Mobile Application Recommendation System: Enhancing User Experience through Personalization

Indraneel Mukhopadhyay
Amity Institute of Information Technology
Amity University, Kolkata, West Bengal, India

Abstract- A The widespread distribution of new ideas through mobile applications has the capacity to greatly influence individuals' lives on a global scale. Currently, there is a significant proliferation of apps across various categories such as work, commerce, and leisure. These apps are being developed and made accessible online in large numbers. The current accessibility of app stores may pose challenges in effectively recommending specific software to individual customers. The importance of customizing app suggestions to align with individual user preferences and limitations cannot be overstated. The proposed approach focuses on the development of a mechanism that suggests appropriate apps to users by considering factors such as user ratings, app size, and required permissions. Apkpure.com, a widely recognized android software market, employs a Web Crawler tool to verify links and gather data pertaining to the website. The Clustering Algorithm is utilized to categorize the programs according to the three criteria: "Favorite," "Allowed," and "Secure." The objective of this paper is to propose a recommendation system that includes important factors such as star ratings, file size, and access control without compromising on simplicity.

Keywords – Mobile App, Recommender System, extraction, security

I. Introduction

Due to their inherent simplicity in both development and upkeep, android applications have experienced a remarkable surge in prominence throughout the preceding decade. Usually, programmers engage in the development of applications with the primary objective of addressing and fulfilling user inquiries and demands. It would be fallacious to make the sweeping claim that every application we employ possesses an unequivocally high degree of safety. It is of utmost importance to acknowledge the presence of malware applications and ascertain their level of security, as certain instances may yield adverse consequences such as system disruption or unauthorized acquisition of personal data. The Android Application Suggestion Engine now commences its operational endeavors. In our ongoing efforts to assist users in identifying reliable applications within the Google Play Store, we are currently engaged in the development of an Android App Recommendation System. It facilitates the process of classifying applications based on specific parameters and providing recommendations for applications that can be deemed reliable. Apkpure.com has emerged as a preeminent hub for procuring Android applications, thus prompting our diligent concentration on this platform for the purpose of data collection. We utilize a web crawler, a sophisticated software application that not only verifies the integrity of hyperlinks but also organizes and catalogues websites along with their interconnected links. The primary challenges we encountered revolved around procuring pertinent information from the play store and amassing the requisite data for a recommendation.

II. RECOMMENDATION SYSTEM ARCHITECTURE

Recommendation system is either Content-based or Collaborative. Content-based recommendations employ techniques that prioritize item descriptions and attributes. These methods generate a content-based profile for each user, which stores characteristics of previously viewed items. The recommender system aims to identify items that align with the user's preferences by comparing the information provided in their profile and item descriptions. Content-based algorithms recommend items that are like items previously viewed by the user [1]. A movie recommendation system analyses a user's preferred movies to identify commonalities in terms of content, such as actors, directors, and genres. The user's preferences will be used to identify movies that closely resemble the ones they liked [2]. Content-based recommenders have limitations. One issue is the "new user problem," where users must view or rate a specific number of items before their preferences can be accurately determined. Occasionally, there is a possibility that the recommender system may suggest items that are excessively like one another, resulting in the provision of superfluous information to the user.

Collaborative recommender systems gather user feedback on item ratings. The platform connects users based on shared ratings and utilizes this data to generate recommendations. Collaborative filtering algorithms recommend items based on the preferences of users who have similar tastes. This type of recommendation is also referred to as personalized recommendation [1]. A movie recommender system identifies peers, i.e., users who exhibit similar rating patterns to the user in question, to provide recommendations. The recommended movies would be those with the highest ratings among peers, which the user has not yet seen [2]. The user's text is too short to be rewritten. Collaborative recommenders utilize human judgments but are also affected by the new user problem. Furthermore, users encounter the "new item problem," which results in new items being disregarded or not recommended until a specific number of users have provided ratings for the item [2].

Within this section, we shall proceed to illustrate and expound upon the envisaged system architecture. Illustration 1 showcases the envisaged framework of the proposed system.

