VATSapp: An Analysis on Google Play Store Apps Using R Shiny App

Table of Contents

# Introduction

Over 1000 apps get uploaded per day on Google Play Store as of the report from first quarter of 2024, with a downloads of over 25.6 billion total. The rapid growth of various apps in mobile has made the platform a competitive environment for developers, ad agencies, entrepreneurs. A brief idea on various apps, ratings, downloads, pricing etc is needed in order to survive in the market.

This project uses Google Play Store data to analyse apps from various categories such as its trends, key factors of popularity, growth over time etc. The app can also analyse an app based on its icon.

# Objective

The main objectives are

* To Create an user friendly app that can analyse google play store apps
* To Point out the different trends and patterns over different categories of apps in different region ( US and India mainly).
* To get an overview on category specific app ratings, downloads, reviews, price.

# Methodology

The project is done using various methodology, mainly data collection, preprocessing, and interactive visualization.

* **Data Collection :** The data has been scrapped from google play store. The api used is [SerpApi](https://serpapi.com/). It is a real time api that allowed ud to scrape SERP results from Google Play Apps Store.
* **Data Preprocessing :** The data is read, cleaned and formatted through different categories such as “GAME\_SPORT”, “ART\_AND\_DESIGN”, “BEAUTY” etc. uniformly. This way each categories get stabilized with similar variables.
* **Analysis Tools :** Interactive components in R Shiny and ggplot2 visualization are used to display trends and relationships in the data.

# Data Description

The “Data” folder contains the data on all the top apps from different categories in each sub folder. Every data set consists of 5 columns.

* **title :** The unique name for each app
* **link :** A play store web link to find the app easiy
* **rating :** The app rating out of 5 based on user data
* **category :** The category to which the app can more or less belong
* **author :** The creator of the app
* **downloads :** An round figure for the number of downloads for an app
* **video :** Link of any video demonstrated on the app profile
* **thumbnail :** Link of the app icon

Every sub folder denotes a categoy. In each category there are 3 .csv data sets for US and India each, namely *top\_selling\_free, top\_selling\_paid* and *top\_grossing*. Each data set contains details on top 50 apps from its field.

## Bias To Think About

In analyzing Google Play Store data, several biases may affect insights. Device preference bias emerges as Android dominates in India, while iOS has stronger usage in the US, potentially skewing app ratings and performance metrics by region. Platform-specific optimization also introduces bias, as apps optimized for iOS might perform better in iOS-dominant markets, while Android-first updates may resonate more in Android-centric regions. Larger developers enjoy brand recognition, leading to higher downloads and ratings that may not reflect app quality alone, disadvantaging smaller competitors. Additionally, cultural and demographic preferences for app categories vary by region, influencing downloads and ratings in ways that may not align with global trends. Recognizing these biases can make insights more balanced and applicable across different markets.

The categories in google play store contains numerous apps. However we are working on only top 50 of these apps. And based on our findings we are going to come to any conclusion. In such case a **survivor bias** can be working in our analysis.

[*Survivorship bias or survival bias*](https://en.wikipedia.org/wiki/Survivorship_bias) *is the logical error of concentrating on entities that passed a selection process while overlooking those that did not. This can lead to incorrect conclusions because of incomplete data*.

# Let’s Ask The Data

These are some questions we thought about that helped us to approach forward.

* Analyze how app ratings are distributed across categories or app types
* Understand which app categories or types consistently receive higher or lower ratings
* Analyzing the number of downloads for various categories
* Understand the distribution of apps across different categories (e.g., games, education, lifestyle)
* Does Design of icons affect ranking?
* Distribution of Price range of top paid apps
* Analysis of how many Apps a particular company makes and does the name of the manufacturing company affect the number of downloads
* Determine if lengthy app names are linked to worse performance metrics
* Identify leading competitors and assess where an app stands relative to others in the same category.
* Discover which categories are more competitive or saturated and which may have growth opportunities.
* Compare the above features for apps in US and India
* Similar analysis for Games as well

# App Overview

The R shiny app contains 2 main components : the user interface (UI) and the server.

### User Interface

The UI uses a modern dark theme (bs\_theme(bootswatch = "solar")) that makes the app visually stunning and user appealing. It has mainly 5 tabs.

* **Home :** This is an introductory page. It says what the app is about.
* **Data Surfer :** This provides a choice based visual representation of the data. You can choose from different region, category, file type, metric and it will give you the bar diagram/histogram of the metric for your choices.
* **Trend Compare :** Here we see a comparison analysis between apps of 2 different choices. Choice option remains same as in the Data Surfer.
* **Icon Analysis :** App Icon can be uploaded from your local storage here. And then based on the icon it gives a predictive analysis such as how the app can be rated, expected number of downloads, expected price of the app.
* **App Search :** The app search gives you a side bar menu from which you can select your choise of category/sub-category and it will give you the list of all the apps available in the intersection of your choices.

