BUSINESS INTELLIGENCE AND
DATA ANALYTICS
– WE PREDICT THE FUTURE –

University of Petra

IOA Institute of Analytics

307401
Big Data and Data Warehouses

Practical Examples

hadoop

Simple Map and reduce in Python (No Big Data)

The Map Function

This simple python code (no big data) demonstrate the idea of map where we map a function to a list of numbers.

- **Method 1:**

```
lst = [1, 2, 3, 4]
list(map(lambda x: x*x, lst))
```

```
[1, 4, 9, 16]
```

- **Method 2:**

```
def square(x):
      return x*x
list(map(square, lst))
```

```
[1, 4, 9, 16]
```

Simple Map and reduce in Python (No Big Data)

The Reduce Function

This simple python code (no big data) demonstrate the idea of reduce, where we reduce a group of numbers into a single number.

- **Method 1:**

```
from functools import reduce
reduce(lambda x, y: x + y, lst)
10
```

- **Method 2:**

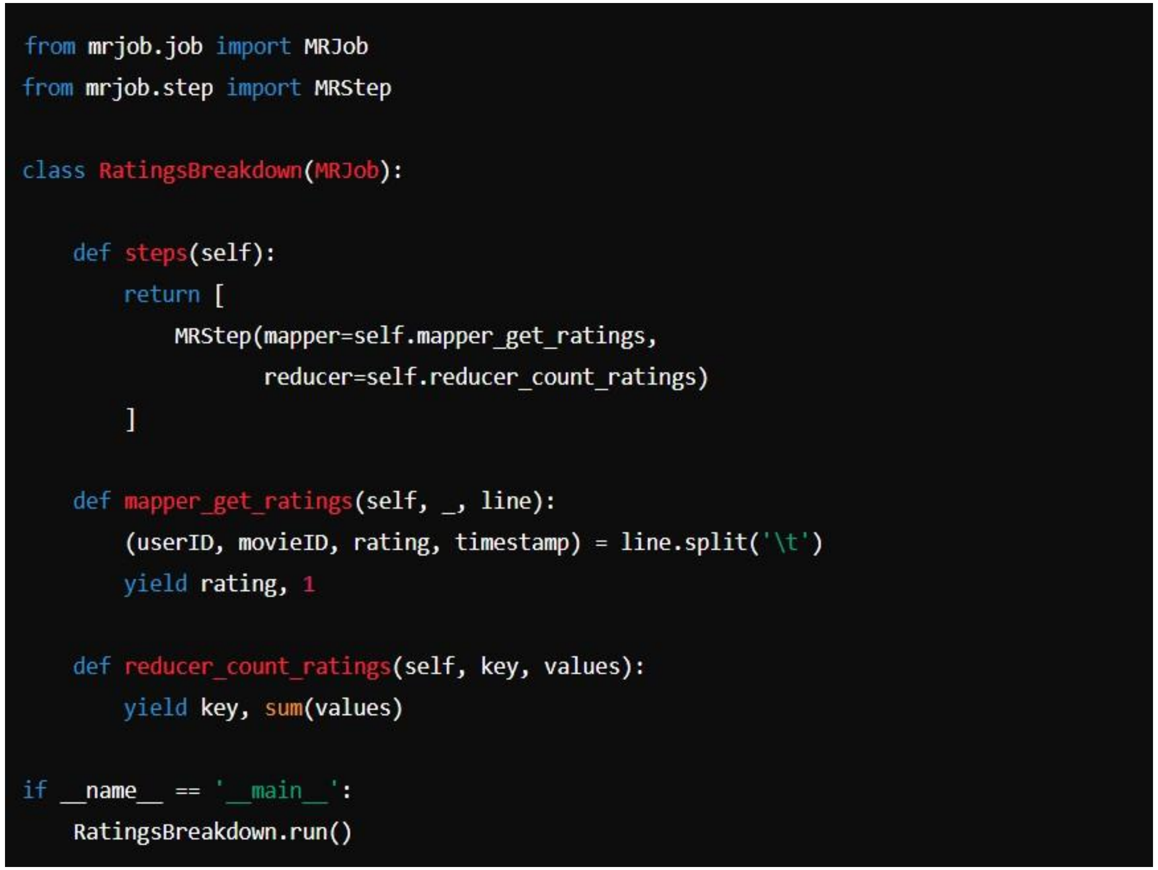
```
def add_reduce(x, y):
  out = x + y
  print(f"{x}+{y}-->{out}")
  return out
reduce(add_reduce,lst)

3<--2+1
6<--3+3
10<--4+6
10
```

