iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 **Issue:** VI **Month of publication:** June 2020

DOI: http://doi.org/10.22214/ijraset.2020.6273

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue VI June 2020- Available at www.ijraset.com

Mobile App Success Prediction

Akanksha Singh[1], Drishya Tyagi[2], Brijesh Yadav[3], Apurv Gupta[4], Kumud Alok[5]
[1, 2, 3, 4, 5]*Department of Computer Science, Dr. A.P.J. Abdul Kalam Technical University. Uttar Pradesh, Lucknow, India*

Abstract: ***Google play store has thousands of apps and several thousand apps are added to its list every day. The competition is so high that it is difficult for a developer to find out whether he is working in the direction of making a successful app or not. The success of an app can be determined by factors like ratings, number of installs and reviews. In this paper we have applied Exploratory Data Analysis to discover relationships among features of an app to predict which app will succeed. Data from google play store was used to train three different models to predict the success of the app- Random Forest, Support Vector Machine and Linear Regression.***

Keywords: Mobile app, Google Play Store, Apple App Store, Exploratory Data, Support Vector Machine, Random Forest and Linear Regression

I. INTRODUCTION

App Stores are not a forgotten achievement as Apple App store and Google Play Store came merely a decade back in 2008. Apps were originally intended for productivity assistance such as email, calendar and contact databases, but the public demand for apps caused rapid expansion into other areas such as mobile games, factory automation, GPS and location-based services, order-tracking, and ticket purchase, so that there are now millions of apps available.[1] These app stores provide us with numerous amounts of entertaining facilities with just few clicks in our Pcs. From that time till now we have only seen the rise in the demands of the apps these stores provide to users. Digital technology market is greatly influenced by mobile apps growth over the years. Mobile applications are enormous and profitable business. Looking at the market of mobile apps we can see that there is a huge difference between the success rate of iPhone mobile apps and google mobile apps. Studies show that the iPhone has over 1.84 million apps in their store and Google play Store has over 2.57 million apps, as of 2019. Looking at the figures, we can conclude that there is a huge difference in numbers of apps between the two. On the other side, even though there are less iPhone users compared to Android users, Apple stores still make 75% more revenue than the Google Play Store, as studies show in 2016. Resting aside the revenue factor, we observe that Google App Store has always had a greater number of downloads, popularity, apps available, more free apps, and different genres of apps. With the growth of the mobile app market, the number of app developers has also significantly increased. With each passing day, the demand for more advanced and updated apps keeps on rising and the number of apps available in the app stores increase by thousands. This many apps confuse the user into considering which app they can use. Developers too loose track and only one question remains, what will make an app a success? Creating and developing apps without considering current trends and previous data only builds up the number of apps available and burns a hole in the pocket of the developer. We have made it a little simpler for the users and the developers, to consider which or what app to use and what kind of apps they can deploy, respectively. Developers can see and understand what changes or advances they can bring in their ideas and applications that will help them give a more efficient app for users and a successful app for them.

In this study, features extracted from the data made available on Google's Play Store website is used as input to different models. Each model understands the features of a given application, and interesting observations about their behaviour are discussed. In the next step, we filter the data and remove the factors that can result in ambiguity or errors. We came across a paper [2] that explained what causes an app to fail and what are bad apps. So, we perform Exploratory Data Analysis to understand the features of apps that result in better success of an app. We use machine learning models like Random Forest Model, Support Vector Machine and Linear Regression Model to find the correlation and understand the data.

There have been some seminal studies [2][3][4][5]. We hope to have brought some advancement in the field for the upcoming research.

II. LITERATURE REVIEW

In paper [4], the author mined and processed the data available by the Google Play Store. He extracted the features available and used three models to come to final conclusions. He also trained the data under those models. Using the key features, number of installations and average user rating, allowed the author to come closer to predictions. The author used the linear model for prediction of the average rating for the whole presented data. Principal component analysis, by using inputs of Generalized Linear Model and Linear Regression, presented strong patterns between the features.

The author worked on the description presented by the creators of the application that presented the conclusion that about thirty-five percent of all successful applications contain the stem “photo” and thirty-one percent of all successful applications contain the steam “share” somewhere in their description. Successful in concluding what genres of applications interest most of the users. For his Future Work author suggested using revenue to predict and for success metrics.

