.78043955493e+12,"J.K. Rowling, Mary GrandPré",1997.0,Haz
3,41865,41865,3212258,226,316015849,9.78031601584e+12,Stephenie Meyer,2005.0,Twilight
4,2657,2657,3275794,487,61120081,9.78006112008e+12,Harper Lee,1960.0,To Kill a Mockir
```

```
data.first()
```

```
oneRecord = data.first()
columns = oneRecord.split(',')

```

columns

```
['id',
 'book_id',
 'best_book_id',
 'work_id',
 'books_count',
 'isbn',
 'isbn13',
 'authors',
 'original_publication_year',
 'original_title',
 'title',
 'language_code',
 'average_rating',
 'ratings_count',
 'work_ratings_count',
 'work_text_reviews_count',
 'ratings_1',
 'ratings_2',
 'ratings_3',
 'ratings_4',
 'ratings_5',
 'image_url',
 'small_image_url']

```

```
import pyspark
spark = pyspark.sql.Session.builder.getOrCreate()
type(spark)

```

pyspark.sql.session.Session

```
books_df = spark.read.csv('books.csv', header=True, inferSchema=True)

```

```
books_df.printSchema()

```

```
root
|-- id: integer (nullable = true)
|-- book_id: integer (nullable = true)
|-- best_book_id: integer (nullable = true)
|-- work_id: integer (nullable = true)
|-- books_count: integer (nullable = true)
|-- isbn: string (nullable = true)
|-- isbn13: double (nullable = true)
|-- authors: string (nullable = true)
|-- original_publication_year: double (nullable = true)
|-- original_title: string (nullable = true)
|-- title: string (nullable = true)
|-- language_code: string (nullable = true)
|-- average_rating: string (nullable = true)
|-- ratings_count: string (nullable = true)

```

```
|-- work_ratings_count: string (nullable = true)
|-- work_text_reviews_count: string (nullable = true)
|-- ratings_1: double (nullable = true)
|-- ratings_2: integer (nullable = true)
|-- ratings_3: integer (nullable = true)
|-- ratings_4: integer (nullable = true)
|-- ratings_5: integer (nullable = true)
|-- image_url: string (nullable = true)
|-- small_image_url: string (nullable = true)
```

```
type(books_df)
```

```
pyspark.sql.dataframe.DataFrame
```

```
len(books_df.columns)
```

```
23
```

```
ratings_df = spark.read.csv('ratings.csv', header=True, inferSchema=True)
```

```
type(ratings_df)
```

```
pyspark.sql.dataframe.DataFrame
```

```
ratings_df.printSchema()
```

```
root
 |-- book_id: integer (nullable = true)
 |-- user_id: integer (nullable = true)
 |-- rating: integer (nullable = true)
```

```
ratings_df.first()
```

```
Row(book_id=1, user_id=314, rating=5)
```

```
ratings_df.show(5)
```

```
+-----+-----+-----+
|book_id|user_id|rating|
+-----+-----+-----+
|      1|    314|      5|
|      1|    439|      3|
|      1|    588|      5|
|      1|   1169|      4|
|      1|   1185|      4|
+-----+-----+-----+
only showing top 5 rows
```

```
ratings_df.head(5)
```

```
[Row(book_id=1, user_id=314, rating=5),
 Row(book_id=1, user_id=439, rating=3),
 Row(book_id=1, user_id=588, rating=5),
 Row(book_id=1, user_id=1169, rating=4),
 Row(book_id=1, user_id=1185, rating=4)]
```

```
ratings_df.select('book_id', 'rating').show(5)
```

```
+-----+-----+
|book_id|rating|
+-----+-----+
|      1|      5|
|      1|      3|
|      1|      5|
|      1|      4|
|      1|      4|
+-----+-----+
only showing top 5 rows
```

```
ratings_df.filter('rating <= 3').show(5)
```

```
+-----+-----+-----+
|book_id|user_id|rating|
+-----+-----+-----+
|      1|    439|      3|
|      1|   5461|      3|
|      1|   7563|      3|
|      1|   9246|      1|
|      1|  20076|      3|
+-----+-----+-----+
only showing top 5 rows
```

```
ratings_df.select('book_id', 'rating').filter('rating <= 3').show(5)
```

```
+-----+-----+
|book_id|rating|
+-----+-----+
|      1|      3|
|      1|      3|
|      1|      3|
|      1|      1|
|      1|      3|
+-----+-----+
only showing top 5 rows
```

```
ratings_df.count()
```

```
981756
```

```
print(f'Total number of Ratings Records : {ratings_df.count()}')
```

Total number of Ratings Records : 981756