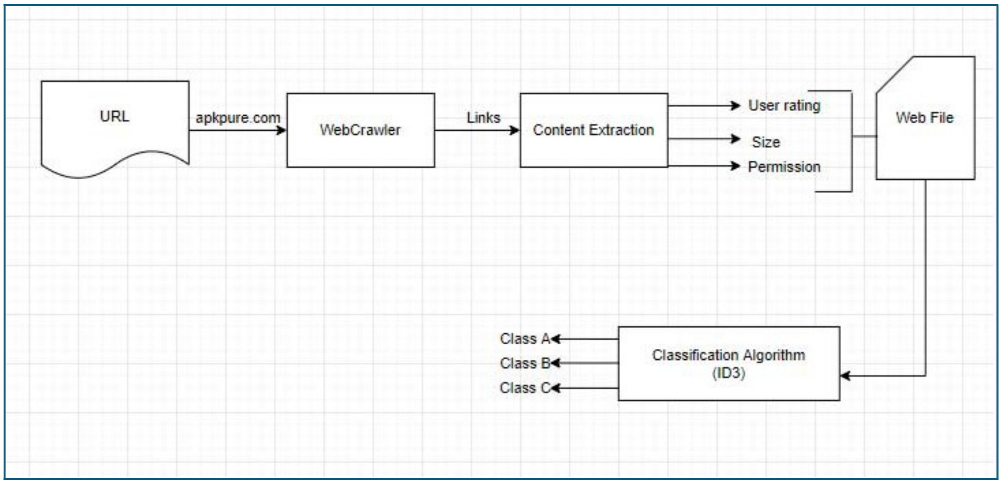

Figure 1 depicts the suggested system architecture.

a. Crawling apkpure.com is utilized to collect the dataset. A user inputs a website's address (URL). We employ a Web crawler to index the application, extract links to relevant material, and verify all external linkages.
b. The crawler module is responsible for gathering all the site's links, which it then stores in two local files before filtering. One is for connections directly to the base pages, while the other is for links to the basic pages themselves.
c. Size, user ratings, and permissions are the three criteria we utilize after Content Extraction. We do this with the help of scrappy and other software libraries for feature extraction. To extract the necessary features from each app, the crawler will provide the URLs to the extraction module.
d. The information will then be recorded in a structured file. Relational databases are used to store the gleaned information.
e. Data is sorted and used in the categorization process might be stored in the cloud.

III. ARCHITECTURAL REASONING

Utilizing a web crawler, we extract the requisite feature sets from the Android market. A Web Crawler is a sophisticated software application that performs the crucial task of verifying the integrity of hyperlinks and meticulously cataloguing websites along with their interconnected links. The Search Engine has devised a sophisticated web crawler with the purpose of expediting users' access to the specific content they seek. When presented with a compilation of Uniform Resource Locators (URLs), a web crawler shall proceed to traverse each individual link and procure all digital material that may be discovered within said destinations. Systematically identifies and validates each hyperlink present on the webpage, subsequently appending it to the compilation of suggested websites for further exploration, meticulously documented in a well-organized repository. Web crawling, commonly referred to as web spidering, entails the automated extraction of information from the vast expanse of the

World Wide Web. The website undergoes the process of crawling and indexing. The URL of the application is collected and stored within files, subsequently employed as a parameter for the purpose of extracting the data. It is imperative to devise a proficient algorithm that can effectively classify sets of features and subsequently draw conclusive classifications.

