FIT9136 Algorithms and Programming Foundations in Python

Assignment 3

OCT 2022

Table of Contents

1. Key Information

Purpose	This assignment will develop your skills in designing, constructing, testing, and documenting a Python program according to specific programming standards. This assessment is related to the following learning outcome (LO): ● **LO2 -** Restructure a computational program into manageable units of modules and classes using the object-oriented methodology ● **LO3 -** Demonstrate Input/Output strategies in a Python application and apply appropriate testing and exception handling techniques ● **LO4 -** Investigate useful Python packages for scientific computing and data analysis; ● **LO5 -** Experiment with data manipulation, analysis, and visualisation technique to formulate business insight.
Your task	This assignment is a individual task where you will write Python code for a simple application whereby you will be developing an analyser application as per the specifications.
Value	**35**% of your total marks for the unit.
Due Date	**Friday, 4 November 2022, 4:30 PM (AEST)**
Submission	● Via Moodle Assignment Submission. ● FIT GitLab check-ins will be used to assess the history of development ● MOSS will be used for similarity checking of all submissions.
Assessment Criteria	The following aspects will be assessed: 1. Program functionality in accordance to the requirements 2. Code Architecture and Adherence to Python coding standards 3. The comprehensiveness of documented code
Late Penalties	● 10% deduction per calendar day or part thereof for up to one week ● Submissions more than 7 calendar days after the due date will receive a mark of zero (0) and no assessment feedback will be provided.
Support Resources	See Moodle Assessment page and Section 7 in this document.
Feedback	Feedback will be provided via one formats: ● specific student feedback ten working days post submission

2. The Assignment

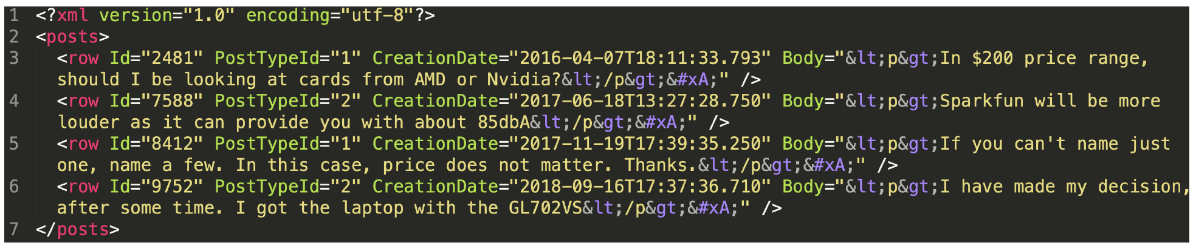
In this assignment, you will implement a basic parser to investigate the natural-language posts from Q&A (Question and Answering) site. The parser is able to perform basic data extraction, statistical analysis on a number of linguistic features and also to present the analysis results using some form of visualisation.

2.1. The Dataset: HardwareRecs

Before you get started with any of the programming tasks, you should read through the description of the dataset that we will be using for the purpose of this assignment.

The dataset is known as HardwareRecs [https://hardwarerecs.stackexchange.com] which is a Q&A site for people seeking specific hardware recommendations. The Q&A site is a platform for users to exchange knowledge by asking and answering questions such as Quora, Zhihu, and Stack Overflow. Within HardwareRecs, users can ask questions about hardware recommendations, while other users can also answer those questions with corresponding suggestions.

The data is written in XML (Extensible Markup Language) format. Apart from the first two lines and the last line which are XML specific format, each line in the dataset represents a record of a post in the Q&A site, i.e., the row beginning with "<row " while ending with "/>" is a piece of date in this assignment.



```xml
<?xml version="1.0" encoding="utf-8"?>
<posts>
  <row Id="2481" PostTypeId="1" CreationDate="2016-04-07T18:11:33.793" Body="&lt;p&gt;In $200 price range,
  should I be looking at cards from AMD or Nvidia?&lt;/p&gt;&#xA;" />
  <row Id="7588" PostTypeId="2" CreationDate="2017-06-18T13:27:28.750" Body="&lt;p&gt;Sparkfun will be more
  louder as it can provide you with about 85dbA&lt;/p&gt;&#xA;" />
  <row Id="8412" PostTypeId="1" CreationDate="2017-11-19T17:39:35.250" Body="&lt;p&gt;If you can't name just
  one, name a few. In this case, price does not matter. Thanks.&lt;/p&gt;&#xA;" />
  <row Id="9752" PostTypeId="2" CreationDate="2018-09-16T17:37:36.710" Body="&lt;p&gt;I have made my decision,
  after some time. I got the laptop with the GL702VS&lt;/p&gt;&#xA;" />
</posts>
```

