Apache Spark Assignment (LDSA)

Sections A B & C are the mandatory part, and give 2 points in total.
Sections D & E awards points towards higher grades, and give 1 point each. Only proceed to this if you have answered the questions from the earlier sections.

Work individually for all sections.

You will need to use the lecture material, Spark documentation and code examples, to answer the questions. Remember the handout is intended to help you to navigate the Spark documentation!

If you use resources other than the official Spark documentation and lecture material, you must cite your sources. Unless otherwise stated, you must use Spark for all the calculations (not e.g. Excel or Linux tools!).

Learning Outcomes

Learning Outcome	Section Assessed
Use Apache Spark appropriately for interactive analysis of large datasets, using a range of Map/Reduce operations (Lecture 1)	A, B
Deploy Apache Spark to the Cloud (Lecture 2)	C
Describe key components in the Spark architecture/execution model (Lectures 1, 2 & 3)	C, D, E
Apply knowledge of Spark internals to develop applications effectively, and avoid common performance pitfalls. (Lectures 2 & 3)	C, D
Create stream processing pipelines using Apache Spark (Lecture 4)	-
Relate key ideas in the implementation of Spark to concepts in the theory of distributed computing and big data. (Lectures 2 & 3)	D, E

Be able to monitor the performance of Spark Applications; and (using the documentation as a guide) tune configuration settings. (Lecture 4)	C

Submission Guidelines

You should submit a single document for all sections (PDF), with separate answers for each question clearly marked. Combine concise written answers (in English please), with code snippets (showing output, or a sample of output) as appropriate to each question.

You should also submit separately a complete copy of all your source code for all sections as a single zip file (or similar). Use a separate file or folder for each section, and identify with comments which code relates to which question.

Section A - Working with the RDD API

For this section, it is recommended to work with Spark in local mode, using Python 3, PySpark with a Jupyter notebook. You will need to setup Spark locally before you start (see lectures slides and links within).

Use the PySpark RDD API for this section.

We'll work with a parallel corpus of transcripts from the European Parliament
http://www.statmt.org/europarl/

These documents are small enough to process on your own machine, (so Spark is perhaps overkill), but we can use it to assess our understanding nonetheless.

Download and extract a parallel corpus for a language of your choice (use a utility from your operating system).

Question A.1

A.1.1 Read the English transcripts with Spark, (cache the resulting RDD), and count the number of lines.
A.1.2 Do the same with the other language (so that you have a separate lineage of RDDs for each).
A.1.3 Verify that the line counts are the same for the two languages.
A.1.4 Count the number of partitions.

Question A.2

A.2.1 Use Spark to pre-process the data according to the recommendations on the web page (“*To use the parallel corpora with tools like GIZA++, you want to:....*”)
(Note: **don’t** do the part about removing empty lines until Section A.4 below - the number of blank lines are different in the corpora)
A.2.2 Inspect 100 entries from your RDD to verify your pre-processing.
A.2.3 Verify that the line counts still match after the pre-processing.

Question A.3

A.3.1 Use Spark to compute the 10 most frequently according words in the English language corpus. Repeat for the other language.
A.3.2 Verify that your results are reasonable.

Question A.4

A.4.1 Use this parallel corpus to mine some translations in the form of word pairs, for the two languages. Do this by pairing words found on short lines with the same number of words respectively. (We incorrectly assume the words stay in the same order when translated)

Make a copy of your code from the previous sections. Here is one approach you might like to try:

- Record the line numbers associated with each line (hint: ZipWithIndex())
- Swap the key and value - so that the line number is the key
- Match the lines in each corpus, so you have pairs of matching lines. (hint: join())
- Pre-process the lines, to split the words as before, but don’t flatten yet.
- Filter to exclude line pairs that have an empty/missing “corresponding” sentence.
- Filter to leave only pairs of sentences with a small number of words per sentence (this should give a more reliable translation (you can experiment).
- Filter to leave only pairs of sentences with the same number of words in each sentence.
- For each sentence pair, map so that you a pair each (in order) word in the two sentences (we no longer need the line numbers). (hint: use python’s built in zip() function)
- Use reduce to count the number of occurrences of the word-translation-pairs.
- Take some of the most frequently occurring pairs of words.

A.4.2 Do your translations seem reasonable? Use a dictionary to check a few (don’t worry, you won’t be marked down for incorrect translations!).

Section B - Working with DataFrames and SQL

For this section, again, work with Spark in local mode, and use Python 3, PySpark with a Jupyter notebook. We can do this because the dataset is small.

Use the PySpark DataFrames/SQL API for this section (as always, do not use Excel!). It's recommended to do Section A first.

We use the 2017-2018 gender equality statistics from UK organizations:
https://gender-pay-gap.service.gov.uk/viewing/download

Detailed explanation of the statistics can be found here:
http://www.acas.org.uk/media/pdf/m/4/Managing_gender_pay_reporting_04_12_17.pdf

Question B.1 - Analysis with DataFrames / SQL

B.1.1 Which organization has the largest gender pay gap? Which the least?

B.1.2 What is the mean gender pay gap across all organization?

B.1.3 Export the results of B.1.2 to a CSV file. Inspect the output file to check it looks reasonable.

B.1.4 What proportion of organizations pay women more than men on average?
Explain your calculation.

