Part b

Question 1

Describe your data-

Cmd 3

```
# File location and type
path = "/FileStore/tables/movies.csv"

df = spark.read \
  .format("csv") \
  .option("inferSchema", True) \
  .option("header", True) \
  .option("path", path) \
  .load()
```

▸ (2) Spark Jobs

▸ df: pyspark.sql.dataframe.DataFrame = [movieId: integer, rating: integer ... 1 more fields]

Command took 6.64 seconds -- by vishal.kanna@mail.utoronto.ca at 6/18/2021, 10:32:33 AM on firstCluster

```
display(df)
```

▸ (1) Spark Jobs

	movieId	rating	userId
1	2	3	0
2	3	1	0
3	5	2	0
4	9	4	0
5	11	1	0
6	12	2	0
7	15	1	0

Truncated results, showing first 1000 rows.

Command took 0.73 seconds -- by vishal.kanna@mail.utoronto.ca at 6/17/2021, 5:10:28 PM on firstCluster

Cmd 5

```
df.printSchema()
```

```
root
 |-- movieId: integer (nullable = true)
 |-- rating: integer (nullable = true)
 |-- userId: integer (nullable = true)
```

Command took 0.06 seconds -- by vishal.kanna@mail.utoronto.ca at 6/17/2021, 5:10:34 PM on firstCluster

```
1  # check the count of null values for each column
2  df.select([count(when(col(c).isNull(), c)).alias(c) for c in df.columns]).show()
```

▸ (2) Spark Jobs

```
+-------+------+------+
|movieId|rating|userId|
+-------+------+------+
|      0|     0|     0|
+-------+------+------+
```

Command took 1.95 seconds -- by vishal.kanna@mail.utoronto.ca at 6/17/2021, 5:10:37 PM on firstCluster

Our input data is a csv file that has three columns named- movieId, userId and ratings. All these 3 columns have only integers in them and have no null values. MovieId is a unique number given to each movie and userId is a unique number given to each user. And the rating is given to the movies by the user.

Top 10 movie-

```
1  movie_avg=df.groupBy("movieId").agg(F.mean('rating'), F.count('rating'))
```

▸ movie_avg: pyspark.sql.dataframe.DataFrame = [movieId: integer, avg(rating): double ... 1 more fields]

Command took 0.09 seconds -- by vishal.kanna@mail.utoronto.ca at 6/17/2021, 6:43:58 PM on firstCluster

Cmd 33

```
1  movie_avg.sort(col("count(rating)").desc()).limit(10).show()
```

▸ (2) Spark Jobs

```
+-------+------------------+-------------+
|movieId|       avg(rating)|count(rating)|
+-------+------------------+-------------+
|     22|              2.05|           20|
|     50|               1.8|           20|
|     29|               2.4|           20|
|     51|               2.0|           20|
|      6|              1.45|           20|
|      2|2.1052631578947367|           19|
|     15|1.1578947368421053|           19|
|     55|1.7894736842105263|           19|
|     68|2.1578947368421053|           19|
|     94| 2.473684210526316|           19|
+-------+------------------+-------------+
```

Command took 0.99 seconds -- by vishal.kanna@mail.utoronto.ca at 6/17/2021, 6:44:23 PM on firstCluster

Top 10 users-

```
1  user_avg=df.groupBy("userId").agg(F.mean('rating'), F.count('rating'))
```

▸ user_avg: pyspark.sql.dataframe.DataFrame = [userId: integer, avg(rating): double ... 1 more fields]

Command took 0.09 seconds -- by vishal.kanna@mail.utoronto.ca at 6/17/2021, 6:46:45 PM on firstCluster

Cmd 35

```
1  user_avg.sort(col("count(rating)").desc()).limit(10).show()
```

▸ (2) Spark Jobs

```
+------+------------------+-------------+
|userId|       avg(rating)|count(rating)|
+------+------------------+-------------+
|    14|1.7894736842105263|           57|
|     6|1.4385964912280702|           57|
|    22|2.1607142857142856|           56|
|    11|2.2857142857142856|           56|
|     4|1.5636363636363637|           55|
|    12|1.8545454545454545|           55|
|     7|1.6296296296296295|           54|
|     9|1.7924528301886793|           53|
|    18|1.7307692307692308|           52|
|    23|2.1346153846153846|           52|
+------+------------------+-------------+
```

Command took 0.92 seconds -- by vishal.kanna@mail.utoronto.ca at 6/17/2021, 6:46:59 PM on firstCluster