ISS Lab Task: Python

Tathagata Raha

July 9, 2021

1 Task

This task will cover all the concepts taught in the lab including OOPS, File I/O and parsing of JSON file along with some additional concepts like word clouds and bar plotting.

You have been provided with the *hyderabad_restaurant_reviews.json* file. The task is to fulfil the requirements listed below. Your code should take input from the terminal to fulfil each of the tasks. Look at the dry run below for more details.

1.1 RestaurantReviews class - 5 marks

You need to create a class called *RestaurantReviews*. The class needs to have *__init__* functions. The init function will load the provided json in a variable(make sure that variable can be accessed in all the methods of the class).

Input The program should initiate the class at the start of the program.

Output After loading the json, the program should print *JSON has been loaded.*

1.2 Average ratings - 15 marks

You need to create a method called *avg_rating* which will take a restaurant name as an argument and print the average rating of the restaurant.

Input The program should take input 1 along with the restaurant name separated by space.

Output After calculating the average rating, the program should print *Rating: <avg_rating>*. If the given restaurant name doesn't exist, it should print *Restaurant not found*

1.3 All reviews to file - 10 marks

You need to create a method called *reviews_to_file* which will concatenate all the reviews using newline of all the restaurants and store them to a file.

Input The program should take input 2 from the terminal.

Output The concatenated reviews should be stored in a file called *all_reviews.txt*. After storing the reviews, the program should print *All reviews have been stored.*

1.4 Barplot of ratings - 25 marks

You need to create a method called *plot_bargraph* which will take a restaurant name as an argument and plot a bar graph which will contain the rating in X-axis and the frequency of the rating in the Y-axis. Check the *matplotlib* module for this task.

Input The program should take input 3 along with the restaurant name separated by space.

Output After generating the bar plot, the plotted graph needs to be stored in a file called *plot.png*. After storing the barplot, the program should print *Bar plot has been generated.* If the given restaurant name doesn't exist, it should print *Restaurant not found.*

1.5 Word Cloud - 25 marks

Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. Significant textual data points can be highlighted using a word cloud. You need to create a method called *create_wordcloud* which will visualize the word cloud of all the reviews of all the restaurants. Check the *wordcloud* module for this task.

Input The program should take input 4 from the terminal.

Output After generating the word cloud, the word cloud needs to be stored in a file called *cloud.png* and the program needs to print *Word cloud has been generated.*

1.6 Exiting the program - 5 marks

On giving 5 as an input, the program should end.

1.7 Proper usage of classes and objects - 15 marks

Make sure the program is code properly in object-oriented design. This carries 15 marks.

2 Instructions

- Please read the entire document carefully to get a good idea on how it is to be implemented before commencing. Your **input/output should exactly match what is shown in the dry run**.

- **Complete the given tasks before the lab ends i.e. 5PM.**

- Submission to be made on Moodle only. **Submit a zip file as <roll_number>.zip, containing only the .py file.** No need to submit the other files like the plots, all_reviews, etc.

- For the filepaths, you can assume that the file names will be same as mentioned in the doc. There's no need to take input for different filenames. Also the files will be present in the same directory as the python file while evaluating.

- The submissions are most probably going to be evaluated automatically. So make sure you follow the instructions and I/O format correctly. There might be upto **20% penalty** for not following correctly.

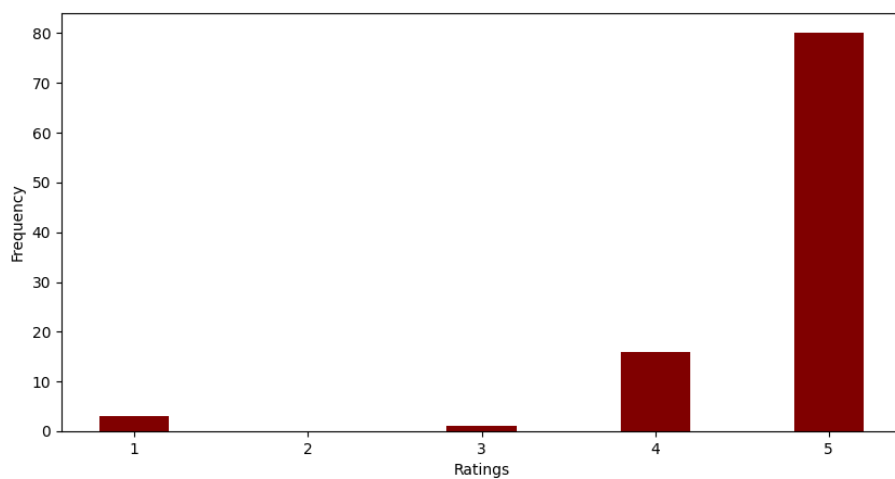- For the Love of God **don't copy**! If you have any doubts feel free to ask the TAs in your respective channel.

3 Dry Run

Figure 1: Dry run example

Figure 2: Output of *plot.png*

Figure 3: Output of *cloud.png*