Predicting Application Download Volumes for Google App Store

Contents

1. Project Overview

1.1 The Problem:

Google Play Store hosts an ever-growing number of mobile apps, each vying for user attention and downloads. It is an online store on Android that makes it easy for more than 2.5 billion monthly users across 190+ countries worldwide to discover millions of high-quality apps and delightful content. App developers and publishers face challenges in estimating the potential download volumes for their apps. The absence of quick access to insights generated to predictive tools limits their ability to make timely informed decisions regarding application development, marketing strategies, and revenue projections. This impacts the profitability and sustainability of the mobile apps industry as a whole.

1.2 Solution:

The primary objective of our big data analytics project is to develop a predictive model that estimates the number of downloads for apps available on the Google Play Store based on genres, sizes and other attributes. This model will serve as a valuable tool for app developers and publishers to gain insights into the potential user base for their apps and serves as an industry benchmark to evaluate app download performances. Especially, it will be helpful for:

1. Google App Store who seeks for a more robust application publishing system for developers who try to launch their apps.
2. Small to mid-tier app developers and publishers seeking to optimize their app releases and marketing strategies.
3. Consumers who rely on google app stores to discover new apps.
4. Investors and stakeholders in the process of launching applications and looking for data-driven decision support.

This idea can further be extended to a valid monitoring tool where if a developer enters certain characteristics, a predictive dashboard can be created, giving them insights as to how the app will perform.

2. Data Collection and Analysis

Collect comprehensive data on games available in the Google Play Store from Kaggle. This data initially had 10,841 rows and 14 columns. After preprocessing the data for null values, special characters and data type conversion, we ended up with 7726 rows and 13 columns such as:

- App name
- Category
- Reviews
- Size
- Installs
- Type
- Price
- Content Rating
- Genre

- Last Update
- Android version
- Current version

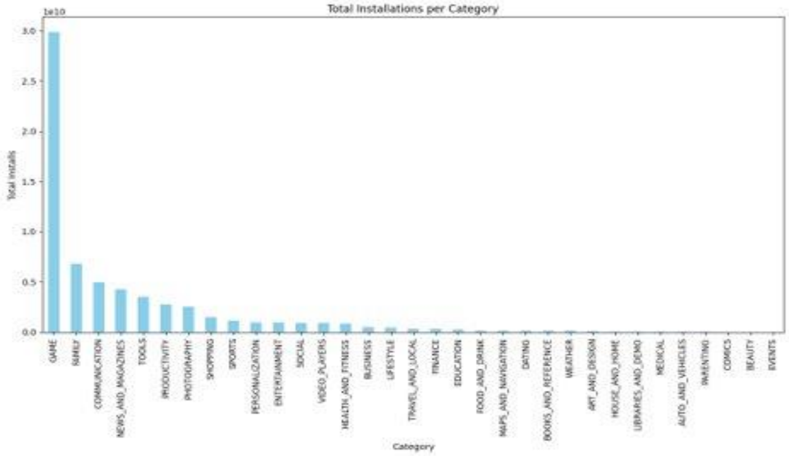
3. Data Exploration

Preliminary data exploration informed us that:

1. The initial shape of the data is 10,841 rows and 14 columns.
2. Out of 13 variables, there is 1 float and 12 object type variables. We cleaned the data because some of the numeric columns were object type. After cleaning them we converted them into float type for analysis. This led to final dataset shape of 7726 rows and 13 columns.

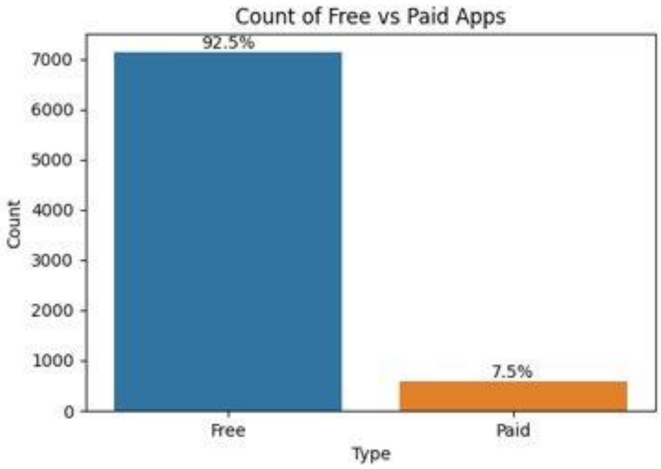
Null values, outliers and ranges:

3. There were 1463 null values in ‘Ratings’ column.
4. The range of Rating is from 1-5.
5. There are 34 unique categories of games and 112 genres with ‘family’ games having the higher proportion.



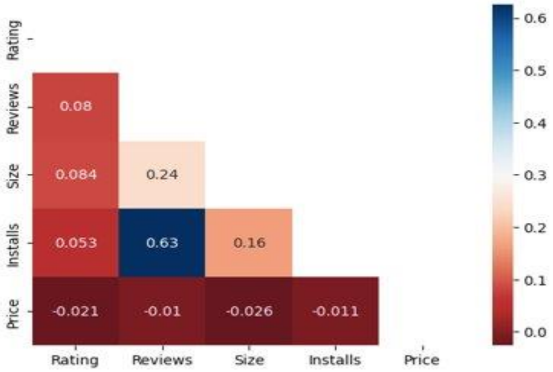
Plot 1: Histogram of Installation by category

6. Most of the games have around 10 million installs, with few exceeded 1000 million.
7. Prices range from free to $400. However, there are very few games that are prices above $50 and most of them are free.



