MidTerm Review

Anurag Nagar

Big Data Class

Outline

1. Topics Covered
2. Introduction to Big Data
3. Hadoop Distributed File System
   - HDFS Storage
   - HDFS Architecture
4. MapReduce
   - Basics
   - Scala Questions
   - Apache Spark
   - DataFrame Questions
5. Machine Learning

Topics Covered

List of topics covered so far:

- Introduction to Big Data
- Hadoop Distributed File System (HDFS)
- MapReduce Programming Concepts
- Scala Programming
- Apache Spark and RDD
- Spark DataFrames
- Machine Learning using Spark

Outline

1 Topics Covered

2 Introduction to Big Data

3 Hadoop Distributed File System
- HDFS Storage
- HDFS Architecture

4 MapReduce
- Basics
- Scala Questions
- Apache Spark
- DataFrame Questions

5 Machine Learning

Introduction to Big Data

What is Big Data?

- Remember the 3V definition
- Examples of Big Data
- Characteristics of Big Data e.g. raw data, log data, etc that needs to be processed to derive information
- Go through the slides and reading assignment

Outline

1 Topics Covered

2 Introduction to Big Data

3 Hadoop Distributed File System
- HDFS Storage
- HDFS Architecture

4 MapReduce
- Basics
- Scala Questions
- Apache Spark
- DataFrame Questions

5 Machine Learning

Hadoop Distributed File System

Properties of HDFS as a storage medium:

- Distributed
- Partitioned
- Fault-Tolerant by using replication
- Write-once, read-many
- Commodity Hardware
- File stored as blocks
- Designed for high latency, high throughput batch processing.

HDFS Architecture

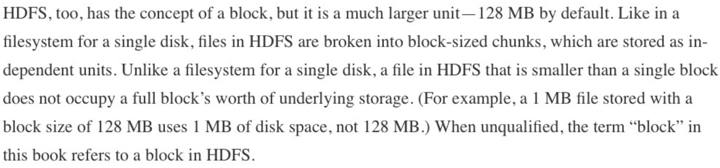
HDFS Architecture

- Master/Slave
- Master: NameNode
- Slaves: DataNodes
- NameNode takes care of metadata (not actual data) storage, and resource management
- DataNodes store actual data in units called blocks.
  In Hadoop 2, default block size = 128 MB
- Locality of computation - computation is scheduled where data is located, so there is less data movement.

See `https://hadoop.apache.org/docs/r2.6.0/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html#HDFS_Architecture` for details

Block Size

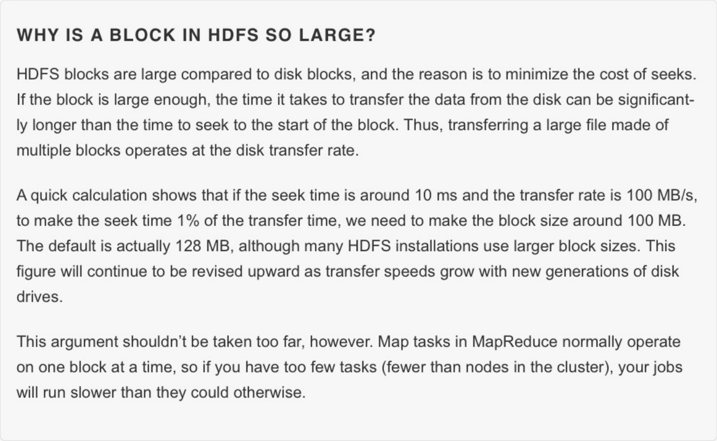
Read the details below about block size. Note that a file of size 129 MB would occupy two blocks - one of size 128 MB, and second one of size 1 MB and not 128 MB



HDFS, too, has the concept of a block, but it is a much larger unit—128 MB by default. Like in a filesystem for a single disk, files in HDFS are broken into block-sized chunks, which are stored as independent units. Unlike a filesystem for a single disk, a file in HDFS that is smaller than a single block does not occupy a full block's worth of underlying storage. (For example, a 1 MB file stored with a block size of 128 MB uses 1 MB of disk space, not 128 MB.) When unqualified, the term "block" in this book refers to a block in HDFS.

Block Size

Why are block sizes in HDFS so large? Read below



WHY IS A BLOCK IN HDFS SO LARGE?

HDFS blocks are large compared to disk blocks, and the reason is to minimize the cost of seeks. If the block is large enough, the time it takes to transfer the data from the disk can be significantly longer than the time to seek to the start of the block. Thus, transferring a large file made of multiple blocks operates at the disk transfer rate.

A quick calculation shows that if the seek time is around 10 ms and the transfer rate is 100 MB/s, to make the seek time 1% of the transfer time, we need to make the block size around 100 MB. The default is actually 128 MB, although many HDFS installations use larger block sizes. This figure will continue to be revised upward as transfer speeds grow with new generations of disk drives.