### Server

The server-side script is responsible for reading, processing, and dynamically rendering data.

* **Data Loading :** Data is loaded from category-specific files. The app sets a base directory, scanning the “Data” folder for category folders and reading each category into a separate data frame.
* **Reactive Data Handling :** The app uses reactive functions to update visualizations and tables based on user selections.
* **Interactive Features :** The app provides filtering options for attributes like rating and installs. Additionally, it allows sorting within tables for an in-depth view of app performance.

# Data Visualization

The app uses ggplot2 for visualisation provided a clear sight of rough numbers. Some examples are :

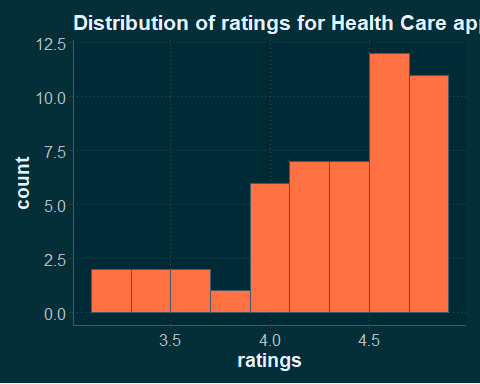
* **Rating Distribution :**  A histogram to identify the distribution of rating of apps in different categories. This helps to track down the apps with consistently high or low rating.
* **Price Chart :** The bar chart for the price of apps can easily identify the tendecy of users towards paid/ non-paid apps. If paid what amount most of the users willing to pay.
* **Install Counts :** The histogram for downloads of apps can give a trend analysis of the app category. How often users are following new trends and shifting towards new apps.
* **Icon Analysis :**  In the “Icon Analysis” tab, users upload an app icon, which is analyzed as follows:

Icon Upload and Preview: Users upload an icon, which is previewed in the app. Color Histogram Calculation: The app calculates a color histogram for the uploaded icon, representing the distribution of RGB intensities across the image. Histogram Plotting: A bar plot of the histogram is displayed, showing RGB channel distributions in intensity bins, which reveals the color balance of the uploaded icon. Category Comparison: For comparison, the app retrieves icons from the selected Play Store category, computes their histograms, and calculates the Euclidean distance between each icon’s histogram and the uploaded icon’s histogram. The Euclidean distance measures how close the color distributions are—smaller distances indicate higher similarity. Closest Match and Predicted Metrics: The icon with the smallest Euclidean distance is identified as the closest match, and its performance metrics (rating, downloads, and price) are shown as the expected metrics for the uploaded icon. This approach helps users estimate how well their icon’s color profile aligns with successful icons in the category, providing an indirect prediction of potential app performance

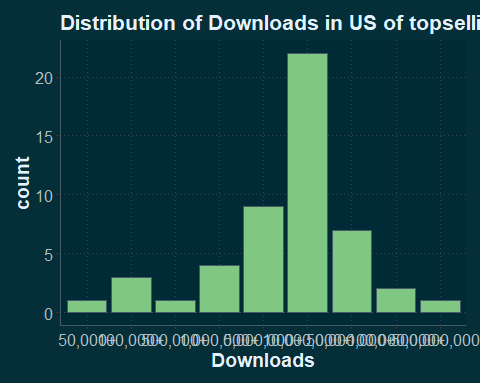
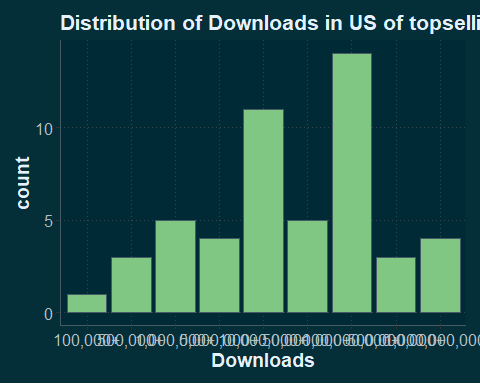
### A Key Insight

Through the app, users can find following information:

* **Top-Rated Categories :** Some categories like “Education” and “Health & Fitness,” have consistently higher ratings. This indicates users are more or less satisfied in this section.



* **Popular Free Apps :** Categories like “Games” especially free/non-paid games shows a higher number in installation.

# Conclusion

This R Shiny app provides an interactive and intuitive tool for exploring Google Play Store data. Through data visualizations and filtering options, it enables users to gain insights into app performance across various categories. However, there is room for improvement:

* **Enhanced Visualizations :** Adding more complex visualizations, such as correlation heatmaps and trend lines.
* **Predictive Analysis :** Implementing machine learning models to predict app success based on rating, installs, and other features.
* **Real-Time Data Integration :** Incorporating real-time data from Google Play, if available, to provide up-to-date insights.

# References

* [Google Play Store](https://play.google.com/store/games?hl=en)
* [SerpApi](https://serpapi.com/)
* [Wikipedia](https://www.wikipedia.org/)