In another paper [14], the author reviews the ratings for the applications. It shines light on the fact that how store-ratings after reaching a certain value does not affect the overall store rating even after users rate it. But on the contrary the following above is not observable in the store-rating. It also highlights that an updated app rating depends on their versions. Thus, came the idea of rating the versions. Version ratings can be calculated by calculating the available store-rating. The idea behind it was to help the developers develop an app even when it reached a prominent value. They share and advise the idea of displaying the current version ratings for the app. Similarly, in paper [13], the author works around the feature- review. The author quotes the unfruitful combination of reviews and rating. The author furthermore works on this and builds multiple systems that detects the inconsistency between the above two features. The system mainly revolves around the approaches of Naïve Bayes Classifier, Decision Tree, Decision Stump, Decision Table and other few Machine Learning Algorithms and Deep Learning Approaches. They held several surveys in the process to learn the opinion of the user and also the developers regarding the above two features. Concluding the likes of the author, the end users and developers agreed to the idea of matched relevancy between the review and the rating of the app.

Following the research around the same sentiment analysis, in this paper [15], provides an understanding that there is a difference between the starred ratings on the app store and the reviews by the users. So, the author proposes a new rating system which might remove the ambiguity and difference created by the user between the reviews and the ratings. The author states that the user is interested in downloading an app according to their rating. The author sums up the problem in 2 parts- the ambiguity and the biases. The author's proposed system will perform a sentiment analysis on the descriptive review by the user and then generates a numeric rating. In this way the final rating generated will be the average of the ratings. This proposed system is said to reduce the confusion of the users and allow them to have a final rating based on both review and star rating. This paper concludes a very strong relationship between the review and the ratings and works to help the user get the better and best app according to their likes.

In another paper [17], the authors bring up the question of why an app is not desired by the user and answers the reason behind their failure. It can be observed and seen that not all reviews can be treated and considered under the study. Such reviews create noise and add up to the number of errors. The authors worked to remove such reviews resulting in reduction of noise in the dataset and providing a better performance in sentiment analysis. It results in generating more accurate polarity value. Li proposed WisCom, a system that is able to analyse at least ten million of user ratings and comments in the app markets on three different levels. This system is said to be firstly accomplished in identifying inconsistency in reviews and then looked for the reasons why people do not like a particular app and how the reviews change over time according to user demands. Their proposed system and the valued research provide an overwhelming walk through the most fascinating concerns. It generates techniques for summarizing and mining the reviews, which will help the end users to choose the best app that too without going through the previous user descriptive reviews. This paper allows the user to understand what went wrong in the demand or the usefulness of the app and what features and areas they might consider to update their app to reach out to more user and move their app ranking to a higher place.

Questioning the failure same as above, in the paper [16], the authors comment on the idea of reviews and consider it the more quantitative value and that combination of star rating with the text review will result in better quantitative estimation of service satisfaction rating. Their average rating for the dataset was 3.6 so the author placed the rating above 3.6 to positive sentiment and below it to negative sentiment. He used some common and popular Machine Learning Algorithms like Naïve Bayes and SVM along with the most unusual but effective learning algorithm consisting of two processes: capturing semantic similarities and modelling after sentiment. He used seven different training set sizes that predicted ratings with the respective sentiment.

All the papers mentioned above helped us to get all the ideas that we incorporated in our work, which made our work more organized and eventually helped us to produce a better analysis of the features that are present in our dataset.

III.ALGORITHM/ MODEL USED

A. Random Forest Model

Random forests or random decision forests are ensemble learning methods for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set. Assembling is nothing but a combination of weak learners (individual trees) to produce a strong learner.[7]

Random Forest Algorithm: The following are the basic steps involved in performing the random forest algorithm:

1) Pick N random records from the dataset.
2) Build a decision tree based on these N records.
3) Choose the number of trees you want in your algorithm and repeat steps (i) and (ii).
4) In case of a classification problem, each tree in the forest predicts the category to which the new record belongs.[3] In the end category with majority is defined as a new record. In short, we can conclude that Random Forest forms multiple decision trees and cumulates them together to get better and accurate results/ predictions. The trees we obtain are basic and easier to understand than before but they tend to be noisy.

