**Group Members:**

Andrew Staniforth

Shuchun “Fred” Wang

Connie Wu

**CIS192 – Final Project Summary**

Searching for Under-Monetized Sites

*Overview and Description*

Using various Python packages such as:

urllib2

html5lib

Beautiful Soup

Scrapy

JSON

Lxml

a)

We want to search a given list of websites on the internet for ones that are undermonetized based on the current ad revenue generated versus what can be potentially obtained based on site analytics data. Based on user preferences, they can input a custom list, or can use provided lists such as the top ~100,000 by activity provided by Alexa. What this means is that some sites get a steady stream of visitors, but because of several reasons such as site age, owner, and unawareness, these sites can be optimized to generate more ad revenue based on the number of users that visit. Our goals involve obtaining the most current data on incumbent ad presence on theses sites. Then interface these data with the analytics to allow users to gain an intuitive insight on what sites are indeed under-monetized, from which they can take action!

Features:

1. Scrape sites on the internet to obtain information on number of ads that exist on the front page and at a desired level down.
2. Goals: to generate a number output of ads, and consolidate with published data on the number of visitors to the site, click-through rate, frequency of visitors, and retention
3. Develop an algorithm that will rank these sites based on the data obtained
4. Have a sorter UI that will allow the user to rank and sort through criteria used when searching for advertisements
5. Features of the UI will be site preferences, giving users quantifiable valuation and site potential for monetization
6. In terms of data structure, our project uses threads to parse through each site. Then a file is made out of the site data, which can then be parsed to get the number of ads that currently exist on the site. Depending on how time permits we will either take site data analytics from Alexa to obtain the number of visitors of the particular site using a similar threading data structure to parse through the data that exists on that site. If we have time will try to obtain this data ourselves, but it is not essential to do so.
7. Major Challenges:
   1. The major challenges include the strategy for parsing the site visitor data. We want to figure out what is the incremental benefit of looking for the metrics ourselves versus using what is already provided on the analytic sites.
   2. We want to make sure that when we are parsing data, there are as few possible errors in the parsed files, this entails understanding to a certain extent what Python packages are doing/not doing
   3. Quantify undermonetization: We are currently in the process of coming to a consensus on how to interpret the data for the user in the most objective way possible. This requires figuring out the weight of the various metrics and what their potential impact on monetization of the site is.
   4. Coming into this project we had no web to python experience, so coming up with an efficient way to read the web and parse that data took time. As it stands now we are in the process of migrating away from a system where the entire html is read into a file, and that file is again parsed.
      1. The speed of the program due to the web reading was incredibly hard to optimize. To tackle this, threads and processes were looked into. At this stage, we are using threading which in CPython has little optimization benefits due to the GIL.

1. Extended Features:
   1. As mentioned before, if there were more time we would have liked to created a custom analytics platform to obtain our data
   2. It would have been nice to have a way to figure out the type of ads that currently that exist on the site and figure out a way to maximize the consumer preferences from the visitor data demographic information
   3. In addition, some ad exchanges may have better CTR than others, we would have like to build that in to the calculations as well.
   4. If time permitted, we would have liked to make a nice graphic user interface for demonstration purposes
2. Breakdown of Responsibilities:
   1. Andrew, Primarily the Back End person: Analyzer, Scanner
   2. Connie, Primarily the User person: UI, Optimization of code
   3. Fred, Primarily the web person: Scraper, Analytic Research