This argument shouldn't be taken too far, however. Map tasks in MapReduce normally operate on one block at a time, so if you have too few tasks (fewer than nodes in the cluster), your jobs will run slower than they could otherwise.

Questions

What is metadata in Hadoop?

1. Data in txt format
2. Information about data stored in Datanodes
3. User data
4. Copy of data stored in Datanodes

Questions

What is metadata in Hadoop?

1. Data in txt format
2. Information about data stored in Datanodes
3. User data
4. Copy of data stored in Datanodes

Questions

What is the major advantages of larger block sizes in HDFS?

1. It saves disk seek time i.e. time taken to locate the block on disk
2. It saves disk access time
3. It saves disk processing time
4. It saves disk latency time

Questions

What is the major advantages of larger block sizes in HDFS?

1. It saves disk seek time i.e. time taken to locate the block on disk
2. It saves disk access time
3. It saves disk processing time
4. It saves disk latency time

See `https://stackoverflow.com/questions/22353122/why-is-a-block-in-hdfs-so-large` for details

Questions

A file of size 1028 MB needs to be stored in HDFS having block size $= 128$ MB. Assuming replication factor $= 1$, how many blocks will be created and what will be their sizes.

Questions

A file of size 1028 MB needs to be stored in HDFS having block size = 128 MB. Assuming replication factor = 1, how many blocks will be created and what will be their sizes.

8 full blocks of size 128 MB, and the last block of size 4 MB.

Questions

A file of size 8 PB (petabytes) needs to be stored in HDFS. Assuming block size=128 MB and replication factor of 4, find the total number of blocks needed.

Questions

A file of size 8 PB (petabytes) needs to be stored in HDFS. Assuming block size=128 MB and replication factor of 4, find the total number of blocks needed.

8 PB $= 8 \times 2^{50}$ bytes $= 2^{53}$ bytes
128 MB $= 2^{7} \times 2^{20}$ bytes $= 2^{27}$ bytes

Number of blocks needed $= 2^{2} \times \frac{2^{53}}{2^{27}}$
$= 2^{28}$ blocks

Outline

1 Topics Covered

2 Introduction to Big Data

3 Hadoop Distributed File System
- HDFS Storage
- HDFS Architecture

4 MapReduce
- Basics
- Scala Questions
- Apache Spark
- DataFrame Questions

5 Machine Learning

MapReduce Phases

Two phases:

- **Map** - Transformation from one list to another
- **Reduce** - Aggregates data

Questions

What is the output of the following code in Scala?

```
var  result  = 1
val odds = List(3,  5,  7)
odds.map(x => result *= x)
print ( result )
```

Questions

What is the output of the following code in Scala?

```
var  result  = 1
val odds = List(3,  5,  7)
odds.map(x => result *= x)
 print ( result )
```

105

Questions

What is the output of the following code in Scala?

```
var  result  = 1
val odds = List(3,  5,  7)
odds.map(x => result *= x)
print(odds)
```

MidTerm Review

Anurag Nagar

Topics Covered

Introduction to Big Data

Hadoop Distributed File System

HDFS Storage

HDFS Architecture

MapReduce

Basics

Scala Questions

Apache Spark

DataFrame Questions

Machine Learning

Questions

What is the output of the following code in Scala?

```
var result = 1
val odds = List(3, 5, 7)
odds.map(x => result *= x)
print(odds)
```

List(3, 5, 7)

Questions

What is the output of the following code in Scala?

```
var result = 1
val odds = List(3, 5, 7)
odds.map(x => x*x)
print(odds)
```

Questions

What is the output of the following code in Scala?

```
var  result  = 1
val odds = List(3,  5,  7)
odds.map(x => x*x)
 print (odds)
```

List(3, 5, 7)

Questions

We would like to find the sum of elements of a list in Scala. Which of the following lines of code does this?
val list = List(2, 4, 8)

1. list.reduce((x, y) => x + y)
2. list.map ((x, y) => x + y)
3. list.reduceByKey ((x, y) => x + y)
4. list.groupByKey ((x, y) => x + y)

Questions

We would like to find the sum of elements of a list in Scala. Which of the following lines of code does this?
val list = List(2, 4, 8)

1. list.reduce((x, y) => x + y)
2. list.map ((x, y) => x + y)
3. list.reduceByKey ((x, y) => x + y)
4. list.groupByKey ((x, y) => x + y)

Questions

What will be the output of the following lines of code in Scala:

```
val nums = List(1, 2, 3)
nums = nums.map(x => x*x)
print(nums)
```

1. List(1, 2, 3)
2. List(1, 4, 9)
3. It will produce an error
4. List(0, 0, 0)

Questions

What will be the output of the following lines of code in Scala:

```
val nums = List(1, 2, 3)
nums = nums.map(x => x*x)
print (nums)
```

1. List(1, 2, 3)
2. List(1, 4, 9)
3. It will produce an error
4. List(0, 0, 0)

Questions

What will be the output of the following lines of code in Scala:

```
val languages = List("spanish", "french", " farsi ")
val languagesRdd = sc. parallelize (languages)
def myMap(s: String):(Char, Int) = {(s(0), 1)}
languagesRdd.map(myMap).reduceByKey((x, y) => x+y).collect()
```

