Count the no. of; how many times a movie has been rated & getting them sorted in ascending order

from mrjob.job import MRJob

from mrjob.step import MRStep

class RatingsBreakdown(MRJob):

def steps(self):

return [

MRStep(mapper=self.mapper\_get\_ratings,

reducer=self.reducer\_count\_ratings),

MRStep(reducer=self.reducer\_sorted\_output)

]

’’’The declarative/main function that invokes & store the function’s value in mapper and reducer variable.

The mapper variable traverses across the data file and transforms the data in key-value format. The mapper variable stores the value in the form:

‘3443’ : 1

‘3443’ : 1

The reducer variable counts the no. of 1’s against each Movie ID so as to find the no. of times; as movie has been rated, and sorts them in ascending order.

‘3443’ : 2

’’’

def mapper\_get\_ratings(self, \_, line):

(userID, movieID, rating, timestamp) = line.split('\t')

yield movieID, 1

def reducer\_count\_ratings(self, key, values):

yield str(sum(values)).zfill(5), key

def reducer\_sorted\_output(self, count, movies):

for movie in movies:

yield movie, count

if \_\_name\_\_ == '\_\_main\_\_':

RatingsBreakdown.run()

Before executing code, Python interpreter reads source file and define few special variables/global variables.  
If the python interpreter is running that module (the source file) as the main program, it sets the special \_\_name\_\_ variable to have a value **“\_\_main\_\_”**. If this file is being imported from another module, \_\_name\_\_ will be set to the **module’s name.** Module’s name is available as value to \_\_name\_\_ global variable. So, sometimes you want to write a .py file that can be both used by other programs and/or modules as a module, and can also be run as the main program itself.

The yield statement suspends function’s execution and sends a value back to the caller, but retains enough state to enable function to resume where it is left off. When resumed, the function continues execution immediately after the last yield run. This allows its code to produce a series of values over time, rather than computing them at once and sending them back like a list.

Output:

