Google Play Store Filter

# Project Overview:

This is a program written in Python and partially completed in Java that allows a user to search for Google Play Store apps based on certain criteria. The data is from 2018 and is loaded from a public data set available on the internet. A user can search for up to three key terms on the data set based on which attribute they are interested in. The program is simple, and the searches are rudimentary. Terms typically need to be exact in order to successfully return the subset you are interested in as most columns are processed as strings. All searches return a subset of the data by printing out all rows and column values of matching records. *All columns will return substring matches if the column contains the key term the user entered*.

# Attributes to Search On:

* App Name – the name of the app itself. User may search for any term and if the app name contains the term at all (even as a substring), the row is returned to the user.
* For example, searching “Ho” on this field will return all apps containing that substring.
* Category – the Google Play Store’s classification of the app. Searches on this field must be in entered as fully upper-case strings, but will return substring matches as well. Categories with multiple words in the title are separated with an underscore character instead of a space.
* Ex: “SOCIAL,” “ART\_AND\_DESIGN,” “AUTO\_AND\_VEHICLES,” “BEAUTY,” “BOOKS\_AND\_REFERENCE,” etc.
* Rating – The rating the app had at the time the data set was made. Must enter values between 1 and 5. The filter will include any apps with a partial score of that number.
* Ex: Searching for “4” will also return apps with a rating of 4.1 – 4.9
* Reviews – The number of reviews the app had on the Store. Must be exact, most apps have more than just handful of reviews. Will also return substring matches.
* Ex: Searching for “275” will return all apps that have that sequence of numbers in their review count.
* Size – The amount of storage the app requires to be installed.
* Ex: Searching for “60” will return all apps that required 60MB of storage space as well as apps with that substring match.
* Installs – The number of installs the app has recorded on the store. Also returns substring matches.
* Type – Whether the app is a “Free,” or “Paid” (first letter must be capitalized).
* Price – If the app is a “Paid” app, this column contains the price of the app. The most common values in this list are 0 (for free), $0.99, $1.99, $2.99, and $4.99. You do not need to enter the dollar sign.
* Content Rating – The age group the app is targeted at.
* The Google Play Store categories these common age groups: Everyone, Everyone 10+, Teen, Mature 17+, and Adults only 18+.
* Genres – While an app has one category, it can belong to multiple Genres. The data set only lists the primary genre from the Google Play Store, so only search by one term for this column.
* Values in this column include Business, Personalization, Tools, Entertainment, Education, Medical, and several others. Looking at the data source link below can provide more terms.
* Last Updated – The date the app last had an update pushed to the Store from the developer.
* The date format must be the following: Month Day, Year. Ex: September 29, 2016.
* Current Version – The current version number that was on the app store at the time the data set was created. Must be exact, will also return substring matches.
* Ex: Search “2.1.8” instead of searching for just “2”
* Android Version – The minimum required version of the Android Operating System that will be needed to install and run the app. Must also be exact, will return substring matches.
* Ex: Search “4.1” instead of just “4”

# Objectives:

* To work with Lists and ArrayLists to load data from a source into a primitive data structure
* To process the data from the data source once it is loaded in
* Report subset string matches to the user

# Requirements/Task(s):

Requirement 1: Find data source and load it into a List or ArrayList

Requirement 2: Allow user to search on up to 3 terms

Requirement 3: Return matching subsets and record counts based on user’s search

# Further Improvements and Iterations

As the project was developed, there became some opportunities for future improvements that could enhance the application.

1. Load numeric data from data source as integers or floating-point values instead of strings to avoid misleading substring matches
2. Provide user an option to enter a “minimum” and a “maximum” value to search within a particular column in the case of numeric data
3. Create a GUI that would enhance user experience of the application that doesn’t require using shells or command lines
4. Create a working Java version of the program

# Relevant Links to Project and Data Source:

1. [GitHub Project Repository](https://github.com/matthew-steuerer/Google-Play-Store-Filter), made under MIT learning license
2. [Google Play Store Data Source](https://www.kaggle.com/lava18/google-play-store-apps)