ISSN 1816-6075 (Print), 1818-0523 (Online)
Journal of System and Management Sciences
Vol. 12 (2022) No. 6, pp. 300-314
DOI:10.33168/JSMS.2022.0619

Predicting Mobile Apps Performance using Machine Learning

Sabreen Abulhaija, Shyma Hattab, Ahmad Abdeen, Wael Etaiwi

King Talal School of Business Technology, Jordan
Princess Sumaya University for Technology, Jordan

abu20208045@std.psut.edu.jo

Abstract. The number of mobile applications is expanding daily along with new upgrades. It might be difficult for developers to identify the key factors that affect an application's rating. This paper aims to help application developers in improving their decisions and adding new features to mobile applications. 2.24 million Records from the Google Play Store were used in the research to create a dataset for the purpose of predicting the performance of mobile applications. The dataset included different features such as rating count, minimum installs, category, maximum installs, and price; add support, in app purchasing, total size and content rating. Four machine learning methods were used: decision trees, naive bayes, ANN, and random forest. The random forest outperformed the other algorithms in terms of precision, recall, and f-measure. Additionally, the top three features influencing the performance of the app were category, minimum installations, and rating count. This research offers practical recommendations for tech entrepreneurs and app developers to better comprehend the features affecting the performance of the mobile apps and give them the ability to predict the app performance based on their assumptions of the number of installs, price, category, and other features, which will help them properly, plan their development initiatives to avoid failure.

Keywords: machine learning, mobile applications, rating performance, predictive modeling

1. Introduction

The Mobile Application (AKA: Mobile-App) industry has been developed radically in the last ten years. These applications provide users with unlimited functions that make users' life more entertaining, comfortable, and excited, by delivering services such as online shopping, food ordering, gaming, health management, etc. However, some of these applications are not useful or are not working properly. Hence, users are always looking for an application with a high rating and positive reviews to decide whether to download this application or not. Reference to the recent statistics, more than 2.5 billion people are using a smartphone, and more than twelve million developers have developed applications for these smartphones, also developers were access to more than 5 million apps on electronic stores such as Apple, Google, and Amazon, gaining over 200b downloads (Dhinakaran et al., 2018).

This trend is followed by a growing number of mobile software businesses delivering a massive number of mobile applications. Specifically, there are two giant platforms in the market provided by Google Play Store (GPS) and Apple Store (iOS). Mobile applications are offered for free or subscription-based. The app stores have a massive number of free applications, making the market very competitive and providing many alternatives for users. Additionally, as a part of customer service management, these two platforms allow their users to evaluate the applications and provide reviews and opinions (Dhinakaran et al., 2018).

Application rating represents all reviews received from users' responses. However, not all applications have excellent ratings, and users would instead download applications with top ratings since they expect them to be more effective and of higher quality. Mobile applications on electronic stores receive on average 22 ratings daily, which depend on the popularity of the application and might reach to few thousand daily. Moreover, only one-third of these reviews are useful to analysts and developers (Dhinakaran et al., 2018). These reviews and feedback play a vital role in both the developers' business and users' experience in this competitive world (Sarro et al., 2018).

The previous literature analyzed the rating issues through testing datasets that include thousands of records and a limited number of features and this research used a large dataset with 2.24 million records to predict the performance of the apps.

This research used the Google Play Store dataset to predict the mobile apps application performance using different features that affect the application ratings, and it has been applied machine learning techniques to utilize these features.

The motivation to introduce this research, as it will add value to businesses and developers before launching their mobile applications, by giving them background about mobile stores rating analysis and an overview about the rating analysis categories of mobile applications, available datasets in the market, machine learning performance on these datasets and different models. Moreover, this research will help developers to understand which attributes are affecting user satisfaction to take them into account during the mobile applications development phase.

This research is organized as the following: the second section includes the literature review. The third section includes the methodology, fourth section illustrates the experiments and results, the fifth section includes the recommendations and finally, the sixth section shows the conclusion.

2. Literature Review

The researchers reviewed the literature to identify the research papers that analyze mobile applications performance using a set of predefined keywords including Mobile Apps, Apple Apps, Android Apps, and Google Play Store.

The research of Magar et al. (2021) used the GPS dataset to classify the overall popularity of an app and use the number of installs as the measure. They used six machine learning (ML) algorithms; Logistic Regression (LR), Random Forest (RF), Stochastic Gradient Descent (SGD), KNN, SVM, DT, and the experimental results showed that the SVM classifier produced best results. The models were based only on the top five important features external to the app and they did not include features internal to the app such as the software features and performance of the app.

