Associate Prof. Dr. Mehmet Aktaş BLM4120 Big Data Processing and Analytics

Deadline: 31.05.2023 9.00

Project Report

OĞUZHAN TOPALOĞLU
Ç19052025 – Group 1

Computer Engineering,
Faculty of Electrical-Electronics,
Yıldız Technical University

Istanbul, 2023

1. General Information

For my project, I created 2 Gradle projects called “gui-project” and “jar-project”. The GUI project only contains the commands and GUI code and the JAR project only contains the 5 Java classes that implements the map reduce programming model to calculate the min, max, average, std-dev of Python question scores. As well as determining all the Python questions that contains a specific string as a substring in their “Body” column.

Since I am not using a CSV reader, the functions I implemented calculates wrong results. This is because the database I chosed for this project was very complex and hard to parse using basic string methods or regexes. I tried adding Apache Commons CSV library to the project but Hadoop kept giving me weird bugs so I decided to keep it as it is.

The word counting function works even with the wrongly parsed dataset but the min/max/avg/stddev functions finds wrong values. I created a Dummy.csv file and filled it with 4 rows of random data. It has the same sequence and number of columns as the Questions.csv file. I used this Dummy.csv file to prove that my function do indeed work properly. If while testing and grading my project, anyone uses this Dummy.csv file they will see that all the map-reduce programming files do indeed work properly.

I have exported both of my projects as JARS and put them in a folder called “executables” along with many batch scripts I have written. These scripts contain the command-line commands that I use to execute Hadoop commands. These commands are also called from the GUI but since I am working on Windows I couldn’t get the GUI to work properly too. If it has access permissions, it will work (since the batch files and their contents are indeed executable from the GUI, just press the ”Open Hadoop” button and see it opening Hadoop).

2. Technical Challanges

Installing Hadoop on Windows was a true nightmare. I followed this tutorial to finally set it up correctly: https://medium.com/@pedro.a.hdez.a/hadoop-3-2-2-installation-guide-for-windows-10-454f5b5c22d3

I had issues with virtualization associated things in the past, since the version of Windows I am using does not allow my to virtualize any environment. I had issues with Docker in the past similar to this.

After I set Hadoop up, I ran the following command to upload my dataset(s): hadoop fs -put <full_path>\Questions.csv /input.

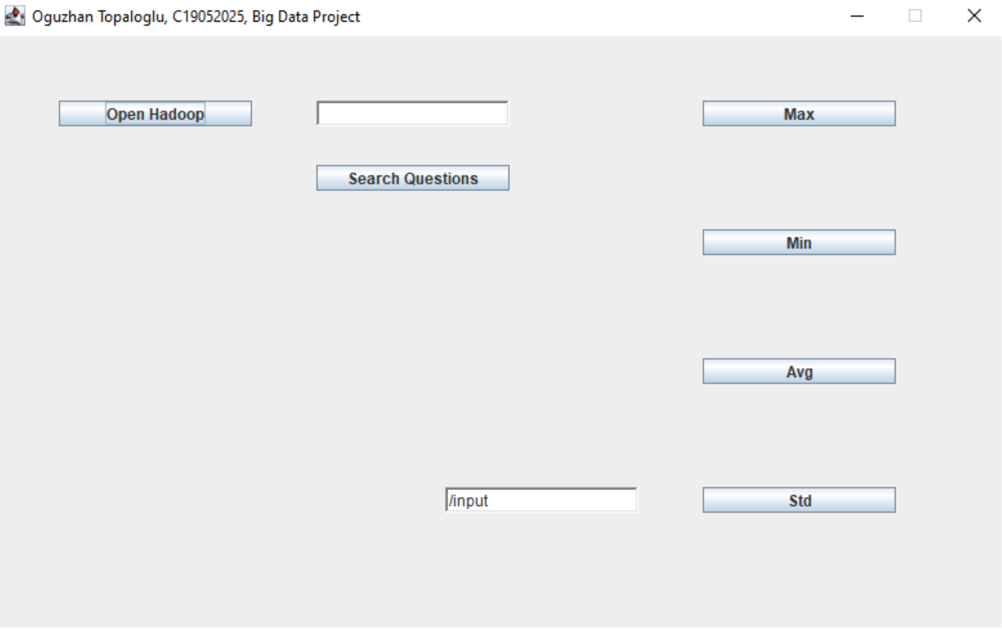
3. Explanation of functions

Contains.java file, contains a Java class that has a custom Mapper class with an overriden map(...) method. This file basically searches for substrings and maps them with “1” (found).

Max.java, Min.java, Avg.java and Std.java looks up all the Python questions and finds the respected statistical values.

4. GUI

The GUI can be seen in the figure below:



/input

Figure 1: The GUI of my application

5. Performance Testing

Instead of the following line in each jar-project file, we can remove System.exit() and add two System.currentTimeMillis() to measure the elapsed execution time.

```java
System.exit(job.waitForCompletion(true) ? 0 : 1);
```

I didn’t do anything like this since the GUI doesn’t have access permissions and the console is very messy to find a console print command. :)

But something like this can be done to measure it:

```java
long starting, ending;
starting = System.currentTimeMillis();
job.waitForCompletion(true);
ending = System.currentTimeMillis();
System.out.println("Elapsed time: " + (ending-starting)); // or show it in GUI.
```