Tutorial 3

Hadoop with neonatology dataset

Step 0: If you have not done so already: *claim a virtual machine from infomdssLabB*

Login to the azure portal (https://portal.azure.com/),

Go to **Dashboard → infomdssLabB → infomdssLabB** and claim a virtual machine.

Login to your lab B vm (the default password is the same as in tutorial 1) and **change your password**.

```
$ passwd # choose your new password wisely
```

These actions are explained in tutorial 1.

Step 1: Fetch the neonatology data

Within your Azure VM, go to the hadoop folder, create a new directory named `neonatology` and download `neonatology.zip` from google drive.

```
$ mkdir /usr/local/hadoop/neo && cd "$_" # shortcut for mkdir and cd
$ wget -O neonatology.zip \
https://github.com/Infomdss2018/infomdss/blob/master/tutorial_4/neonatology.zip?raw=true
```

***WARNING**: these data are **STRICTLY CONFIDENTIAL**. By downloading the data file, you agree to not redistribute the data, or the URL to the data, under any circumstances.*

Step 2: Extract the neonatology data

Note the neonatology data is a zip-file and needs to be extracted before you read the content. To do so, you first have to install `unzip` (a bash-program to unzip zip-files):cd .

```
$ sudo apt-get install unzip
```

After unzip is installed, you can run `unzip` (a password is required which can be found on the teams page of this course):

```
$ unzip neonatology.zip # you can find the password on teams
```

Finally you have to upload the data files to HDFS, the same way as you did in the wordcount tutorial.

```
$ [...] # upload the neonatology data to HDFS
```

After the data is extracted, you are ready to start with the assignments, where you have to write mappers and reducers yourself. As a starting point, you can download the `neo_mapper.py` and `neo_reducer.py` from github and edit them with `nano` (tutorial 1).

Note that `neo_mapper.py` and `neo_reducer.py` are both python scripts, however you can also use another scripting languages, for example `r`.

https://raw.githubusercontent.com/Infomdss2018/infomdss/master/tutorial_4/neo_mapper.py
https://raw.githubusercontent.com/Infomdss2018/infomdss/master/tutorial_4/neo_reducer.py

*Note: these scripts will **not** run out-of-the-box as the wordcount tutorial, they are intended to give you a starting point to implement a mapper and a reducer yourself.*

Assignments

Part I: Hadoop preprocessing

Create small scripts for each step below to preprocess the data files

1. Merge the two data files "`Opname-en-lab-2.csv`" and "`Opname-en-lab-3.csv`" within Hadoop into one integrated raw data file titled "`neo.csv`" (hereafter: "data file").
2. Clean the data file by removing any empty lines.
3. Clean the data file by removing identical rows.

Hints:

- You can use the `neo_mapper.py` and `neo_reducer.py` as a starting point.
- Running large datafiles will take a while, so for *testing and debugging*, you can create a sample dataset:

  ```
  $ shuf -n 100 neo.csv > neo_test.csv # select 100 random lines
  ```
- You can save a lot of time by debugging your scripts before running Hadoop:

  ```
  $ cat neo_test.csv | neo_mapper.py | sort | neo_reducer.py
  ```
- There are a lot of examples for map reduce scripts which you can use, for example https://stackoverflow.com/search?q=hadoop+mapreduce+example.

Part II: Map/Reduce processing

Create mappers and reducers to answer the following questions:

1. How many measurements (that is rows) are available for each patient in the data file?
   Note: that the patient id is denoted as `PseudoID_voorkeur`.
   The output might look something like:

```
Patient_ID\tNUM_OF_MEASURES
0\t<number of measures for patient 0>
1\t<number of measures for patient 1>
[...]
```

2. How many patients are in the data file?
3. What is the hospitalisation duration for each patient, given that '`Opnamedatum_OPN`' records the moment hospitalisation and '`Ontslagdatum_OPN`' denotes the discharge date?
   The output might look something like:

```
Patient_ID\tHOSP_DURATION
0\t<hospitalisation duration for patient 0>
1\t<hospitalisation duration for patient 1>
[...]
```

4. What is the average hospitalisation duration for a patient?
5. Which patients are hospitalized the longest? Make a top ten.

Hints:

- Store the map/reduce output result in a separate file for possible reuse as an input file for a subsequent map/reduce job.

Part III: MapReduce and/or Spark (Optional: for honor students)

There are so many variables left to explore, that we are sure that you, by using your creative imagination backed by your rudimentary domain knowledge obtained through the guest lecture, can uncover more interesting knowledge from the data file!

Therefore, in this third part of this assignment, we invite you to hypothesise and implement potentially interesting assumptions within the domain of neonatology. Try to think of several WHAT IF scenarios and then implement each WHAT IF scenario in a script to subsequently explore the data to try answer it. Here are some suggestions to get you started. Feel free to select a minimum of three questions from the list below or explore your own:

1. Building on Part II, are there differences in average hospitalisation times per ward (‘`AfdelingID_OPN`’)?
2. Building on Part II, does refining the hospitalisation duration calculations by including the time of day (‘`Opnamedatum_OPN`’), influence the findings in Part II?
3. What happens with the patients after discharge (‘`BestemmingID`’, ‘`BestemmingOmschrijving`’)? What percentage of patients went home (‘`BestemmingID=H`’)?
4. Is there a relation between discharge time and reason of discharge?
5. At what time of day are patients being hospitalised?
6. What is the quality of the lab results? Is it complete, standardised/usable, …
7. Do the lab results relate to hospitalisation duration?
8. Do seasonal influence, day of week, moon phases, or other factors play a role in admission frequencies or hospitalisation outcomes?
9. To what extent can we predict, based on these variables, whether a patient will or won’t pass away?

As with all Honors assignments, please submit your console output for the command above as a PDF file at http://bit.ly/infomdss-honors, preceded by a layman’s explanation of what happens in all the warning and information messages triggered by the command, and a brief interpretation of the final outcome.

Stop your virtual machine at the end of the workshop