SIT742 Modern Data Science 2018 T1
Assignment 1 (Three Tasks)

Prepared by Guangyan Huang

Assignment 1: General data processing and using big data (15% marks)

This is an individual work on the understanding of data science, big data and their applications. It contains written answers and some programming-related tasks based on topics presented during Weeks 1 to 3. Assignment 1 is broken down to three tasks below. You can use Google to find the data sources (i.e., websites). After your practice, please write down your executable Python codes and put the collected data in tables for above demonstration. You also need to write several paragraphs to explain your comparison and make a conclusion.

Due: 11:59pm, Saturday, ~~14 April 2018~~ 28 April 2018 (Postpone two weeks due to many late enrolments)

Task 1. Data Acquisition (7 marks)

Design a web scraping program by Python to collect weather forecast report data of a city (e.g., Melbourne) from a website, such as temperature, humidity, weather status (cloudy, sunny etc.), and store the data in a csv file. Please do this task in both the following ways:

(1) Collecting data by regular interval sampling. You need to find the best sampling interval in terms of space efficiency and demonstrate using numeric results why it is the best solution.

(2) Collecting data by change detection. You store one data object only when any of the weather forecast report data is changed at the website.

Both you need to record weather data with their timestamps. Then, compare the two collection methods, conclude the optimal one and demonstrate using numeric results.

(Please refer to Lecture 2.)

Task 2. Data Integration (5 marks)

Use the optimal method you demonstrated above to collect weather report data from more than one websites and integrate the data from different sources (websites) and write the integrated data into a csv file. Please demonstrate

(1) how to do schema alignment and

(2) how to determine which is correct if two data from different sources do not agree with each other.

(Please do a survey about the existing techniques and use one to resolve the problem, Lecture 2 provides you some basic concepts and you may do a broader search by yourselves.)

Task 3. Missing Data Prediction (3 marks)

Use the data you collected in Tasks 1 and 2, please design a method to predict a missing data object, for example, between two consecutive data objects (time, temperature) in your csv file as below:

11:00AM, 15

12:00PM, 17

the user want to query about the temperature at *11:30AM*.

(Please do a survey about the existing techniques and use one to resolve the problem. Lecture 3 provides you some basic concepts and you may do a broader search by yourselves.)

Assignment 1 Marking Criteria

Assignment 1 (15%=15 Marks)			
Criteria	**Excellent**	**Good**	**Not Shown**
Task 1: Data Acquisition (7 marks)			
Collect data by regular interval sampling	Provide python codes that automatically collect data from static web pages (such as html), use a reasonable interval (e.g., sampling 5 times a day). At least three attributes (e.g., temperature, weather status and humidity) are used and at least ten records are collected (3 marks)	Provide python codes that automatically collect data from static web pages (such as html), but cannot continuously collect data. Less than three attributes are used and less than ten records are collected (1-2 marks)	0 mark
Collect data by change detection	Provide python codes that automatically collect data from static web pages (such as html) and develop a change detection scheme. At least three attributes (e.g., temperature, weather status, humidity) and ten records (2 marks)	Provide python codes that automatically collect data from static web pages (such as html), but no change detection scheme is used. Less than three attributes and less than ten records (1 mark)	0 mark
Compare the two collection methods and conclude the optimal one	Demonstrate using automatically generated numeric results by python codes (2 marks)	Demonstrate using manually collected numeric results (1 mark)	0 mark
Task 2: Data Integration (5 marks)			
Collect weather report data from multiple sources	Select high quality data sources/websites (at least two). For each data source, at least three attributes and at least ten records are collected (3 marks)	Collect data from at least two data sources/websites without quality selection, and less than three attributes and less than ten records are collected from each data source (1-2 marks)	0 mark
Do schema alignment	Manually get the schema alignment rules for data from two sources and write the rules into python codes for continuous update of data (1 mark)	Manually get the schema alignment rules for data from two sources and no automatic and continuous update of data (0.5 mark)	0 mark
Do data fusion	Compare data records from different data sources (records' attributes are aligned) and develop a simple fusion scheme. (1 mark)	Compare data records from different data sources (records' attributes are aligned) and use mean to fuse data. (0.5 mark)	0 mark
Task 3: Missing Data Prediction (3 marks)			
Develop a data prediction method	Predict the missing data point using its consecutive neighbors or the whole trends or the value in the similar time bin. (3 marks)	Use simple techniques, such mean and median (1-2 marks)	0 mark

A Ladder for Assignment 1

Approach Instruction for Task 1.