B. Support Vector Model

Support-vector machines are supervised learning models with associated learning algorithms that analyse data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier.[8]

Support Vector Machines acts as one of the best approaches to data modelling. They combine generalization control as a technique to control dimensionality.[9] SVM chooses the decision boundary that maximizes the distance from the nearest data points of all the classes. An SVM doesn't merely find a decision boundary; it finds the most optimal decision boundary. The most optimal decision boundary is the one which has maximum margin from the nearest points of all the classes. The nearest points from the decision boundary that maximize the distance between the decision boundary and the points are called support vectors as seen in Figure. The decision boundary in case of support vector machines is called the maximum margin classifier, or the maximum margin hyperplane. [3]

C. Linear Model

Linear regression is a linear approach to modeling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). The case of one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regression. [10] It is also called the Linear Regression Model that assumes a linear relationship between the input variables (x) and the single output variable (y). More specifically, that y can be calculated from a linear combination of the input variables (x).

D. Decision Tree Model

The decision tree model is a classification-based model that helps the user to segregate the features of the data and allows them to understand better. Under its hood comes the Decision Tree Classifier. The Decision Tree Classifier performs the same function of classification of data. It forms a tree in which the internal nodes are labelled by features. The root and internal nodes perform the test functions to separate the data that has different characteristics. The model breaks down the data into smaller and smaller subsets while at the same time a decision tree is increasingly developed. This model allows to draw a pattern between the data i.e., shows the variation of similar data.

E. XGBoost Algorithm

This algorithm is considered to be dominating in now times. It is a further implementation of the gradient boosted decision trees designed for speed and performance. The algorithm is created by Tianqi Chen. It comes under the hood of the Distributed Machine Learning Community. XGBoost has model features like: Gradient Boosting, Stochastic Gradient Boosting and Regularized Gradient Boosting. Its system features are: Parallelization, Distributed Computing, Computation of very large dataset that don't fit memory and Cache Optimization. This algorithm results in giving a good execution speed and astonishing model performance. This model is also capable of building random forests.

F. K-Nearest Neighbor Algorithm

It is a simple supervised learning algorithm that is used to solve both- classification and regression. This algorithm delivers similar data that lies in close proximity. If we break it down to simple words, it states that similar data is near/ close to each other. The K in the KNN algorithm stands for nearest K neighbors, which is selected by the user on the basis of assumption. To select the right K for the data it is advised to run KNN a few times with different K values that results in a reduced number of errors and enhances the accuracy.

IV. DATA COLLECTION

An app store (or app marketplace) is a type of digital distribution platform for computer software called Applications, often in a mobile context. Apps provide a specific set of functions which, by definition, do not include the running of the computer itself. Complex software designed for use on a personal computer[6]. In July, 2008 iPhone brought the concept of app store to users. Not so long after Google launched its Google play store in the same year, proving many android apps on the same platform. However, Google and iPhone are not the only one providing various user-friendly apps to customers. There are others like Samsung Galaxy Apps, Huawei App Store, Sony Apps, Amazon Appstore and many others. While there are many app stores it is right to say they are not as dominant as the iPhone and Google app stores.

We picked to work on the Google Play store as it provides larger content, different variety and they have a greater number of users as compared to iPhone leaving aside the revenue factor. Google makes data available about its applications on http://play.google.com. We extracted the raw data from the official Play Store. There were 10,841 of the data available to us on which we worked further and applied various algorithms.

