CSC1109 Data at Speed & Scale - Assignment 1

Luqman Rostam - 22371746
https://github.com/LuqmanRostam/CSC1109-Assignment-1—Luqman-Rostam

1 Introduction

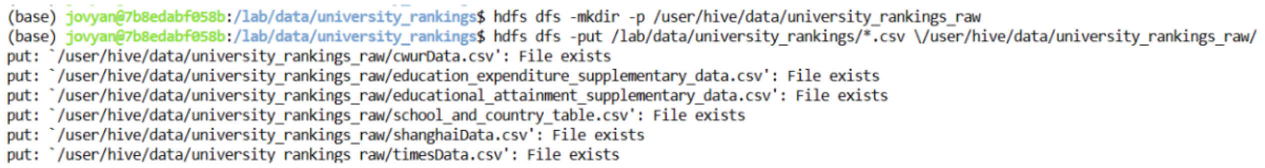
Apache Hadoop is a framework which allows large amounts of data to be processed in a distributed way across multiple computers, it does this using a MapReduce model. Tools such as Apache Pig and Apache Hive work on top of Hadoop, they are high level scripting tools that make working with big data a lot easier, these tools can be used to clean, transform and process data.

In this assignment I used both Pig and Hive to clean and analyze the University Rankings dataset. The data which was used, includes the rankings from the Center for World University Rankings(CWUR) and the Time Higher Education, both datasets were loaded, cleaned and saved using pig.

2 Data Preparation and HDFS Storage

For this analysis, the original files were originally downloaded in a compressed format(zipped folder) the files were extracted using a simple command "unzip". In order to make use of Hadoop's distributed processing capabilities, the data first had to be uploaded to Hadoop's Distributed File System (HDFS). This is especially important when working with larger datasets so that multiple machines can process the data in parallel, rather than overloading a single computer.

I created a new dedicated directory called "university_rankings_raw" in HDFS, this is where the original, unprocessed datasets are stored. The data is uploaded to this HDFS directory using the "put" command, by doing this it allows both Pig and Hive to efficiently access and process the data using Hadoop processing. This is an essential step in the overall data pipeline as it allows us to use Hadoop's scalability and performance benefits.



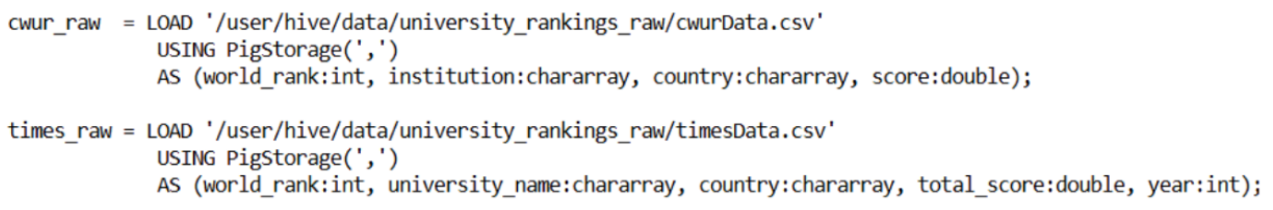
```
(base) jovyan@7b8edabf058b:/lab/data/university_rankings$ hdfs dfs -mkdir -p /user/hive/data/university_rankings_raw
(base) jovyan@7b8edabf058b:/lab/data/university_rankings$ hdfs dfs -put /lab/data/university_rankings/*.csv \/user/hive/data/university_rankings_raw/
put: `/user/hive/data/university_rankings_raw/cwurData.csv': File exists
put: `/user/hive/data/university_rankings_raw/education_expenditure_supplementary_data.csv': File exists
put: `/user/hive/data/university_rankings_raw/educational_attainment_supplementary_data.csv': File exists
put: `/user/hive/data/university_rankings_raw/school_and_country_table.csv': File exists
put: `/user/hive/data/university_rankings_raw/shanghaiData.csv': File exists
put: `/user/hive/data/university rankings raw/timesData.csv': File exists
```

Figure 1: HDFS directory structure

I also created another directory called "university_rankings_clean" dedicated to storing the processed and cleaned datasets, which we will later use to query on. Having two separate directories is good practice as it keeps both the original and the processed data safe.

3 Data Cleaning with Pig

With the raw data now in the HDFS, the next step is to clean and prepare the data set for further analysis, using Pig scripts. I was able to prepare the data set by using these essential data cleaning procedures. The script first begins with loading the two main datasets cwurData.csv and timesData.csv and then defining the schema accordingly to their data types.

