Predict New App Quality By Using Machine Learning

Zhenghuan Zhong\textsuperscript{1}, Yuhang Bao\textsuperscript{2}, Shiyao Shen\textsuperscript{3}, Entong Zhou\textsuperscript{4}

\textsuperscript{1}Shanghai Guanghuan College, 200040, PRC
\textsuperscript{2}Xiamen University TAN KAH KEE College, 230000, PRC
\textsuperscript{3}Auburn University, 36832, USA
\textsuperscript{4}National University of Singapore, 11842, SG

Email of all the authors: 2262849816@qq.com; 18949890972@189.cn; szs0243@auburn.edu; e0424905@u.nus.edu

Abstract: Increasingly, individuals and companies are developing applications and selling them online. Competition between different companies has led to tens of thousands of applications being put on the market for users to choose from. There are numerous functions on the market to give app evaluation. For users, they do not need to contemplate how to select an app for a long time, because user comments and ratings can give them a clear instruction for user to select an app they need from abundant apps. As a result, we would like to explore the possibility. Hence, we think that we can make a prediction function for the market of whether we can make a prediction function for the app market, akin to a weather forecast or stock market forecast. Our group not only wants to know what the future of the app market is, but also want to find out why some high-quality apps are not getting the downloads they should. With the development of AR and VR technology, the future application will combine with these technology to bring user a immersive experience. In addition, high quality application without AR or VR are less competitive. We plan to answer these questions by using machine learning techniques.

1. Introduction
In the past 20 years, users have already started to become accustomed with all kinds of mobile phone apps, such as education, games, film, and social apps. With the popularization of the Internet, Internet users are blooming, manufacturers have also spotted business opportunities they launch app on different platforms, based on the popularity of the app (downloads, rating), some proverbial app such as Facebook can offer the company a huge economic profits [2].

Moreover, people tend to share their thoughts freely through comments on the internet - psychologists Moon Garrett pointed this out in 2017 [3]: people love to share their thoughts - which give others instruction about which apps to download.

In this paper, we surveyed data from 10,841 mobile applications (9660 after removing duplicates) with various features such as app’s size, free or paid. Our model will exclude duplicate values and special values by using our particular algorithms and plugging in data such as application properties and user review into the model to predict the possible rate when they be lunched. The higher the possible rate, the more popular it is.
2. Related Work

In recent years, numerous studies have utilized machine learning to form forecasts. For example, Meghawat and colleagues [4] utilize the SMP-T1 dataset and use visual and social features to foresee the popularity of social media photographs. Abbass et al. (2020) [5] many models (e.g. MNB, KNN, SVM) to train a Twitter dataset in order to anticipate social media violations. Chen, M., & Liu, X. (2011) [6] predict the popularity of songs by employing Classification And Regression Tree (CART). Yang, Y., Lin, Y., Su, Y., & Chen, H. H. (2007) [7] construct Typical music emotion classification (MEC) as multiple linear regression and vector regression problem to evaluate the prediction accuracy. Similar to these studies, our group wants to analyze preexisting data to predict its future.

3. Data Overview

We collected data in two steps: first, we look for a suitable website. Of the one we identified, the authors stated that “The website among the many open network data sets, and then we select the data set we want. Initially we were just looking around for the right data set, Play Store apps data has enormous potential to drive app-making businesses to success. Actionable insights can be drawn for developers to work on and capture the Android market!” This data set, taken from the Google play store, is originally from Kaggle. Our data set includes Category, Rating, Size, and among others.

4. Material and Methods

![Methodology overview](image)

Figure 1. Methodology overview

4.1. Dataset Extraction

We obtained the dataset from Kaggle, an online community for open datasets, scraped from the Google Play Store database. The dataset contains information on 10841 mobile apps (9660 after removing duplicates) with various features such as app name, last updated time, current version, genre, category, size, reviews, installs, type, price, content rating and android version.