Attributes	Description
Category	Category or genre
Content	Rating Suitable content for the audience
Installs	Number of installs e.g. 1+, 5+
Rating	Average rating of the app
Last Updated	Date of last update
#Ratings	Total number of ratings
Rating Distribution	Number of 5-star ratings
Type	Free or Paid
Review	The comment text entered by a user
Size	Size of app in k (kilobytes), M (megabyte), . . .
App	Name of the app

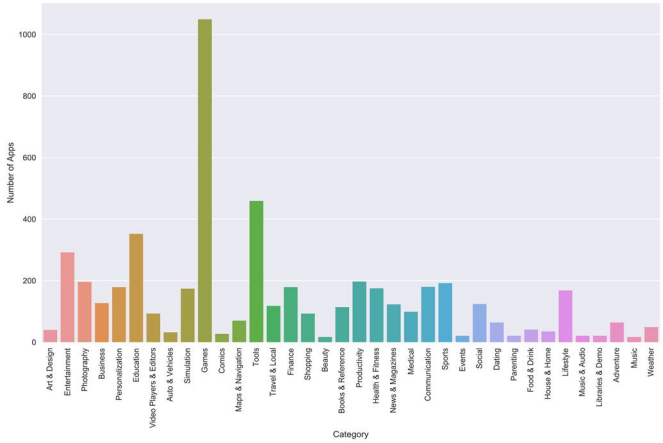


Fig 1 Number of apps in each category

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue VI June 2020- Available at www.ijraset.com

V. EXPLORATORY DATA ANALYSIS AND DATA PREPROCESSING

EDA is the first and foremost important step in every prediction. Exploratory Data Analysis is used to analyse our data and understand it better. Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypotheses and to check assumptions with the help of summary statistics and graphical representations [18]. EDA allows users to select the best predicting algorithms and models for the data by looking at the analysis.

Before working on our data, it is important to understand it like by understanding the data-type of features, range of specific categories/ features and also finding the number of missing/null/redundant values in each feature. We firstly try to reduce down all the null data that could result in creating the noise in our prediction. We install Pandas-Profiling that will help us in EDA analysis. As the data is not well-adjusted yet and still need to undergo several analysis steps pandas-profiling guides better with such adjustments. Instead of just giving you a single output, pandas-profiling enables its user to quickly generate a very broadly structured HTML file containing most of what you might need to know before diving into a more specific and individual data exploration.[19] It helps speed up in EDA.

We summarize our data and see that there are over 33 different categories of Apps available on the Google-app store. Ranking the highest in number of count is the category of Family-based apps followed by Games. We reduce the categories of our data which are lesser in count and as we believe will not affect our prediction or accuracy that much but instead might result in creating noise and errors. We removed over 19 categories of apps with the remaining only 14 apps that have higher count.

We identify the numerical and categorical features. We take our features like Rating and convert it to a float value to help in smooth processing further and remove all the noises in the Rating data. We do the same with feature Price and change it to int value. We remove any noise causing values or the part of values by splitting the data and removing it. We perform the same with the rest of data, which needs to be changed and remove the noise.

Now that we have successfully removed noise and changed our data for better processing, we can plot graphs and understand correlation between features to find the independent variable. From the box plot we understand that features: Installs and Reviews are highly correlated.

Next, to understand our data better, we perform various graph functions to see the followings: frequency of average rating between 0-5, categories with most installs, categories with most reviews, categories with most space consumption.

Moving on to our Data Preprocessing, Data Preprocessing is a Data Mining Technique which is useful to change and modify our data into a well understood format. The remaining errors left before are reduced and removed here. We understand the behaviour of the data. We did the following steps: Scaling and cleaning of feature- Installations, cleaning categories into integers, filling empty sized with NA, dropping of unrelated and in necessary items, cleaning of feature- Genres and Content Rating classification

VI. PREDICTION ALGORITHMS

Firstly, we applied the Random Forest Model that allowed us to form decision trees and segment or data into the smallest bits possible. This allowed us to draw a clear pattern among the similar data and process them together as a whole.

Next, we performed SVM, again a classification model that chooses decision boundaries to maximize distance from nearest data points. Next up was the Linear Model, which allowed us to divide the success of an app on the basis of the value of x. when x=1 i.e., success. We used the model to measure the performance factor but it was soon for us to define the success of an app so we discarded this model.

We have divided our apps in two categories to help us understand their demand better: Paid and Free. We found what frequency of both the categories were getting updated over the years, when was the highest rate, in months, when both categories were updated, rating of both categories by the defined age groups, free app content rating by the defined age groups, rating difference between both the categories, android version frequency in both categories, frequency and number of installed apps in both categories, rating over content by the defined age groups, rating over android version.