The study of Sarro et al. (2018) analyzed 11,537 apps from BlackBerry and Samsung World app stores, they used Natural Language Processing (NLP) technique to obtain from current app types, feature data that capture some of the operations of these apps. They also used Case-Based Reasoning (CBR) to predict the rating of the apps by relying on the claimed features. The results indicated that the ranking of 89% of those apps can be predicted 100% accurately. The findings of the study could help in the requirements engineering of the app stores and provide chances to encourage the needs induction procedure for developers.

The study of Bashir et al. (2019) proposed a modern structure that offers developers an efficient approach to effectively discover in the competitive Mobile-App industry. By comparing the predicted ranking and downloads numbers with the original dataset. They analyzed the GPS dataset using ML techniques to predict rating and download number before going live with the app on the store to help the developers assess their work. The result showed that SVM and KNN can deliver better accuracy than RF.

The research by Suleman et al.(2019) which aimed to predict rating on GPS using a real-time dataset collected in 2018 contained 10839 records and 8 attributes with the following names; app name, number of reviews, downloads volume, size of the app, categories, content ranking, android version and with a class named as rating. They used several ML techniques including DT, LR, SVM, Naïve Bayesian (NB), K-Means, KNN, and Artificial Neural Networks (ANN). Their methodology contained many processes, including collecting, cleaning, and feature reductions.

The authors used MATLAB for the data visualizations. The result showed that DT has the best results in making rating predictions among other techniques.

Daimi et al (2019) pointed out that there are many users who are not technically oriented and do not have much deep knowledge about the mobile applications, therefore they depend on the applications rates to choose the most appropriate one, hence the aim of their paper research was to predict user rating of mobile applications using iOS dataset. The authors downloaded the dataset from Kaggle contains 7197 rows and 16 attributes. They used 7 ML techniques including SVM, NN, RF, M5 Rules, LR, and Random Tree, all of them employed by using WEKA (a ML software). The result showed that the RF has yielded the best results for predicting the user rating for the iOS dataset among other techniques.

The proposed paper of Umer et al. (2021) intended to forecast the numeric ranking of GPS apps using ML classifiers including gradient boosting classifier (XGB), RF, gradient boosting classifier (GBM), extra tree classifier (ET), and the extreme AdaBoost classifier (AB). The dataset was aggregated from the GPS utilizing the Beautiful-Soup (BS) web scraper contained 658 records and included attributes such as: “App_category, App name, App_id, App_review, and App_ rating”. The dataset for this paper was semi-structured which requires several preprocessing techniques to analyze it including features selection. The result showed that GBM and ET have produced the most exact numeric rating predictions. Future work included applying a Deep Learning (DL) algorithm for numeric rating prediction.

The research of Zhang and Kim (2020) reviewed the influence of financial service characteristics on use intention through customer satisfaction with mobile fintech and the study revealed that to acheive the statisfaction, greater degree of flexibility is needed for different customer products.

The study of Kayalvily et al. (2022) aimed to predict GPS apps rating using the simplest ML technique which is DT. The dataset was collected from GPS in 2019. They used KDD Methodology to understand and analyze the data and used Tableau for visualization. They concluded that the price and number of downloads have a strong influence on the user ratings.

Sadiq et al. (2021) aimed in their research to predict numeric reviews and ratings of GPS apps by using DL approaches including Bidirectional Long Short-term Memory (BiLSTM), Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), Long short-term memory (LSTM), and Gated Recurrent Unit (GRU). The dataset collected from GPS contained fourteen different sorts of mobile apps including 658 records that was scrapped by using BeautifulSoup (BS) web scraper, which is a Python package for parsing HTML and XML documents. Different attributes of the dataset are used; “App_id, Appname, App_category, App_review, and App_rating”. The dataset was unstructured and noisy which required preprocessing like cleaning, removing duplicated data, etc. The results showed that the CNN gave the most numeric rating exact predictions than others with the results of 89% recall and 82% precision

In their research paper, Sandag et al. (2020) created a prediction model using user rating (dataset 2019) for Android apps on GPS utilizing the KKN algorithm. The results showed that education is the most reviewed and book & reference is the highest-rated and dating category is the lowest. On the other hand, KNN performed well in predicting Android applications based on the fivefold cross-validation dataset.

