DOUGLAS COLLEGE

COMMERCE AND BUSINESS ADMINISTRATION

CSIS 3290-001: FUNDAMENTALS OF MACHINE LEARNING

Project #1: Data Wrangling and Regression Analysis

Learning Objectives

1. Extract data from online sources and prepare it into a form suitable for analysis
2. Apply regression analysis techniques to analysis the data and make predictions

Background

Given intense competition in the business environment, many companies utilize several strategies to stay competitive. Nearly every company now has an online presence. One the strategies companies use to win customers is through online search advertising. Using the search advertising strategy, companies often utilize two options for driving traffic to their website: (1) using **organic search** (i.e., using non-paid keywords), and (2) **paid search** or **Adwords keywords**(i.e., buying keywords from Google Adwords). Companies often spend hundreds of thousands of dollars on paid search, but the value they obtain from these strategies is not obvious.

The purpose of this project is to examine the business value of online search advertising strategies.

To do so, we are going to use Fortune1000 companies dataset since it provides outcome variables which can be applied to analyze the business value of online search advertising.

The project is divided into two parts. In part 1, you will collect data from online sources and prepare it into a form suitable for analysis. In part 2 of the project, you'll develop regression models, evaluate them, and make predictions with the chosen model.

This document provides a description for part 1 od the project.

Part 1: Data Collection and Preparation

Step 1: Read the process the Fortune1000 data file ("fortune1000.txt")

In Project1 folder on Blackboard, you are provided with a text file named 'fortune1000.txt which contains a list of the top 1000 Fortune companies in the U.S.

1. Read the file using "with open" statement, like:
   with open("fortune1000.txt, "r") as file:
   …your statement to process the file

   Store each line of text read in a list variable.
2. After you have completed processing the file, reshape the list into a DataFrame and provide appropriate column headers (**note**: there are 1000 companies and 11 variables. The DataFrame should be of the shape [1000x11]. The original data has the following columns: Rank, Name, Revenue($m), % Change in Revenue, Profits($m), % Change in Profits, Assets($m), Market Value($m), Change in Rank (1000), Employees, and Change in Rank (500 Only). Please use the above as column names/headers for the DataFrame. Note: you can use df.head() and df.tail() methods to examine the first 5 or last 5 rows of data, respectively (here 'df' is the user-defined name for the DataFrame).
3. Drop the following columns from the DataFrame since they are not necessary for our analysis: (% Change in Revenue, % Change in Profits, Change in Rank (1000), and Change in Rank (500 Only).
4. The data is recorded with dollar signs ($) and commas (,) which renders it unsuitable for analysis. Please strip '$' and ',' from the relevant columns. You can do this either column by column using *str.replace()* method or for a group of columns using df(cols).replace() method, where cols is a variable containing the list of columns to be processed. Note: if you choose to use *df(cols).replace()*, you'll need to specify regex=True as a second parameter since this approach uses a regular expression.
5. Save the DataFrame to a CSV file (use df.to_csv('filename.csv') to save df to CSV file).

Step 2: Add a column of Company URLs to the DataFrame

Obtain the URL for each company and append a column of URLs to the DataFrame. Note: I tried to search for urls that match the current list of Fortune 1000 companies, but I did not see a free list. The following link (https://gist.github.com/hrbrmstr/ae574201af3de035c684) provides a list of Fortune 1000 companies and their URLs, albeit not the current ranking! Please use it to fill in the URLs of the current list of companies contained in the DataFrame file (hint: you may find this step easier to do using VLOOK function in Excel).

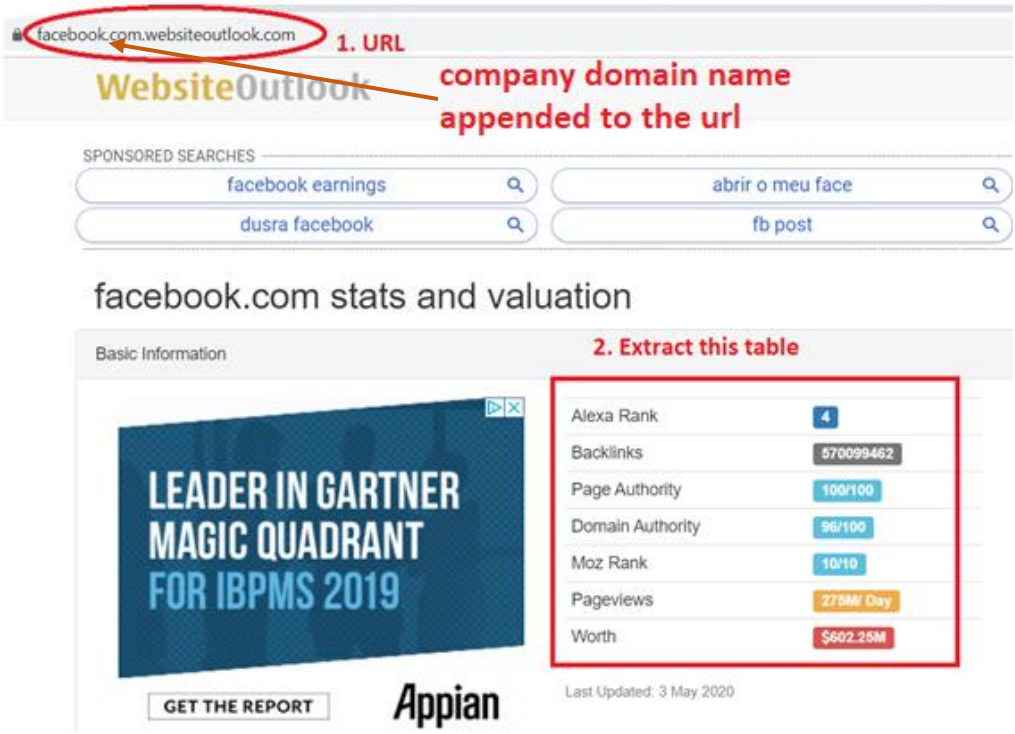
For companies that do not have a matching URL from that link, perform a web search to obtain their URL.

