Proceedings of the Fourth International Conference on Computing Methodologies and Communication (ICCMC 2020)
IEEE Xplore Part Number:CFP20K25-ART; ISBN:978-1-7281-4889-2

Mobile App Recommendation System Using Machine learning Classification

Jisha R C, Amrita J M, Aswini R Vijay, Indhu G S
Department Of Computer Science And Applications
Amrita Viswa Vidhyapeetham
Amritapuri
India.
jisha@am.amrita.edu,indhuambadi720@gmail.com,jmamrita9@gmail.com,aswini.r.vijay@gmail.com

***Abstract*—The introduction of new ideas with mobile applications can bring great change to people around the world. Nowadays Thousands of apps are developed to satisfy different needs of people such as for doing jobs, transactions, entertainment etc.and distributed over the Internet. So most of the existing app stores available might face difficulties for recommending a particular app to a particular user. So there is a need for recommending apps for the users according to their personal preferences and various other limitations. We made a mobile application recommendation system with ratings, Size, and Permission as parameters and we will recommend suitable apps to the user by evaluating these parameters. Here we are using Apkpure.com which is one of the famous android application markets and also makes use of Web Crawler which helps in collecting information about the website and helps in validating hyperlinks. After that by using the Clustering Algorithm, applications are grouped or clustered based on Popularity, Permission and Security aspects. This paper aims to provide a simple recommendation system without compromising rating, size and Permission aspects.**

***Index Terms*—WebCrawler,Mobile,Security.**

I. Introduction

Since the last decade, android applications are gaining wide usage owing to the easiness of building and updating them. Generally, applications are developed to satisfy the different needs of users. We cannot claim that all the applications which we are using highly secured.Some of them are under the category of malware which will affect ourselves by making system down or misusing our personal details etc.. so it is very necessary to identify the malware apps and also ensure that if it is safe or not. Here comes the role of the Android Application Recommendation System. We are developing an Android App Recommendation System for recommending safe apps from the play store. It helps to classify the apps based on some parameters and suggesting the credibility of apps. We are using apkpure.com which is one of the famous android application markets for collecting the dataset by crawling. we use web crawler which is used for indexing websites and the links related to them, and also helps in validating the hyperlinks. The main challenges we faced -is collecting required data for a recommendation, dataset for list of apps and comparison, Difficulty in accessing details from the play store.

II. System Architecture

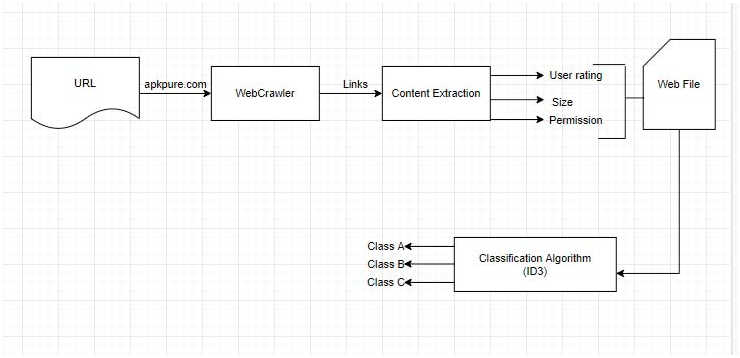

fig. 1. The proposed system architecture

- apkpure.com is used for collecting the dataset by crawling. The URL of the website is given as input. Here, a module program is developed for the purpose
- Here we use Web crawler which indexes the application and it extracts the links get content from the link and it also helps in validating hyperlinks. This is done by spider module which collects all the links from the site with filtering later and stores in two local files. The one for base links and the other one for the links from the base links
- After Content Extraction, We take three parameters i.e., size, user ratings, and permissions. For this we use software libraries like scrappy for the essential feature extraction. All the links collected by crawling are given to the extraction module so that the needed features are extracted from each app following the links.
- Then it will be stored in a well-defined document. The extracted data are stored as relational database.
- Here, we are using the ID3 Algorithm which is a classifying algorithm. A cloud is used for storing of information during classification.

III. Background Study

A. Web Crawler: The essential feature sets are obtained from the android market using a web crawler. Web Crawler is a program that collects content from the web and is used for indexing websites and links related to

them and also helps in validating hyperlinks. Web crawler is created by the Search Engine, which allows the users to find the requested pages very quickly. Once the Web crawler is given a list of URLs to checkout, it will visit each site and also download content from it. and identify and validates all the hyperlinks on the website and add them to the list of URLs to visit and it is stored in a well-defined document. This extraction of pieces of information from websites is called web spidering or web crawling. Crawler indexes the website. The URL of the applications is collected and stored in files later which are passed as parameters for extracting the necessary data. Classification Algorithm: An efficient algorithm should be devised to classify feature sets and finalizing the classification. Here it is ID3 Decision Tree Algorithm.