Additionally, the research of Mahmud et al. (2019) proposed a category prioritizing method by studying the rankings and the reviews of the apps, in order to recognize the most important features to consider for a future better release. The authors used the NB and the J48 decision tree classifier, where NB had better results and the results also showed that utilization of resources and application performance had the highest priority rank.

Moreover, the paper of Lengkong and Maringka (2020) used RF, KNN, GB, and DT to identify the most influential characteristics of high rated apps in GPS, such as the features of Size, installs, reviews, types (free vs paid), rating, category, content rating, and Price. The research summarized that the GB has the greatest performance with 100% accuracy, 100% Precision, and 100% Recall, it was also noted that the factors install and reviews are the most influential in predicting high-ranked apps.

The study of Mahmood (2020) ensured that users seek to download applications that have a high rate as they considered high quality and will be more satisfied. His research question was to know which aspects influence the apps’ ranking in GPS. The dataset collected from Kaggle contains 10,840 records, including the following attributes: app name, current version, app id, installs, reviews, size, category, rating, type, content rating, price, last updated, and android version. He used RF, Support Vector Regression (SVR), LR, and Pearson Correlation to know which attributes influence the rating of the app. RF defines the significance of all the factors and their impact on the rating, and it shows that the number of evaluations, genre, app size, and character count in the name are the most important variables than others. The results for SVR showed that content rating and word counting in the name are the most influential factors of rating.

The paper of Dhinakaran et al. (2018) proposed an active learning technique to decrease the human effort in the review analysis. The proposed app review classification approach utilizes three active learning tactics based on uncertainty sampling. It was found that active learning with comparing to a randomly selected training dataset, generates a much greater prediction accuracy using multiple scenarios.

According to the literature reviewed, the number of mobile applications is enormous and most of the users depend on the mobile applications' rating in their decisions, scholars analyzed the rating issues through testing datasets that include thousands of records, to the best of our knowledge this paper is among the first to use a large dataset with 2.24 million records to predict the performance of the apps.

3. Methodology

The methodology of this research is based on Data Mining and Knowledge Discovery in Databases (KDD) methodology, which is a set of processes for discovering useful knowledge from collecting data, data cleaning, noise detection and extracting useful patterns from raw data which leads to making discovered patterns understandable (Kayalvily et al., 2022).

This part illustrates a dataset, preprocessing steps, data visualization and classification models.

3.1. Dataset Discerption

The dataset contains 23 different attributes and contains 2.24 million of records, it is derived from Google Play Store. Different attributes of the dataset involve App Name, App Id, Category, Minimum Installs, Maximum Installs and Price, etc. and Rating attribute as a target. Table 1 shows in details the data types for all attributes.

Table 1: Attributes data types

Attributes Name	Data Type
App Name	String
App Id	String
Category	String
Rating	Number
Rating Count	Number
Installs	Number
Minimum Installs	Number
Maximum Installs	Number
Free	Boolean
Price	Number
Currency	String
Size	Number
Minimum Android	Number
Developer Id	String
Developer Website	String
Developer Email	String
Released	Date
Last Updated	Date
Content Rating	Number
Privacy Policy	String
Ad Supported	Boolean
In App Purchases	Boolean

3.2. Data Preprocessing

To prepare data for the training and analysis phase; inconsistency, incomplete and noisy data need preprocessing. Noise is something that does not hold useful information in prediction and will consume a lot of time in the next steps in the analysis process (Sadiq et al., 2021). Moreover, inconsistency, duplicate and incomplete data often reduces the accuracy of the models. Literature shows that preprocessing plays a major role in accurate prediction and leads to good quality results (Sadiq et al., 2021). Therefore, various steps of preprocessing have been performed on the dataset before starting the analysis phase as shown in details in table 3.

In the first step, for the Installs column, the symbol of “+” has been removed by using a string manipulation node, substr (join) expression and replacing the symbol of “+” with null. Moreover, the inconsistency has been resolved for the size column by eliminating the “M” for million and “K” for thousand and replaced them with zeros using column filter and string manipulation nodes then regexReplace expression for each one. Also, the round double node has been applied on the size attribute to convert the type from double to an integer. In addition, all the stop words (i.e., and up from Minimum Android Column), punctuation, and duplicate commas have been deleted using the string manipulation node and regexReplace expression.

