Run mapreduce job with mrjob library:

Installed “mrjob” library
Make a python file named “1.py”

```
from mrjob.job import MRJob
class WordCount(MRJob):
   def mapper(self, _, line):
       for word in line.split():
           yield (word, 1)

   def reducer(self, word, counts):
       yield (word, sum(counts))
```

Make a text file named “abc.txt”:

```
This is an orange.
This is an apple.
```

Open terminal and write command

python 1.py abc.txt

Output:
(venv) stackintel@Stack-Intel-MA:~/PycharmProjects/mlpyspark$ python 1.py abc.txt
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory /tmp/1.stackintel.20240206.050652.927806
Running step 1 of 1...
job output is in /tmp/1.stackintel.20240206.050652.927806/output
Streaming final output from /tmp/1.stackintel.20240206.050652.927806/output...
"is" 2
"an" 2
"This" 2
"orange." 1
"apple." 1
Removing temp directory /tmp/1.stackintel.20240206.050652.927806...

Run mapreduce job without mrjob library:

Make mapper.py file:

#!/usr/bin/env python
import sys

```
# Read each line from stdin

for line in sys.stdin:
    # Get the words in each line
    words = line.split()
    # Generate the count for each word
for word in words:
    # Write the key-value pair to stdout to be processed by
    # the reducer.
    # The key is anything before the first tab character and the
    #value is anything after the first tab character.
    print ('{0}\t{1}'.format(word, 1))
```

Make reducer.py file:

```
#!/usr/bin/env python
import sys
curr_word = None
curr_count = 0
# Process each key-value pair from the mapper
for line in sys.stdin:
    # Get the key and value from the current line
    #print("hello world")
    word, count = line.split('\t')
    #print("word",word)
    # Convert the count to an int
    count = int(count)
    # If the current word is the same as the previous word,
    # increment its count, otherwise print the words count
    # to stdout
    if word == curr_word:
        curr_count += count

    else:
            print( curr_word, curr_count)
            curr_word = word
#           curr_count = count
# Output the count for the last word
#if curr_word == word:
#print '{0}\t{1}'.format(curr_word, curr_count)
```

Open terminal run this command:
echo 'jack be nimble jack be quick' | ./mapper.py | sort -t 1 | ./reducer.py

Output:
None 0
Be 2
Jack 2
nimble 1
Quick 1