1. Array((s,2), (f,1))
2. Array((s,1), (s,1), (f,1))
3. It will produce an error
4. Array((f,2), (s,1))

Questions

What will be the output of the following lines of code in Scala:

```
val languages = List("spanish", "french", " farsi ")
val languagesRdd = sc. parallelize (languages)
def myMap(s: String):(Char, Int) = {(s(0), 1)}
languagesRdd.map(myMap).reduceByKey((x, y) => x+y).collect()
```

1. Array((s,2), (f,1))
2. Array((s,1), (s,1), (f,1))
3. It will produce an error
4. Array((f,2), (s,1))

Questions

Consider the Spark code snippet below:

```
val storeAddress = sc.parallelize(List(
("Ritual", "1026 Valencia St"), ("Philz", "748 Van Ness Ave"),
("Philz", "3101 24th St"), ("Starbucks", "Seattle")))
```

Which of the following will return the count of each stores:

1. storeAddress.countByKey.toSeq
2. storeAddress.count()
3. storeAddress.keys.count()
4. storeAddress.keys.map(x => (x, 1)).
   reduceByKey((x,y) => x+y)

Questions

Consider the Spark code snippet below:

```
val storeAddress = sc. parallelize (List(
("Ritual", "1026 Valencia St"), ("Philz", "748 Van Ness Ave"),
("Philz", "3101 24th St"), ("Starbucks", "Seattle")))
```

Which of the following will return the count of each stores:

1. storeAddress.countByKey.toSeq
2. storeAddress.count()
3. storeAddress.keys.count()
4. storeAddress.keys.map(x => (x, 1)).
reduceByKey((x,y) => x+y)

Questions

Consider the Spark code snippet below.

```
val storeAddress = sc. parallelize (List(
("Ritual", "1026 Valencia St"), ("Philz", "748 Van Ness Ave"),
("Philz", "3101 24th St"), ("Starbucks", "Seattle")))

val storeRating = sc. parallelize (List(
("Ritual", 4.9), ("Philz", 4.8)))
```

How many elements will be there in the following:
storeAddress.join(storeRating)

1. 2
2. 3
3. 4
4. 0

Questions

Consider the Spark code snippet below.

```
val storeAddress = sc.parallelize(List(
("Ritual", "1026 Valencia St"), ("Philz", "748 Van Ness Ave"),
("Philz", "3101 24th St"), ("Starbucks", "Seattle")))

val storeRating = sc.parallelize(List(
("Ritual", 4.9), ("Philz", 4.8)))
```

How many elements will be there in the following:
storeAddress.join(storeRating)

1. 2
2. 3
3. 4
4. 0

Questions

Consider the Spark code snippet below.

```
val storeRating = sc.parallelize (List(
("Ritual", 4.9), ("Philz", 4.8), ("Philz", 4.0),
("Ritual", 2.5), ("Starbucks", 4.0)
))
```

You would like to find the **maximum** rating for all the stores. Which line accomplishes this?

1. storeRating.reduceByKey(max)
2. storeRating.max.reduceByKey()
3. storeRating.reduceByKey{case (x: Double, y:Double) => Math.max(x, y) }.collect()
4. storeRating.reduceByKey(x, y => Math.max(x, y)).collect()

Questions

Consider the Spark code snippet below.

```
val storeRating = sc. parallelize ( List (
("Ritual", 4.9), ("Philz", 4.8), ("Philz", 4.0),
("Ritual", 2.5), ("Starbucks", 4.0)
))
```

You would like to find the **maximum** rating for all the stores. Which line accomplishes this?

1. storeRating.reduceByKey(max)
2. storeRating.max.reduceByKey()
3. storeRating.reduceByKey{case (x: Double, y:Double) => Math.max(x, y) }.collect()
4. storeRating.reduceByKey(x, y => Math.max(x, y)).collect()

Apache Spark

Important features of Apache Spark project[1]:

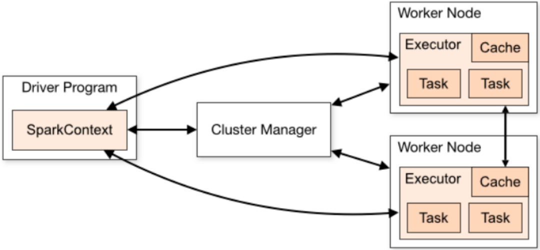
- Open-source cluster computing framework
- Developed to provide real-time, low latency queries on data that is stored in a cluster, such as Hadoop
- Uses partitioned, and distributed in-memory datasets, known as Resilient Distributed Datasets (RDD) to speed up computation.
- Disk I/O, which is the limiting factor in case of traditional MapReduce algorithms, is avoided by using RDDs
- Runs programs up to 100x faster than Hadoop MapReduce in memory, or 10x faster on disk.