Plot 2: bar plot showing proportion of ‘free’ and ‘paid’ games

8. There’s a strong correlation between ‘Installs’ and ‘Reviews’



Plot 3: Correlation matrix

```
□df.printSchema()
root
 |-- App: string (nullable = true)
 |-- Category: string (nullable = true)
 |-- Rating: float (nullable = true)
 |-- Reviews: integer (nullable = true)
 |-- Size: float (nullable = true)
 |-- Installs: integer (nullable = true)
 |-- Type: string (nullable = true)
 |-- Price: double (nullable = true)
 |-- Content Rating: string (nullable = true)
 |-- Genres: string (nullable = true)
 |-- Last Updated: date (nullable = true)
 |-- Current Ver: string (nullable = true)
 |-- Android Ver: string (nullable = true)
□
```

4. Modelling and Predictions

Four supervised models were build to predict the number of installations. These are multivariate regression, Decision tree with numeric variables, Decision tree with categorical variables, Random forest with all variables, Gradient Boost.

We also tried using Apriori rules such as unsupervised learning method, to make see under which 'Category' an app could get more downloads.

4.1 Supervised Machine Learning

First, we divided the data into training, validation and testing set by 60%, 20% and 20%.

4.1.1 Multivariate Regression Model

The first model we built using numeric variables: ratings, reviews, size and price which were assembled in a feature set using VectorAssembler. 'Installs' was our target variable. The regression equation we get is:

Installs = 5780403.39 - 652434.40 * Ratings + 20.93 * Reviews -3860.36 * Size - 10493.48 * Price

It indicates that as the number of reviews increase, installations go up; while the ratings go up, installation volume goes down, which is counter intuitive. This might be because, the apps in the dataset might be using highly targeted approach serving specific audience, appealing to limited users (there are 650 million average weekly visitors (newsroom, 2023)).

The RMSE of the multivariate regression model is 44,019,633, meaning that our prediction of installation volume is off by 44 million, which is not a good model. However, considering that we're working with highly unbalanced data and the range of output is large, this evaluation is not totally unexpected. Next on, we're trying to employ models that can deal with such data better to see if we get better predictions.

4.1.2 Decision Tree Models

The first decision tree model we built using numeric variables significantly improved the performance by reducing the RMSE to 37,400,492. The second decision tree model contains not only numeric variables, but also categorical variables including: Category, Type, Content Rating, Genres, which were assumed relevant to the installation volume. One hot encoding was used to encode the categorical variables. The RSME is reduced down to 33,449,809.

4.1.3 Random Forest Model

Next on, we used a random forest model to fit in all of the variables from decision tree models 2. The RMSE is reduced down to 31,573,385.

4.1.4 Gradient Boost Tree

Considering we're working with weak predictors for the installation volume, gradient boost was used to build a model that could learn from each weak predicting decision tree. Taking the numerical columns (Rating, Reviews, Size and Price) only into consideration, a model was build after splitting the data into training and testing set in the ratio of 80:20 and with following hyper parameters:
maxDepth=4, maxIter=100, stepSize=0.1
The RMSE value of the model came out to be 23112197.87.
The model help in producing list of feature importance that helped in identifying which attributes were more important for installations:

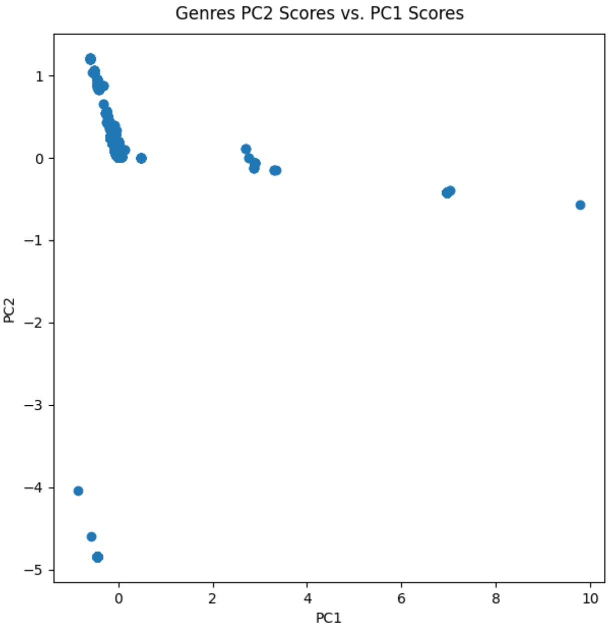
- Feature: Size, Importance: 0.507

- Feature: Reviews, Importance: 0.285
- Feature: Rating, Importance: 0.207
- Feature: Price, Importance: 4.132e-05

Note: due to high imbalance of dataset and presence of too many categorical variables, gradient boost was not modelled with categorical features, due to computational complexity

4.2 Unsupervised Machine Learning: Apriori rules

Given that we have 112 genres, we build a PCA model to explore, which genres have more importance. This is evident from the position of the 'scores' of genre of each app that not genre is important. The figure below shows that PC2 captures more apps with higher 'scores'.