Additionally, by using the column filter node, some irrelevant attributes have been excluded like App-Id, Developer-Id, and Developer-email, as it will affect the analysis process and increase the overhead of training by increasing time and decreasing the accuracy.

Moreover, it is important to mention that the analysis phase and results have been affected greatly by missing values and outliers, therefore, the statistics node has been applied to the dataset to know the number of the outliers and the columns that included missing data. Hence, it has been noticed that some attributes contained missing values i.e., Rating, Rate Count, Minimum Installs and Maximum Installs, etc. and they have been handled by replacing them with mean for the columns that have not contained outliers, median for the columns that have outliers and most frequent value for sting column like a minimum android.

Furthermore, some feature engineering techniques have been applied to the dataset. According to Jones et al. (2019), future engineering is the process of manipulating and converting raw data into preferred features using statistical or machine learning techniques. Also, Anderson et al. (2013) mentioned that feature engineering is a machine learning technique that helps in creating new features, for both supervised and unsupervised learning, which does not exist in the current training set, in addition, it can assist in transforming the data type either from numerical to categorical using binning or numerical to categorical using label encoding technique in a purpose of simplifying the dataset, enhancing the result of the model and increasing the accuracy, recall and precision measures. In this dataset, some of the features have been transformed from one representation to another such as the number of installs and size from categorical to numeric. Finally, the class attribute "Rating" has been converted from numeric to categorical and has been categorized into 5 categories using the numeric binner node.

Table 2: Preprocessing tasks

Column Name	Column Value Before Preprocessing	Task Required	Node Name & Expression for Preprocessing	Column Value After Preprocessing
Installs	10+	Removing the Symbol of "+"	String Manipulation Node and substr (join) Expression	10
Size	10 M 50 K	Removing the Value of K and M and Replacing them with Zeros	String Manipulation, Column Filter and regexReplace Expression	10000000 50000
Size	1000.5	Converting the type from Double to Integer	Round Double Node	1001
Minimum Android	7.1 and up	Deleting the Value of "and up"	String Manipulation, regexReplace Expression and String to Number Node	7.1
Developer Website	NA	Need to aggregate it	Group By	NA
App-Id, Developer-Id, Developer-email, etc.	These columns were included in the dataset	Need to exclude it	Column Filter Node	Excluded from the Dataset
Rating	Included Missing Date	Replacing the Missing date with the Mean	Missing value Node	Missing data has been excluded from the dataset
Minimum Installs	Included Missing Date	Replacing the Missing date with the Median	Missing value Node	Missing data has been excluded from the dataset
Minimum Android	Included Missing Date	Replacing the Missing date with the Most Frequent Value	Missing value Node	Missing data has been excluded from the dataset
Rating	Numeric Data	Converting the	Numeric Binner	Five Categories as

	Type	data type from number to categorical and Creating Five categories	Node	follows: Category 1: -∞ to 1.1 Category 2: 1.1 to 2.1 Category 3: 2.1 to 3.1 Category 4: 3.1 to 4.1 Category 5: 4.1 to ∞

4. Experiment

Machine learning is the field of study that combines statistics and computer science to introduce algorithms that get more efficient when they are subject to relevant data rather than being given explicit instructions (Charbuty & Abdulazeez, 2021). The experiment of this research is based on supervised machine learning algorithms which are used to learn a model that creates the same labeling for the data offered and functions well on unseen data.

The following evaluation measures were used; recall, precision, and F-measures respectively to test the effectivity of the models. The recall determines the number of positive category predictions made from all positive instances in the data set (i.e., the total of the True Positives (TP) and False Negatives (FN), however, the precision defines how many predictors of the positive category belong to the positive category (i.e. the sum of True Positives (TP) and False Positive (FP), but the F-Measure provides one score that assesses both concerns of precision and recall in one single value. (Kynkäänniemi et al., 2018; Tatbul et al., 2018).

Moreover, the experiment of this research includes four machine learning technique; Decision trees, Naïve Bayes, ANN and Random Forest.

Decision Trees; which is a tree-based technique in which any path beginning from the root is described by a data separating sequence until the outcome at the leaf node is achieved. Decision tree classifiers are known for their enhanced view of performance outcomes. Because of their strong precision, optimized splitting parameters, and enhanced tree pruning techniques (Charbuty & Abdulazeez, 2021). In the model of this research, the evaluation measures for the decision tree; the precision, recall and F-measure were 59%, 62.5% and 61% respectively.

