Students with difficulty in meeting the deadline because of illness, etc. must apply for an assignment extension (up to 3 days) no later than 12:00pm on 16/04/2021 (Friday).

Instructions

Six files are provided for this assessment task:

`HTWebLog_p1.zip` The compressed `zip` file is for Part I of this assessment task, and it is a sample of *Hotel TULIP* Web log dataset, which contains the web access log information from 11/2006 to 02/2007. [1].

`Professor-list.csv` This `CSV` file is for Part II of this assessment task, and it contains three columns: the professor *name*, the professor *title* and also the *university*.

`Professor-citation-information.csv` This `CSV` file is for Part II of this assessment task, and it has 8 columns: the professor *name*, the professor *title*, th e`citation-all`, the `citation-since2016` (citations after 2016), the `h-index-all` [2], the `h-index-since2016`, the `i10-index-all` [3] and also the `i10-index-since2016`.

`SIT742Task1.ipynb` This is the notebook file for the `Python` code in `ipynb`, and the latest notebook is also released in `SIT742Task1.ipynb`.

Web log This code snippet contains all the coding requirements and also hints for Part I of this assessment task.

Web crawling This code snippet contains all the coding requirements and also hints is for Part II of this assessment task.

You will need to complete the code in the notebook and make it run-able. The results on running the notebook will help you to develop your report, as well as generate the required files: `Professor-list.csv` and `Professor-citation-information.csv`.

`SIT742Task1-DataDictionary-Template.xlsx` This is the `Excel` template file for the *data dictionary*, and it is for Part I of this assessment task.

`SIT742Task1-Report-Template.docx` This is the `Word` template for your report `SIT742Task1-Report.pdf`.

What to Submit?

You are required to submit the following completed files to the corresponding *Assignment* (Dropbox) in CloudDeakin:

`SIT742-DataDictionary.xlsx` The data dictionary for the *Hotel TULIP* Web log dataset.

`Professor-list.csv` The csv file of all professors in Deakin University School of IT.

`Professor-citation-information.csv` The csv file of all citation information on professors.

`SIT742Task1.ipynb` The completed notebook with all the run-able code on all requirements.

`SIT742Report.pdf` Your report for the both Part I and Part II of this assessment task.

[1] This file is exclusively for `SIT742` educational purpose only. You are not allowed to further distribute it.

[2] h-index is the largest number h such that `h` publications have at least `h` citations. The second column has the "recent" version of this metric which is the largest number `h` such that `h` publications have at least `h` new citations in the last 5 years.

[3] i10-index is the number of publications with at least 10 citations. The second column has the "recent" version of this metric, which is the number of publications wit at least 10 new citations in the last 5 years.

Part I
Data Manipulation — *Web Log Data*

Here is the hypothetical background:

> *Hotel TULIP* (a hypothetical organisation) is a five star hotel that locates in Australia. It is a very special hotel with an equally special purpose: Not only does it embody all the creative energy and spirit of TULIP-Lab, it's a "learning environment" on which the tourism and hospitality students are trained for future hoteliers.
>
> In the past two decades, the Web server of *Hotel TULIP* has logged all the web traffic to the hotel website, and stored large amount of data related to the use of various web pages. The hotel's CIO, Dr *Bear Guts* (not *Bill Gates*!), believes that those log files are great resources to help their *Information Technology Division* improve their potential customers' online experience, and help their *Market Promotion Division* to identify potential customers and their behaviour patterns. Hence, *Hotel TULIP* would like to outsource the web usage mining task to *Group-SIT742* (a hypothetical data analytics group with up to 3 data analysers) to analyse web log files and discover user accessing patterns of different web pages.
>
> The Web server is using *Microsoft Internet Information Service* (IIS), and the Web log format can be found at: `https://msdn.microsoft.com/en-us/library/ms525807(v=vs.90).aspx`

You are employed within *Hotel TULIP* working in the *Information Technology Division*. Your manager, Dr *Beer Guts* (also not *Bill Gates*!), has asked you to prepare a set of documents for *Group-SIT742* so that they can have an initial understanding of the data to be analysed.

Task Description

This task requires you to construct a *data dictionary* and develop a data exploration report for the provided *Hotel TULIP* Web log dataset.

Without exploration or further analysis, 'raw' Web log data hardly reveals any insightful information. In this part, you are required to complete the `Python` code snippets to generate suitable numeric and visual description in the *Hotel TULIP* Web log dataset based on the detailed requirements in `SIT742Task1.ipynb`, and develop the report `SIT742Task1Report.pdf` to summarise the descriptive statistics information. The detailed requirements can also be found in the notebook `SIT742Task1.ipynb`, here we summarise them as follows:

1 ETL

1.1 Data Loading (4 marks)

Complete the `Python` code snippets in `SIT742Task1.ipynb` as required in notebook, and complete the data dictionary and report.

Code Load (may need unzip first) the *Hotel TULIP* Web log data `HTWebLog_p1.zip` into `dataframe` `df_ht`, and check how many files are loaded. Then check data statistics and general information by printing its top 5 rows.

