Introduction

This task comprises of setting up and exploring the yelp dataset. The yelp dataset contains over 6 million reviews from 200,000 businesses in 10 metropolitan areas using 8.69 GB of storage[1]. The data is contain in json files which have seen a rise in popularity over the past decade, over taking xml file formats as the most popular format in late 2012[2].

Using Ackoff's DIKW hierarchy, the aim of the task is to generate information from data using the means of understanding and connectedness[3]. By asking data science questions and using statistical techniques to answer them, a higher definition of the data is uncovered called information, where trends, patterns and relationships are shown. This newly gained information can allow for richer decision making - understanding the past to affect the future.

As the nature of the dataset is so large, normal methods of exploration would be too costly for both the computer and user, therefore the exploration task must be distributed. One method of doing this is to use two pieces of software made by Apache; Hadoop and Spark. In order to store the files in a distributed way, I will employ the Hadoop Distributed File System (HDFS) which stores data on commodity machines, providing a very high aggregate bandwidth across the cluster[4]. In order to explore the data to uncover a higher definition, Spark's Core API will be used. Apache Spark provides an easy to use interface for distributed computing that follows the Map Reduce model and has built in fault tolerance by redundantly storing the data in blocks[5]. Apache Spark has a data structure called a Resilient Distributed Dataset which allows for exploratory task to be run in parallel.

The MapReduce model contains two tasks, Map and Reduce. Map takes a set of data and converts it into another set of data, where individual elements are broken down into key value pairs. Then the reduce which takes the output from a map as an input and combines those key value pairs into a smaller set of tuples or aggregates them into values[6].

In order to achieve the aim, the yelp dataset was stored on the HDFS on the Poincare cluster at MMU, the commands used can be found in the Hadoop Scripts file. Once the dataset was stored in a distributed way, Apache Spark will be employed to analyse and explore the data, which can be found in the python script. Once run the outputs are produced giving a higher definition of the data, found in the output file.

Conclusion

The first question was related to reviews, having two parameters on which the data is filtered funny and useful. In order to attain the information, the json file must first be mapped into a key value pair which then can be filtered based on a usefulness of 30 or more and a funniness of 20 or more. In order to aggregate the information into something meaningful such as count which produced 4,011. To validate the result, .take was used to look at a sample of the results.

The second question wanted to find all Night Life businesses in Las Vegas that were rated 4.5 stars or higher. The business json file is loaded into an RDD, which is then mapped into a key value pair. The key value pairs are then filtered based on categories, city and stars. I then sorted the results in order of star rating and produced a list of the top 10 names. In order to see the full extent of the results, I aggregated them using a count which showed that there are 381 Night-life businesses in Las Vegas rated 4.5 stars or more.

The third question wanted to find the top 10 'useful' reviewers of nightlife in Urbana-Champaign. In order to attain the information, the business and review files had to be joined together. This was done by first mapping the two files into key value pairs with the business id being the key. Once the two files were joined

together several filters were applied to get a list of the top ten useful reviewers. The top reviewer had 22 useful reviews but the mode of the top 10 was 19.

The last question, which I have proposed was to see if there is a relation between useful reviews and active users. In order to gain such information a few assumptions were made. Firstly, to distinguish between active and non active I set a boundary of 5 reviews such that a reviewer with 5 or more reviews is classed as active and less than 5 is non active. Secondly, in order to compare between the two groups I employed the central tendency measure of the mean to find the average usefulness of reviews from active and non active members.

As the mean isn't an associative or communicative process a distributed method has to be implemented. This is done by using key value pairs which use both sum and count - being both associative or communicative process.

Looking at the results, there is a clear relationship. As a base, I found the average usefulness of all reviews in the dataset which returned a value of 29.34. Comparing that to non active users, which got 1.00 usefulness showing that non active users produce less than the average of all users. Finally, comparing active and non active users we see that active users had a usefulness of 60.36 having a difference of over 59 votes.

References

[1] "Yelp Dataset", *Yelp.com*, 2019. [Online]. Available: https://www.yelp.com/dataset. [Accessed: 02- Apr- 2019].

[2] "The Rise and Rise of JSON", *Twobithistory.org*, 2017. [Online]. Available: https://twobithistory.org/2017/09/21/the-rise-and-rise-of-json.html. [Accessed: 02- Apr- 2019].

[3] "Data, Information, Knowledge, & Wisdom", *Systems-thinking.org*. [Online]. Available: http://www.systems-thinking.org/dikw/dikw.htm. [Accessed: 02- Apr- 2019].

[4] "What is Hadoop?", *SAS.com*, 2019. [Online]. Available: https://www.sas.com/en_gb/insights/big-data/hadoop.html. [Accessed: 02- Apr- 2019].

[5] "Apache Spark", *En.wikipedia.org*, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Apache_Spark. [Accessed: 02- Apr- 2019].

[6] "MapReduce Tutorial", *Hadoop.apache.org*, 2019. [Online]. Available: https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html. [Accessed: 02- Apr- 2019].