Additionally, Naive Bayes algorithm is a probability classifier which calculates the probability by counting the frequency and combinations of values in the data set and it uses Bayes ‘theorem and assumes that all the variables are independent of each other while considering the value of the class variable (Saritas & Yasar, 2019), the evaluation measures for the naive bayes; the precision, recall and F-measure were 59%, 67% and 63% respectively.

ANN is a common machine learning technique derived from the biological neural network in the human brain, ANN sends the weighted values of each artificial neuron as an output to the next layer after processing the inputs from neurons in the previous layer (Saritas & Yasar, 2019). According to the paper of Abiodun et. al (2018) ANNs are helpful models for classification, clustering, pattern recognition and prediction in many disciplines such as agriculture, science, medical science, education, finance, and management. Additionally, they mentioned that ANNs applications include; modeling, classification, pattern recognition, and prediction, the evaluation measures for the ANN; the precision, recall and F-measure were 58%, 64% and 61% respectively.

Random forest is a machine learning algorithm consisting of decision trees, each of which provides a vote for a specific class. Combining a large number of trees in a random forest leads to reliable predictions, while a single decision tree may overfit the data. Random forest has many advantages such as; producing high accuracy rates, processing data with a large number of features, robust to mixed variable types and building random forests is quite fast (Devetyarov & Nouretdinov, 2010), the evaluation measures for the random forest; the precision, recall and F-measure were 64%, 69% and 66% respectively.

In this research each algorithm has been performed in a different aspect due to the parameters configured with a cross validations of (10) partitioner. Two major nodes have been used (DT learner and DT Predictor) with a pruning method of (MDL) and (Gini Index) quality measure which led to better performance when it comes to categorical data. However, in Naive Bayes algorithm two major nodes have been used (NB Learner and NB Predictor) utilizing default probability and minimum standard deviation, which usually lead to better performance when it comes to numeric data. And although ANN is a common machine learning technique that mimics the biological neural network in the human brain, an important remark from this research noted that it is not essentially that complex ML models such as ANN can permanently beat the precision of simpler models such as DT and it can perform less. Finally, Random Forest worked well in this research as a classification model, a split criterion of information gain ratio has been used.

5. Results and Discussion

As indicated above, the algorithm had approximately close performance results according to the evaluation measures and the random forest overperformed them. A summary of the evaluation measures is indicated in table number 4.

Additionally, according to the decision tree algorithm, the priority features were; rating count, minimum installs, category, maximum installs, price, add support, in app purchasing, total size and content rating.

When selecting a machine learning technique, no specific algorithm performs well in all situations. Hence, some of them perform well with huge datasets and some perform well with high dimensional datasets, accordingly, it is essential to choose and evaluate the model's effectiveness based on the nature of each dataset.

In this research where the dataset contains 23 different attributes from which 9 of them are numbers and 8 strings and contains 2.24 million of records, the random forest technique worked well and outperformed all tested techniques, as it easily dealt with this research big dataset which contains categorical and numerical features.

Moreover, although the dataset of this research was large, random forest was able to perform the prediction in quickly and train the dataset in short time. Also, it handled the outliers efficiently by binning them. It was also noted that the deeper the tree, the extra complex the decision rules and the fitter the model. However, RF deals very well with over fitting issue (Sakr et al., 2017). Noting that overfitting the data is a major drawback of DT technique.

However, the result of Naive Bayes algorithm was not desirable as this algorithm does not perform well in this research dataset because most of the data are categorical and it is usually deal better with numerical data.

It was also noted that ANN took long time in the training phase and data development of this research and requires hardware/ processors with parallel processing power which is also computationally expensive, in addition to unexplained behavior of the network.

Tab. 3: Evaluation Measures

Algorithm / Statistics	Precision	Recall	F-Measures
Decision Tree	59%	62.5%	61%
ANN	59%	67%	63%
Naive Bayes	58%	64%	61%
Random Forests	64%	69%	66%

6. Conclusion

The use of mobile phones is increasing all the time. These phones have become increasingly vital and beneficial in all parts of the lives, including social and business sides. The number of mobile applications on the Google Play Store is significantly growing every single day. However, not all the applications succeed and become widespread among the users as each mobile application should go through systematic and organized procedures to know if it will be popular and successful or not, these processes need specific knowledge of successful mobile applications’ attributes and rating prediction (Kayalvily et al., 2022).

