**EDGAR Log Dataset Report**

The EDGAR Log File Data Set contains information in CSV format extracted from Apache log files that record and store user access statistics for the SEC.gov website. The problem statement was to perform missing data analysis and visualization using Tableau.

We referred to the rules for each column from the following location:

<https://www.sec.gov/files/EDGAR_variables_FINAL.pdf>

**Steps:**

1. Fetch the year from the user, for which he/she wants to get the log files
2. Dynamically generate the URL’s to fetch all the zip log files from the website
3. Download data from the zip files and load them into pandas dataframe
4. Merge the dataframes of every month to form one consolidated dataframe which contains data of the whole year
5. Once we have the data, the first step is to handle the missing information. We first replace all the NaN or missing values with the following appropriate values:
6. IP address: Replace by default max IP address of 255.255.255.255
7. Date: Using forward fill, we would fill this missing value with the next valid entry
8. Time: Using backward fill, we would fill this missing value with the previous valid entry
9. Zone: Replace by ‘Not available’ since we cannot identify from which zone the request could be arriving
10. Extension: Replace by default value “-index.htm”. The default page from where the user must have landed here
11. Size: Replace by default value “0”
12. Noagent: Replace by default value “1” (no user agent)
13. Find: Replace by default value “0” (no character string found)
14. Browser: Replace by “Not Available”, since we cannot get this information
15. IDX:

Rule: Takes on a value of 1 if the requester landed on the index page of a set of documents (e.g., index.htm), and zero otherwise

So we are computing the check, if the extension value for the record is “index.htm”, we set the value as 1 else 0

1. Norefer:

Rule: Takes on a value of one if the Apache log file referrer field is empty, and zero otherwise

We refer to the column “find” which takes a value 0 if the referrer field is empty. Thus we check if the find column has a value 0, we set Norefer’s value 1 otherwise 0

1. Crawler:

Rule: This variable takes on a value of one if the user agent self-identifies as one of the following webcrawlers or has a user code of 404. Below are the actual Perl regular expressions used: a. if($agent=~m/(wget|Googlebot|polybot|Yahoo\!\s\*Slurp|spider|robot|perl|python|lwp|crawl er)/i){$crawl=1}; b. if($code==404){$crawl=1};

Thus we are checking the value of code, if it is 404, we set the value as 1 else 0

We referred to the Github repository of EDGAR and fetched a txt file which consisted of CIK and Company name. We fetched this information and joined it with our dataset. Thus now along with the rest of the information, we also have the company associated to each CIK.

As per the rule, CIK and accession number are related. The accession number is made up of three sections, CIK-Year-Number\_of\_filings\_listed. So to ensure validation, we are checking if the the CIK and the accession number’s 1st part is same. If not this is an anomaly and we have indicated this by setting the CIK\_Accession\_Anamoly\_Flag as “Y”.

As per the rule the file name comprises of accession\_number.extension. We created a new column named file\_name which is the concatenation of accession\_numaber and extension.

To identify the anomalies in the dataset. We traced cik’s which have sudden rise in ip hits and presented it in a separate dataframe.