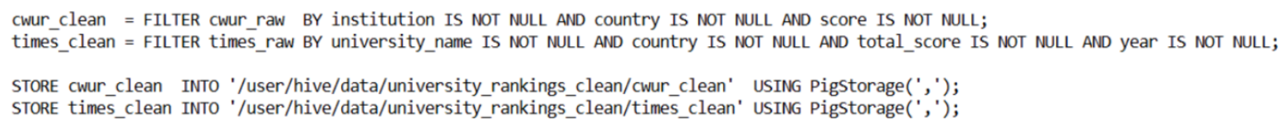


```
cwur_raw  = LOAD '/user/hive/data/university_rankings_raw/cwurData.csv'
            USING PigStorage(',')
            AS (world_rank:int, institution:chararray, country:chararray, score:double);

times_raw = LOAD '/user/hive/data/university_rankings_raw/timesData.csv'
            USING PigStorage(',')
            AS (world_rank:int, university_name:chararray, country:chararray, total_score:double, year:int);
```

Figure 2: Loading datasets and defining schema

To clean both datasets I used the "FILTER" command to remove any incomplete records from the datasets, for the CWUR dataset I dropped any rows missing a university name, country or score, and for the Times dataset I checked for missing data in the same columns as well as an extra check to ensure that the year data is also present.



```
cwur_clean  = FILTER cwur_raw  BY institution IS NOT NULL AND country IS NOT NULL AND score IS NOT NULL;
times_clean = FILTER times_raw BY university_name IS NOT NULL AND country IS NOT NULL AND total_score IS NOT NULL AND year IS NOT NULL;

STORE cwur_clean  INTO '/user/hive/data/university_rankings_clean/cwur_clean'  USING PigStorage(',');
STORE times_clean INTO '/user/hive/data/university_rankings_clean/times_clean' USING PigStorage(',');
```

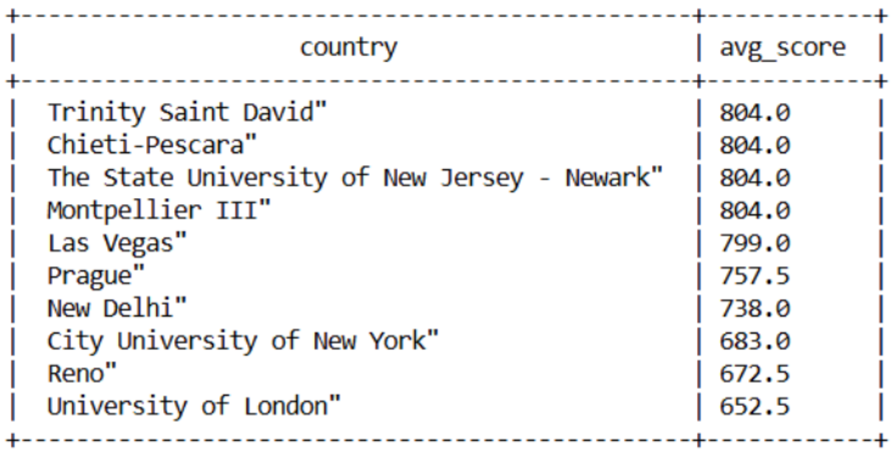
Figure 3: Filtering commands for data cleaning

As our mostly numerical dataset does not require much processing I went ahead and used the "STORE" command to save the newly cleaned datasets to the new HDFS directory, where it will be available for query analysis. This cleaning Pig script is executed using "pig -x mapreduce clean_universities.pig"

4 Simple Hive Queries

4.1 Highest Average Score by Country

This query was pretty straightforward, it required the data to be grouped by country with the average score calculated for each.



