CS 498: Computational Advertising	**Fall 2018**
Homework 4	
Handed Out: November 7, 2018	*Due: December 5, 2018 11:59 pm*

1 General Instructions

- This assignment is due at 11:59 PM on the due date. We will be using Compass (http://compass2g.illinois.edu) for collecting the assignment. Contact TAs if you face technical difficulties in submitting the assignment. We shall NOT accept any late submission!
- The homework MUST be submitted in pdf format. Handwritten answers are not acceptable. Name your pdf file as YourNetid-HW4.pdf
- This is a two-part homework. This pdf contains details and instructions about the first part. We will release the second part of the homework within a week.
- It is OK to discuss with your classmates and your TAs regarding the methods, but it is NOT OK to work together or share code. Plagiarism is an academic violation to copy, to include text from other sources, including online sources, without proper citation. To get a better idea of what constitutes plagiarism, consult the CS Honor code (http://cs.illinois.edu/academics/honor-code) on academic integrity violations, including examples, and recommended penalties.There is a zero tolerance policy on academic integrity violations; Any student found to be violating this code will be subject to disciplinary action.
- Please use Piazza if you have questions about the homework. Also feel free to send TAs emails and come to office hours.

2 Part 1 - Reading Assignment

In the first part of the homework, you need to read two survey papers on privacy. The paper pdf's are attached on the wiki. Both papers contain a list of subtopics related to various privacy issues. After reading the survey papers, you need to pick one subtopic based on your interest. You need to find two newspaper articles and one research paper on that subtopic from the reference list provided in the survey papers. Submit a review of these two articles and the research paper. The reviews should be submitted in a pdf format.

While writing the reviews, you need to follow the exact modular structure provided below. Please note your submitted pdf must follow this exact structure to be graded properly.

- Name of subtopic chosen:
  - State in one sentence what is the main idea/issue/problem statement of this topic.
  - State briefly why is the problem hard.

  - State briefly why is the problem interesting.
  - State in one sentence what interests you specifically about this problem.
- Name of research paper [properly cited]
  - State in one sentence the main idea of this paper.
  - Mention three strengths of this paper.
  - Mention three weakness of this paper.
  - Suggest one improvement over the paper.
- News Article 1 : follow same structure as the research paper.
- News Article 2 : follow same structure as research paper.
- Conclusion : Briefly state the main lesson learned about the topic after reading these three contents.

Please organize your submission in the list structure provided above. Make a separate sub-point for each item in the list and properly highlight each item name in your submission.

3 Part 2

In this part, you will install a browser extension, run the extension in the background and collect ad data. You will then download this collected data to a local json file into your system and write a code to perform basic analytical task on this data.

You need to follow these steps sequentially to collect the ad data.

1. Download the zip file *cs498_hw4_chrome.zip* from wiki Assignments page
2. Unzip the file.
3. Go to chrome://extensions/
4. Turn on developer mode
5. Select *Load unpacked* and navigate to the unzipped folder *cs498_hw4_chrome*, then press *Select*.
6. Adtracker extension should now be installed in your system. To check , please open your browser and the AdTracker icon **A** should appear near the top right corner of your web-page.

7. Once the **A** button appears on the top-right corner of your page, it will automatically start collecting data on the background. You can press on the **A** button at any time to store the data that has been collected till now in a local file. The data will be saved as *ads.json*

ads.json file will contain the data in the following format.

{ ads:

{ visited_page1_url:[ad1_url, ad2_url], visited_page2_url:[ad1_url, ad2_url, ad3_url], ... } }

The ads will be stored in a dictionary format where you will have the *urls* of the pages you visited as keys and the corresponding ad urls shown on those pages as values. For example, in the case above you visited a page *visited_page1_url* and viewed two ads— *ad1_url* and *ad2_url*. Here the ads are represented as URLs. You can trivially use existing python libraries like *urllib* to fecth the corresponding ad image into your local folder.

There are a few important requirements that you need to follow to collect the ad data properly.

1. It is preferable that you use the extension with *Google Chrome* browser. While the extension may work with other browsers, it has not been extensively tested for the others.

2. Ads must be **legitimate**— non-tracker and non-logo type. On manually evaluating, you will see that there are two types of ads that you need to filter out from ads.json file before performing any analysis. For certain ads, the corresponding image will contain only a single pixel. These ads correspond to the trackers for that webpage and we need to remove them. Also, for some other ads, you will see the keyword **logo** in their URL. These are some generic images displayed on a page and we will filter out all ads containing **logo** in their URL before doing our analysis.

3. For this assignment, you need to collect **at least 100 legitimate ads** in your data. So you need to keep the extension in your browser and browse for a sufficient amount of time so that you can collect 100 different ads. Try to browse a lot of different pages during your browsing session which will help you collect these ads in a shorter period of time.

4. Please note that before you quit your browser or shut down your system, you need to press the **A** button. This will save all the data that has been collected till now to your local file. If you quit the browser without pressing the **A** button, all the data collected till now will be erased.

5. Since you need to collect at least 100 legitimate ads in your dataset, you can either do this over multiple browsing sessions or over one long session. If you collect data over

just one session, you need to press **A** button only once at the end and get one ads.json file. If you collect data over multiple sessions, you need to press the **A** button once at the end of each session to save a new ads.json file. You can then append all these ads.json files created after each session to create a combined ads.json file.

If you have followed all the above steps correctly, now you should have an *ads.json* file. Now as a part of the assignment, you need to do the following things.

1. Parse the ads.json file
2. Filter out ads containing keyword *logo* in their URL.
3. Retrieve images for all these ad URLs using methods from libraries like *urllib*. Note some ads may have an expiry time on the validity of the ad url. So we suggest that you do this step soon after you have collected all the ads.
4. Remove images that contain a single pixel. You can do this either manually or programmatically.
5. Check whether the number of ad images that you currently have is at least 100. If not, then you need to repeat the data collection step.

After the completion of the above five steps, you should now have 100 ad images. Now create an ontology of 10 attributes that are important to you. Attributes can be various like gender, ethnicity, location etc. Now you need to go over each ad image and find all the relevant attributes in your ontology that this image covers. For each of these relevant attributes, see whether the ad conforms with your value of that attribute. If yes, then you will increment a counter value for that attribute by one. Else, you will do nothing.

For example, suppose person A is a male who lives in Urbana. Say he sees an ad image which is targeted towards males living in New York. Lets further assume that A had attributes Gender, Ethnicity and Location in his ontology. So after seeing this ad, he will increment only the Gender counter by one. The ad does not cover Ethnicity attribute. For Location attribute, the ad does not match with the person's location.

After repeating the above process for all 100 images, you should now have a different count value for all 10 attributes in your ontology. Create a bar chart with these 10 values. On the x-axis, you will have the attribute names in your ontology. On the y-axis, you will have the count value or in other words, the number of ads that were shown to you based on this attribute of your ontology. Note that we are not asking you to reveal your value for a particular attribute, but rather the attribute name. In other words, you do not need to reveal your gender, you only need to say that you included Gender as an attribute in your ontology.

Output/Deliverables:

1. You need to submit the code that you used to process the ads.json file and fetch legitimate ads.
2. You need to submit a bar chart on attribute names vs the number of ads shown based on that attribute. You should have 10 bars in your chart corresponding to 10 attributes in your ontology.