JIGSAW ACADEMY
THE ONLINE SCHOOL OF ANALYTICS

Movie Lens Data Case Study

Business Scenario:
MovieLens data sets were collected by the GroupLens Research Project at the University of Minnesota. The data sets contain reviews of movies by different users.

This data set consists of 100,000 ratings given by 943 users on 1682 movies. Each user has rated at least 20 movies.

Data Sets and Data Dictionary:
The data is given in delimited files. The data dictionaries for each of the file are given below:

u.data contains 100,000 records in tab limited format as show below. The data file given does not have the header row.

UserId	MovieId	Rating	Timestamp
196	242	3	881250949
186	302	3	891717742
22	377	1	878887116
244	51	2	880606923
166	346	1	886397596

u.user contains details of 1682 users with columns of each row delimited by | character. The given file does not contain the header row.

UserId	Age	Gender	Occupation	Zipcode
1	24	M	technician	85711
2	53	F	other	94043
3	23	M	writer	32067
4	24	M	technician	43537
5	33	F	other	15213

Problem Statement:
Use PySpark SQL and load the data into data frames. Write code to perform the following tasks.

1. Give gender-wise breakup of the users
2. Give the top 5 movies which are reviewed maximum number of times
3. Give the top 5 users who reviewed maximum number of movies
4. List the top 10 movies which received highest number of 5 star ratings
5. List the top 10 users who gave highest number of 5 star ratings
6. Prepare a dataframe from user data that has the records transformed as below.


```
UserId|Age|AgeGroup|Gender|Male|Female|Occupation|OccupationId|Zipcode
1      |24 |2        |M     |1   |0     |technician|1.0          |85711
2      |53 |5        |F     |0   |1     |other     |2.0          |94043
3      |23 |2        |M     |1   |0     |writer    |3.0          |32067
4      |24 |2        |M     |1   |0     |technician|1.0          |43537
5      |33 |3        |F     |0   |1     |other     |2.0          |15213
```

The transformations to be done are as below:

1. Binning of data in the column Age as 2, 3, 4, 5 etc.
2. Adding a dummy variable for the column Gender and filling in 1 or 0 accordingly
3. Generating an index number for the list of all categorical values in the column Occupation and use the number in place of the Occupation name string. (The occupation ids shown above are for example. They can be any number need not be the same numbers as shown above.)

Solution:

The solution is provided in the iPython notebook file. The approach we took is explained below. There are other ways of performing the above tasks.

1. Read and load the input data files as a delimited file. As we do not have a header row PySparkSQL `read.load` method assigns default column names such as `_c0, _c1` etc.
2. We can rename the columns as per our convenience so that we can refer to them easily in the later steps in the code. We have two different methods to do this and both of these are shown in the code.
3. We create a temp view for each dataframe (users and movies) and write SQL queries especially for the first five questions.
4. For the sixth question for first two transformations we use SQL features like an arithmetic function and CASE-WHEN structure.
5. For third transformation i.e. get unique id number associated with each occupation name string, we use `StringIndexer` API.

You can try out other ways of doing the same tasks as an exercise. For example:

- Step 1 above can be changed to read the data with `testFile` method as an RDD and convert it into a dataframe as shown in the video on building dataframes using RDDs.
- For Step 2 we used the PySpark's DSL (Domain Specific Language) methods like `.alias` and `.withColumnRenamed`. Similarly for Step 3 also we can use methods instead of using plain SQL queries to get familiar with the methods of the DSL.
- In Step 4 also instead of SQL you can explore the methods available for dataframes to do the transformations.