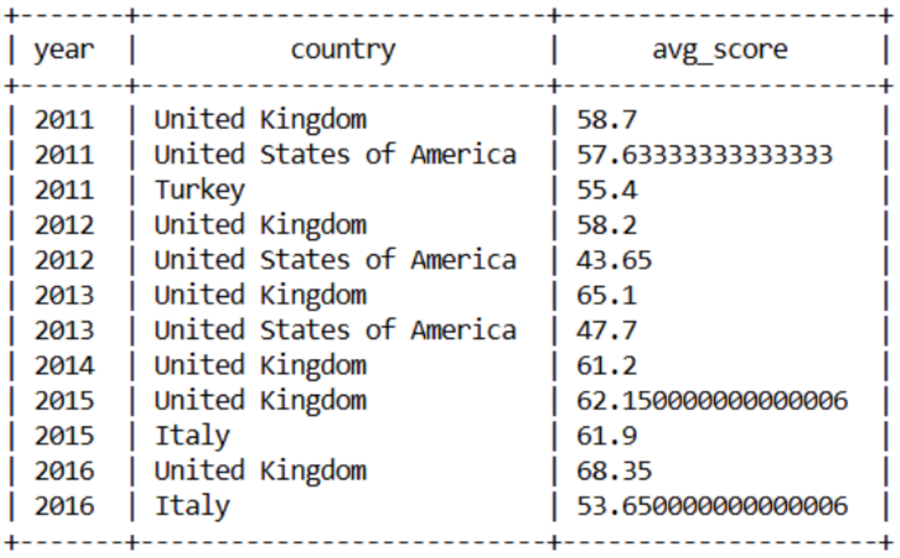
```
+-------------------------------------------------+-----------+
|                     country                     | avg_score |
+-------------------------------------------------+-----------+
| Trinity Saint David"                            | 804.0     |
| Chieti-Pescara"                                 | 804.0     |
| The State University of New Jersey - Newark"    | 804.0     |
| Montpellier III"                                | 804.0     |
| Las Vegas"                                      | 799.0     |
| Prague"                                         | 757.5     |
| New Delhi"                                      | 738.0     |
| City University of New York"                    | 683.0     |
| Reno"                                           | 672.5     |
| University of London"                           | 652.5     |
+-------------------------------------------------+-----------+
```

Figure 4: Hive results for highest average score by country

The list above shows the average university score for different countries, based on the CWUR ranking, the higher the score the better the performance. A group of institutions, including Trinity Saint David and Montpellier III, can be seen to share the highest average score of 804.0, indicating top-tier performance.

4.2 Average Times score per year and country

This query required grouping the data by both year and country to calculate the average score for each combination.



```
+-------+--------------------------+--------------------+
| year  |         country          |     avg_score      |
+-------+--------------------------+--------------------+
| 2011  | United Kingdom           | 58.7               |
| 2011  | United States of America | 57.63333333333333  |
| 2011  | Turkey                   | 55.4               |
| 2012  | United Kingdom           | 58.2               |
| 2012  | United States of America | 43.65              |
| 2013  | United Kingdom           | 65.1               |
| 2013  | United States of America | 47.7               |
| 2014  | United Kingdom           | 61.2               |
| 2015  | United Kingdom           | 62.150000000000006 |
| 2015  | Italy                    | 61.9               |
| 2016  | United Kingdom           | 68.35              |
| 2016  | Italy                    | 53.650000000000006 |
+-------+--------------------------+--------------------+
```

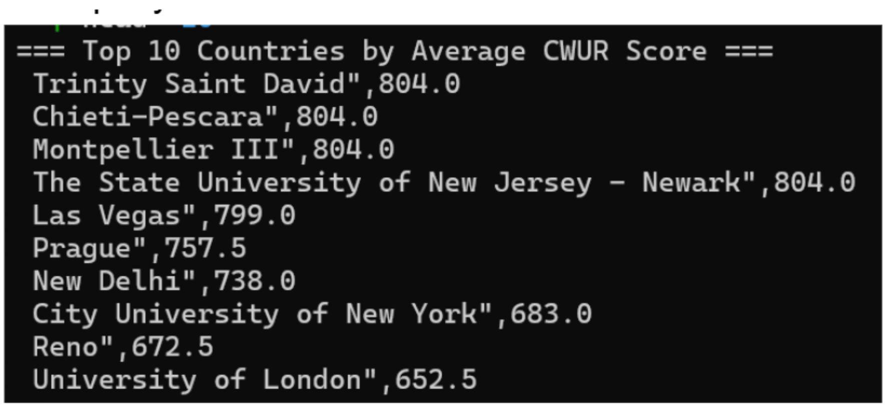
Figure 5: Hive results for average Times score per year and country

From the results, the United Kingdom can be seen to have a consistently high average score, the results show a general upward trend from 58.7 in 2011 to 68.35 in 2016. In contrast to that, the United States of America's average score appears to have decreased significantly between the years 2011 and 2012.