IV. Related Work

The more the data, the more efficient the system is. We have a data set of apps from the website apkpure.com. First, the website is being crawled [1]. We have written a module for crawling through the pages of the website. The module is written in python. Later the crawled files are used for feature extraction.

The previous works on recommendation did not deal with the future scope of classification of new apps based on the system derived. We have included an efficient way of expanding the scope of the preciseness of recommendation by including various parameters.

Selection of Classification algorithm is another important part of the system[2],[3],[4]. After getting the data we need to classify the apps by machine learning techniques. We have chosen the ID3 Decision Tree Algorithm. It gives us the best attribute of features to classify. Hence that attribute decides the feasible range of the given app. Web scraping is used to collect large information from websites. Web scraping is an automated method used to extract large amounts of data from websites. The data on the websites are unstructured. Web scraping helps collect these unstructured data and store it in a structured form. We have implemented web scrapping through python.

V. Methods and Algorithms

The proposed system uses the web crawler for feature extraction. The major technology for implementing selection is by Data extraction through crawling. The requirements for the system are user ratings, permissions, and the size of mobile applications. Extracted data are stored in files. The ID3 algorithm is adapted for reasonable classification. The advantage of this method is that it takes particular features of apps, which recognizes underrated apps while keeping the essential user-relevant parts[5],[6],[7]. These are the process involved in the extraction of the features from apkpure.com:

1.Begins with giving input as 'https://apkpure.com/'

2. Create a directory apkpure and two files inside(.text files).

3. Create queue and spider,then get the header of the content type="text/html" and bytes="utf-8".

4.After this, we have search for tag="a" and attribute ="href".

5. Then add to queue and check for redundancy and then add to queue files.

6. Create multiple spiders, links and append to files checking for redundancy

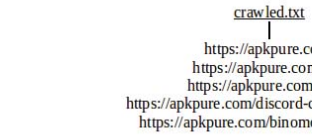

fig. 2. The crawled file with the base links

fig. 3. The queue file with the liks from each base links

VI. Extraction of features using web scrapping

We collect all the links of apps and use these for feature extration.The links are passed on to a web scrapping program that collects the datas available in the source file .Import important modules like pandas,numpy, seaborn etc. Pass each link from the directory that we stored the crawled data .Get the featues using a suitable library. Here it is Beautiful Soup.

A. Title Extraction

We extract the text between the tags h1 and /h1 using Beautiful Soup. The words are converted to pandas (or any other module) dataframe. The path of the title is noted

as: html →body →div.main.page-q →div.left →div.box →dl.ny-di.ny-dl.n →dd →div.title-like →h1

B. Rating Extraction

We extract the ratings between the tags span class=”average” and /span. Convert into dataframe. The path of the rating is noted as: html →body→div.main.page-q →div.left →div.box →dl.ny-di.ny-dl.n →dd →div.details-ratings →div.rating-info →span.rating →span.average

Table I: The table of user ratings of 10 apps

App	Ratings
Whatsapp	9
Instagram	9.3
Youtube	8.2
Forza Street	8
Subway Surfers	9.3
MX Player	9.6
LuluBox	7.5
Video Player	6
FX Motion	10
PUBG Mobile Kr	8.4

C. Permissions Extraction

We extract Permissions between the tags div class=”content” and /div. Convert into dataframe. The path of the permissions are noted as: html →body→div.main.page-q →div.left →div.box →div.describe →divdescribe →div.description →div.content

Table II : The table of app size of 5 apps

App	Permissions
Whatsapp	19
Instagram	20
Youtube	17
Forza Street	9
Subway Surfers	1.5

D. Size Extraction

We extract Permissions between the tags span class=”f size” and /span. Convert into the data frame. The path of the title is noted as: html →body →div.main.page-q →div.left →div.box →dl.ny-di.ny-dl.n →dd →div.ny-dowm →a..da →span.fsize

Table III : The table of app permissions of 10 apps

App	Size
Whatsapp	37.6
Instagram	28.3
Youtube	35.9
Forza Street	1433.6
Subway Surfers	94.1
MX Player	30.4
LuluBox	13
Video Player	7.4
FX Motion	30.6
PUBG Mobile Kr	1683.4

