BAX-452 Final Project

"*Rating Prediction of Mobile Application using Machine Learning Techniques*"

Author:

Kexin Fu, Parth Patel, Ruihan Zhou, Yushan Liu

Mar.20th.22

Executive Summary

We obtained a dataset of information about various types of applications in the mobile app market from kaggle. The dataset contains features like app type, rating, number of reviews, size, installs, free or not, price, content rating, genres, last updated, current version, and android version.

We wish to predict the app's rating based on the above 13 features and identify features that are essential in order to app rating. Based on this prediction and the important features found, we can thus provide business value to software stores (like Google Play, App Store) and app developers. For software stores, they can use our predictions to give users potential high-score apps. For app developers, they can use the essential features we found to modify their apps to get more users' favorites.

In terms of data modeling, we first cleaned and encoded the dataset. We replaced NA with mean or median, and deleted a series of symbols that cannot be entered into modeling with complete formatting. The year, month, and day were encoded. After the first step of data preprocessing, we performed an exploratory data analysis. The distribution of each feature and the correlation between each feature were observed. Then we fitted the app rating using Lasso regression and random forest regressor. Based on this, the three most important features were found: number of reviews, last updated time and app size.

1. Background, Context, and Domain Knowledge

Starting from the year of 2010, the term ‘app’ has become a hot topic around the world. It stands for ‘application’ and is a software application/computer program for mobile devices.[1] Consumers often go to mobile app markets on their mobile devices to download or purchase apps. According to Statista, more than 93% of apps for both Android and iOS are free for download. Most of the app markets will have the rating for an app displayed and it becomes one of the most essential features to indicate app performance and likability from user 's side. The rating of apps could be based on many features such as its price, number of reviews, and its current version’s performance.

While the consumers place a high value on the importance of rating during their application hunt, the ratings could be fraud and intentionally increased by the app’s developers. Apple reveals that during the year of 2020, it removed/rejected more than $1 million fraudulent apps and managed to stop around $1.5 billion malicious purchases on the App Store.[2]

The same case applies to Google Play Store that an application had been downloaded by around 300,000 times until they found out it was actually a banking trojan for obtaining users’ password and logged keystrokes.[3] Placing customers’ privacy and identification safety concerns in the first

[1] https://en.wikipedia.org/wiki/Mobile_app
[2] https://www.dailymail.co.uk/sciencetech/article-9567253/Apple-rejected-removed-1M-malicious-apps-App-Store-2020.html
[3] https://arstechnica.com/information-technology/2021/11/google-play-apps-downloaded-300000-times-stole-bank-credentials/#:~:text=Researchers%20said%20they've%20discovered,logged%20keystrokes%2C%20and%20took%20screenshots.

place urges the mobile app market developers to identify fraudulent applications as soon as possible to prevent information leakage.

2. Industry Responses and Strategy

The App review team serves as the first barrier to block out the fraudulent app. They make sure that the new mobile apps submitted for review and version updates follow the App guidelines. They will carefully examine for hidden and undocumented features in order to reject spam and plagiarism. The second barrier would be the “Report Problem” feature that is customer facing and encourages them to report any suspicious activities they find while using the apps. For the review comments, Apple engages machine learning technique, artificial intelligence, and human examination for identification of deliberate comments to help consumers make unbiased app purchasing decisions based on the moderated app reviews.

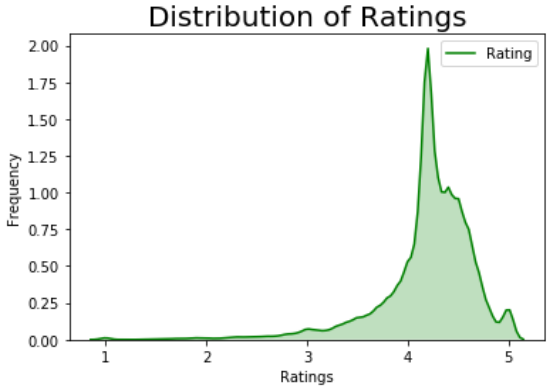
The machine learning model introduced in this paper will study what are the essential features contributing to the rating of an app and therefore could encourage app markets to incorporate the model into their algorithm when looking for fraudulent mobile applications.