5 Simple Pig Queries

5.1 Highest Average Score by Country

This query was as straightforward as the hive version, based on the same CWUR dataset. As the query and dataset was identical, the Pig query returned identical results and figures as the Hive query.



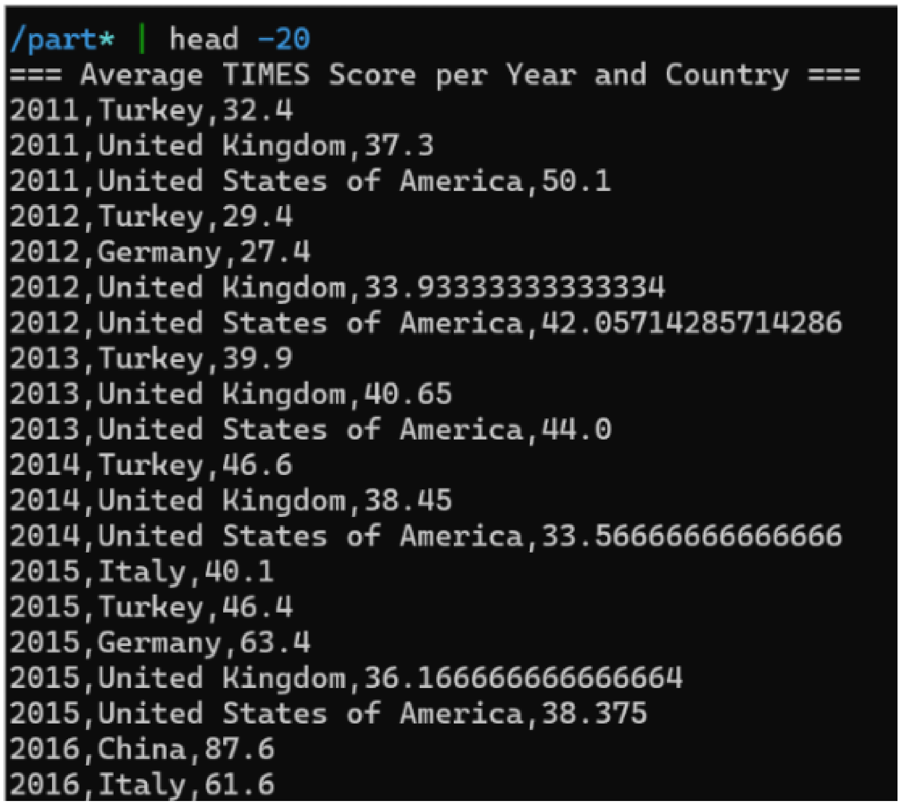
```
=== Top 10 Countries by Average CWUR Score ===
 Trinity Saint David",804.0
 Chieti-Pescara",804.0
 Montpellier III",804.0
 The State University of New Jersey - Newark",804.0
 Las Vegas",799.0
 Prague",757.5
 New Delhi",738.0
 City University of New York",683.0
 Reno",672.5
 University of London",652.5
```

Figure 6: Pig results for highest average score by country

The results showed that the Trinity Saint David, Montpellier III, The State University of New Jersey and Chieti-Pescara share the highest average score of 804.0, of course indicating top-tier performance.

5.2 Average Times score per year and country

This query, although identical to the previous ”Average Times score per year and country” hive query, returns different results. This is due to the fact that I decided to use the data from times dataset rather than the CWUR data which was used in the hive query.



```
/part* | head -20
=== Average TIMES Score per Year and Country ===
2011,Turkey,32.4
2011,United Kingdom,37.3
2011,United States of America,50.1
2012,Turkey,29.4
2012,Germany,27.4
2012,United Kingdom,33.93333333333334
2012,United States of America,42.05714285714286
2013,Turkey,39.9
2013,United Kingdom,40.65
2013,United States of America,44.0
2014,Turkey,46.6
2014,United Kingdom,38.45
2014,United States of America,33.5666666666666
2015,Italy,40.1
2015,Turkey,46.4
2015,Germany,63.4
2015,United Kingdom,36.166666666666664
2015,United States of America,38.375
2016,China,87.6
2016,Italy,61.6
```