As seen in above image, each post contains four attributes:

- **Id**: the unique identifier to represent each post
- **PostTypeId**: the type of the post:
  - 1 = Question
  - 2 = Answer
  - 3 to 8 = Others
- **CreationDate**: the creation date and time of the post (format as yyyy-mm-ddThh:mm:ss)

- **Body**: the content of the post

You should note that there are many different "PostTypeId" recorded in the dataset. However, for the purpose of this assignment, the data required for processing and analysis are the questions and answers in the site, which are those rows indicated by the "PostTypeId" as 1 or 2.

Note: You should download the dataset from the FIT9136 S2 2022 Moodle site before attempting the following tasks. The dataset is named data.xml.

2.2. Task 1: Handling with File Contents and Preprocessing

In the first task, you will begin by reading in all the posts of the given dataset. You will then conduct a number of pre-processing tasks to clean the post content (Body) needed for analysis in the subsequent tasks (Task 2 and 3) in this assignment. Upon completing the pre-processing tasks, the content of questions and answers should be saved as two individual output files. This would be a more efficient approach whenever we need to manipulate the cleaned dataset without having to repeat the pre-processing task, especially for large-scale data analysis.

For each post, you should first extract the content/body of it i.e., the string embedded within "Body:"..."" in each row of the XML file. Then you need to carry out some preprocessing steps to it as follows:

A. In HTML, XML documents, the logical constructs known as character data and attribute values consist of sequences of characters, in which each character can manifest directly (representing itself), or can be represented by a series of characters called a character reference. We need to convert those special character references back to its original form by following the rules in Table 1.

Table 1: Character Reference Transformation.

Character reference	Original form
&amp;	&
&quot;	"
&apos;	'
&gt;	>
&lt;	<

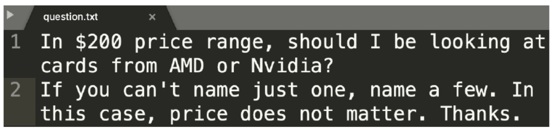
Example:

<u>**Before filtering:**</u> &lt;p&gt;In $200 price range, should I be looking at cards from AMD or Nvidia?&lt;/p&gt;&#xA;

<u>**After filtering:**</u> <p>In $200 price range, should I be looking at cards from AMD or Nvidia?</p>

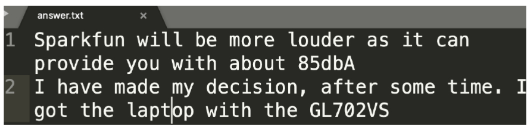
B. Replace special characters including “&#xA;”, “&#xD;” by a single empty space.

C. Remove all HTML tags. All data within the body attribute are content of posts in HardwareRecs site, and it is rendered by HTML (Hypertext Markup Language) format. Within HTML, there are many tags to annotate the content such as <p>, <a>. All tags contain start tags like <p>, and end tags like </p>. All of these tags are written as the format “<*>”, and some tags even have detailed attributes like ‘<a href=”www.google.com”>’. You should remove all these HTML tags (including their attributes inside) accordingly.



```
In $200 price range, should I be looking at
cards from AMD or Nvidia?
If you can't name just one, name a few. In
this case, price does not matter. Thanks.
```

(a) question.txt



```
Sparkfun will be more louder as it can
provide you with about 85dbA
I have made my decision, after some time. I
got the laptop with the GL702VS
```

(b) answer.txt

You should remove all these HTML tags (including their attributes inside) accordingly. Note that we assume that the content in the body contain complete tags i.e., all start tags are also accompanied by related end tags.

Example:

Before filtering: <p>In $200 price range, should I be looking at cards from AMD or Nvidia?</p>

After filtering: In $200 price range, should I be looking at cards from AMD or Nvidia?