3.1 Exploratory Data Analysis

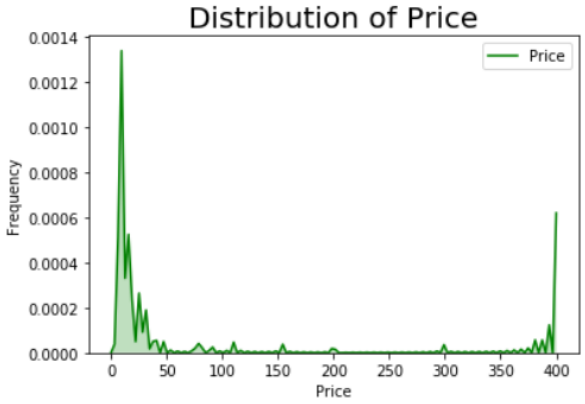
We did many exploratory data analyses to grasp the most important and straightforward distributions and relationships in the dataset.

As the rating is our target variable, we first analyzed the distribution of ratings, so we plotted the Kernel Density plot for reviews, which shows the probability density function of the rating. We

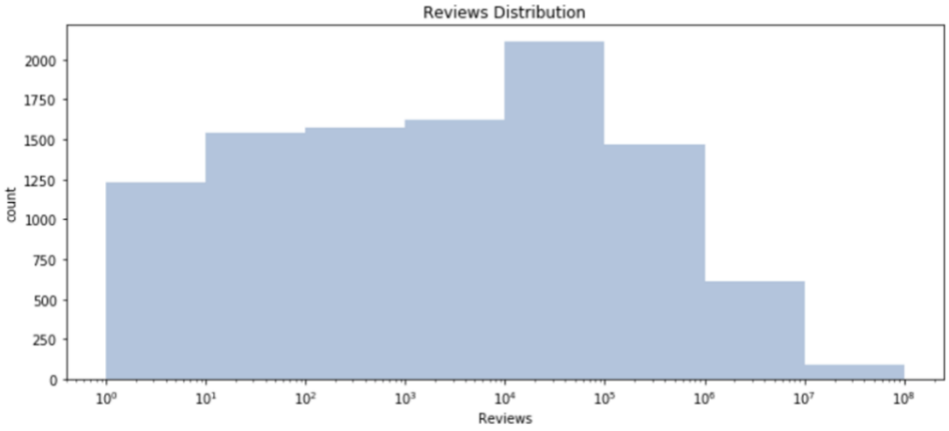
can see that ratings are mainly around 4.1 - 4.3, and the distribution is left skewed and has a long tail of ratings spreading from 1 - 3.



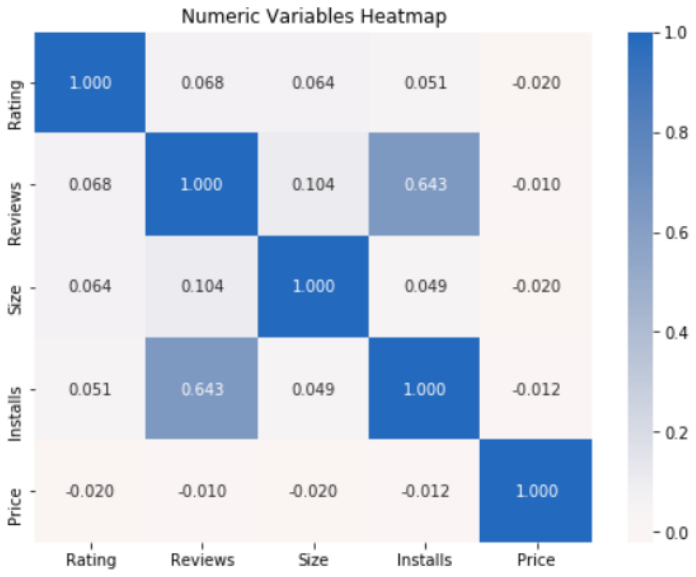
Then we looked at the Kernel Density plot of the APP prices, which are mainly in the range 0 - 20 dollars, and some outlier prices are over 350.



We further explore the distribution of reviews, since the reviews are highly left skewed, so we used a log-transformation on the X-axis to display the distribution more properly. Most APP reviews are within the range of 1,000 to 100,000.

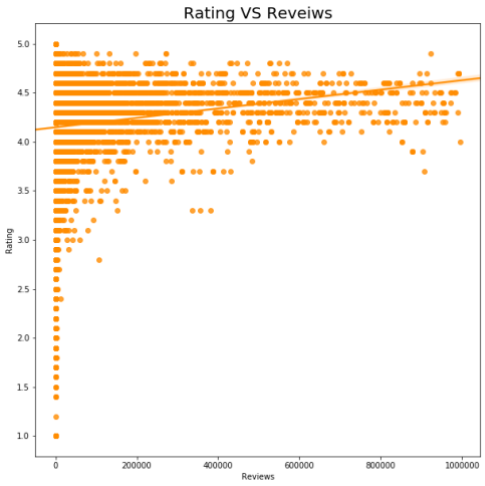


We then looked at bivariate correlation among numeric variables. As the heatmap indicates, reviews, size and installs are slightly correlated with our target variable rating, while installs and reviews are moderately correlated. We can have some assumptions that these numeric variables might not be really helpful in predicting the rating, but we still want to verify whether we can infer any causal relationship between these variables and our target variable rating.



Just to dissect the above graph a little, we tried fitting a linear regression using seaborn's regplot to see whether there is a linear relationship between rating and reviews, and we can observe

some linear patterns, the higher the number of reviews, the higher the rating of an APP, which is consistent with our business understanding.



3.2 Model Building and Evaluation

We then started the feature selection part. The target variable is *Rating*. The following 9 variables are selected as explanatory variables: *Category, Reviews, Size, Installs, Type, Price, Content Rating, Genres,* and *Last Updated*. We removed *Current Ver* and *Android Ver* since *Current Ver* varies by APPs and it is hard to define and rank *Android Ver*. For categorical variables as *Category, Type, Content Rating,* and *Genres*, we used LabelEncoding to encode them into integers. For the timestamp variable *Last Updated*, we use mktime() to transform the time into a float point number, so that it can be fed into the model. Then we did a train test split, and after that did standardization through StandardScalar() to ensure future betas estimates will roughly be in the same magnitude, and also prevent data leakage by applying standardization separately to the train and test dataset.

Then we fitted a variety of models to the data, including:

1) Linear Regression without interaction

2) Linear Regression with interaction*

3) Lasso without interaction

4) Lasso with interaction*

5) Ridge without interaction

6) Ridge with interaction*

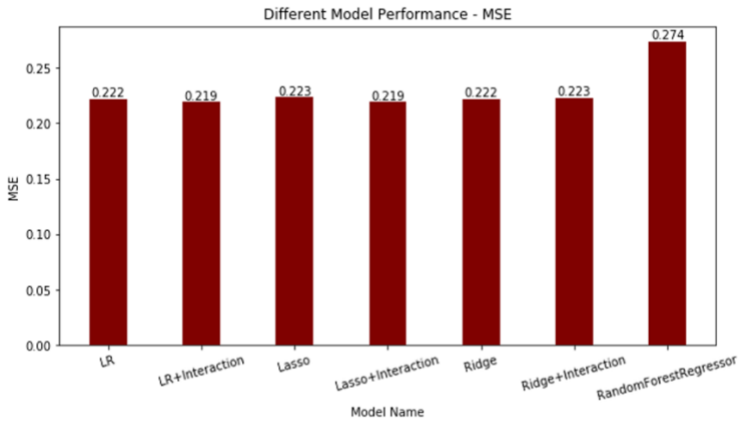
7) Random Forest Regressor**

We used MSE (Mean Squared Error) as our metric to evaluate the model performance.

*with interaction: We used PolynomialFeatures(degree=2) to find 2-way interaction.

**We tried GridSearchCV and RandomizedSearchCV for hyperparameter tuning and cross validation for Random Forest Regressor, and ended up choosing GridSearchCV because of the slightly better performance.

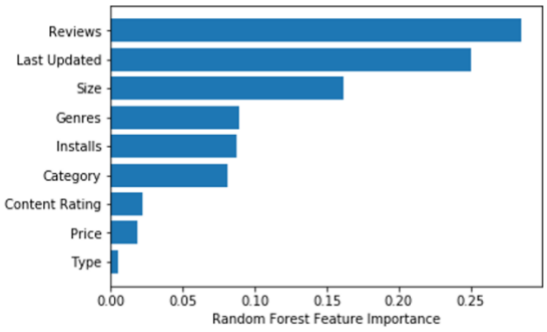
Different models' performance (MSE) are as below:



The best performance model is Linear Regression with interaction and Lasso with interaction. So we can conclude that there is some interaction effect: one variable has some effect on other variables.

3.3 Feature Importance

We looked at the feature importance plot by Random Forest Regressor, and combined with the coefficient of Lasso regression, we can infer the most important attributes for a high-rating APPs are: 1. Reviews 2. Last Updated 3. Size.




```
('Category', 0.0003492591431359316)
('Reviews', 0.018790545137149797)
('Size', 0.002479476500040324)
('Installs', 0.0)
('Type', 0.0)
('Price', -0.0)
('Content Rating', -0.0)
('Genres', -0.0005900810121726546)
('Last Updated', 0.05782715717447565)
```

Lasso Coefficient Graph

4. Recommendations and Business Value Provided:

In the current day, the various application marketplaces such as the Google Play Store and the Apple App Store are home to a plethora of applications that cover every function or need a user may have when using their smartphones. According to recent data, there are approximately 3.5 million apps on the Google Play Store itself. The application domain has become an industry itself, with many large companies' business strategies being reliant on their app offerings. Notable examples include multi billion dollar companies such as Uber, Airbnb and Lyft.

Due to the competitive nature of these application marketplaces, the companies that run them (namely Apple and Google) need to regulate and monitor them for any suspicious activity such as the presence of duplicate, fraudulent and/or malicious applications. This is a very difficult task due to the extremely large quantity of applications present. One way to measure the quality of an application is the actual rating given to that application. However, there are various factors that contribute to an application's rating. This makes it difficult to gauge the reasons why a particular application may be considered illegal, duplicate, fraudulent, malicious or a combination of these factors.

The analysis we have provided tries to resolve some of these issues for marketplace owners by indicating the most important aspects that contribute to the overall rating of an application. Additionally, the inferences made from these analyses can also inform marketplace owners on trends that indicate whether an application is illegitimate. When looking at the features that contribute to a high rate application, we can see from section 3.3 that *Reviews* and *Last Updated* are considered the most important according to the Random Forest Regressor model. These two features can inform us on applications that are fake or illegitimate. For example, there are multiple duplicate applications on both the Google Play Store and Apple App Store. These are usually copies of popular applications and are hard to discern as duplicates. As seen in the *Ratings vs. Reviews* chart in section 3.1, Ratings are positively correlated with the number of reviews. This indicates that duplicate applications can be identified as those without as many reviews as their legitimate counterparts.

On the other hand, many fake applications also rely on fake reviews or “bot” reviews created by fake users. These reviews are always positive but do not match up with the app's rating. This indicates that we can isolate outliers with a high number of positive reviews but a low rating as those that can be potential fraudulent apps. Finally, *Last Updated* indicates how recently an app has been updated on the backend by the developer. This is useful for detecting illegitimate applications as most of these are never updated after their creation, as compared to regular applications which are regularly updated for bug fixing and enhancements.

In addition, this analysis can also help app developers with understanding what aspects make an app highly rated. As mentioned earlier, *Reviews* and *Last Updated* are the two most important features. While having a high number of reviews is intuitively justifiable, the recency of the last update is an interesting finding. This indicates that users are more likely to give an application a higher rating if it is updated for UI enhancements, bug fixes or other changes.

5. Summary and Conclusions:

In this report, we have highlighted the various aspects that contribute to an applications rating. The dataset we obtained contained 13 features. As part of the process, we first cleaned the data and applied transformation wherever we felt it was necessary. After this, we performed multiple machine learning models and techniques, with the goal to determine the most important features with a moderate to high degree of accuracy. The models predicted that *Last Updated* and *Review Count* were the most important features. This allowed us to make some inferences about the business value that such an analysis could provide to application marketplace owners who are trying to determine the legitimacy of an application.