Map Reduce Using Hadoop and MRJob Python Library

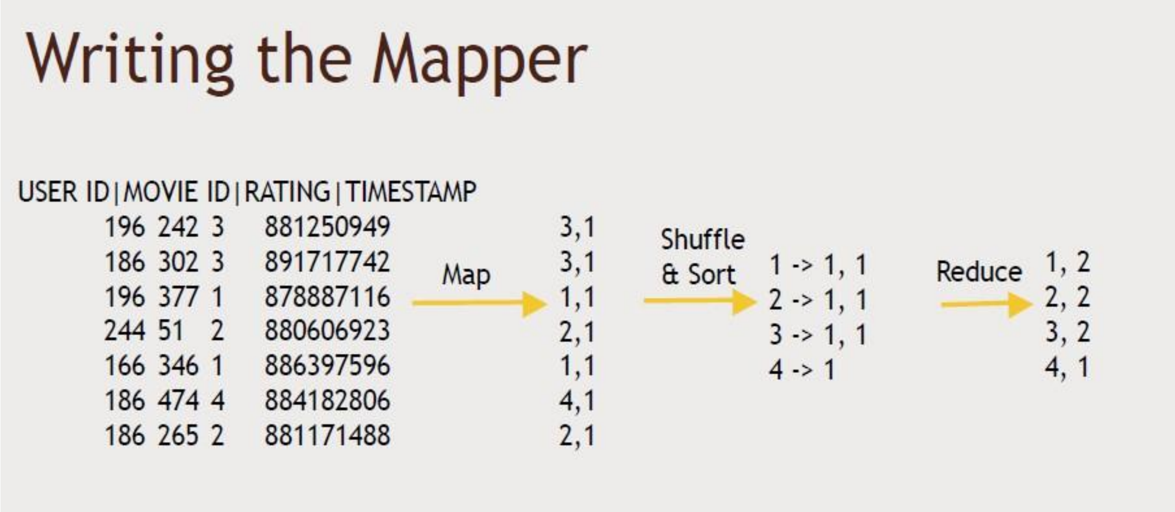
Map Reduce Movie Ratings Count Example

User ID	Movie ID	Rating	Time Stamp
0	50	5	881250949
0	172	5	881250949
0	133	1	881250949
196	242	3	881250949
186	302	3	891717742
22	377	1	878887116
244	51	2	880606923
166	346	1	886397596
298	474	4	884182806
115	265	2	881171488
253	465	5	891628467
305	451	3	886324817

Map Reduce Movie Ratings Count Example



```
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)
        ]

    def mapper_get_ratings(self, _, line):
        (userID, movieID, rating, timestamp) = line.split('\t')
        yield rating, 1

    def reducer_count_ratings(self, key, values):
        yield key, sum(values)

if __name__ == '__main__':
    RatingsBreakdown.run()
```



Writing the Mapper

```
USER ID|MOVIE ID|RATING|TIMESTAMP
      196  242  3    881250949          3,1
      186  302  3    891717742    Map   3,1   Shuffle    1 -> 1, 1   Reduce   1, 2
      196  377  1    878887116   ---->  1,1   & Sort     2 -> 1, 1   ---->    2, 2
      244  51   2    880606923          2,1   ---->      3 -> 1, 1            3, 2
      166  346  1    886397596          1,1              4 -> 1               4, 1
      186  474  4    884182806          4,1
      186  265  2    881171488          2,1
```

Map Reduce Example – Two Steps (Sorting after Counting)

```
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)
        ]

    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()
```

Word Count Example