Note that If you wish to read more about Character References, Wikipedia has a page on the topic https://en.wikipedia.org/wiki/Character_encodings_in_HTML#XML_character_references. More details about the HTML tags can be seen as https://www.w3schools.com/tags/. However it is **not necessary** to read these or understand HTML/XML to finish the assignment.

Finally, once you have completed with the filtering process, you should identify if the post is a question or answer. You should then save the data into two different files “**question.txt**” and “**answer.txt**” according to the post type shown in the data. The cleaned body/content for each post need to be saved in one line in the output file. Examples can be seen in Figure 2.

Note: You should write your code within the given template file “preprocessData_studentID.py”, and name the file with your own ID. There are two functions in the file: preprocessLine(inputLine) for dealing with the each valid data row from the file, and splitFile(inputFile, outputFile_question, outputFile_answer) for reading the input file, calling preprocessLine function to process the line, and saving the cleaned questions and answers into output files. All files should be saved in the current folder as that of source code file i.e., not using the absolute path.

2.3. Task 2: Building a Class for Data Analysis

The second task is about collating the required data for analysis. Apart from extracting the clean body as achieved in Task 1, the main task here is to further parse the given row of the data in XML format with object-oriented programming.

Your class "Parser" should contain the following methods:

- __init__(self, inputString):
  This is the constructor required for creating instances of this class. The inputString will be the row of data from the XML file.

- __str__(self):
  Re-define this method to present your data (the instance variables) in a readable format. You should return a formatted string in this method. The order of output should be "ID, post type, creation date quarter, the cleaned content".

- getID(self):
  Get Id of the post (indicated by "Id" attribute)

- getPostType(self)
  Get the post type of the post (indicated by "PostTypeId" attribute) with 1 as the question, 2 as the answer, and 3-8 as others.

- getDateQuarter(self)
  Get the date quarter of the creation date (indicated by the "CreationDate" attribute). One year has four quarters including Q1 (Jan to Mar), Q2 (Apr to Jun), Q3 (Jul to Sep) and Q4 (Oct to Dec). For example, given "2016-04-07T18:11:33.793" as the CreationDate, your program should return a string named "2016Q2".

- getCleanedBody(self)
  Get the cleaned body of the posts (indicated by "Body" attribute) which is the extracted cleaned body as that of task 1. You can import the function preprocessLine() in the template of Task 1 to reuse the pre-processing functionality of Task 1. But different from Task 1, we do not require splitting of the question/answers or saving to the file.

- getVocabularySize(self)
  Get the number of unique words in the cleaned body converted in the lower case. Note that we do not count space or punctuation as the word. For example, given the sentence "Although I use Mac, I do not like Mac.", there are 7 unique words including {"although", "i", "use", "mac", "do", "not", "like"}. The counting process may involve splitting the words from the cleaned body returned from the getCleanedBody() method. Note, just using str.split(" ") is not enough, as it may mistakenly recognize "mac," as a word instead of "mac".

When instantiating this class with the data row from the XML file "data.xml" as input, e.g., <row Id="2481" PostTypeId="1" CreationDate="2016-04-07T18:11:33.793" Body="&lt;p&gt;In $200 price range, should I be looking at cards from AMD or Nvidia?&lt;/p&gt;&#xA;" /> , your class can extract the ID, post type, creation date quarter, cleaned body, vocabulary size, and nicely print the input data.

Note: You should write your code within the given template "parser_studentID.py", and name the file with your own ID.

2.4. Task 3: Analyzing the File for Data Visualization

In this task, based on the class defined in Section 2.3 (Task 2) , you will implement two functions to visualise the statistics as some form of graphs. The implementation of these two functions should make use of the external Python packages, including NumPy, SciPy, Pandas, and/or Matplotlib in order to create the suitable graphs for comparing the statistics collected for posts.