[1] https://spark.apache.org/

Apache Spark

Important features of Apache Spark project[2]:

- Uses lazy evaluation for efficient processing
- RDDs are immutable i.e. they cannot be updated once created
- Spark core is the base engine for computation
- Spark workflow is shown below:



[2] https://spark.apache.org/

Questions

In Apache Spark, what is the use of the SparkContext (sc) object?

1. It represents a container for all the objects in memory
2. It represents all RDDs that are in your program
3. It represents an active connection to the Spark cluster and can be to request resources using the cluster manager
4. It represents the Hadoop file system

Questions

In Apache Spark, what is the use of the SparkContext (sc) object?

1. It represents a container for all the objects in memory
2. It represents all RDDs that are in your program
3. It represents an active connection to the Spark cluster and can be to request resources using the cluster manager
4. It represents the Hadoop file system

DataFrame Questions

Which of the following are true about DataFrames in Spark?[3]

1. They are part of the Spark SQL library
2. A DataFrame is a structured dataset organized into named columns
3. DataFrames can be constructued from a variety of sources, such as JSON files, CSV files, Hive tables or external databases
4. In Scala, a DataFrame is represented by a dataset of Rows

[3] See https://spark.apache.org/docs/latest/sql-programming-guide.html#datasets-and-dataframes for more details.

DataFrame Questions

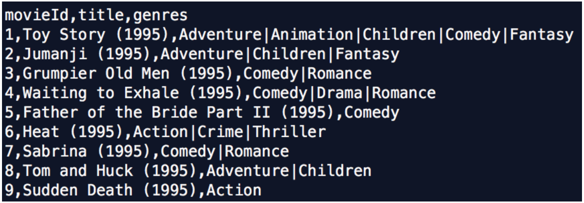
Which of the following are true about DataFrames in Spark?[3]

1. They are part of the Spark SQL library
2. A DataFrame is a structured dataset organized into named columns
3. DataFrames can be constructued from a variety of sources, such as JSON files, CSV files, Hive tables or external databases
4. In Scala, a DataFrame is represented by a dataset of Rows

[3] See `https://spark.apache.org/docs/latest/sql-programming-guide.html#datasets-and-dataframes` for more details.

DataFrame Questions

Suppose you have a file "movies.csv" :



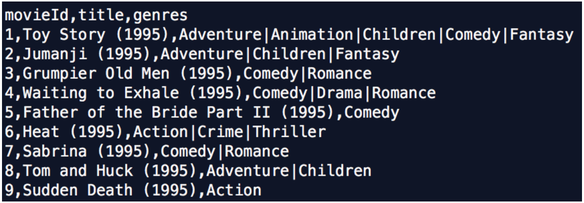
```
movieId,title,genres
1,Toy Story (1995),Adventure|Animation|Children|Comedy|Fantasy
2,Jumanji (1995),Adventure|Children|Fantasy
3,Grumpier Old Men (1995),Comedy|Romance
4,Waiting to Exhale (1995),Comedy|Drama|Romance
5,Father of the Bride Part II (1995),Comedy
6,Heat (1995),Action|Crime|Thriller
7,Sabrina (1995),Comedy|Romance
8,Tom and Huck (1995),Adventure|Children
9,Sudden Death (1995),Action
```

Which of the following is the correct way to load this file into a DataFrame?

1. val movies = spark.read.option("header","true").csv("movies.csv")
2. val movies = spark.read.option("header","false").csv("movies.csv")
3. val movies = spark.textFile.csv("movies.csv")
4. val movies = spark.csv("movies.csv")

DataFrame Questions

Suppose you have a file "movies.csv" :



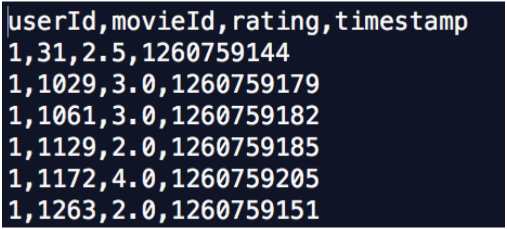
```
movieId,title,genres
1,Toy Story (1995),Adventure|Animation|Children|Comedy|Fantasy
2,Jumanji (1995),Adventure|Children|Fantasy
3,Grumpier Old Men (1995),Comedy|Romance
4,Waiting to Exhale (1995),Comedy|Drama|Romance
5,Father of the Bride Part II (1995),Comedy
6,Heat (1995),Action|Crime|Thriller
7,Sabrina (1995),Comedy|Romance
8,Tom and Huck (1995),Adventure|Children
9,Sudden Death (1995),Action
```

Which of the following is the correct way to load this file into a DataFrame?