VII. TRAINING AND TESTING

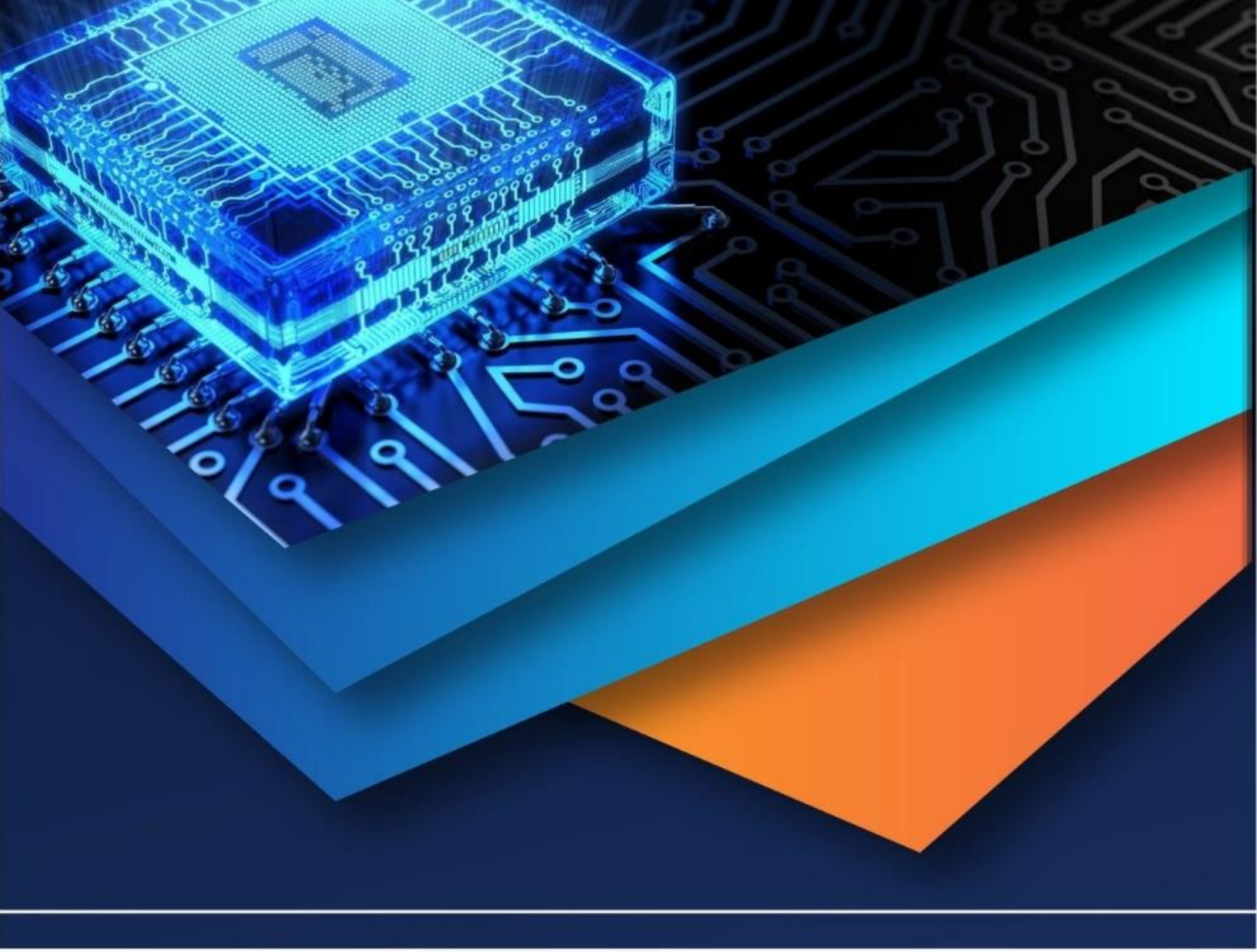
Training and testing is considered as one of the last procedures in prediction analysis. It is understood that training or testing is never done on the actual data but instead on the dummy data and then seen but to what accuracy the both the data's fittings. We use features: Category Rating and Rating to create our dummy data as we are now working on the feature- Rating. We drop the rest of the data. Importing the sklearn directory allows us to use its selection model and preprocessing model. We box plot the following models: Decision Tree Classifier, SVM, Random Forest, XGBoost and KNN to find their accuracy rates as: 70.49%, 80.34%, 75.59%, 79.99% and 77.26%, respectively.

