INF 559 – Spring 2018

Homework #1: CSV & JSON

Due: February 6, Tuesday 11:59 pm PT
100 points

The goal of this assignment is to get you familiar with the Python libraries for reading and writing different file types and to start to understand some of the differences and issues that show up in reading and writing different types of data files.

We will be working with the same information encoded in two different file formats that we will be covering in class. The first is called a comma separated value or CSV format. In this format, the data is represented as a row with contains multiple values, each value is separated by a comma. Sometimes a different character like a '|' might be used to separate the column. Often a "header" row is used to say what the names of the columns are. Usually the same number and type of values are stored in every row, but this is not always the case as sometimes two tables might be stored in the same CSV file which can complicate figuring out what is what.

We will also be using files that use the JSON Object Notation, or JSON format. A JSON format is basically a standard representation of a Python dictionary, that can be read by other programming environments, such as JavaScript programs on the web. JSON is the format that is used to exchange much of the data used by cloud services.

Python provides a library to read and write CSV files a row at a time, or the whole file at once and read it into a Python structure. To handle JSON, Python provides a library to convert a Python dictionary into a string in JSON format which can then be written into a file. You can find information about these libraries in the Python manual, and lots of examples of using them on the web, in places like stack overflow.

In this assignment, we are going to read and write information in CSV and JSON format. One particular aspect we are going to explore is how to provide information about what is in the file we write (metadata) so when we give the file to someone else, they can figure out what is inside of it. In this example, the metadata we are going to be concerned with is the name of each of the columns, a description of the file contents and information about who created the data (the author, date, and organization of the author).

You have been provided with two files: called movies.csv and movies.json that have the same data in different formats. The movies.csv is a very small subset of the data from MovieLens data set from the web - https://grouplens.org/datasets/movielens/ . These files contain movie ratings given by users and have below information – userID, movieID, rating and timestamp. We will be doing simple data manipulation on two columns – userID and rating.

The assignment has five parts. You should put the solution in a single python file (Version – 3.5) **FirstName_LastName_hw1.py** with functions for each part described below.

Part 1 (30 points): Write a function part1 and read in the CSV file - movies.csv. For every unique userID, calculate the average rating. Write the results sorted by userID to a CSV file named part1.csv separated by commas as below. This file should have just the data in it without column headers

Example output (Not the actual output)

```
1,3.5
2,2.5
3,4.0
4,3.0
5,2.7
```

Part 2 (10 points): Write function part2 and repeat steps of part1. In addition, add a header row that includes the column names. Save the result in a file called part2.csv

Example output

```
userID,avg_rating
1,3.5
2,2.5
3,4.0
4,3.0
5,2.7
```

Part 3 (20 points): We would now like to include information about the author, creation time and a description of the data. Figure out how to add this data to the CSV file. Edit the part2.csv to include this additional data in a file called part2_metadata.csv. Write a new function called part3 that can read your modified CSV file using the CSV file reading library and **print out the metadata values to the console**. Provide comments in your new function that explain what you are doing. **EXTRA CREDIT** (**10 points):** Write a function write_csv_metadata to create the file part3_extra.csv with both the data and metadata.

Part 4 (30 points): Write a function part4 that reads in the JSON version of the data, computes the new values (As part 1) and writes out a JSON version in a file called part4.json which includes all of the **output data and metadata.**

Example output:

```
{
    "metadata": {
        "info": {
            "author": "name",
            "organization": "org name",
            "creation_date": "date"
        },
        "columns": {
            "userID": "description of the field",
            "avg_rating": "description of the field"
        }
    },
    "data": [
        [1, 3.5],
        [2, 2.5],
        [3, 4.0]
    ]
}
```

Part 5 (10 points): Sometimes the metadata is stored in a different file from the actual data. Write a function part5 that augments the results of part1 by creating a JSON file part5.json that only has the metadata in it and a key called 'datafile' which has the name of the CSV file in it. **EXTRA CREDIT (10 points)**: Write a function part5_extra that will read in the JSON metadata description and the CSV data file part1.csv to create a python dictionary that has both the data and metadata in it. Write output to a file called part5_extra.json .

What to Submit on Blackboard?

A single zip file **FirstName_LastName_hw1.zip** containing below files

a. FirstName_LastName_hw1.py
b. All output files – part1.csv, part2.csv, part2_metadata.csv, part4.json, part5.json, and **optional** - part3_extra.csv and part5_extra.json

Grading Criteria

- If program does not run, there will be 50 % penalty. In that case, grading will be done based on the output files submitted.
- If resulting output – userID and avg_rating are not sorted by userID in the ascending order, there will be 20% penalty.
- Late homework will be deducted 10% of its points for every 24 hours that it is late. No credit will be given after 72 hours of its due time.

Important Note: Submitted work must be your own. Don't share your code with anyone, and start early!