Step 1: Google a city's weather data. From the Google hit/result list, manually select several (e.g., 5) web pages, which comprise the weather data in html pages.

Step 2: Learn "Web Scraping Codes" from Slides 20-21 of Lecture 2. You can observe source codes from https://github.com/hmcuesta/PDA_Book/tree/master/Chapter2

Step 3: Understand two concepts: "regular interval sampling" and "discrete sampling based on change" from Slides 15-17 of Lecture 2. Design both a regular interval sampling scheme and a change detection sampling scheme. Then implement your two schemes by modifying the example web scraping codes learnt in Step 2. Collect data from **only one website** (at least ten records using at least three weather attributes) for each scheme.

Step 4: Put the data collected by the two schemes into a table and manually compare/analyse the results difference and discuss which one is optimal in terms of space efficiency and collecting efficiency.

Approach Instructions for Task 2.

Step 1: Repeat Task 1 but only using the optimal sampling scheme by considering at least one more website.

Step 2: Manually compare the attributes from different data sources/websites (at least two) and pair two attributes if they are the same meaning (one such pair we call a schema alignment rule). Write the schema alignment rules into python codes for continuously collecting updated data for at least ten records.

Step 3: Put (at least) ten data records into a column for each data source, so you get two columns and (at least) ten rows. Observe and compare the data at each row and think out a scheme to fuse them into one result that you can put at the third column. You may consider some optimisation in your scheme to ensure higher accuracy and higher confidence of your final fused results.

Approach Instructions for Task 3.

Step 1. Observe your data in the table at Step 3 of Task 2. Consider methods in Slide 12 of Lecture 3. You can assume a scenario, if a user query a weather data at a time that is not in your table. You design a scheme to answer this kind of user query. For example, in your table, you may have

8:00AM, 13℃
9:00AM, 13.5℃
10:00AM, 14℃
11:00AM, 15℃
12:00PM, 16℃
13:00PM, 17℃
14:00PM, 25℃
15:00PM, 22℃
16:00PM, 23℃
17:00PM, 20℃

but the user want to query about the temperature at *11:30AM*.

Possible scheme 1. Based on consecutive neighbors
Possible scheme 2. Based on whole trends
Possible scheme 3. Based on the value in the similar time bin in another day.
And other possible schemes developed by yourself. (Note: try your best model the data trends and build prediction models by some survey and we understand you may not learn data analytics methods at this stage.)

Submission: You must submit an electronic copy of your assignment either in Acrobat (.pdf) or Microsoft Word (.doc) via CloudDeakin.
Delays caused by student's own computer downtime cannot be accepted as a valid reason for late submission without penalty. Students must plan their work to allow for both scheduled and unscheduled downtime.
It is the student's responsibility to ensure that they understand the submission instructions. If you have ANY difficulties ask the Lecturer for assistance (prior to the submission date).

Copying, Plagiarism:
This is an individual assignment. You are not permitted to work as a part of a group when writing this assignment.
Plagiarism is the submission of somebody else's work in a manner that gives the impression that the work is your own. For individual assignments, plagiarism includes the case where two or more students work collaboratively on the assignment. The School of Information Technology treats plagiarism very seriously. When it is detected, penalties are strictly imposed. Deakin University uses *Turnitin* as the program that allows you to check whether there is any unoriginal material in your work, please refer to
http://www.deakin.edu.au/students/clouddeakin/help-guides/assessment/plagiarism.

Additional Requirements and Notes

1. Any text, table, figure, and code adapted from any source must be clearly referenced.
2. All assignments must be submitted through CloudDeakin. Assignments will not be accepted through any other manner without prior approval. Students should note that this means that email and paper based submissions will ordinarily be rejected.
3. Submissions received after the due date are penalised at a rate of 10% (out of the full mark) per day, no exceptions. Late submission after 3 days would be penalised at a rate of 100% out of the full mark. Close of submissions on the due date and each day thereafter for penalties will occur at 11:59 pm **Australian Eastern Time (UTC +10 hours) with Daylight Saving**.
4. No extension will be granted.