VII. ID3 Decision Tree Algorithm

The obtained feature set is used to classify the app. We assign a risk class based on the analysis on the app by various sources. For each feature, we classify it into class or range. We first obtain the information of the entire data set. Then we calculate the information of each feature. Later we calculate the information gain in each feature. The one with the highest information gain is the most probable feature to classify the app[8],[9],[10],[11],[12]. Each feature is given a range of classes after the extraction.

input:F=f1,f2,f3,....,fn
output:Class
foreachRows,do
v1=p$_i log_2 p_i + v1$
foreachfeature,do
v2=p$_i log_2 p_i + v2$
foreachDataset,do
array=v1-v2
Get the maximum value in the array

App	Ratings	Permissions	Size	Risk Class
App1	A	B	B	A
App2	A	A	C	B
App3	B	A	A	B
App4	C	B	A	C
App4	B	B	C	C
App5	A	C	B	A
App6	C	A	B	C
App7	A	C	A	A
App8	A	C	A	B
App9	C	A	B	C
App10	B	A	B	C

fig. 4. The proposed ID3 for our system

Our data for the classification consists of feature set of about 10 apps.

VIII. Range of features

A. Ratings

Ratings can range from 0 to 10. Hence the classification

- Class A: 8-10
- Class B: 5-7.9
- Class C: 0-4.9

B. Size

Sizes are ranged as

- Class A:less than 20 MB
- Class B: more than 20.1 MB and less than 50 MB
- Class C: more than 50 MB

C. Permissions

Permissions can be of different aspects. We give points to different permissions depending on the importance.

- 2 Points for Photos/Medias/Files, Location, Storage, Phone , Camera
- 1.5 points for Microphone, Device ID and call info, Contacts, Wifi connection, Device and App history, SMS

– 1 point for others

So the total scores can be classified as Class A: 0-5 Class B: 5.1-10 Class C : more than 10.1

IX. Future Enhancement

To develop the system more precisely and with very low entropy, as a future enhancement, we consider more features of the app. As a part of future enhancement, we aim to perform analysis on Android app traffic consumption and collect the detailed network traffic components such as unwanted traffic . When a new app comes it's feature is compared with the already existing data sets of apps and classified accordingly.

X. Result Analysis

Our data for the classification consists of feature set of about 10 apps.

Table IV: The data extracted combined to be classified by ID3 classification

App	Size	Ratings	Permissions	Risk Class
Whatsapp	B	A	C	A
Instagram	B	A	C	A
Youtube	B	A	C	A
Forza Street	C	A	B	B
Subway Surfers	C	A	A	B
MX Player	B	A	B	A
LuluBox	A	B	A	C
Video Player	A	B	C	C
FX Motion	B	A	B	A
PUBG Mobile Kr	C	A	B	A

The target classification can be class A, class B and class C.
Gain(Size)=0.115
Gain(Ratings)=0.649
Gain(Permissions)=0.151
Since, rating attribute has maximum gain it can be the root node.
Entropy(Size | A)=0.65
Entropy(Size | B)=0
Entropy(Size | C)=0
Entropy(Permissions | A)=1
Entropy(Permissions | B)=1
Entropy(Permissions | C)=1
Therefore, if Ratings=A ll Size=A , the result is A and if Ratings=B ll Size=A, the result is B

XI. Conclusion

This paper gives a simple outlook on how ratings, permissions, and size of applications can be used for an application recommendation system. The essential features such as user ratings, permissions and sizes are extracted using the method of Web crawling. The security aspect of smartphones is gaining popularity. To enable future improvement, this study focuses on the model of android apps by considering the selected feature and more features can be added for more efficient and accurate results of app recommendation.

XII. Reference

[1]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

[2]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

[3]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

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

[5]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)

[6]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

[7]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

[8]Osisanwo F.Y, Akinsola J.E.T., Awodele O, Hinmikaiye J. O., Olakanmi O.,Akinjobi J., Supervised Machine Learning Algorithms: classification and comparison , International Journal of Computer Trends and Technology (IJCTT) Volume 48 Number 3 June 2017

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

[10]Kotsiantis S B, Zaharakis I and Pintelas, P, Supervised machine learning: A review of classification techniques, emerging artificial intelligence applications in computer engineering

[11]V Karthick, Dr Sumathy, A Similarity study of Techniques in data mining and big data, International Journal of Computer Trends and Technology (IJCTT)

[12]Moath Alluwaici, Ahmad KadriJunoh, Farzana KabirAhmad, Raed Alazaidah, Mohamad Farhan MohamadMohsen, Open Research Directions for multi label learning, IEEE Symposium on Computer Applications Industrial Electronics (ISCAIE)