4.2. Dataset Understanding

Preliminary data visualization and correlation analysis were conducted to understand the data distribution before selecting features. To predict the popularity of mobile apps, we chose “Rating” as the prediction goal: the higher the rating, the more accurate the prediction of the future.

4.3. Data Preprocessing

The dataset needed to be cleaned before data analysis as it contained multiple duplicates, missing values, and inappropriate data types.

4.3.1. Remove duplicates

In the dataset there are many of the same apps showing up numerous times with distinctive custom reviews number. We deleted all duplicate data, which decreased the app number from 10841 to 9660. Therefore, even though we deleted all duplicate values, the number of apps was still enough to analyze.

4.3.2. Replace null value

The following features contained missing values: Rating, Type, Content Rating, Current Version and Android Version. Since the Type and Content Rating are categorical rather than numerical and the average can only apply to the numerical object, using the mode or median is a more suitable way to replace the null value. We choose median to replace the missing values of Rating and use mode for the
other features.

4.3.3. Remove outliers

We used a Box-plot to display the data, and then removed outliers. Finally, we repeated operations to ensure that there are no outliers by using box-plot to see if it includes any possible outliers.

4.4. Choosing Feature:

There are 13 features in the Google app database. After understanding the meaning of each feature, we dropped the following feature types: App, Last Updated, Current Ver and Genre. Since the names of different apps are made up by different languages, it does not apply to general situations. Current version and Genres can be replaced by Android Version and Category, because they are the similar. The Last Updated Time could be better expressed through the Current Version feature, since the time date is usually not considered a feature. Therefore, our final features are Category, Reviews user, Size, Installs, Type, Price, Content Rating and Android Ver along with the target feature – Rating.

| Feature      | Description                                      |
|--------------|--------------------------------------------------|
| Category     | Category that the app belongs to, such as Family, Game, etc. |
| Reviews      | Number of user reviews at the time of scraping. |
| Size         | With the unit of bytes.                         |
| Installs     | Number of users downloads                       |
| Type         | Free or paid.                                  |
| Price        | Price of the app                               |
| Content Rating | Age group, such as ‘everyone’,'Mature 21+'   |
| Android Ver  | Minimum required android version for using.     |

Figure 2. Feature selected

5. Algorithms

We select the target column (Rating), and divide the training set and test set (30%, 70%).

5.1. Random forest

Among the foremost broadly utilized ensemble supervised learning procedure for predictive analytics. It consists of multiple decision tree classifications and outputs the mode of each trees to fix the overfitting problem that decision trees are likely to encounter [8].

| rating | precision | recall | f1-score | support |
|--------|-----------|--------|----------|---------|
| 1      | 0.00      | 0.00   | 0.00     | 2       |
| 2      | 0.03      | 0.20   | 0.05     | 20      |
| 3      | 0.12      | 0.40   | 0.19     | 324     |
| 4      | 0.95      | 0.78   | 0.86     | 5497    |
| 5      | 0.12      | 0.38   | 0.19     | 60      |

accuracy: 0.75 total: 5903

Fig.3 The result of using Random Forest model

5.2. Logistic Regression

Utilized to portray information and to clarify the relationship between one subordinate double variable and one or more ostensible, ordinal, interim or ratio-level free factors. Sigmoid function is used in mapping predictions [9].

\[ f(x) = \frac{1}{1 + e^{-x}} \]
5.3. Gradient Boosting
A sort of machine learning support. It depends on the instinct that the best conceivable following model, when combined with previous models, minimizes the in general forecast mistake [10].

| rating | precision | recall | f1-score | support |
|--------|-----------|--------|----------|---------|
| 1      | 0.00      | 0.00   | 0.00     | 0       |
| 2      | 0.00      | 0.00   | 0.00     | 0       |
| 3      | 0.00      | 0.00   | 0.00     | 0       |
| 4      | 1.00      | 0.76   | 0.86     | 5903    |
| 5      | 0.00      | 0.00   | 0.00     | 0       |

