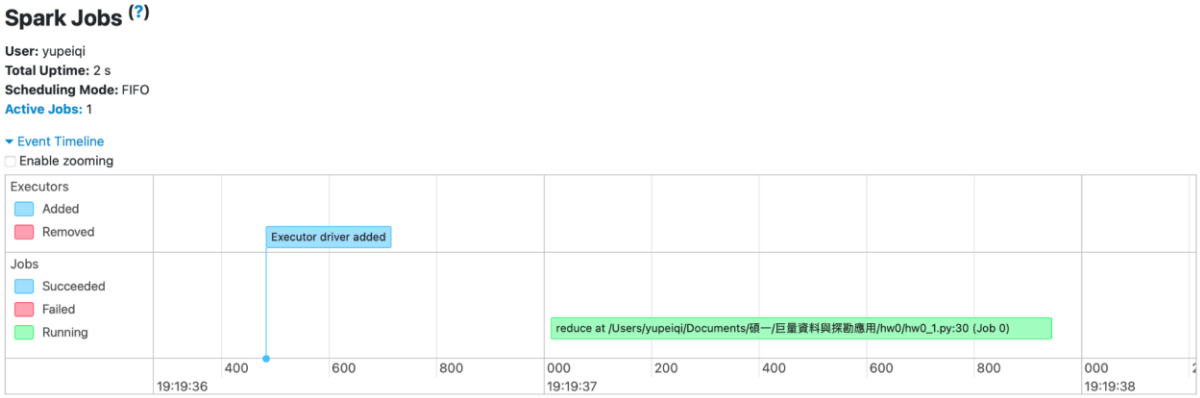
Big DataMining HW#0　112598028 余珮綺

- Programming language: Python on Spark
- A document showing your environment setup: PCs/VMs, platform spec, CPU cores, memory size, …

as for CPU cores, my computer has 8 cores



- Your source codes: All in "hw0" files, separate as hw0_1.py, hw0_2.py, hw0_3.py

The generated output (or snapshots):

First, I let the missing values become -1.0, which can see in my source code.

(30pt) (1) Output the minimum, maximum, and count of the following columns: 'global active power', 'global reactive power', 'voltage', and 'global intensity'

```
23/10/18 16:07:58 INFO DAGScheduler: Job 2 fi
hw0_1.py:32, took 0.493656 s
Minimum values: (0.076, 0.0, 223.2, 0.2)
Maximum values: (11.122, 1.39, 254.15, 48.4)
Count: 2049280
```

(30pt) (2) Output the mean and standard deviation of these columns.

```
w0_2.py:45, took 1.110310 s
Mean values: (1.0916150365006068, 0.12371447630387154, 240.83985797450583, 4.627759310587101)
```

```
hw0_2.py:62, took 1.143138 s
STD values: (1.05729390312673, 0.11272195204788779, 3.2399858884915984, 4.444395175406103)
```

(40pt) (3) Perform min-max normalization on the columns to generate normalized output.

