COMP6235 Coursework 2 – Data Management

Module	Foundations of Data Science			Lecturers	MS, ES
Assignment	Data Management			Weight	30%
Deadline	14/12/2018	Feedback	11/01/2019	Version	1.0

Description

This coursework will assess your understanding of using two classes of data management technologies: NoSQL databases and Big Data processing. In COMP6235, we are using MongoDB and Hadoop to illustrate these concepts. We recommend you revisit the lecture and tutorial notes to familiarise yourself with them before starting the coursework.

The coursework consists of three tasks:

- **Task 1: MongoDB**, using Enron email data.
- **Task 2: Hadoop**: using YouTube spam-comments data.
- **Task 3: a technical report**, in which you will discuss the advantages and disadvantages of different types of data management technologies, including centralized and distributed options, and present alternatives to the technologies used in the tutorials.

For tasks 1 and 2, we provide a **Jupyter notebook**. You must use it for your submission.

The total marks for the coursework add up to **100 points**, split across the three tasks as explained below. For each individual question, we enclosed the points available in square brackets.

Task 1: MongoDB (40 points)

This task will assess your ability to populate a MongoDB database, and query data from it.

Download the JSON version of the Enron dataset, available at http://edshare.soton.ac.uk/19548/. Import it into a collection called *messages* in a database called *enron*. You do not need to set up any authentication. Perform the following tasks in the notebook, using the Python library PyMongo. For each function, we have included answer templates in the notebook.

1. Write a function, which returns a MongoDB connection object to the *messages* collection. **[4 points]**
2. Write a function, which returns the total number of emails in the *messages* collection. **[4 points]**
3. Write a function, which returns each person who was **Bcc'**ed on an email.[1] Include each person only once. Display only their name, according to the **X-To** header. **[4 points]**
4. Write a function with parameter *subject*, which returns all emails in the email thread with that *subject*, and order them by date in ascending order.[2] **[4 points]**

1 If you are not familiar with BCC, check for instance: https://en.wikipedia.org/wiki/Blind_carbon_copy

2 *"An email thread is an email message that includes a running list of all the succeeding replies starting with the original email"*, according to Technopedia. Check further details at https://www.techopedia.com/definition/1503/email-thread

5. Write a function, which returns the percentage of emails that have been sent on a weekend (i.e., Saturday and Sunday). The output should be a float between 0 and 1 (i.e. 20% corresponds to 0.2). **[6 points]**
6. Write a function with parameter *limit*. The function should return for all email accounts: the number of emails sent; the number of emails received; and the total number of emails (sent and received). Use the following format: [{"contact": "michael.simmons@enron.com", "from": 42, "to": 92, "total": 134} and the information contained in the **To**, **From**, and **Cc** headers. Sort the output in descending order by the total number of emails. Use the parameter *limit* to specify the number of results to be returned. If *limit* is null, the function should return all results. If *limit* is higher than null, the function should return the number of results specified as *limit*. *limit* cannot take negative values. [**10 points]**
7. Write a function to find out the number of users who are sender and direct receivers. A sender is someone who has sent one email or more. A direct receiver is someone who has received at least one email as a recipient of the **To** header, and not via **Cc** or **Bcc**. **[4 points]**
8. Write a function with parameters *start_date* and *end_date,* which returns the number of email messages that have been sent between those dates, including *start_date* and *end_date* **[4 points]**

For all functions, optimise for speed. Answers that are less efficient will not get full marks.

Task 2: Hadoop (20 points)

This task will assess your ability to use the Hadoop Streaming API and MapReduce to process data. For each of the questions below, write two Python scripts, one for the Map phase and one for the Reduce phase. You should provide the correct parameters to the *hadoop* command to run the MapReduce process.

Just as in task 1, you will need to use the Jupyter notebook provided to write your answers.

First download and unzip the YouTube data available at http://edshare.soton.ac.uk/19547/) onto the machine where you have installed Hadoop. The archive consists of five datasets.

1. Using the Youtube01-Psy.csv dataset, write a function to find the hourly interval, in which most spam was sent. The output should be in the form of a **single key-value pair**, where the value is a datetime at the start of the hour with the highest number of spam comments. **[9 points]**
2. Using all five datasets, write a function with parameter *username* to find all comments associated with that user. Return a JSON array of all comments associated with the user. **[11 points]**

Task 3: Report (40 points)

Starting from your experience in the tutorials and in this coursework, write a **technical report** of no more than **four pages** covering the following aspects:

1. The difference between a relational and a NoSQL database. **[4 points]**
2. The pros and cons of using different NoSQL architectures, both in general, and with respect to the email data used in this coursework. **[6 points]**
3. The relational or non-relational database technology of your choice for Task 1, including, but not restricted to MongoDB, with clear rationales for your preference. **[6**

points]

4. The scenarios in which Hadoop is used and an overview of the steps of the Hadoop MapReduce process. **[4 points]**
5. The costs and benefits of parallel and distributed computation vs. centralised solutions in data science **[8 points]**

In addition to covering these aspects, in Task 3 we will also give up to **8 points** to account for the quality of the report (i.e. narrative, style, presentation, grammar); and up to **4 points** for good academic practice (i.e. good citations, Academic Integrity, see below).

This is a technical report, which should include your experiences and observations alongside insights from **high-quality external sources** (no blogs, Wikipedia or other online sources unless they are the main authority in their field). Literature references do not count towards the page limit.

You may also include a limited number of tables and figures, as long as they support the text. Both count towards the page limit.

The report must use the 2 017 ACM Master Article Template .

Academic Integrity

At the end of your report, you should include the following statement regarding academic integrity, and adhere to its spirit:

"I am aware of the requirements of good academic practice, and the potential penalties for any breaches."

This text snippet does not count towards the page limit.

Please check the Academic Integrity Regulations in Section IV of the University Calendar: http://www.calendar.soton.ac.uk/sectionIV/academic-integrity-regs.html for further details.

Submission

You will submit two files on HandIn:

- The completed Jupyter Notebook in .ipynb format. Please use this exact notebook, as it contains additional metadata, which helps us with the marking. To download it, open the Notebook in Jupyter and click File -> Download As -> Notebook (.ipynb)
- A PDF version of your report.

Standard ECS late submission penalties apply.