The implementation of two methods should follow the requirement below:

- visualizeVocabularySizeDistribution(inputFile, outputImage):
  Given the input file "data.xml", you should count the vocabulary size for each post. Then you should draw a bar chart in Python to visualize the distribution of the vocabulary size of all posts. The x-axis is the vocabulary size, and the y-axis represents the number of posts with a certain vocabulary size. Note that for the x-axis, the vocabulary size interval is 10 and once the vocabulary size is larger than or equal to 100, you should put them into "others", i.e., 0-10, 10-20, 20-30, 30-40, 40-50, 50-60, 60-70, 70-80, 80-90, 90-100, others (left inclusive). You should save your visualization figure into a png file named as "vocabularySizeDistribution.png".
- visualizePostNumberTrend(inputFile, outputImag):
  This function displays the trend of the post number in the Q&A site. Given the input file "data.xml", you should first get the number of questions and answers in each quarter. Then following the time order, you should draw a line chart to annotate the number of posts in each quarter. Note that you should draw two lines for question number and answer number respectively, and add a legend in the figure to tell which

line is for which type of posts. You should save your visualization figure into a png file named “postNumberTrend.png”.

Note: Please import the class defined in Section 2.3 (Task 2). Apart from the defining these two functions, you should also call these two functions and obtain the png files. You should put your code for this final task into the template file “dataVisualization_studentID.py”, and name the file with your own ID.

2.5. User Manual

Apart from the comments along with the code, you should also prepare a user documentation (at least 4 pages but not more than 8 pages) in the PDF format with clear and complete instructions on how to run your programs and the analytical outputs (tables and graphs). For the graphs in Task 3, it is required to add an explanation to describe the graph.

3. Do and Do NOT

Do	Do NOT
● Maintain appropriate citing and referencing[1], ● Get support early from this unit and other services within the university, ● Apply for special consideration or for extensions[2] early if needed.	● Leave your assignment in draft mode (assignments in draft mode will not be marked), ● Submit late (10% daily penalty applies)[3], ● Attempt to submit after 7 days of the due date (they will not be accepted), unless you have special consideration.

3.1. Important NOTES

- For all tasks, we provide the template source code files, please fill in your code within the template files following the instructions. Not following the rules in the template file may invite mark penalties.
- Please make sure your file reading/writing operations include encoding type **utf8**. Changing the application running environment could cause issues if you do not have the encoding type. Any character encoding issues/exceptions will cause serious mark deduction.
- If one method is not working and it hinders the program from continuing running to show other functionalities, the following functionalities will get no mark. Therefore, it is important to finish the methods one by one and make sure they work properly.
- If any exception/errors happen when running your program, you will lose 50% of allocated function logic marks.
- Add correct validation and output messages to make your code more user-friendly to users.
- The assignment must be done using the **Pycharm**, **Python Version 3.9**.
- The Python code for this assignment must be implemented according to the PEP 8-Style Guide for Python Code.
- The allowed libraries are **Math**, **re**, **Numpy, Scipy, Matplotlib, Pandas**. You will not receive marks for the components if you use any other libraries apart from the mentioned library.

[1]https://www.monash.edu/library/help/citing-and-referencing/citing-and-referencing-tutorial
[2] https://www.monash.edu/exams/changes/special-consideration
[3] e.g.: The original mark was 70/100, submitting 2 days late results in 50/100 (20 mark deduction). This includes weekends and public holidays.

- Commenting on your code is an essential part of the assessment criteria. In addition to inline and function commenting on your code, you should include comments at the beginning of your program file which specify your name, student ID, the creation date, and the last modified date of the program, as well as a high-level description of the program.
- This assignment cannot be completed in a few days and requires students to apply what we learn each week as we move closer to the submission date. Please remember to show your progress weekly to your tutor.
- You must keep up to date with the Moodle Ed Assignment 3 forum where further clarifications may be posted (this forum is to be treated as your client).
- Please be careful to ensure you do not publicly post anything which includes your reasoning, logic or any part of your work to this forum, doing so violates Monash plagiarism/ collusion rules and has significant academic penalties. Use private posts or email your allocated tutor to raise questions that may reveal part of your reasoning or solution.

4. Submission Requirements

The assignment must be submitted by **Friday, 04 November 2022, 4:30 PM (AEDT)**.

The following files are to be submitted and must exist in your individual FITGITLab server repo:

- A series of **.py** files (i.e., dataVisualization_studentID.py, parser_studentID.py, preprocessData_studentID.py)**.**
    - A template, A3_student_template.zip, is available on the Moodle Assessments page. You have to use this template.
- A **userManual_studentID.pdf** file**.**

The above files must be compress to a .zip file named **ass3_studentID#.zip** and submitted via Moodle.