```
unique_user_count = ratings_df.select('user_id').distinct().count()  
unique_user_count
```

53424

```
book_rating_less_or_three_count = ratings_df.filter('rating <= 3').count()  
book_rating_less_or_three_count
```

331429

```
ratings_df.describe('book_id').show()
```

	summary	book_id
count	981756	
mean	4943.275635697668	
stddev	2873.207414896114	
min	1	
max	10000	

```
ratings_df.count()
```

981756

```
aa = ratings_df.dropDuplicates()
```

```
aa.count()
```

980112

```
rating_without_null = ratings_df.dropna().count()  
rating_without_null
```

981756

```
ratings_df.dropna('any').count() # drop a row if it contains any nulls
```

981756


```
ratings_df.dropna('all').count() # drop a row if it contains any nulls
```

981756

```
# Maximum value of any column  
ratings_df.agg({'rating': 'max'}).show()
```

```
+-----+
|max(rating)|
+-----+
|          5|
+-----+
```

```
ratings_df.groupby('rating').count().toPandas()
```

	rating	count	
0	1	19575	
1	3	248623	
2	5	292961	
3	4	357366	
4	2	63231	

```
ratings_df.groupby('rating').count().show()
```

```
+-----+-----+
|rating| count|
+-----+-----+
|      1| 19575|
|      3|248623|
|      5|292961|
|      4|357366|
|      2| 63231|
+-----+-----+
```

```
# join 2 csv datasets
ratings_df.join(books_df, books_df.book_id == ratings_df.book_id).select(
  'user_id', 'title').show(5)
```

```
↳ +-----+-----+
|user_id|          title|
+-----+-----+
|    314|Harry Potter and ...|
|    439|Harry Potter and ...|
|    588|Harry Potter and ...|
|   1169|Harry Potter and ...|
|   1185|Harry Potter and ...|
+-----+-----+
only showing top 5 rows
```

```
ratings_df.orderBy('rating').show(5)
```

```
+-----+-----+-----+
|book_id|user_id|rating|
+-----+-----+-----+
```

6628	39907	1
6631	24498	1
6629	47480	1
6627	52717	1
6630	27769	1

+-----+-----+-----+

only showing top 5 rows

```
ratings_df.orderBy('rating', 'book_id').show()
```

book_id	user_id	rating
1	9246	1
1	51480	1
2	6063	1
2	48687	1
2	17643	1
2	13794	1
3	52036	1
3	49298	1
3	48687	1
3	15604	1
3	33065	1
3	11854	1
3	10751	1
3	21733	1
3	588	1
3	9246	1
3	32305	1
3	37284	1
3	10509	1
3	29703	1

+-----+-----+-----+

only showing top 20 rows

```
# Change the value of an existing columns
ratings_df.withColumn('rating', ratings_df.rating*10).show(5)
```

book_id	user_id	rating
1	314	50
1	439	30
1	588	50
1	1169	40
1	1185	40

+-----+-----+-----+

only showing top 5 rows

```
# Add new columns
new_dataset = ratings_df.withColumn('rating_ten', ratings_df.rating*10)
new_dataset.show(5)
```



```

+-----+-----+-----+-----+
|book_id|user_id|rating|rating_ten|
+-----+-----+-----+-----+
|      1|    314|      5|        50|
|      1|    439|      3|        30|
|      1|    588|      5|        50|
|      1|   1169|      4|        40|
|      1|   1185|      4|        40|
+-----+-----+-----+-----+
only showing top 5 rows

```

```
ratings_df.show(5)
```

```

+-----+-----+-----+
|book_id|user_id|rating|
+-----+-----+-----+
|      1|    314|      5|
|      1|    439|      3|
|      1|    588|      5|
|      1|   1169|      4|
|      1|   1185|      4|
+-----+-----+-----+
only showing top 5 rows

```

```

# Drop a column
ratings_df.drop('rating').show(5)

```

```

+-----+-----+
|book_id|user_id|
+-----+-----+
|      1|    314|
|      1|    439|
|      1|    588|
|      1|   1169|
|      1|   1185|
+-----+-----+
only showing top 5 rows

```

```
ratings_df.show(5)
```

```

+-----+-----+-----+
|book_id|user_id|rating|
+-----+-----+-----+
|      1|    314|      5|
|      1|    439|      3|
|      1|    588|      5|
|      1|   1169|      4|
|      1|   1185|      4|
+-----+-----+-----+
only showing top 5 rows

```

✓ 0s completed at 10:38 AM