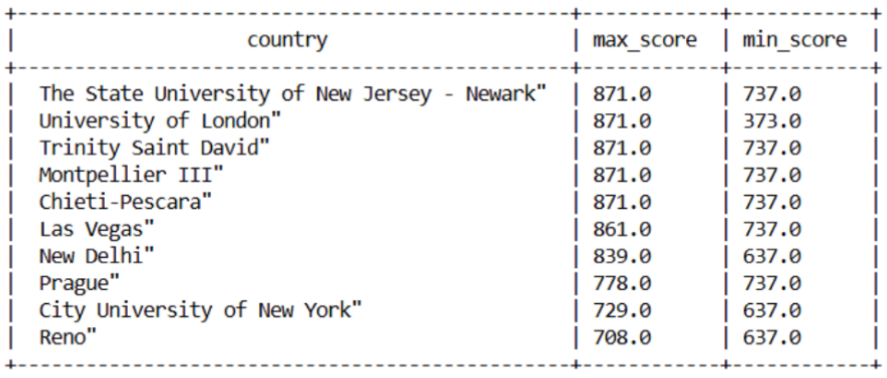
Figure 7: Pig results for average Times score per year and country

Nevertheless, results show a fairly similar trend to the hive query, the results indicate that the United States and United Kingdom consistently perform and score a high average across the years.

6 Hive Complex Queries

6.1 Highest and Lowest Scores by Country (Aggregate)

This query utilized the MAX and MIN aggregate functions to calculate the highest and lowest university scores for each country.



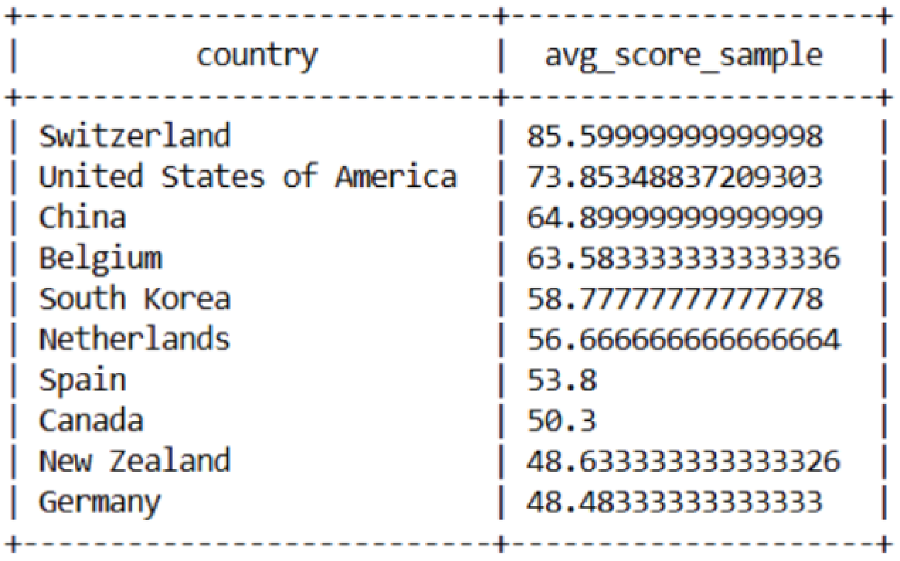
```
+----------------------------------------------------+------------+------------+
|                       country                      | max_score  | min_score  |
+----------------------------------------------------+------------+------------+
|  The State University of New Jersey - Newark"      | 871.0      | 737.0      |
|  University of London"                             | 871.0      | 373.0      |
|  Trinity Saint David"                              | 871.0      | 737.0      |
|  Montpellier III"                                  | 871.0      | 737.0      |
|  Chieti-Pescara"                                   | 871.0      | 737.0      |
|  Las Vegas"                                        | 861.0      | 737.0      |
|  New Delhi"                                        | 839.0      | 637.0      |
|  Prague"                                           | 778.0      | 737.0      |
|  City University of New York"                      | 729.0      | 637.0      |
|  Reno"                                             | 708.0      | 637.0      |
+----------------------------------------------------+------------+------------+
```

Figure 8: Hive results for highest and lowest scores by country

As you can see from the results, everything is ordered by the maximum score in descending order. From the table you can see a group of five countries share the highest possible score of 871 and from among the five countries, there are four who also share a minimum score of 737, this indicates that these universities are some of the most consistent in their performance.

6.2 Sampling with TABLESAMPLE (Subquery)

This query calculates which country has the best university on average, however it makes an educated guess based only on a tablesample of the data(10%), this is often done when you have a massive dataset as it saves you both time and computing power. Of course the trade off for this increase in efficiency is a loss in precision and accuracy.



