

Design/Practical Experience [CSN1020]

(Department of Computer Science & Engineering)

Project Report

Academic Year: 2021-2022

Semester: 2

Date of Submission of Report: 01-05-2022

Project Title: Android Play Store Data Analysis.

Project Mentor: Dr Sumit Kalra (Department of Computer Science & Engineering).

Project Deliverables: Insight on the kind of bugs occurring in a set of selected android applications to aid new developers.

Submitted by: Dev Goel (B20CS090).

Objective:

The aim of this project is to analyze the kind of bugs that occurs in different categories of mobile apps. Some of the different types of bugs that occur in apps are compatibility crash bugs, performance issues, UI and UX bugs, network-related issues, memory leakage, slow responses, permission issues (camera, video, audio), general functionality issues, and module features, OS version issues etc. Hence, the objective is to analyse the different kinds of bugs occurring in different categories of apps like Social

media, Educational, Utility, Gaming, Finance/Banking apps using data collected from web scraping different websites that contain this data about the version histories and bug histories of mobile apps to aid new developers. This will assist the developers to develop state-of-the-art apps that the public deserves.

Study/Research:

The project's prerequisite involves the knowledge of writing web crawlers to scrape the data from the websites and the knowledge of metadata. So, I first learned about the metadata which is the index that gives the clue that where the data is stored to the user. It has three types i.e., operational, extraction and transformation, and end-uses. Operational gives a source of data from where it is extracted, extraction and transformation keep information of all the extracted data from the different data warehouse and their transformation and end-use is the index (overall view for metadata) tells the user how to access data/information. Then, I also learned how to write the web crawlers in python to scrape the data from the websites for my project. I learned the BeautifulSoup which is a Python package for parsing HTML and XML documents. I read the documentation and watched the tutorials for this package. It basically builds a parse tree for the parsed pages, which may be used to extract data from HTML and hence it is useful for web scraping.

Technologies Used:

- Google Colab
- Python
- BeautifulSoup
- NumPy
- Matplotlib
- Pandas
- Scikit-learn

The methodology used to implement the problem:

The methodology that I used for implementing the given project is that I first gathered the list of websites present on the internet that contains the version histories data, the date of update, and the updates on bug fixes. I gathered all the websites that contain the data for different categories of apps. Then, I applied the concepts of web scraping and scraped the data of at least 5 apps from each of the categories of apps chosen. For this part, I have to modify the scripts for some websites and also for scraping the websites I read about the different tags of HTML. Then, after scraping the data from different websites I stored them in a list and then created a data frame using the pandas library. I created three columns the first one contains the name of the version of the apps, the second one contains the date of release and the last one contains the bug fixes comments. Then,

with the help of some libraries of python, I analyzed the bug comments count and frequent bugs that appeared in each category of the app.

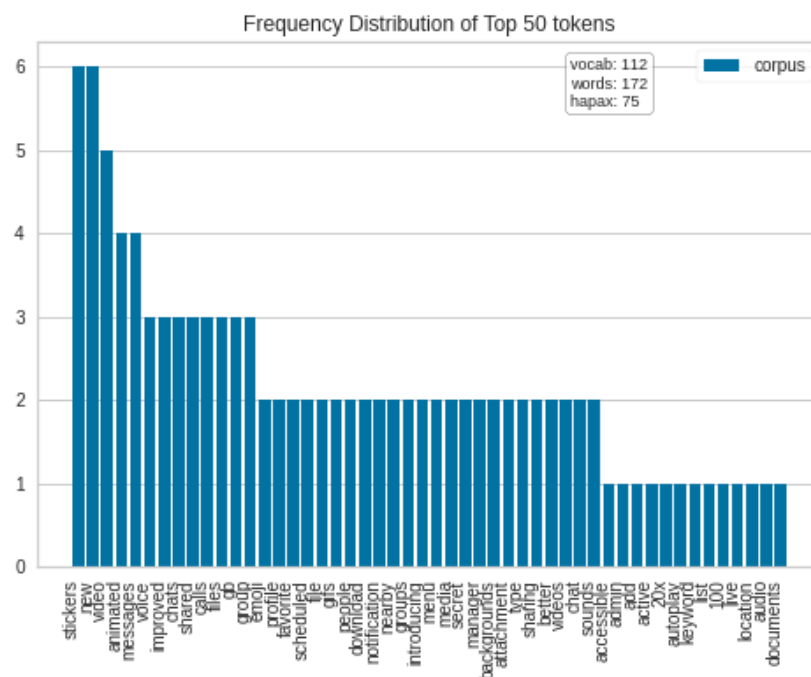
Issues Faced:

