Data Cleaning and Analysis

1. Brief

Due date: 11PM on 02/01/2018

Late submission is not acceptable. The submission link in Blackboard will disappear immediately at the deadline. You can submit multiple times, only the latest version will be graded.

This is an individual assignment. Do not discuss or collaborate or share your ideas/code with other students.

What you turn in must have a `SOURCES.txt` file as follows:

1. If you use snippets from the internet or from any other source, document your sources in the `SOURCES.txt` file as well as inside the code.
2. If the work is yours alone, then the `SOURCES.txt` file will contain the following: I certify that all the work submitted for this assignment is exclusively by (your name, your UIC email address).

Refer to the course website for detailed information about plagiarism and cheating policies.

2. Introduction to Python

Knowledge of python and basic programming concepts is essential to complete the assignments and projects in this course. If you are not familiar with python, go through the Python 3 Essential Training course on Lynda: https://www.lynda.com/Python-tutorials/Python-3-Essential-Training/62226-2.html. This is a beginner's 6-hour course that teaches how to install python, introduces basic programming concepts and how to use them in Python.

3. Reformatting Data: Super Bowl Champions

The Super Bowl is an annual American football game that determines the champion of the National Football League (NFL). In this assignment, you will fetch a Wikipedia page and extract data from it. We would like to transform the extracted data into a usable format. There are few tables in the mentioned web page; however, the objective is to use python and the *BeautifulSoup* library to scrape the data from the second table and save it into a CSV file `transformed.csv`. Each line should contain 6 fields: game number, year, winning team, score, losing team, and venue. Note that the number after every team (and venue) indicates the number of times that the team (or venue) has been in the Super Bowl.

3.1. Data Extraction

First, fetch the HTML from https://en.wikipedia.org/wiki/List_of_Super_Bowl_champions and extract relevant portions from it. Note that an HTML table is divided into rows with the <tr> tag, and each row is

divided into data cells with the <td> tag. Using the converted HTML file, write a python script `transform.py` to pull data out of the relevant HTML portion (second table) and save it into a CSV file, named `transformed.csv`, which should match the output shown below. The CSV file headings should match the following and the rest rows in the file would be the rows of the mentioned table on Wikipedia.

```
Game,Year,Winning team,Score,Losing team,Venue
I,1967,Green Bay Packers 01,35–10,Kansas City Chiefs 01,Los Angeles Memorial Coliseum 01
II,1968,Green Bay Packers 02,33–14,Oakland Raiders 01,Miami Orange Bowl 01
III,1969,New York Jets 01,16–7,Indianapolis Colts 01,Miami Orange Bowl 02
```

... and so on, up to the last row containing a valid score.

Note that the Super Bowl this year is on February 4, but the assignment is due on 2/2. Your submission may be evaluated before or after 2/4. Make sure that your solution is generic enough to accommodate both cases.

3.2. What to turn in

`transform.py and transformed.csv`

4. Entity Resolution: UIC Courses

In an unfortunate series of events, the UIC registrar's course catalog has been corrupted, and must be rebuilt. You have been selected to help with rebuilding it, using various data sources that were acquired from various websites. Your job is to develop (and implement) a series of transformation rules that can be used to recreate the registrar's catalog.

In this assignment, you are given a file `class.txt` that shows the courses taught by UIC professors. However, since data is not clean, there might be cases that multiple versions of professor/course names actually refer to the same professor/course. Your task is to clean the dataset using proper transformation rules, and analyze the cleaned dataset by answering some queries below.

First, take a look at the dataset and get a sense of why this dataset is dirty. Then, try to come up with some transformation rules that you would like to use when you clean the dataset. The format of the input data is: `professor_name - course_1|course_2|...course_n.`

4.1. Data Cleaning

After you have determined the appropriate transformation rules, write a Python script, `clean.py`, to read the dataset, apply transformation rules, and output a cleaned dataset `cleaned.txt`. Note that the cleaned dataset should have the same format as the dirty dataset, and professors should be listed in alphabetical order based on their last name (do not include their first name in the cleaned dataset). For each professor, the courses that he/she teaches should also be listed in alphabetical order.

4.2. Data Analysis

Using the cleaned dataset, write a python script `query.py` with 3 functions (one for each of the following queries, named Q1, Q2, and Q3) to answer the following questions. Each of the functions should take only the cleaned, raw `cleaned.txt` file as input, and print the solution to the console one below the other.

Q1: How many distinct courses does this dataset contain?

Q2: List all the courses (in alphabetical order) taught by Professor x in comma-separated form. X will be a command line argument that will be supplied, for example:

```
python3 query.py cleaned.txt "Patrick Troy"
```

Q3: For professors who have taught at least 5 courses, implement the Jaccard distance to determine which two professors have the most aligned teaching interests based on course titles. Note that you should implement the function to calculate the Jaccard distance. Do not use an existing package or library.

4.3. Note

`class.txt` provided is only a subset of the full dataset, and we will run your `clean.py` on our full dataset to evaluate its quality. i.e., it has to be run with the input:

```
python3 clean.py [any_file].txt
python3 query.py [any_cleaned_file].txt [any_professor]
```

Make sure that your solution is generic.

4.4. Hints

You can assume that the professor's last name is a unique identifier for the name of the professor.

Normally, the professor's last name comes after their first name. But when the name is in comma separated form, the first name comes after the last name.

Note that since the dataset is made up, it does not have to represent the real situation in our department.

Pay attention to abbreviations!

4.5. What to turn in

`clean.py, cleaned.txt and query.py`

5. HANDING IN

Create a folder as follows.

Folder name: `<your_netid>`

Folder contents:

- `README.txt` - Explain your solution to both problems, note any issues/bugs with your solution, sources (if any), and any other relevant information.
- `SOURCES.txt` - As detailed in Section 1, BRIEF.
- All your python and result files, as mentioned in Sections 3.2 and 4.5 (What to turn in).

Compress the folder into zip file `<your_netid>.zip` (zip only, not rar, 7z or other formats).

Upload the zip file to Blackboard.