SI 630: Homework 0 – Regular Expressions

Due: Wednesday, February 4, 11:59pm

1 Background

Email addresses are everywhere online. Especially in personal web pages, people provide their email as way of easily getting in touch with them. However, unscrupulous spammers also look for these addresses to send unwanted email to people. As a result, some web page authors have resorted to obfuscating their address so that a human could still figure out what the address is without a machine being able to easily detect it. For example, someone might write `myname@domain.edu` as `myname at domain dot edu`.

2 Task

You've been asked to perform a security audit for a large university. They want to know what kinds of email addresses might be recoverable from each web page. Conveniently, they've already put together all of the web pages for you into a single file, where the HTML page for each page is on one line. Further, every page is guaranteed to have one email on it at most, since no one lists two email addresses for themselves on a page. However, not everyone lists their email on a page, so some pages have no email addresses! The big challenge is that there is no consistency in how the addresses are formatted!

Problem 1. Write a program that uses regular expressions to extract and canonicalize email addresses from web pages. Hint: regex groups may come in handy here. You will be provided with a large file of web pages on Canvas, `W21_webpages.csv`, where each page is on a separate line. Your program will produce a new file the canonicalized email address found on each page or the word `None` if no email address was found. By canonicalized, we mean that if the author wrote `myname at domain dot edu`, you would output `myname@domain.edu` in your file. Your output should have the same number of output lines as the input file. Note there is no space after the comma and the line number starts from 0. Besides, the first line of your answer file should be 'Id,Category' denoting two field names of each line.

3 Submission Procedure

For this (and some future) homeworks, we'll use Kaggle. You should follow the instructions at `https://www.kaggle.com/t/a72905816bb74e08816740b96582129d`, but roughly, you'll generate your solution and upload it to Kaggle, which will show you a score after your file has been compared with a *small fraction* of the total test set. Be sure to follow the formatting guidelines (specified above) so that Kaggle can correctly compare your file with the gold standard output. Homework 0 is intentionally easy so everyone is expected to get close to a 100% on Kaggle. If you don't get this right away, you can always modify your code and resubmit. Due to Kaggle's submission limitation, you can submit up to 20 times per day.

4 Grading Guidelines

Students will be evaluated based on the number of correctly recovered email addresses. Students close to retrieving all the emails (and not finding others) will receive full credit. Due to the automated nature of generating the data, there *may* be a chance that a full 100% is not possible, so don't worry if you're missing (or adding) a few extra than the exact solution.

In addition to Kaggle, you should submit two items as a part of your solution to Canvas:

- Your code
- A csv file named `email-outputs.csv` with the emails you extracted (what you uploaded to Kaggle as well)

5 Academic Honesty Policy

Unless otherwise specified in an assignment all submitted work must be your own, original work. Any excerpts, statements, or phrases from the work of others must be clearly identified as a quotation, and a proper citation provided. Any violation of the University's policies on Academic and Professional Integrity may result in serious penalties, which might range from failing an assignment, to failing a course, to being expelled from the program. Violations of academic and professional integrity will be reported to Student Affairs. Consequences impacting assignment or course grades are determined by the faculty instructor; additional sanctions may be imposed.