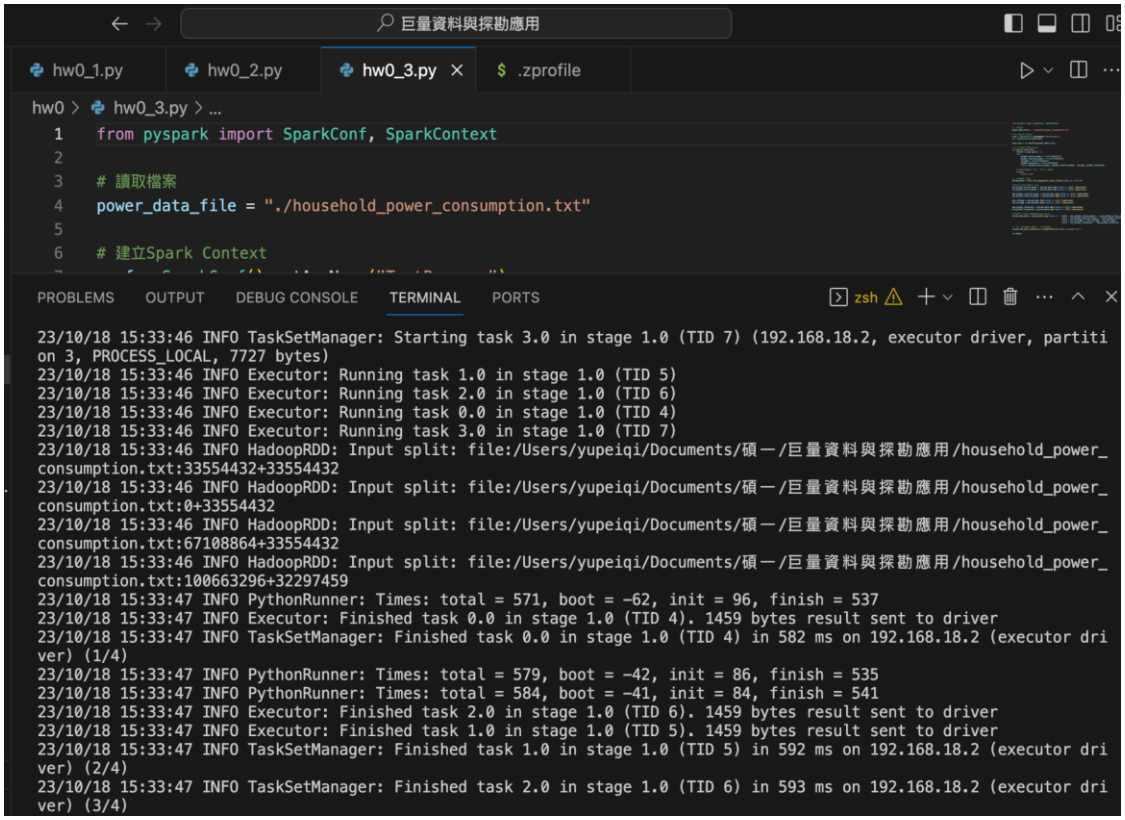
```
(0.3747963063552418, 0.30071942446043165, 0.376090468497577, 0.37759336099585067)
(0.4783632084012313, 0.31366906474820144, 0.33699515347334413, 0.47302904564315357)
(0.4796306355241717, 0.35827338129496406, 0.32600969305331173, 0.47302904564315357)
(0.48089806264711216, 0.3611510791366907, 0.3405492730210021, 0.47302904564315357)
(0.3250045265254391, 0.3798561151079137, 0.4032310177705981, 0.32365145228215775)
(0.311787072243346, 0.37553956834532376, 0.3819063004846531, 0.3070539419087137)
(0.3282636248415716, 0.3741007194244605, 0.3841680129240713, 0.32365145228215775)
(0.32808256382400874, 0.3741007194244605, 0.38836833602584825, 0.32365145228215775)
(0.32518558754300203, 0.36690647482014394, 0.3486268174474964, 0.32365145228215775)
(0.32464240449031323, 0.36690647482014394, 0.34442649434571954, 0.32365145228215775)
(0.39579938439254037, 0.35827338129496406, 0.31211631663974215, 0.40248962655601667)
(0.48307079485786714, 0.3381294964028777, 0.30953150242326355, 0.47717842323651455)
(0.46605105920695283, 0.34388489208633094, 0.3163166397415191, 0.46058091286307057)
(0.470034401593337, 0.2863309352517986, 0.3137318255250405, 0.4647302904564316)
(0.36013036393264536, 0.3035971223021583, 0.3890145395799681, 0.3609958506224067)
(0.29947492304906753, 0.2028776978417266, 0.4504038772213244, 0.29045643153526973)
(0.28915444504798116, 0.10935251798561152, 0.43715670436187376, 0.28215767634854777)
(0.30363932645301467, 0.11223021582733814, 0.4478190630048467, 0.29460580912863077)
```

Above is part of examples, complete outcome is in the files" path-to-output", has a part-00000.txt

```
23/10/18 16:14:22 INFO FileOutputCommitter: Saved output of task 'attempt_202310181614186297878755637286145_0014_m_
000000_0' to file:/Users/yupeiqi/Documents/碩一/巨量資料與探勘應用/path_to_output.txt/_temporary/0/task_20231018161
4186297878755637286145_0014_m_000000
23/10/18 16:14:22 INFO SparkHadoopMapRedUtil: attempt_202310181614186297878755637286145_0014_m_000000_0: Committed.
 Elapsed time: 0 ms.
23/10/18 16:14:22 INFO Executor: Finished task 0.0 in stage 8.0 (TID 32). 1749 bytes result sent to driver
23/10/18 16:14:22 INFO TaskSetManager: Finished task 0.0 in stage 8.0 (TID 32) in 4349 ms on 192.168.18.2 (executor
 driver) (1/1)
23/10/18 16:14:22 INFO TaskSchedulerImpl: Removed TaskSet 8.0, whose tasks have all completed, from pool
23/10/18 16:14:22 INFO DAGScheduler: ResultStage 8 (runJob at SparkHadoopWriter.scala:83) finished in 4.367 s
23/10/18 16:14:22 INFO DAGScheduler: Job 8 is finished. Cancelling potential speculative or zombie tasks for this j
ob
23/10/18 16:14:22 INFO TaskSchedulerImpl: Killing all running tasks in stage 8: Stage finished
23/10/18 16:14:22 INFO DAGScheduler: Job 8 finished: runJob at SparkHadoopWriter.scala:83, took 4.368192 s
```

time efficiency about min-max normalization.

- Documentation on how to compile, install, or configure the environment:



I use visual studio code to write down my source code, and use the terminal to run "spark-submit hw0/hw0_1.py", "spark-submit hw0/hw0_2.py", spark-submit hw0/hw0_3.py".

As for installing or configuring the environment, I tried the following steps to set up my environment.

安裝hadoop: https://www.youtube.com/watch?v=inDC9jgwpWY

安裝scala: https://www.scala-lang.org/download/

安裝spark: https://zhuanlan.zhihu.com/p/473313901