The efficacy of the system is enhanced proportionally to the augmentation of the quantity of information at hand. We possess the privilege of accessing the application data set procured from the esteemed online platform known as apkpure.com. The preliminary measure in the procedure entails conducting a comprehensive examination of the designated website [3]. A novel module has been devised to streamline the process of navigating the website. Python is widely employed for the creation and implementation of modules. The employment of crawled files is subsequently observed in the procedure of feature extraction. The extant body of literature pertaining to recommendation systems has regrettably neglected to thoroughly investigate the prospective future applications that may arise from the classification of novel applications utilizing the derived approach. One must not overlook the significance of the selection of the classification algorithm as an integral element within the system [4,5,6]. To effectively classify the applications at hand, it is imperative that we employ machine learning methodologies. Web scraping is a method employed to amass copious quantities of data from various websites. Web scraping is a sophisticated method employed to autonomously amass copious quantities of data from the vast expanse of the internet. The data present on the websites exhibits a lack of coherence and structure. The practice of web scraping proves to be highly advantageous in the realm of data acquisition, as it allows for the systematic collection and storage of information in a more structured and orderly manner. Python has been employed as a tool for the implementation of web scraping techniques.

IV. ARCHITECTURAL REASONING

Utilizing a web crawler, we extract the requisite feature sets from the Android market. A Web Crawler is a sophisticated software application that performs the crucial task of verifying the integrity of hyperlinks and meticulously cataloguing websites along with their interconnected links. The Search Engine has devised a sophisticated web crawler with the purpose of expediting users' access to the specific content they seek. When presented with a compilation of Uniform Resource Locators (URLs), a web crawler shall proceed to traverse each individual link and procure all digital material that may be discovered within said destinations. Systematically identifies and validates each hyperlink present on the webpage, subsequently appending it to the compilation of suggested websites for further exploration, meticulously documented in a well-organized repository. Web crawling, commonly referred to as web spidering, entails the automated extraction of information from the vast expanse of the

World Wide Web. The website undergoes the process of crawling and indexing. The URL of the application is collected and stored within files, subsequently employed as a parameter for the purpose of extracting the data. It is imperative to devise a proficient algorithm that can effectively classify sets of features and subsequently draw conclusive classifications.

The efficacy of the system is enhanced proportionally to the augmentation of the quantity of information at hand. We possess the privilege of accessing the application data set procured from the esteemed online platform known as apkpure.com. The preliminary measure in the procedure entails conducting a comprehensive examination of the designated website [3]. A novel module has been devised to streamline the process of navigating the website. Python is widely employed for the creation and implementation of modules. The employment of crawled files is subsequently observed in the procedure of feature extraction. The extant body of literature pertaining to recommendation systems has regrettably neglected to thoroughly investigate the prospective future applications that may arise from the classification of novel applications utilizing the derived approach. One must not overlook the significance of the selection of the classification algorithm as an integral element within the system [4,5,6]. To effectively classify the applications at hand, it is imperative that we employ machine learning methodologies. Web scraping is a method employed to amass copious quantities of data from various websites. Web scraping is a sophisticated method employed to autonomously amass copious quantities of data from the vast expanse of the internet. The data present on the websites exhibits a lack of coherence and structure. The practice of web scraping proves to be highly advantageous in the realm of data acquisition, as it allows for the systematic collection and storage of information in a more structured and orderly manner. Python has been employed as a tool for the implementation of web scraping techniques.

V. METHODS AND ALGORITHMS

The suggested method extracts features using a web crawler. Crawling for data extraction is the primary technology used in the selection process. User ratings, permissions, and mobile application size are necessary components of the system. The collected information is saved in files. For more accurate categorization. This approach is beneficial because it singles out underappreciated apps by focusing on certain aspects while maintaining the most important ones from the users' perspective [7,8,9]. Feature extraction from apkpure.com entails the following steps:

Pseudocode:

a. The user starts by providing the input as "https://apkpure.com/".
b. The user wants to create a directory named "apkpure" and add two text files inside it.
c. To perform the task, a queue and a spider need to be created. The objective is to retrieve the header of the content with the attribute's "type" set to "text/html" and "bytes" set to "utf-8".
d. After completing the previous step, the next task is to conduct a search for the tag "a" and the attribute "href".
e. The process involves adding items to a queue, checking for redundancy, and adding files to the queue.
f. One approach to achieve the desired outcome is to develop a program that utilizes web scraping techniques to create multiple spiders. These spiders can be designed to crawl through websites and extract relevant information such as links. The extracted links can then be appended to files for further analysis. It is important to implement a redundancy check to ensure that duplicate links are not included in the files.