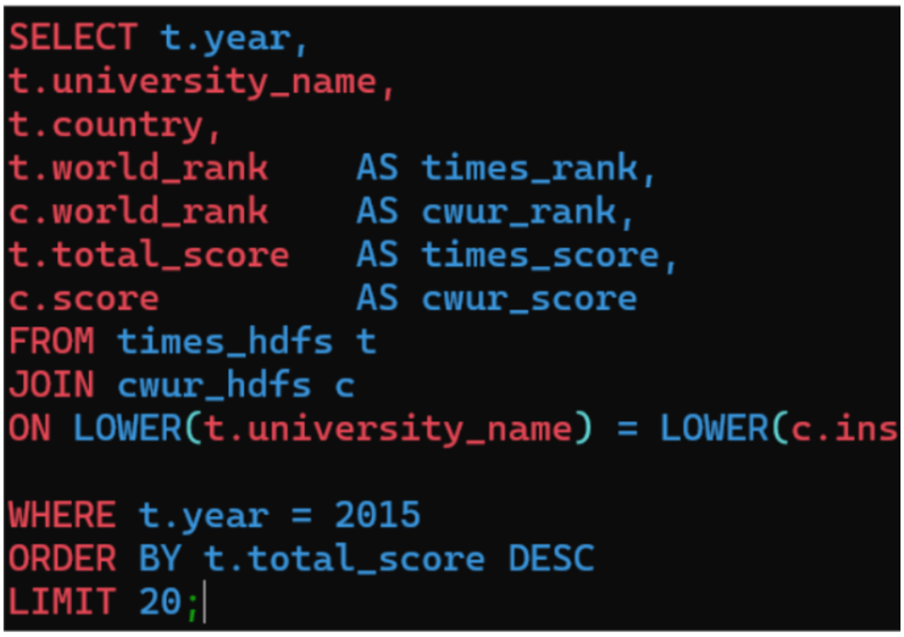
```
+--------------------------+--------------------+
|         country          |  avg_score_sample  |
+--------------------------+--------------------+
| Switzerland              | 85.59999999999998  |
| United States of America | 73.85348837209303  |
| China                    | 64.89999999999999  |
| Belgium                  | 63.583333333333336 |
| South Korea              | 58.77777777777778  |
| Netherlands              | 56.666666666666664 |
| Spain                    | 53.8               |
| Canada                   | 50.3               |
| New Zealand              | 48.633333333333326 |
| Germany                  | 48.48333333333333  |
+--------------------------+--------------------+
```

Figure 9: Hive results using TABLESAMPLE

From the results we can clearly see that from our tablesample Switzerland has the best universities on average followed by USA and China, this is a good example of the trade off you might get when using tablesamples.

6.3 JOIN between CWUR and TIMES

When trying to run this query it returns empty results everytime. I joined the two tables on t.university_name and c.university.



```
SELECT t.year,
t.university_name,
t.country,
t.world_rank     AS times_rank,
c.world_rank     AS cwur_rank,
t.total_score    AS times_score,
c.score          AS cwur_score
FROM times_hdfs t
JOIN cwur_hdfs c
ON LOWER(t.university_name) = LOWER(c.ins

WHERE t.year = 2015
ORDER BY t.total_score DESC
LIMIT 20;
```

Figure 10: Hive JOIN query returning empty results

Note that in the query I used a function which ignores capitalization differences, it still required the names to be perfectly identical. The core issue is that the TIMES and CWUR ranking systems use different naming conventions for some of the same institutions. For example, one of the list might use "University of California, Berkeley" while the other uses "UC Berkeley"

7 Conclusion

Throughout this assignment I was able to demonstrate a practical application of Apache Pig and Hive in processing and analyzing large datasets within the Hadoop ecosystem. I was also able to get hands-on experience with data cleaning procedures, such as using tools like Pig, this includes filtering incomplete records and preparing datasets for analysis. The comparison between Pig and Hive queries showed me that while both tools are able to achieve similar results, each have their own strengths and use cases.

The analysis of university rankings data provided valuable insights into global educational performance trends. In processing the data, it highlighted the importance of data standardization in big data projects. In short, I really do feel that after completing this assignment with all its troubles and headaches, it has really strengthened my understanding of distributed data processing frameworks and their real-world applications.