Google Store App Rating Prediction

Ensemble Techniques - Graded Project

Domain

Mobile Apps

Context

The Play Store apps data has enormous potential to drive app-making businesses to success. However, many apps are being developed every single day and only a few of them become profitable. It is important for developers to be able to predict the success of their app and incorporate features that make an app successful.
We can collect app data and user ratings from the app stores and use it to extract insightful information.
A machine learning model can be used to predict a rating for a given app, which can be used to estimate success and scope of improvement. Actionable insights learned through such analysis can be used by developers to make a successful app and capture the Android market.

Objective

To predict the rating for a mobile app given features like size, number of downloads, etc.

Data Description

Shape - 10841 records and 13 columns

Attributes -

- App: Application name
- Category: Category the app belongs to
- Rating: Overall user rating of the app
- Reviews: Number of user reviews for the app
- Size: Size of the app
- Installs: Number of user downloads/installs for the app
- Type: Paid or Free
- Price: Price of the app
- Content Rating: Age group the app is targeted at - Children / Mature 21+ / Adult
- Genres: An app can belong to multiple genres (apart from its main category). For eg, a musical family game will belong to Music, Game, Family genres.
- Last Updated: Date when the app was last updated on Play Store
- Current Ver: Current version of the app available on Play Store
- Android Ver: Min required Android version

Steps

1. Install the necessary libraries and read the provided dataset. (1 point)

2. EDA and Preprocessing (27 points)
    a. Check the info and summary statistics of the dataset. List out the columns that need to be worked upon for model building. (2 points)
    b. Check if there are any duplicate entries for the apps (1 point)
    c. Check if there are any wrong values in the ‘Category’ column and impute them with relevant values. (2 points)
    d. Which category has the highest number of apps? (2 points)
    e. Check the distribution of rating column and convert ratings into two categories and save it in the data frame as ‘Rating_cat’ (high = +>3.5 and remaining as low) (2 points)
    f. Convert the ‘Review’ column to a numerical column and impute invalid values if there are any. (1 point)
    g. Name the top 5 apps which have the highest number of reviews and their genre? (1 point)

h. Make the values of ‘Size’ as integers by replacing M and K with correct values. Convert all the values to numeric and make invalid values to NaN. (3 points)
i. Remove “,” and “+” from the values of the “Installs” column and change the datatype. (3 points)
j. What is the percentage of paid apps in the data? (2 points)
k. Remove the “$” sign the “Price” column values and make it a numerical column. (2 points)
l. Which is the most expensive app and how much does it cost? (2 points)
m. Drop columns that you feel can not be used for model building. Example- App, Content Rating, Genre, Last updated, Current Ver, and Android Ver columns from the final data frame. (2 points)
n. Encode categorical column (Type, Rating_categories, Category) [Hint - use get_dummies] (2 points)

3. Prepare data for modeling. (2 points)
    a. Segregate dependent variable and independent features into two separate variables and split the data into train and test set [Use 70:30 split]

4. Build a classifier model to predict the rating category (Rating_cat - high or low) using the following algorithm and make predictions on the test data. Evaluate the model and report your results. (16 points - 4 points each)
    a. Decision Tree Classifier
    b. Random Forest model
    c. Gradient Boosting model
    d. Stacking model

5. Check the importance of different features by using model.feature_importances_ function in Python (2 points)

6. Comment on your results and findings from the above analysis. What can you infer about how to make a highly rated mobile App from this project? (2 points)

7. Further exploration (Optional- non-graded)
    a. See if you can utilize any of these columns to improve the prediction - Content Rating, Genre, Last updated, Current Ver, and Android Ver
    b. Fine-tune the models by trying out a different set of hyperparameters
    c. Try to balance the classes to get better precision for the minority class.

Learning Outcomes

- Exploratory Data Analysis - removing test, changing data types, dealing with mixed data, removing duplicates
- Decision trees
- Random forest
- Gradient Boosting
- Stacking
- Confusion Matrix