1. val movies = spark.read.option("header","true").csv("movies.csv")
2. val movies = spark.read.option("header","false").csv("movies.csv")
3. val movies = spark.textFile.csv("movies.csv")
4. val movies = spark.csv("movies.csv")

DataFrame Questions

Suppose you have a file "ratings.csv", which you have loaded into a **Dataframe** called **ratings**



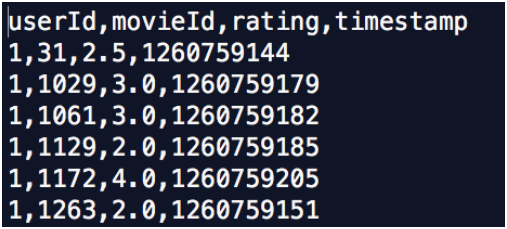
```
userId,movieId,rating,timestamp
1,31,2.5,1260759144
1,1029,3.0,1260759179
1,1061,3.0,1260759182
1,1129,2.0,1260759185
1,1172,4.0,1260759205
1,1263,2.0,1260759151
```

How can you find out the number of ratings for each movieId?

1. ratings.reduceByKey("movieId").count()
2. ratings.groupBy("movieId").count()
3. ratings.groupBy("movieId").keys
4. ratings.groupBy("movieId").keys.count()

DataFrame Questions

Suppose you have a file "ratings.csv", which you have loaded into a **Dataframe** called **ratings**



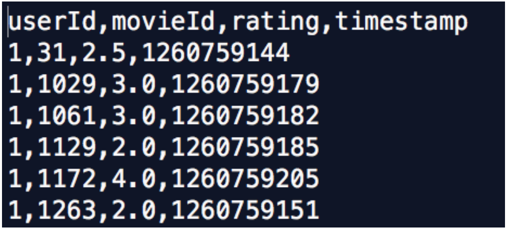
```
userId,movieId,rating,timestamp
1,31,2.5,1260759144
1,1029,3.0,1260759179
1,1061,3.0,1260759182
1,1129,2.0,1260759185
1,1172,4.0,1260759205
1,1263,2.0,1260759151
```

How can you find out the number of ratings for each movieId?

1. ratings.reduceByKey("movieId").count()
2. ratings.groupBy("movieId").count()
3. ratings.groupBy("movieId").keys
4. ratings.groupBy("movieId").keys.count()

DataFrame Questions

Suppose you have a file "ratings.csv", which you have loaded into a **Dataframe** called **ratings**



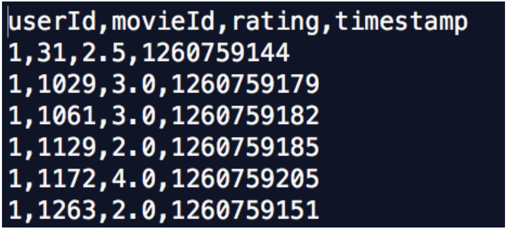
```
userId,movieId,rating,timestamp
1,31,2.5,1260759144
1,1029,3.0,1260759179
1,1061,3.0,1260759182
1,1129,2.0,1260759185
1,1172,4.0,1260759205
1,1263,2.0,1260759151
```

You would like to find the **count** of ratings for each movieId sorted by descending order of count,

1. ratings.groupBy("movieId").agg(desc("count"))
2. ratings.groupBy("movieId").desc("count").show()
3. ratings.groupBy("movieId").count().orderBy(desc("count"))
4. ratings.groupBy("movieId").orderBy(desc("count"))

DataFrame Questions

Suppose you have a file "ratings.csv", which you have loaded into a **Dataframe** called **ratings**



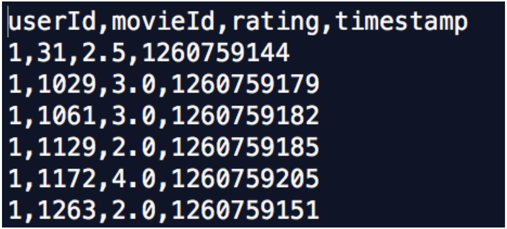
```
userId,movieId,rating,timestamp
1,31,2.5,1260759144
1,1029,3.0,1260759179
1,1061,3.0,1260759182
1,1129,2.0,1260759185
1,1172,4.0,1260759205
1,1263,2.0,1260759151
```

You would like to find the **count** of ratings for each movieId sorted by descending order of count,

1. ratings.groupBy("movieId").agg(desc("count"))
2. ratings.groupBy("movieId").desc("count").show()
3. ratings.groupBy("movieId").count().orderBy(desc("count"))
4. ratings.groupBy("movieId").orderBy(desc("count"))

DataFrame Questions

Suppose you have a file "ratings.csv", which you have loaded into a **Dataframe** called **ratings**



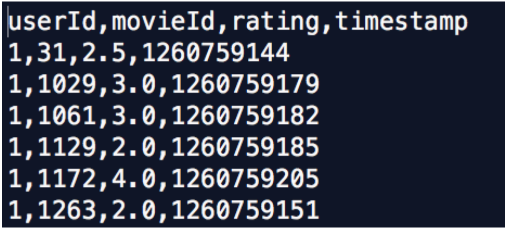
```
userId,movieId,rating,timestamp
1,31,2.5,1260759144
1,1029,3.0,1260759179
1,1061,3.0,1260759182
1,1129,2.0,1260759185
1,1172,4.0,1260759205
1,1263,2.0,1260759151
```