The competition between different mobile applications is expanding dramatically, hence every single minute, new upgrades and editions have been developed and increased. This increase makes it more difficult for the users, particularly those who are not technologically minded, to determine which applications have to install, used, and will serve his/her needs (Bashir et al., 2019). Therefore, the proposed rating prediction model of this research can aid the users by knowing the maximum number of downloading, price, and rating, which mobile application is the best and will serve his/her requirements and avoid priceless applications.

Moreover, this proposed rating prediction model will assist the companies specifically the small and the starter ones to provide them with the probability of successful development of mobile applications and if it will be profitable or not. In addition, knowing the priority features will help the developers to understand the customers' needs, attitudes, trends, and desires and try to develop an application that has characteristics based on the users' needs.

Furthermore, users play a critical role in determining future technology and trends. Therefore, an in-depth understanding of the users' needs and knowing the features that will affect the rating, assist the mobile applications companies to reduce the chance of failure and able to achieve target goals.

This research used Google play Store dataset to predict the mobile apps application performance using different features such as; rating count, minimum installs, category, maximum installs, price, add support, in app purchasing, total size and content rating. Four machine learning algorithms were used; decision trees, naïve bayes, ANN and random forest, and according to the evaluation measures (precision, recall and f-measure), the random forest outperformed the other algorithms. Moreover, the top three features affecting the app performance were; rating count, minimum installs and category.

This research provided practical recommendations for tech entrepreneurs and app developers to better understand the features affecting the app performance and provide them the potential to predict the app performance based on their assumptions of the number of installs, price, category and other features, which in turn will help them properly plan their development initiatives to avoid failure.

To the best of our knowledge, this paper is among the first to use a large dataset with 2.24 million records to predict the performance of the apps, while other papers discussed the issue with a dataset that includes thousands of records and limited number of features.

Future work may consider technical attributes of the apps as this study focused only on the features related to customers' experience, it could also consider applying the same models on other app stores such as; Apple Store dataset and Huawei AppGallery. Additionally, this research applied machine learning algorithms, future work may consider applying deep learning or ensemble methods.

7. References

Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., & Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11), e00938

Anderson M, Antenucci D, Bittorf V, Burgess M, Cafarella M, Kumar A. (2013). Brainwash: A data system for feature engineering. In: *Proceedings of the Conference on Innovative Data Systems Research. 2013 Presented at: CIDR'13*; January 6-9, 2013; Asilomar, CA, USA

Aralikatte, R., Sridhara, G., Gantayat, N., & Mani, S. (2018, January). Fault in your stars: An analysis of android app reviews. *In Proceedings of the acm india joint international conference on data science and management of data*, 57-66

Bashir, G. M. M., Hossen, M. S., Karmoker, D., & Kamal, M. J. (2019, December). Android apps success prediction before uploading on google play store. In 2019 *International Conference on Sustainable Technologies for Industry 4.0* (STI), 1-6, IEEE

Charbuty, B. & Abdulazeez, A. (2021). Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, 2(01), 20-28

Daimi, K. & Hazzazi, N. Using apple store dataset to predict user rating of mobile applications

Devetyarov, D. & Nouretdinov, I. (2010, October). Prediction with confidence based on a random forest classifier. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, 37-44, Springer, Berlin, Heidelberg

Dhinakaran, V. T., Pulle, R., Ajmeri, N., & Murukannaiah, P. K. (2018, August). App review analysis via active learning: reducing supervision effort without compromising classification accuracy. In 2018 *IEEE 26th international requirements engineering conference (RE)*, 170-181. IEEE

Jha, N. & Mahmoud, A. (2019). Mining non-functional requirements from app store reviews. *Empirical Software Engineering*, 24(6), 3659-3695

Jones, M., Johnson, M., Shervey, M., Dudley, J. T., & Zimmerman, N. (2019). Privacy-preserving methods for feature engineering using blockchain: review, evaluation, and proof of concept. *Journal of medical Internet research*, 21(8), e13600

Kayalvily, T., Denis, A., Mohd Norshahriel, A. R., & Sarasvathi, N. (2022). Data analysis and rating prediction on google play store using data-mining techniques. *Journal of Data Science*, 2022(01)

Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., & Aila, T. (2019). Improved precision and recall metric for assessing generative models. *Advances in Neural Information Processing Systems*, 32

Lengkong, O., & Maringka, R. (2020, October). Apps Rating Classification on Play Store Using Gradient Boost Algorithm. In 2020 *2nd International Conference on Cybernetics and Intelligent System (ICORIS)* (pp. 1-5). IEEE