Question B.2 - Advanced DataFrames / SQL

B.2.1 Create a new column for the industry sector (for each company) using the SIC code:

The UK SIC 2007 sector group ranges (the first two digits) are listed in the appendix.
Where a company has multiple SIC codes, use the first one.
Ignore SIC codes of '1'.

Use a broadcast variable to represent the SIC code mapping.

B.2.2 Compute the mean gender pay gap per sector.
Inspect a sample - and check your calculations look reasonable?

B.2.3 How does gender pay equality compare per sector? Compute some additional statistics. Discuss briefly.

Section C - Spark Clusters and Deployment

In this section, we deploy a Spark cluster in the SNIC cloud, run our application again, and inspect the web GUIs to understand how our application is being run.

For written answers, use full sentences, (1 or a few) per question, as appropriate. Use the handout to help you. Marks will be awarded based on the correct use of terminology for Spark and distributed computing concepts.

For relevant questions, your submission should Include the commands you used in the Linux shell for the deployment, with short explanation of each (1 sentence max).
Omit any steps included in the Spark Setup guide for sections A and B.

C.1 Deploy a Spark cluster consisting of a master node and one or more worker nodes.
Use standalone deployment mode. Deploy Jupyter on the master node.

Additional Tasks (don't submit anything for them):

- Run a basic example (e.g. the very first lecture example) to check that your cluster works.
- Ensure you can access the Web GUI for the master node, and your application. You will either need to configure public IPs and the firewall, or SSH port forwarding.
- Get the dataset from question 1 onto all the nodes.

C.3 Modify a copy of your code from Section A, so that it runs on your cluster. Verify this. (You shouldn't submit the code again, just the lines that you changes with a short explanation).

C.4 Run your code first without and then with .cache() - and look under the storage tab in the web GUI for your application. What do you notice? Explain briefly what's going on.
(If you don't notice a difference, try restarting your application)

C.5 Use the Web GUI to explore your cluster and examine jobs, stages, and tasks.
Create an example that requires a job with more than one stage. Explain, with reference to the Spark API methods you invoke in your code, why this is so.

Section D - Concepts in Apache Spark and Distributed Computing

Please answer in full sentence(s) (max 3), per question, as appropriate. Use the handout to help you. Marks will be awarded based on the correct use of terminology for Spark and distributed computing concepts. You do not need to submit any code for these questions, but you might want to experiment with your code to help you answer them.

D.1 Why do we use MapReduce?

D.2 Are partitions mutable? Why is this advantageous?

D.3 Where is lazy evaluation used in Spark? Why? Explain briefly how it works.

D.4 Why might I use Spark instead of Hadoop?

D.5 *“in Spark, the driver node needs enough memory to load my entire dataset”*
Is this statement true or false? Explain.

D.6 Give an example of how RDDs are ‘resilient’? How is this achieved?

D.8 Can I use, say, Python 2 for my driver application, and Python 3 for my executors? Explain briefly why/why not?

Section E - Essay Questions

“A colleague has mentioned her Spark application has poor performance, what is your advice?”

Give 5 clear recommendations, answer in full sentences.

Marks will be awarded based on the correct use of Spark terminology. Do not submit code for this question (but you might want to mention API methods, for example).

Appendix: UK 2007 SIC Codes

```
,Sector,Min,Max
A,AGRICULTURE FORESTRY AND FISHING,1,3
B,MINING AND QUARRYING,5,9
C,MANUFACTURING,10,33
D,ELECTRICITY GAS STEAM AND AIR CONDITIONING SUPPLY,35,35
E,WATER SUPPLY SEWERAGE WASTE MANAGEMENT AND REMEDIATION
ACTIVITIES,36,39
F,CONSTRUCTION,41,43
G,WHOLESALE AND RETAIL TRADE REPAIR OF MOTOR VEHICLES AND
MOTORCYCLES,45,47
H,TRANSPORTATION AND STORAGE,49,53
I,ACCOMMODATION AND FOOD SERVICE ACTIVITIES,55,56
J,INFORMATION AND COMMUNICATION,58,63
```

```
K,FINANCIAL AND INSURANCE ACTIVITIES,64,66
L,REAL ESTATE ACTIVITIES,68,68
M,PROFESSIONAL SCIENTIFIC AND TECHNICAL ACTIVITIES,69,75
N,ADMINISTRATIVE AND SUPPORT SERVICE ACTIVITIES,77,82
O,PUBLIC ADMINISTRATION AND DEFENCE COMPULSORY SOCIAL SECURITY,84,84
P,EDUCATION,85,85
Q,HUMAN HEALTH AND SOCIAL WORK ACTIVITIES,86,88
R,ARTS ENTERTAINMENT AND RECREATION,90,93
S,OTHER SERVICE ACTIVITIES,94,96
T,ACTIVITIES OF HOUSEHOLDS AS EMPLOYERS UNDIFFERENTIATED GOODS-AND
SERVICES-PRODUCING ACTIVITIES OF HOUSEHOLDS FOR OWN USE,97,98
U,ACTIVITIES OF EXTRATERRITORIAL ORGANISATIONS AND BODIES,99,99
```

source: https://www.ons.gov.uk/methodology/classificationsandstandards/ukstandardindustrialclassificationofeconomicactivities/uksic2007