<u>Step 3</u>: Extract/Scrape Search Advertising Related Data from A website

Extract data from https://websiteoutlook.com/

1. Notice that the domain for the company you want to obtain data for is appended to the URL of websiteoutlook.con (refer to no. 1 in the image below where faceboo.com has been appended to the URL. This would help to extract the data for facebook.com and so on for the rest of the Fortune 1000 companies).

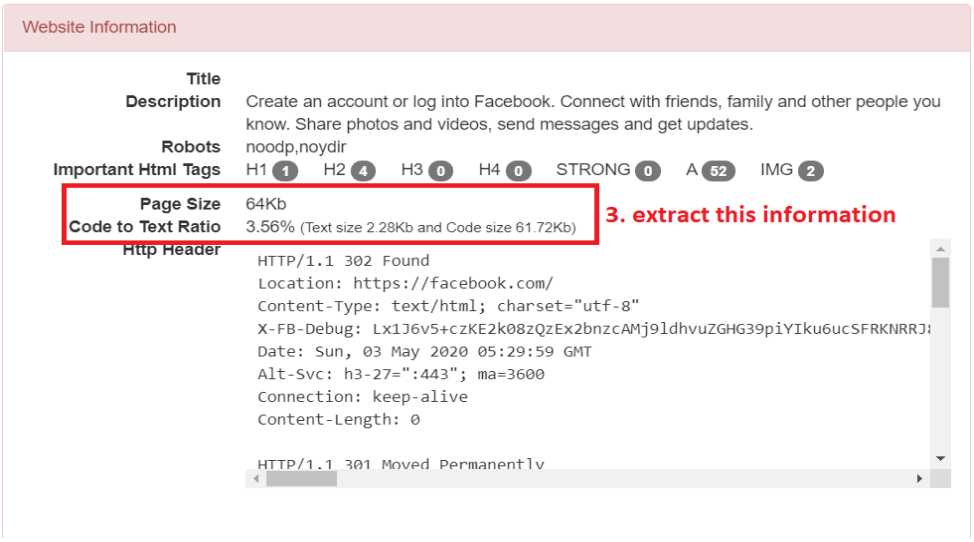
   To iterate through a list of URLs, you'll need to preload them into a list variable (note: since we have a long list of companies, I advise you to work through the URLs a few at time (otherwise you might get blacklisted and barred from extracting the data if you overwhelm their server), say 20 to 50 URLs at a time. You can try using '**BeautifulSoup**' library to extract the data (see for example the answers at this link to see how this is done, especially the third answer which uses batches of URLs: https://stackoverflow.com/questions/40629457/scrape-multiple-urls-using-beautiful-soup).



2. Once url contents have been returned by the get method, you will need to parse the content to extract the necessary. As shown in the image above, please extract basic information from the table indicated (**<u>note</u>**: if you have done web programming, working with selectors, css, html tags, etc., should be straight forward.) You will

need to view the page source from the Browser window, locate where the table is defined and use appropriate selectors/HTML elements to extract the information from the table. Please store each value extractedfor each company in a list variable (**hint**: you could use data labels as list variable names and corresponding values as list items. For example, Alexa_rank would be the name of the list variable and the first entry would be 4).

3. Repeat step 2 to extract the data shown in the image below.



4. Again, repeat step 2 to extract the data shown in the image below.

4. Extract this table

SemRush Metrics

Semrush Rank	5	Rank based on keywords, cost and organic traffic
Keywords	98265060	Number of keywords in top 20 Google SERP
Organic Traffic	559232741	Number of visitors coming from top 20 search results
Cost (in USD)	600225091$	How much need to spend if get same number of visitors from Google Adwords
Adwords Keyword	20565	Keywords a website is buying in Google AdWords for ads that appear in paid search results.
Adwords Traffic	1383442	Number of visitors brought to the website via paid search results.
Adwords budget (in USD)	4820975$	Estimated budget spent for buying keywords in Google AdWords for ads that appear in paid search results (monthly estimation).

5. After you have extracted all the data, combine the list variables into a DataFrame (remember to use head() and tail() methods to verify that everything worked out well).
6. Read the CSV file created from the DataFrame in step 1 on page 2, storing the data read into a DataFrame once again. Join this DataFrame withn the DataFrame from No. 5 above and save the joined DataFrame to a CSV File.