

Lab 2

[Click here to Register Attendance](https://forms.gle/pesShFDe1ibVfS3L7)

|  |  |
| --- | --- |
| Name | Ronan McCormack |
| Date | 17/02/2022 |
| Student No | S00144576 |
| Student Email | S00144576@mail.itsligo.ie |

### **Running a MapReduce Job on your local machine**

1. Go to the UCI Machine Learning Data Repository and explore the datasets available
2. Download a dataset of your choice and extract the files (I used the [Iris dataset](https://archive.ics.uci.edu/ml/datasets/iris))
3. Download the python [file here called MapReduceIris.py](https://github.com/marloftitsligo/ProgrammingForBigData/tree/main/MapReduceIris) and place in the same folder as your dataset
4. Open a Command Line (assuming you have Python installed)
5. Run the following command (replace iris.data with your data filename):  
   python MapReduceIris.py iris.data
6. You will likely get an error message as this python script uses a library called mrjob.   
   To install the library type:  
   pip install mrjob  
   You should get a confirmation message: Successfully installed mrjob-0.7.4
7. Try running the original command again (replace iris.data with your data filename):  
   python MapReduceIris.py iris.data
8. You should get the following in the output:   
   "setosa sepal width avg" 3.418
9. Post a screenshot of the output here:

|  |
| --- |
|  |

1. Try to adjust the Reducer calculation to find the average of other Species  
   Describe the rationales for your changes, your code and screenshot your output here:

|  |
| --- |
| 1. from mrjob.job import MRJob 2. from mrjob.step import MRStep 3. import re 4. DATA\_RE = re.compile(r"[\w.-]+") 5. class MRProb2\_3(MRJob): 6. def steps(self): 7. return [ 8. MRStep(mapper=self.mapper\_get\_sepW, 9. reducer=self.reducer\_get\_avg) 10. ] 11. def mapper\_get\_sepW (self, \_, line): 12. # yield each petal width 13. data = DATA\_RE.findall(line) 14. for ele in line: 15. sep\_W = float(data[1]) 16. flower\_class = data[4] 17. yield (flower\_class, sep\_W) 18. def reducer\_get\_avg(self, key, values): 19. # get max of the petal widths 20. size, total = 0, 0 21. for val in values: 22. size += 1 23. total += val 24. yield (key + ' ' +"width avg", round(total,1) / size) 25. if \_\_name\_\_ == '\_\_main\_\_': 26. MRProb2\_3.run()  * Instead of searching for one species in the mapper function, for each line in the data we get the sepal width and the flower class. * The reducer doesn’t take any changes apart from the output which returns the key which is the species plus the mean. |

1. Find another dataset to use with this MapReduce code and make necessary changes to the code to perform an analysis of your choice.   
   Insert the name and a link to the dataset here and a short description of the analyses you performed:

|  |
| --- |
| I used a crime dataset from the Vancouver Police Department for 2021 available through their portal <https://geodash.vpd.ca/opendata/>  I analysed the type of crime committed and the count of the type of crime for 2021.  Code:  from mrjob.job import MRJob  from mrjob.step import MRStep  import re  WORD\_RE = re.compile(r"[\w.'\s]+")  class MRMostUsedWord(MRJob):  def steps(self):  return [  MRStep(mapper=self.mapper\_get\_words,  reducer=self.reducer\_count\_words)  ]  def mapper\_get\_words(self, \_, line):  # yield each word in the line  data = WORD\_RE.findall(line)  crime = data[0]  yield (crime, 1)  def reducer\_count\_words(self, word, counts):  size, total = 0, 0  for val in counts:  size += 1  yield ('Type of Crime:' + ' ' + word,size)  if \_\_name\_\_ == '\_\_main\_\_':  MRMostUsedWord.run() |

1. Push the code and screenshots from step 11 above to Github and post the link here:

|  |
| --- |
| <https://github.com/ronanmccormack-ca/ProgrammingforBigData/tree/main/Lab%202> |