Luiz, W., Viegas, F., Alencar, R., Mourão, F., Salles, T., Carvalho, D., & Rocha, L. (2018, April). A feature-oriented sentiment rating for mobile app reviews. *In Proceedings of the 2018 World Wide Web Conference*, 1909-1918

Magar, B. T., Mali, S. & Abdelfattah, E. (2021, January). App success classification using machine learning models. In 2021 *IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, 0642-0647, IEEE

Mahmood, A. (2020). Identifying the influence of various factors of apps on google play apps ratings. *Journal of Data, Information and Management*, 2(1), 15-23

Mahmud, O., Niloy, N. T., Rahman, M. A., & Siddik, M. S. (2019, June). Predicting an effective android application release based on user reviews and ratings. In 2019 7th *International Conference on Smart Computing & Communications (ICSCC)*, 1-5, IEEE

Martens, D. & Maalej, W. (2019). Towards understanding and detecting fake reviews in app stores. *Empirical Software Engineering*, 24(6), 3316-3355

Ongsulee, P. (2017, November). Artificial intelligence, machine learning and deep learning. In 2017 *15th International Conference on ICT and Knowledge Engineering (ICT&KE)*, 1-6, IEEE

Pratama, B. T., Utami, E., & Sunyoto, A. (2019, March). A comparison of the use of several different resources on lexicon based Indonesian sentiment analysis on app review dataset. In 2019 *International Conference of Artificial Intelligence and Information Technology (ICAIIT)*, 282-287, IEEE.

Rahman, M., Rahman, S. S. M. M., Allayear, S. M., Patwary, M., Karim, F., Munna, M., & Ahmed, T. (2020). A sentiment analysis based approach for understanding the user satisfaction on android application. *In Data Engineering and Communication Technology*, 397-407, Springer, Singapore

Ranjan, S., & Mishra, S. (2020, July). Comparative sentiment analysis of app reviews. In *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 1-7), IEEE

Sadiq, S., Umer, M., Ullah, S., Mirjalili, S., Rupapara, V., & Nappi, M. (2021). Discrepancy detection between actual user reviews and numeric ratings of Google App store using deep learning. *Expert Systems with Applications*, 181, 115111.

Sakr, S., Elshawi, R., Ahmed, A. M., Qureshi, W. T., Brawner, C. A., Keteyian, S. J., ... & Al-Mallah, M. H. (2017). Comparison of machine learning techniques to predict all-cause mortality using fitness data: the Henry ford exercIse testing (FIT) project. *BMC medical informatics and decision making*, 17(1), 1-15

Sandag, G. A., & Gara, F. (2020, October). Android Application Market Prediction Based on User Ratings Using KNN. In 2020 *2nd International Conference on Cybernetics and Intelligent System (ICORIS)*, 1-5, IEEE

Sarro, F., Harman, M., Jia, Y., & Zhang, Y. (2018, August). Customer rating reactions can be predicted purely using app features. In 2018 *IEEE 26th International Requirements Engineering Conference (RE)*, 76-87, IEEE

Saritas, M. M., & Yasar, A. (2019). Performance analysis of ANN and Naive Bayes classification algorithm for data classification. *International journal of intelligent systems and applications in engineering*, 7(2), 88-91

Soumik, M. M. J., Farhavi, S. S. M., Eva, F., Sinha, T., & Alam, M. S. (2019, December). Employing machine learning techniques on sentiment analysis of google play store bangla reviews. *In 2019 22nd International Conference on Computer and Information Technology (ICCIT*), 1-5, IEEE.11

Suleman, M., Malik, A., & Hussain, S. S. (2019). Google play store app ranking prediction using machine learning algorithm. *Urdu News Headline, Text Classification by Using Different Machine Learning Algorithms*, 57

Tatbul, N., Lee, T. J., Zdonik, S., Alam, M., & Gottschlich, J. (2018). Precision and recall for time series. *Advances in neural information processing systems*, 31

Umer, M., Ashraf, I., Mehmood, A., Ullah, S., & Choi, G. S. (2021). Predicting numeric ratings for google apps using text features and ensemble learning. *ETRI Journal*, 43(1), 95-108

Zhang, L., & Kim, H. (2020). The influence of financial service characteristics on use intention through customer satisfaction with mobile fintech. *Journal of System and Management Sciences*, 10(2), 82-94