Accuracy: 0.76
Total: 5903

Fig.4 The result of using Logistic Regression model

5.4. Multi-layer Perceptron classifier (MLP)
Not at all like other classification calculations such as Naive Bayes Classifier, MLPClassifier depends on a basic Neural Network to perform the errand of classification [7].

| rating | precision | recall | f1-score | support |
|--------|-----------|--------|----------|---------|
| 1      | 0.00      | 0.00   | 0.00     | 15      |
| 2      | 0.03      | 0.17   | 0.06     | 30      |
| 3      | 0.06      | 0.41   | 0.11     | 152     |
| 4      | 0.97      | 0.77   | 0.86     | 5628    |
| 5      | 0.13      | 0.31   | 0.18     | 78      |

Accuracy: 0.75
Total: 5903

Fig.5 The result of using Gradient Boosting model

6. Results
Finally, we tried 7 models, including Random Forest Classifier, Decision Tree Classifier, Logistic Regression, MLP Classifier, AdaBoost Classifier, Bagging Classifier and Gradient Boosting Classifier, because all of these models are more accurate by contrast. As a result, four of them have over 0.75 accuracy.

7. Conclusion and Future works
There are 4 models accuracy are above 0.75 which is relatively and this is owing to that we artificially set the rating grade to 4 as the default, because the score of F1-score and recall is the highest when the rating is 4. Hence, it is feasible that there is some deviation between the practical results and the model, which we need to improve in the future.

We use these models to reach our objective: predict the possible rating of an app when it been put on the market. Machine learning can be extremely powerful, but there are still challenges in finding a suitable model, such as the requirement of numerous experience of machine learning and do a lot of search. It is very important to understand the data and choose the optimal dataset. If there is no complete understanding for every single model, it is impossible to make an effective model. Moreover, feature engineering is also difficult, because it requires some experience to which values can be discarded and which data need to be revised.
Overall, by using different data to train our model, the final accuracy in predicting app rating is stable at about 75%. In the future, the plan is to train on other data sets, such as iTunes data sets, to further improve the accuracy of our predictions.

References
[1]. Published by Patrick Charuza and 2 others on May 1, et al. “How Much Money Can You Earn With an App in 2019?” Fueled, 23 Sept. 2019, fueled.com/blog/much-money-can-earn-app/.
[2]. Moon, Garrett. “Why People Share: The Psychology of Social Sharing.” CoSchedule Blog, 26 Jan. 2017, coschedule.com/blog/why-people-share/.
[3]. Meghawat, M., Yadav, S., Mahata, D., Yin, Y., Shah, R. R., & Zimmermann, R. (2018). A Multimodal Approach to Predict Social Media Popularity. doi:10.31219/osf.io/z8t2b=
[4]. Abbass, Z., Ali, Z., Ali, M., Akbar, B., & Saleem, A. (2020). A Framework to Predict Social Crime through Twitter Tweets By Using Machine Learning, 2020 IEEE 14th International Conference on Semantic Computing (ICSC). doi:10.1109/icsc.2020.00073
[5]. Chen, M., & Liu, X. (2011). Predicting popularity of online distributed applications. Proceedings of the 2011 IConference on - IConference 11. doi:10.1145/1940761.1940859
[6]. Yang, Y., Lin, Y., Su, Y., & Chen, H. H. (2007). Music Emotion Classification: A Regression Approach. Multimedia and Expo, 2007 IEEE International Conference on. doi:10.1109/icme.2007.4284623
[7]. Livingston, F. (2005.). Implementation of Breiman’s Random Forest Machine Learning Algorithm. ECE591Q Machine Learning Journal Paper. Retrieved Fall, 2005.
[8]. Kleinbaum, D. G. (1994). Logistic regression: a self-learning text. Springer.
[9]. Lu, H., Karimireddy, S. P., Ponomareva, N., & Mirrokni, V. (2019). Accelerating Gradient Boosting Machine.
[10]. Marsland, S. (2014). The Multi-layer Perceptron. Machine Learning, 71-110. doi:10.1201/b17476-4