- **<Response [503]> Error:** While web scraping the websites, some of the websites returned this error. And for these websites, I was not able to scrape the data. So, I read about this error that a <Response [503]> status code normally means "Service unavailable". Hence, I was not able to scrape the information from these websites.
- **Extracting/Scraping data from websites:** While scraping the data from the websites I was not able to access some of the text that was in between two different tags. So, I read about this in the documentation and fixed the issue and was able to store the text in between two different tags.

Analysis:

For the analysis of the dataset created from web scraping the data from different sources over the internet. First, I analysed the data frame created after scraping the data. I noticed that there were some duplicated entries present in the dataset. So, I dropped those duplicate rows from the dataset. I also updated the date format in the data frame. Since the analysis of bug fixes includes the bug fixes statements. I used the principles of feature extraction from the text statements. For this I used the count vectorizer

from `sklearn.feature_extraction.text` library to convert a collection of text documents (bug fixes statements) to a matrix of token counts. I passed the hyperparameter `stop_words` as 'English' so that random words that are not much related to the bug fix shall not be counted in the analysis of the statements. I also printed the feature names to visualize the features extracted from the statements. Then, using the `FreqDistVisualizer` library from `Yellowbrick` extends the `Scikit-Learn` API. Hence, I plotted the frequency distributions of the top tokens present in these statements. This helps to identify the most used term in the bug fixes statements. I plotted these for every app in each category. For example, the Plot of the Telegram app (under the Social Media Category)



So, from the plot, we can see that these are the tokens that have the majority of the frequency distribution among others. So, tokens like

‘stickers’, ‘video’, ‘improved’, ‘shared’ and others highlight that these features are resolved while fixing the bugs by the developers. I also plotted some unique statements from the bug comments so that the new developer can know about various bugs that occur while developing a similar app.

- Secret chat voice messages fixed
- PROFILE VIDEOS, IMPROVED PEOPLE NEARBY, 2 GB FILE SHARING, AND MORE
- Improved streaming support for audio and video files.
- PUBLIC GROUPS, PINNED POSTS, 5,000 MEMBERS
- Yes, Video Calls (first version)
- Add emoji to a message by typing the ‘:’ keyword.
- Share your location with friends in real-time with the new Live Locations.
- SCHEDULED MESSAGES
- DOWNLOAD MANAGER, NEW ATTACHMENT MENU, AND MORE
- NEW: View and search all the documents shared in a chat new Shared Files section (accessible in Shared Media). You can now send files of any type up to 1,5 GB in size.

Similar to these fixes statement for the telegram. I have also printed the plots, analyses, and statements for every app in each category in my colab file.

Hence, In total, I scrapped around 200+ web pages to gather a dataset of around 4100+ rows of data. And, for this, I wrote around 2500 lines of python code. Also, created the analysis plot for each of the websites using their individual data.

Acknowledgements:

I would like to express my very great appreciation to my design credits project mentor Dr Sumit Kalra for his valuable and constructive

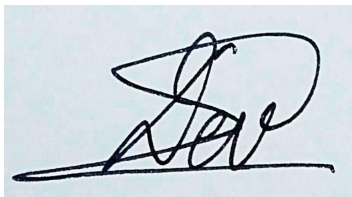
suggestions, assistance with the methodology during the meetings I did with him in order to plan and execute the work required to be done and how it is to be done. His willingness to give his time so generously has been very much appreciated. I learned a lot through this project as I was introduced to many new things like web crawlers, web scraping, new python libraries, metadata of apps etc. I have put in a lot of effort in doing this project individually and had an overall good experience.

References:

1. <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
2. <https://www.zrix.com/blog/common-bugs-found-while-mobile-testing>

Declaration:

I declare that no part of this report is copied from other sources. All the references are properly cited in this report.

A handwritten signature in black ink, appearing to be 'ZAP', written on a light blue background.

Signature of Student