The **.py** files must also have been pushed to your individual FIT GitLab server repo with an appropriate history as you developed your solutions. Please ensure your committed comments are meaningful. **Do NOT** need to push the **history** of the user manual file to the individual FIT GitLab server. Only push the final version of the PDF file. **DO NOT** push the .zip file.

- No submissions will be accepted via email,
- Please note we **cannot mark any work on the GitLab Server**, you need to ensure that you submit correctly via Moodle since it is only in this process that you complete the required student declaration without which work cannot be assessed.
- It is your responsibility to **ENSURE** that the submitted files are the correct files. We strongly recommend after uploading a submission, and prior to actually submitting in Moodle, that you download the submission and double-check its contents.
- Please **carefully** read the documentation under the "**Special Consideration**" and "**Assignment Task Submission**" on the Moodle Assessments page which covers things such as extensions, correct submission, and resubmission.
- Please note, if you need to resubmit, you cannot depend on your tutors' availability, for this reason, please be VERY CAREFUL with your submission. **It is strongly recommended that you submit several hours before due to avoid such issues.**
- **Marks will be deducted for any of these requirements that are not strictly complied with.**

5. Academic Integrity

Students are expected to be familiar with the University Academic Integrity Policy and are particularly reminded of the following:

Section 1.9:

Students are responsible for their own good academic practice and must:

- undertake their studies and research responsibly and with honesty and integrity;
- credit the work of others and seek permission to use that work where required;
- not plagiarise, cheat or falsify their work;
- ensure that their work is not falsified;
- not resubmit any assessment they have previously submitted, without the permission of the chief examiner; appropriately acknowledge the work of others;
- take reasonable steps to ensure that other students are unable to copy or misuse their work; and
- be aware of and comply with University regulations, policies and procedures relating to academic integrity.

and **Section 2.9:**

Unauthorised distribution of course-related materials: Students are not permitted to share, sell or pass on to another person or entity external to Monash:

2.9.1 any course material produced by Monash University (such as lecture slides, lecture recordings, class handouts, assessment requirements, examination questions; excluding Handbook entries) as this is a breach of the Copyright Compliance Policy and such conduct may be a copyright law infringement subject to legal action; or

2.9.2 any course-related material produced by students themselves or other students (such as class notes, past assignments), nor to receive such material, without the permission of the chief examiner. The penalties for breaches of academic misconduct include

- a zero mark for the assessment task
- a zero mark for the unit
- suspension from the course
- exclusion from the University.

Where a penalty or disciplinary action is applied, the outcome is recorded and kept for seven years, or for 15 years if the penalty was excluded.

6. Marking Guide

Your work will be marked as per the following:

- Main Task:
    - 65% for working program — functionality (for all three tasks);
    - 10% for code architecture — algorithms, data types, control structures, use of internal libraries and/or external packages;
    - 10% for coding style — clear logic, clarity in variable/function/class names, code readability;
    - 15% for documentation — program comments and user documentation.

7. Getting help

7.1. English language skills

if you don't feel confident with your English.

- Talk to English Connect: https://www.monash.edu/english-connect

7.2. Study skills

If you feel like you just don't have enough time to do everything you need to, maybe you just need a new approach.

- Talk to a learning skills advisor: https://www.monash.edu/library/skills/contacts

7.3. Things are tough right now

Everyone needs to talk to someone at some point in their life, no judgement here.

- Talk to a counsellor: https://www.monash.edu/health/counselling/appointments (friendly, approachable, confidential, free)

7.4. Things in the unit don't make sense

Even if you're not quite sure what to ask about, if you're not sure you won't be alone, it's always better to ask.

- Ask in Ed: https://edstem.org/au/courses/8843/discussion/

- Attend a consultation: https://lms.monash.edu/course/view.php?id=141449&section=21

7.5. I don't know what I need

Everyone at Monash University is here to help you. If things are tough now they won't magically get better by themselves. Even if you don't exactly know, come and talk with us and we'll figure it out. We can either help you ourselves or at least point you in the right direction.