VI. WEB SCRAPPING ALGORITHIM

The process involves gathering and compiling all the app links, which are then utilized for extracting relevant features. The user's text describes a process where links are provided to a web scraping program, which then collects data from the source file. The user suggests importing important modules such as pandas, NumPy, and seaborn. Iterate through each link within the directory where we have stored the data obtained from crawling. Utilize a suitable library to extract the features. The user has provided the text.

a. Title Extraction

It's a process that involves extracting the title from a given text or document. The text between the h1 tags is extracted. The user's text describes the conversion of words into a pandas data frame or any other module. The path of the title is recognized and acknowledged.

b. Rating Extraction

The ratings are extracted from the tags span class="average" and /span. The user requests to convert data into a data-frame. The user's text describes the path of the rating element in an HTML document. The rating element is located within the following elements: div.details-ratings →div.rating-info→span.rating →span.average

c. Permissions Extraction

The user is referring to the process of extracting permissions from the content section of a webpage, specifically between the div class="content" tags. The user is requesting to convert data into a dataframe. The permissions' path is documented as: div.describe →divdescribe →div.description

d. Size Extraction

The user is referring to extracting permissions from HTML code using the span class "fsize". Create a data frame from the given data. The trajectory of the title is recognized as: a.da →span.fsize

VII. CONCLUSION

This article presents a comprehensive exposition on the potential incorporation of app reviews, access levels, and file sizes within a recommendation system. The practice of web crawling is employed to acquire crucial information encompassing user ratings, permissions, and sizes. There has been a surge in the burgeoning fascination surrounding the realm of smartphone security features. Additional characteristics can be utilization of machine learning methodology algorithm, as its a decision tree algorithm. It offers a comprehensive array of features that are highly advantageous for the task of classification.

REFERENCES

[1] Viljanac V., Ranking of Facebook friends based on user profiles, Faculty of Electrical Engineering and Computing, Zagreb, June 2012.

[2] Adomavicius G., Tuzhilin A, Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions, IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 6, 2005, pp. 734-749

[3] Jisha R C, Ram Krishnan, Varun Vikraman: Mobile Applications Recommendation Based on UserRating and Permission, Department of Computer Science Applications, Amrita School of Engineering, Amrita VishwaVidyapeetham, Amritapuri

[4] Ani R., Augustine, A., Akhil, N. C., and Dr. Deepa Gopakumar O. S., RandomForest Ensemble Classifier to Predict the Coronary Heart Diseases Using Risk Factors, in Proceedings of the International Conference on Soft Computing Systems, 2016

[5] Dr. Deepa Gopakumar O. S., Ani R., Sasi, G., and Sankar, R., Decision Support system for diagnosis and prediction of Chronic Renal Failure using RandomSubspace Classification, in 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Jaipur, India, 2016

[6] HarryKurniawan, YusepRosmansyah, BudimanDabarsyah: Android Anomaly Detection System Using MachineLearning Classification - The 5th International Conference on Electrical Engineering and Informatics 2015

[7] Xin Su , Dafang Zhang , Wenjia Li , Wenwei Li : Android App Recommendation Approach Based on NetworkTraffic Measurement and Analysis - 20th IEEE Symposium on Computers and Communication (ISCC)

[8] JovianLin, Kazunari Sugiyama, MinYen Kan, Tat Seng Chua: New and Improved- modeling versions to improve apps recommendation - SIGIR '14: Proceedings of the 37th international ACM SIGIR conference on Research development in information retrieval

[9] Cao, Hong Lin, Miao. (2017): Mining smart phone data for app usage prediction and recommendation: A survey pervasive and mobile Computing. 37. 1-22. 10.1016/j.pmcj.2017.01.007.d

[10] YogeshWanjari, SanjayNagpure, GokulChute, YogeshwariKamble, Inclusion of efficient rules in PRISM Algorithm for Data Classification, March 2019, International Journal of Computer Applications