HOMEWORK 2: FILE FORMATS
(CSV, JSON & TABLESCHEMA)

Due Date: February 26, Wednesday, 11:59pm PT
100 points

Introduction:
The goal of this assignment is to get you familiar with DIfferent file formats and pros/cons of each of the file formats that we studied in Lecture. We will also get exposure to Python libraries for reading and writing different file types and will try to understand some of the differences and issues that show up in reading and writing different types of data files and how we can validate our datasets using Table schemas.

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 multiple values, each value is separated by a comma. Sometimes a different character like a 'Tab' or '|' 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 Web APIs and cloud services due to its cross-platform independence.

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. We expect you to look online at various online documentations and Stackoverflow to find the appropriate library to achieve Read/Write to and from CSV/JSON file formats.

Assignment Description:
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 - link. 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/three columns – userID, movieID and rating. **Note: All the pictures & screenshots are over dummy data, shouldn't be used to match results in your outputs.**

Task 1 (15 Points): Write a function task1 and read in the CSV file - movies.csv. For every unique movieID, calculate the average rating (**Note:** rounded to 1 decimal place). Write the results sorted by movieID to a CSV file named task1.csv separated by commas as below. This file should have just the data in it without column headers.

```
91,4.5
92,2.2
93,5.0
94,1.2
95,4.7
96,3.4
97,3.2
98,3.5
99,2.8
100,4.2
```

Task 2 (15 points): Write function task2 and repeat steps of task1. In addition, add a header row that includes the column names. Save the result in a file called task2.csv.

```
movieID,avg_rating
91,4.5
92,2.2
93,5.0
94,1.2
95,4.7
96,3.4
97,3.2
98,3.5
```

Task 3 (30 points): Write a function task3 that reads in the JSON version of the data, computes the new values (As task 1) and writes out a JSON version in a file called task3.json which includes all of the output data and metadata.

```json
{
  "metadata": {
    "info": {
      "author": "Your Name",
      "organization": "USC",
      "creation_date": "Date Created"
    },
    "columns": {
      "movieID": "movie id for the movies",
      "avg_rating": "avg rating given to the movies",
    }
  },
  "data": [
    [
      91,3.5
    ],
    [
      92,2.8
    ],
    [
      93,2.9
    ]
  ]
}
```

Task 4 (25 points): Write a function task4 that reads in your json file created in task3 (task3.json) and try to create only one CSV called task4.csv file containing combined information from both ['metadata']['info'] and another for Average rating data where your column header will be obtained from ['metadata']['columns'] and filled with avg_ratings data from ['data']. **Note:** Also include a small note (2-3 lines) about your conclusion which format is more suitable to store together Average Ratings Data and Metadata in this task (CSV or JSON) and Why? (**Extra Credit (10 points)**: If you can come up with a python script that can read data back from task4.csv that you just created and print information out on the Python Notebook Console.)

```csv
author,organization,creation_date
Your Name,USC,Date Created

movieID,avg_rating
91,4.5
92,2.2
93,5.0
94,1.2
95,4.7
96,3.4
```

Task 5 (15 points): We would like you to explore tableschema package to generate schema through CSV file. Write a function task5 that reads your movies.csv file and generate a schema for it in 'schema.json' file. Please refer to https://github.com/frictionlessdata/tableschema-py where you can get documentation on how to use tableschema, what it looks like and steps to install in Google Colab environment (stack overflow).

Coding Environment: You can use any IDE of your choice however its advised to us Google Colab as its easy to install and use packages.
Steps:

1. In your Google Drive upload movies.csv and movies.json files.
2. Use 'New' > 'More' to start a new Google Colab Environment.
3. Use the below two lines of code to setup Colab Environment.

```
from google.colab import drive
drive.mount('/content/drive')
```

4. Now you should be able to read/write files to/from Google Drive using the same functions like the one used in any IDE.

Submission on Blackboard:
A single zip file FirstName_LastName_hw2.zip containing below files
a. FirstName_LastName_hw2.py
b. All output files – task1.csv, task2.csv, task3.json, task4.csv, schema.json

Grading Criteria
• Only Python 3.5+ version submission will be accepted for this coding assignment.
• If the program does not run, there will be a 50% penalty for each task. In that case, grading will be done based on the output files submitted.
• If resulting output – movieID and avg_rating are not sorted by movieID in the ascending order, there will be 20% penalty for each task.
• We can accept Late homework till it is 24 hours after HW Deadline but it will be penalised by 20% points. No credit will be given for submission after 24 hours of its due time.

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