You would like to find the **average** of ratings for each movieId sorted by descending order of average,

1. ratings.groupBy("movieId").avg("rating").sortBy(-1)
2. ratings.groupBy("movieId").agg(avg("rating").alias("avg")).orderBy(desc("avg"))
3. ratings.groupBy("movieId").avg("rating").orderBy(desc("avg"))
4. ratings.groupBy("movieId").avg("rating").orderDesc

DataFrame Questions

Suppose you have a file "ratings.csv", which you have loaded into a **Dataframe** called **ratings**



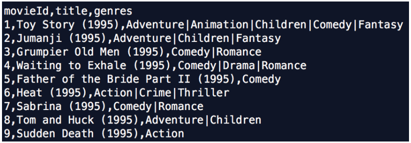
```
userId,movieId,rating,timestamp
1,31,2.5,1260759144
1,1029,3.0,1260759179
1,1061,3.0,1260759182
1,1129,2.0,1260759185
1,1172,4.0,1260759205
1,1263,2.0,1260759151
```

You would like to find the **average** of ratings for each movieId sorted by descending order of average,

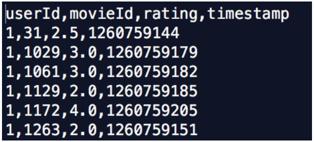
1. ratings.groupBy("movieId").avg("rating").sortBy(-1)
2. ratings.groupBy("movieId").agg(avg("rating").alias("avg")).orderBy(desc("avg"))
3. ratings.groupBy("movieId").avg("rating").orderBy(desc("avg"))
4. ratings.groupBy("movieId").avg("rating").orderDesc

DataFrame Questions

You have loaded the files below into DataFrames **movies** and **ratings**



```
movieId,title,genres
1,Toy Story (1995),Adventure|Animation|Children|Comedy|Fantasy
2,Jumanji (1995),Adventure|Children|Fantasy
3,Grumpier Old Men (1995),Comedy|Romance
4,Waiting to Exhale (1995),Comedy|Drama|Romance
5,Father of the Bride Part II (1995),Comedy
6,Heat (1995),Action|Crime|Thriller
7,Sabrina (1995),Comedy|Romance
8,Tom and Huck (1995),Adventure|Children
9,Sudden Death (1995),Action
```



```
userId,movieId,rating,timestamp
1,31,2.5,1260759144
1,1029,3.0,1260759179
1,1061,3.0,1260759182
1,1129,2.0,1260759185
1,1172,4.0,1260759205
1,1263,2.0,1260759151
```

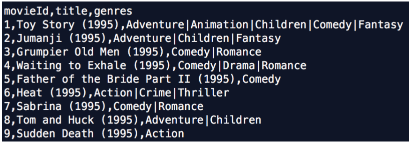
How would you join these two Dataframes? [4]

1. movies.join(ratings, movies.col("movieId") == ratings.col("movieId"))
2. movies.join(ratings, movies.col("movieId") === ratings.col("movieId"))
3. movies.join(ratings)
4. ratings.join(movies)

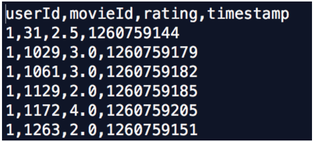
[4] See `https://www.safaribooksonline.com/library/view/high-performance-spark/9781491943199/ch04.html` for more details

DataFrame Questions

You have loaded the files below into DataFrames **movies** and **ratings**



```
movieId,title,genres
1,Toy Story (1995),Adventure|Animation|Children|Comedy|Fantasy
2,Jumanji (1995),Adventure|Children|Fantasy
3,Grumpier Old Men (1995),Comedy|Romance
4,Waiting to Exhale (1995),Comedy|Drama|Romance
5,Father of the Bride Part II (1995),Comedy
6,Heat (1995),Action|Crime|Thriller
7,Sabrina (1995),Comedy|Romance
8,Tom and Huck (1995),Adventure|Children
9,Sudden Death (1995),Action
```



```
userId,movieId,rating,timestamp
1,31,2.5,1260759144
1,1029,3.0,1260759179
1,1061,3.0,1260759182
1,1129,2.0,1260759185
1,1172,4.0,1260759205
1,1263,2.0,1260759151
```

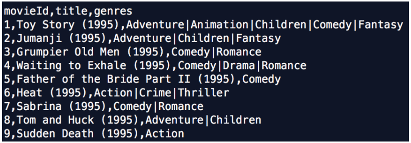
How would you join these two Dataframes? [4]

1. movies.join(ratings, movies.col("movieId") == ratings.col("movieId"))
2. movies.join(ratings, movies.col("movieId") === ratings.col("movieId"))
3. movies.join(ratings)
4. ratings.join(movies)