VIII. CONCLUSIONS

We draw the conclusion that feature like Rating and Content Rating results in better prediction of what can be described as success for a mobile app in today's time when there are hundreds of similar apps on the same platform.We have defined a success parameter for an app based on the number of installs, distribution of ratings 4 and 5 relative to the overall number of ratings and installs to rating ratio. Using features other than the ones used to create that parameter, we created a model that predicts it with 80.34% accuracy.

REFERENCES

[1] Mobile App, https://en.wikipedia.org/wiki/Mobile_app [1]
[2] Venkata N Inukollu, Divya D Keshamoni, Taeghyun Kang and Manikanta Inukollu. "Factors Influencing Quality of Mobile Apps: Role of Mobile App Development Life Cycle" in International Journal of Software Engineering & Applications (IJSEA), Vol.5, No.5, September 2014. [2]
[3] Abdul Mueez, Khushba Ahmed, Tuba Islam and Waqqas Iqbal, "Exploratory Data Analysis and Success Prediction of Google Play Store App," B.Sc. Engineering in CSE thesis, BRAC University, Dhaka - 1212, Bangladesh, December 2018. [3]
[4] Tuckerman, C.J., "Predicting Mobile Application Success," Dec. 2014. [4]
[5] Federica Sarro, Mark Harman, Yue Jia, and Yuanyuan Zhang, "Customer Rating Reactions Can Be Predicted Purely Using App Feature", CREST, Department of Computer Science, University College London, London, UK.[5]
[6] (2002) The IEEE website. [Online]. Available: http://www.ieee.org/ [6]
[7] M. Shell. (2002) IEEEtran homepage on CTAN. [Online]. Available: http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEEtran/ [7]
[8] FLEXChip Signal Processor (MC68175/D), Motorola, 1996. [8]
[9] "PDCA12-70 data sheet," Opto Speed SA, Mezzovico, Switzerland. [9]
[10] A. Karnik, "Performance of TCP congestion control with rate feedback: TCP/ABR and rate adaptive TCP/IP," M. Eng. thesis, Indian Institute of Science, Bangalore, India, Jan. 1999. [10]
[11] J. Padhye, V. Firoiu, and D. Towsley, "A stochastic model of TCP Reno congestion avoidance and control," Univ. of Massachusetts, Amherst, MA, CMPSCI Tech. Rep. 99-02, 1999. [11]
[12] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, IEEE Std. 802.11, 1997. [12]
[13] Liu, B. (2012). Sentiment analysis and opinion mining. Synthesis lectures on human language technologies.[13]
[14] Luiz, W., Viegas, F., Alencar, R., Mourão, F., Salles, T., Carvalho, D., Gonçalves, M. A., and Rocha, L. (2018). A feature-oriented sentiment rating for mobile app reviews. In Proceedings of the 2018 World Wide Web Conference on World Wide Web, pages 1909–1918. International World Wide Web Conferences Steering Committee. [14]
[15] Islam, M. R. (2014). Numeric rating of apps on google play store by sentiment analysis on user reviews. In 2014 International Conference on Electrical Engineering and Information Communication Technology. [15]
[16] Grover, S. (2015). 3 apps that failed (and what they teach us about app marketing). [online] https://blog.placeit.net/apps-fail-teach-us-app-marketing/. [16]
[17] Fu, B., Lin, J., Li, L., Faloutsos, C., Hong, J., and Sadeh, N. (2013). Why people hate your app: Making sense of user feedback in a mobile app store. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 1276–1284. ACM. [17]
[18] Prasad Patil (March 23, 2018). What is Exploratory Data Analysis? [Online]. In blog, Available: https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15. [18]
[19] Lukas Frei (Apr 26, 2019). Speed Up Your Exploratory Data Analysis With Pandasa-Profiling [Online]. In blog, Available: https://towardsdatascience.com/speed-up-your-exploratory-data-analysis-with-pandas-profiling-88b33dc53625. [19]



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY