Applied Data Mining & Predictive Analytics Final Project
Google Play Store Analysis
Ethan Harden Robert Milliner

Abstract

In this study, we utilize logistic regression to examine app success determinants in the Google Play Store, using data from 2010 to 2018. We identify key predictors—rating, reviews, and price—that significantly impact app performance, measured by installs. Our machine learning model achieves a training accuracy of 93.62% and a test accuracy of 94.56%, with an F1 score of 0.94, indicating precise classification. These findings offer actionable insights for developers and marketers, enhancing understanding of app success factors. Future research directions include exploring alternative algorithms and considering external factors like economic trends and technological advancements. Overall, our study advances comprehension of app success drivers and guides future research in this field.

1. Introduction

In the dynamic world of mobile apps, understanding what drives success or failure is crucial. Our dataset, spanning Google Play app data from 2010 to 2018, offers valuable insights into this phenomenon. We've chosen logistic regression, a powerful machine learning tool, to uncover the key factors behind app success. Using logistic regression, we aim to identify the variables that matter most for an app's success, as measured by the number of installs. From app ratings to pricing strategies, we're exploring a range of factors to understand what sets successful apps apart. Our goal is simple: to distill complex data into actionable insights that can inform strategic decisions for developers, marketers, and analysts in the competitive landscape of the Google Play Store. Through our analysis, we're uncovering the secrets of app success and paving the way for future app successes.

2. Proposed Method

We segmented our dataset based on the number of app installs, dividing them into two categories: high installs (coded as 1) and low installs (coded as 0). Apps with over 1,000,000 installs were classified as high installs, while those with fewer were considered low installs. To analyze this data, we employed logistic regression, a powerful tool for predicting categorical outcomes from numerical predictors.

Our chosen predictors were Rating, Reviews, and Price, as these factors were deemed significant in understanding app success. The rating reflects if the user thought the app was good or not, Reviews indicate user engagement and feedback, and Price can influence whether or not a

user is willing to install it because it is not free. By incorporating these numerical features, we aimed to build a robust model that could predict an app's success based on these metrics.

We opted for a Machine Learning approach to model building due to its ease of implementation and interpretability. By leveraging logistic regression, we were able to create a predictive model that provides valuable insights into the relationship between app characteristics and installation success. This approach not only simplifies the analysis process but also enhances our understanding of the factors driving app performance in the marketplace.

3. Experiments

3.1 Dataset

Our project involved the analysis of a dataset spanning Google Play Store app data from 2010 to 2018, encompassing various attributes such as application name, category, rating, reviews, size, installs, type, price, content rating, genres, last updated, current version, and Android version. Our primary objective was to uncover trends, patterns, and correlations within the dataset, with a potential focus on building predictive models, particularly for forecasting app performance metrics like the number of downloads.

Given the diverse nature of the dataset, containing both categorical and numerical columns, we anticipated a rich source of information that could give us valuable insight into the data. To prepare the data for analysis, extensive cleaning was necessary. Notably, the 'installs' column presented a challenge, as it contained categorical labels instead of numerical values. To address this, we opted to categorize app installs into high and low categories using a threshold of 1,000,000 installs. This transformation facilitated the creation of a binary variable suitable for logistic regression analysis.

Furthermore, data cleaning involved converting the 'reviews' column from string to float format, so it could be used in regression analysis. Similarly, the 'price' column required conversion from string to float format, necessitating the removal of currency symbols ('$') for consistency and compatibility with regression models. We also had to remove any rows with null values so that our logistic regression would work properly. This resulted in us removing around 1500 rows that had null values for either rating, price, or reviews.

Subsequently, we narrowed down our focus to the most relevant columns for logistic regression analysis, namely rating, reviews, and price. These features were selected based on their perceived significance in influencing app performance and were expected to yield valuable insights into the factors driving download numbers.

By meticulously preparing and selecting the data, we aimed to ensure the robustness and efficiency of our analysis, thereby enabling us to derive meaningful conclusions and potentially develop accurate predictive models for app performance metrics.

3.2 Software

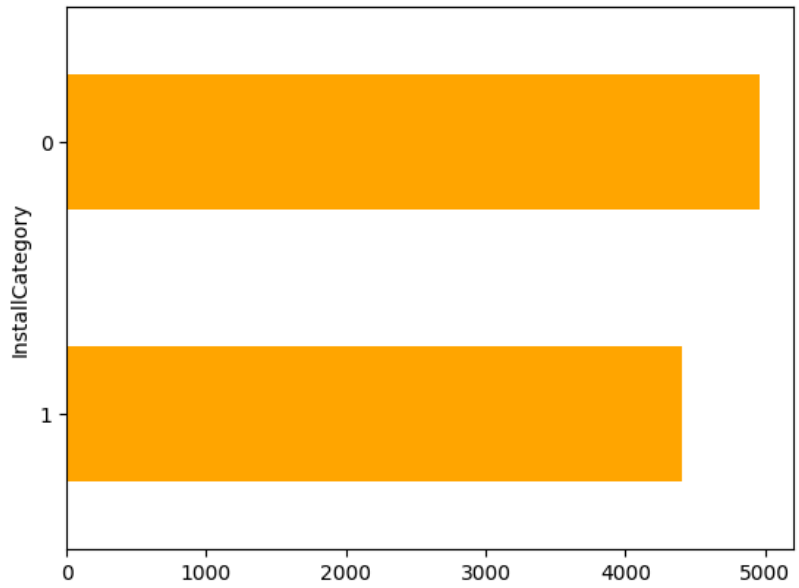
The software we used for this project:

- Python Programming Language (Python Software Foundation)
- Pandas Library (Wes McKinney et al.)
- NumPy Library (Travis Oliphant et al.)
- Statsmodels Library (Statsmodels Developers)
- SciPy Library (SciPy Developers)
- Scikit-learn (sklearn) Library (Scikit-learn Developers)
- Matplotlib Library (John D. Hunter et al.)
- Seaborn Library (Michael Waskom et al.)

3.3 Logistic Regression

We opted for logistic regression due to the categorical nature of our response variable, "installs." This statistical technique is well-suited for predicting binary outcomes, making it a fitting choice for our analysis. Our predictor variables included price, reviews, and rating, while installs served as our dependent variable. Employing a machine learning approach to logistic regression offered advantages in terms of interpretability and usability.

Post-data cleaning, we conducted an exploratory analysis to assess the distribution of high (1) and low (0) installs in our dataset. Visualizing this distribution with a bar chart revealed a balanced distribution, with 4957 low-install apps and 4409 high-install apps. This balanced representation is ideal for logistic regression, ensuring that our model isn't skewed by imbalanced data.



Subsequently, we divided the dataset into predictor variables and the response variable. To evaluate the model's performance, we further split the data into training and test sets.

Additionally, we applied data scaling to mitigate the impact of potential outliers and ensure that all numerical variables contributed evenly to the model.

With the data prepared, we executed logistic regression to derive insights into the relationship between our predictors and the likelihood of high or low app installs. The results of this analysis provided valuable information on the factors influencing app performance and enabled us to assess the efficacy of our predictive model.

4. Results and Discussion

After running the logistic regression with the machine learning approach we got a train accuracy result of 93.62% and a test accuracy result of 94.56%. This means that the predictor's price, reviews, and rating did a good job of predicting the number of installs an app had. We had a test F1 score of 0.94 which means that the model did very well at predicting the if there was a high or low number of installs. The precision for a low number of installs was 0.92. So our model correctly predicted 92% of the low-install rows in the test data. The accuracy for the high number of installs was 0.97. This means that the model correctly predicted 97% of the high-install rows in our test data. Overall, our model demonstrates exceptional predictive accuracy in distinguishing between apps with high and low install numbers. Its ability to accurately classify instances into their respective categories showcases that the predictors selected are well-equipped to make future predictions on unseen data.

5. Conclusions

Through our analysis, we used logistic regression to pinpoint the key variables that contributed to the success of an app for the Google Play Store from 2010 to 2018. While the data is dated, the implications of our study are likely not much different from then to now. The predictors that gave us success were rating, reviews, and price. These proved to be key variables in predicting whether an app had a high or low number of installs. We had a 93.62% train accuracy and a test accuracy of 94.56%. We also received an F1 score of .94 which shows the model's adeptness at classifying instances in the the correct categories. In summary, our findings show us that these variables are important in predicting the success of an app. This information could be useful for anyone developing an app or wanting to improve the success of an already active application.

Further analyses that could be performed could use different machine learning algorithms or several and compare them to see how the results vary. One could also look at factors outside of this dataset such as economic trends and technological advancements. This could provide valuable insights into how outside factors can impact how successful an app is. Looking at how the success of an app changes over time would also be interesting to look at. This could show how some apps went from a low number of downloads to a high number. Any of these would be interesting to explore in the future.

6. Acknowledgments

We would like to thank the person who provided us with the data from Kaggle. Here is the link to where we got it from
https://www.kaggle.com/datasets/lava18/google-play-store-apps?resource=download

7. Contributions

For all of the code, we worked together to write it. For the paper, we split it up into different parts and then collaborated to make sure it all flowed together. Overall we split the work up evenly for this project.