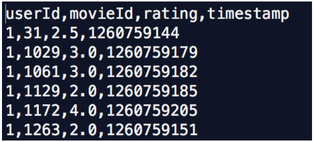
[4] See https://www.safaribooksonline.com/library/view/high-performance-spark/9781491943199/ch04.html for more details

DataFrame Questions

You have loaded the files below into DataFrames **movies** and **ratings**



```
movieId,title,genres
1,Toy Story (1995),Adventure|Animation|Children|Comedy|Fantasy
2,Jumanji (1995),Adventure|Children|Fantasy
3,Grumpier Old Men (1995),Comedy|Romance
4,Waiting to Exhale (1995),Comedy|Drama|Romance
5,Father of the Bride Part II (1995),Comedy
6,Heat (1995),Action|Crime|Thriller
7,Sabrina (1995),Comedy|Romance
8,Tom and Huck (1995),Adventure|Children
9,Sudden Death (1995),Action
```



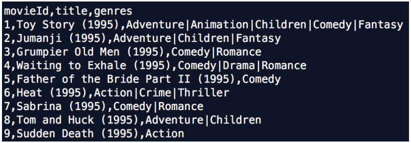
```
userId,movieId,rating,timestamp
1,31,2.5,1260759144
1,1029,3.0,1260759179
1,1061,3.0,1260759182
1,1129,2.0,1260759185
1,1172,4.0,1260759205
1,1263,2.0,1260759151
```

You would like to find the **names** of the **top 5 highest rated movies**. Which of the following approaches would be **most efficient**?

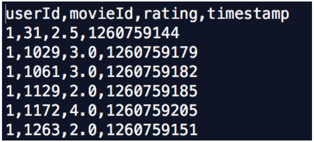
1. First join both Dataframes, compute avg for each movies, then sort by avg in descending order, and finally filter to top 5 rows.
2. First compute the avg for each movie, sort by avg in descending order and filter to top 5 rows, then join the filtered Dataframe to the movies DataFrame

DataFrame Questions

You have loaded the files below into DataFrames **movies** and **ratings**



```
movieId,title,genres
1,Toy Story (1995),Adventure|Animation|Children|Comedy|Fantasy
2,Jumanji (1995),Adventure|Children|Fantasy
3,Grumpier Old Men (1995),Comedy|Romance
4,Waiting to Exhale (1995),Comedy|Drama|Romance
5,Father of the Bride Part II (1995),Comedy
6,Heat (1995),Action|Crime|Thriller
7,Sabrina (1995),Comedy|Romance
8,Tom and Huck (1995),Adventure|Children
9,Sudden Death (1995),Action
```



```
userId,movieId,rating,timestamp
1,31,2.5,1260759144
1,1029,3.0,1260759179
1,1061,3.0,1260759182
1,1129,2.0,1260759185
1,1172,4.0,1260759205
1,1263,2.0,1260759151
```

You would like to find the **names** of the **top 5 highest rated movies**. Which of the following approaches would be **most efficient**?

1. First join both Dataframes, compute avg for each movies, then sort by avg in descending order, and finally filter to top 5 rows.
2. First compute the avg for each movie, sort by avg in descending order and filter to top 5 rows, then join the filtered Dataframe to the movies DataFrame

Outline

1 Topics Covered

2 Introduction to Big Data

3 Hadoop Distributed File System
- HDFS Storage
- HDFS Architecture

4 MapReduce
- Basics
- Scala Questions
- Apache Spark
- DataFrame Questions

5 Machine Learning

Machine Learning

Which of the following are examples of Machine Learning?

1. Programming a home thermostat to start at a fixed time every day.
2. An application automatically learning to classify emails as personal, business, junk, or urgent
3. Creating an email rule that puts every email with "Lottery" in the subject to trash folder.
4. Obtaining movie suggestions from Netflix based on my viewing history
5. A machine that learns to classify clients as high, medium or low risk for default.

Machine Learning

Which of the following are examples of Machine Learning?

1. Programming a home thermostat to start at a fixed time every day.
2. An application automatically learning to classify emails as personal, business, junk, or urgent
3. Creating an email rule that puts every email with "Lottery" in the subject to trash folder.
4. Obtaining movie suggestions from Netflix based on my viewing history
5. A machine that learns to classify clients as high, medium or low risk for default.

Machine Learning

What are the three components of a ML system:

1. Experience (E), Task (T) and Performance measure (P)
2. Experience (E), Time (T) and Practice (P)
3. Work (W), ToDo (T) and Performance measure (P)
4. ELearning (E), Time (T) and Prediction (P)

Machine Learning

What are the three components of a ML system:

1. Experience (E), Task (T) and Performance measure (P)
2. Experience (E), Time (T) and Practice (P)
3. Work (W), ToDo (T) and Performance measure (P)
4. ELearning (E), Time (T) and Prediction (P)

Machine Learning

You are trying to train a machine to predict the amount of rainfall in mm based on weather conditions like humidity, temperature, etc. What type of machine learning is this?