Data Dictionary Fill the *data dictionary* based on the Python code results.

> For a data scientist or business analyst, after obtaining the dataset, the first crucial task is to obtain a good understanding of the data to be analysed. This includes: *examining* the data attributes (or equivalently, data fields), *seeing* what they look like, what is the data type for each field, and from this information, *determining* suitable numerical/visual descriptions.

A systematic approach to this process, as we have learned from the lectures (Week-03), is to construct a *data dictionary* for the dataset. You are required to construct a *data dictionary* for the *Hotel TULIP* Web log dataset using the template: `SIT742Task1-DataDictionary-Template.xlsx`.

SIT742Task1Report Add proper results for Section *Dataset Description* and *Attribute Dictionary*.

1.2 Data Cleaning (2 marks)

Complete the `Python` code snippets in `SIT742Task1.ipynb` as required in notebook, and complete the data dictionary and report.

Code
- Check which columns have `NA`s,
- For each of those columns, display how many records with `NA` values
- Remove all records with any `NA`s.

SIT742Task1Report Add proper results for:

- the number `NA`s for each column.
- the number of rows before removing `NA`s.
- the number of rows after removing `NA`s.

2 Descriptive Statistics

2.1 Traffic Analysis (4 marks)

Analyse the web traffic statistics;

Code
- Discover on the traffics by analysing hourly requests.
- Plot into Bar Chart.
- Filter the hourly requests by removing any below 490,000 and above 400,000. (`hourly_request_amount >= 400000 & hourly_request_amount <= 490000`)

Report
- Please add a figure of Hourly Requests Bar Chart from your Notebook, and elaborate the findings from the figure.
- Please add a table of filter result (`hourly_request_amount >= 400000 & hourly_request_amount <= 490000`)

2.2 Server Analysis (4 marks)

Analyse the server status statistics;

Code Discover on the server status using '`sc-status`' from `DataFrame`, then plot it into Pie Chart.

Report
- How many types of status reported?
- Figure '`Server Status`' in Pie Chart.

2.3 Geographic Analysis (4 marks)

Analyse the server Geographic information statistics;

Code
- Select all requests at `01 Jan 2007` from `20:00:00 pm` to `20:59:59 pm`.
- Discover the geographic information by analysing requests from country and city level.
- Plot countries and cities of all requests in two pie charts.
- List top 3 of both with the request numbers.

Report
- How many requests raised in the period of time?
- How many countries and cities are involved?
- Figure '`Request by Country`' and '`Request by City`' in pie charts.
- List Top 3 countries and cites with the request numbers.

Part II
Data Manipulation — Web Crawling

Google Scholar is a web service that indexes the metadata of research articles on many scientists. Majority of computer scientists choose to use *Google scholar* to track their publications and research development. Therefore, the web crawling on *Google Scholar* can provide the citation information on all professors with a public *Google Scholar* profile.

Task Description

In 2021, to better introduce all the emeritus professors, professors and associate professors in the school of IT, Deakin university wants to collect all the citation information on them. You are required to implement a web crawler, design and complete the code in the notebook and make sure that the web crawling code meets the requirements. You are free to use any `Python` package for Web crawling.

3 Professor list generation

You will need to import the suitable (or your chosen)web crawling library and use the corresponding library to crawl the School of IT staff list page: `https://www.deakin.edu.au/information-technology/staff-listing`.

3.1 Import and install your web crawling library (1 mark)

You could use `selenium` by doing the pip install selenium, download the `webdriver` for chromedriver and define your webdriver for crawling. But you are free to use any other library.

3.2 Crawl and Generate the list (1 mark)

The code must contain the necessary web crawling steps and necessary data save steps. The results of the code running will generate the `Professor-list.csv`. Without using the web crawling steps in the code will incur 0 mark.

4 Professor Citation Information generation

4.1 Professor citation information generation (2 marks)

You will need to use the generated `Professor-list.csv` to identify each professor's google scholar profile page in google scholar platform, and then to crawl the citation information from each google scholar profile. You will need to design your code by using loops and condition statement (as some of the professors did not have google scholar profile) to complete this requirement. The results of code running will generate the `Professor-citation-information.csv`.

4.2 Identify the professor with the most citations (1 mark)

You are required to do the sort and print by using `pandas` function to find out the professor with the most citations (please remove those without a public google scholar page).

4.3 Identify the associate professor with the most i10-index since 2016 (1 mark)

You are required to do the filer, sort and print by using `pandas` function to find out the *associate professor* with the most i10-index since 2016 (please remove those without a public google scholar page).

4.4 Identify those with the `citations-since2016` > 2500 (1 mark)

You are required to do the conditional filter and print to find out those (*professors*, *associate professors*) with the `citations-since2016` > 2500 (please remove those without a public google scholar page).

Note

You will need to complete the notebook and insert the related self-written code and required results into the corresponding place of the report `SIT742Task1-Report.pdf`.