Plot 4: The blue dots are each application 'scores' that was generated after vectorizing the genre of the particular application

Taking this finding we wanted to explore how the genre affect installation for the apps who have more than 500,000 installations. So, the data was subset where the installation for an app was greater than 500,000. This deliberate categorization aimed to isolate apps with a considerable user base, thereby creating a distinct "high download group" for in-depth analysis.

Then Apriori rules were modelled and applied by first transforming the genre category into numerical representations. Then the model was build with minSupport=0.0005 and minConfidence=0.005. On sorting the result with highest confidence we get an output as follows:

```
+----------+----------+----------+-----------------+--------------------+
|antecedent|consequent|confidence|             lift|             support|
+----------+----------+----------+-----------------+--------------------+
|   [0, 16]|      [14]|       1.0| 42.22033898305085|0.002810116419108...|
|       [0]|      [14]|       1.0| 42.22033898305085|0.023685266961059815|
|  [15, 14]|       [0]|       1.0| 42.22033898305085|0.001204335608189...|
|   [53, 0]|      [14]|       1.0| 42.22033898305085|8.028904054596548E-4|
+----------+----------+----------+-----------------+--------------------+
0= App
14 = Action
```

```
15 = Action
16 = Adventure
53 = Racing
```

Findings:

1. The rules suggest strong associations between certain genres and their co-occurrence in high download applications.
2. The associations identified align with a logical understanding of user preferences. For instance, looking at the first output, users interested in Adventure games and general applications (App) are likely to also download Action games. Similarly, the co-occurrence of Racing and App leading to Action downloads reflects a plausible connection between these genres.
3. The lift values indicate that these associations are significantly stronger than random chance.

5. Results and Conclusion

Overall results can be summarized as follows:

Model	Inputs	RMSE
Multivariate Regression Model	Rating, Reviews, Size, Price	44,019,633
Decision Tree with Numeric Variables	Rating, Reviews, Size, Price	37,400,492
Decision Tree with Categorical variables	Rating, Reviews, Size, Price, Category, Type, Content Rating, Genres	33,449,809
Random Forest Model with All Variables	Rating, Reviews, Size, Price, Category, Type, Content Rating, Genres	31,573,385
Gradient Boost	Rating, Reviews, Size, Price	23,112,198

The high imbalance nature of the dataset has not led low RMSE models. Centering the data and more or feature engineering might have led in development of better models. However, the overall insights we get are as follows:

1. From the linear Regression: Apps with high ratings do not necessarily have high installs, whereas higher price indicates lower downloads. However, the fact that more reviews leads to more installations stays as evident from correlation matrix as well.

Installations = 5780403.39 -652434.40 * Ratings + 20.93 * Reviews -3860.36 * Size -10493.48 * Price

2. The Gradient Boost has the best performance. The feature importance from Gradient Boost suggests that ‘size’ is the most important numeric feature, followed by reviews, ratings and price respectively.

3. It appears that using apps from a specific genre generates increased interest in applications of a related genre. This observation can provide valuable guidance for developers with a portfolio of games, helping them identify which genres of applications to develop. Developers can enhance their applications by strategically combining them with complementary genres. This can also b helpful for them to know which genre to tag there application into.

```
+--------------------+----------------+--------------------+
|                Apps|Category_Indices|          prediction|
+--------------------+----------------+--------------------+
|THE KING OF FIGHT...|            [13]|             [14, 0]|
|Zombie Death Shooter|            [16]|             [14, 0]|
|         Life market|            [54]|                  []|
|Where is my Train...|         [7, 60]|                  []|
+--------------------+----------------+--------------------+
13 = Video
54 = Role Playing
7  = Home
60 = Tool
```

For instance, the app "THE KING OF FIGHTER" and "Zombie Death Shooter" are associated with categories [14, 0], where 14 corresponds to "Action," and 0 corresponds to "App." This suggests that the model predicts these apps to be categorized as both Action and general applications. This aligns with the earlier discussion about strong associations between Adventure and App leading to Action downloads.

However, apps with unique categories that may have limited representation in the training set might not receive predictions. For example, "Life market" and "Where is my Train..." have no predicted categories probably because there are not sufficient examples of these unique categories during training, making it challenging to establish reliable associations or patterns for those specific categories.

6. References

- "Developers Generated $1.1 Trillion in the App Store Ecosystem in 2022 - Apple." Apple, 31 May 2023, https://www.apple.com/newsroom/2023/05/developers-generated-one-point-one-trillion-in-the-app-store-ecosystem-in-2022/#:~:text=The%20App%20Store%20attracted%20over,each%20week%20in%202022%2C%20respectively.

- "Google Play Store Apps." Kaggle, n.d. Retrieved from https://www.kaggle.com/datasets/lava18/google-play-store-apps