1. Regression
2. Classification
3. Clustering
4. Recommender Systems

Machine Learning

You are trying to train a machine to predict the amount of rainfall in mm based on weather conditions like humidity, temperature, etc. What type of machine learning is this?

1. Regression
2. Classification
3. Clustering
4. Recommender Systems

Machine Learning

The library in Apache Spark that helps with Machine Learning is called __________

1. MachineLibrary
2. MLlib
3. MAlib
4. MLlibraries

Machine Learning

The library in Apache Spark that helps with Machine Learning is called ____________

1. MachineLibrary
2. MLlib
3. MAlib
4. MLlibraries

Machine Learning

What would be the output of the following lines of Spark MLlib code:

```
val sentenceDataFrame = spark.createDataFrame(Seq(
    (0, "Hi I heard about Spark"),
    (1, "I wish Java could use case classes"),
    (2, "Logistic,regression,models,are,neat")
    )).toDF("id", "sentence")
val tokenizer = new Tokenizer().
    setInputCol("sentence").setOutputCol("words")
val tokenized = tokenizer.transform(sentenceDataFrame)
tokenized.select("words").take(1)
```

1. "Hi I heard about Spark"
2. (hi, i, heard, about, spark)
3. (i, wish, java, could, use, case, classes)
4. None of the above

Machine Learning

What would be the output of the following lines of Spark MLlib code:

```
val sentenceDataFrame = spark.createDataFrame(Seq(
    (0, "Hi I heard about Spark"),
    (1, "I wish Java could use case classes"),
    (2, "Logistic,regression,models,are,neat")
    )).toDF("id", "sentence")
val tokenizer = new Tokenizer().
    setInputCol("sentence").setOutputCol("words")
val tokenized = tokenizer.transform(sentenceDataFrame)
tokenized.select("words").take(1)
```

1. "Hi I heard about Spark"
2. (hi, i, heard, about, spark)
3. (i, wish, java, could, use, case, classes)
4. None of the above

Machine Learning

Logistic Regression represents which type of Machine Learning

1. Regression
2. Classification
3. Recommender Systems
4. Clustering

Machine Learning

Logistic Regression represents which type of Machine Learning

1. Regression
2. Classification
3. Recommender Systems
4. Clustering

Machine Learning

Linear Regression represents which type of Machine Learning

1. Regression
2. Classification
3. Recommender Systems
4. Clustering

Machine Learning

Linear Regression represents which type of Machine Learning

1. Regression
2. Classification
3. Recommender Systems
4. Clustering

Questions

You would like to perform Logistic Regression on a dataset and use the code below:

```
val train = spark.read.csv("train.csv")
val lr = new LogisticRegression().setMaxIter(10)
    .setRegParam(0.3).setElasticNetParam(0.8)
```

Which of the following can be used to train the **lr** algorithm on the **train** dataset and obtain a trained model?

1. lr.train(train)
2. lr.fit(train)
3. lr.doTheTraining(train)
4. train.fit(lr)

Questions

You would like to perform Logistic Regression on a dataset and use the code below:

```
val train = spark.read.csv("train.csv")
val lr = new LogisticRegression().setMaxIter(10)
    .setRegParam(0.3).setElasticNetParam(0.8)
```

Which of the following can be used to train the **lr** algorithm on the **train** dataset and obtain a trained model?

1. lr.train(train)
2. lr.fit(train)
3. lr.doTheTraining(train)
4. train.fit(lr)

Questions

You would like to perform Logistic Regression on a dataset and use the code below:

```
val train = spark.read.csv("train.csv")
val lr = new LogisticRegression().setMaxIter(10)
   .setRegParam(0.3).setElasticNetParam(0.8)
# lr is trained on the train dataset to obtain model object
val test = spark.read("test.csv")
```

Which of the following can be used to test the lr model **model** on the **test** dataset?

1. model.transform(test)
2. model.fit(test)
3. model.doTheTesting(test)
4. test.fit(model)

Questions

You would like to perform Logistic Regression on a dataset and use the code below:

```
val train = spark.read.csv("train.csv")
val lr = new LogisticRegression().setMaxIter(10)
  .setRegParam(0.3).setElasticNetParam(0.8)
# lr is trained on the train dataset to obtain model object
val test = spark.read("test.csv")
```

Which of the following can be used to test the lr model **model** on the **test** dataset?

1. model.transform(test)
2. model.fit(test)
3. model.doTheTesting(test)
4. test.fit(model)

Questions

You have a dataset containing 1 million rows of data, which you would like to put into 10 groups such that items in each group are similar to each other and dissimilar to other groups. Which algorithm can help you accomplish this?

1. K-means
2. Decision Tree
3. Logistic Regression
4. Linear Regression

Questions

You have a dataset containing 1 million rows of data, which you would like to put into 10 groups such that items in each group are similar to each other and dissimilar to other groups. Which algorithm can help you accomplish this?

1. K-means
2. Decision Tree
3. Logistic Regression
4. Linear Regression