Mobile Applications Discover

Web-based Visual Analytics app for Google Play store statistics

Marco Morella
Silvio Dei Giudici

Objectives

We have seen a huge increase in the spread of mobile apps in the past years.

The sheer amount of variety between apps makes it extremely hard to navigate the choices for any kind of user.

Through our visualization environment we want to achieve two main goals:

- Help the developers to understand how the apps market is evolving.
- Help the users to evaluate which app download.

Dataset

The proposed dataset was taken directly from the play store and contains the following features (including the app name and last updated):

Categorical features

- Category
- Installs
- Type
- Content rating
- Android version

Continuous features

- Rating
- Reviews
- Size
- Price

Preprocessing

There are two main phases in the preprocessing:

Filtering:

- The main goal of this phase is to transform every features' entry in a string that can be easily converted to float afterward.

- Also here we delete all the NaN values.

Dimensionality reduction:

- We use MDS with a custom distance function to built the dissimilarity matrix.

- The custom function is a merge between Euclidean and Jaccard distances, that operate with the continuous features and categorical features respectively.

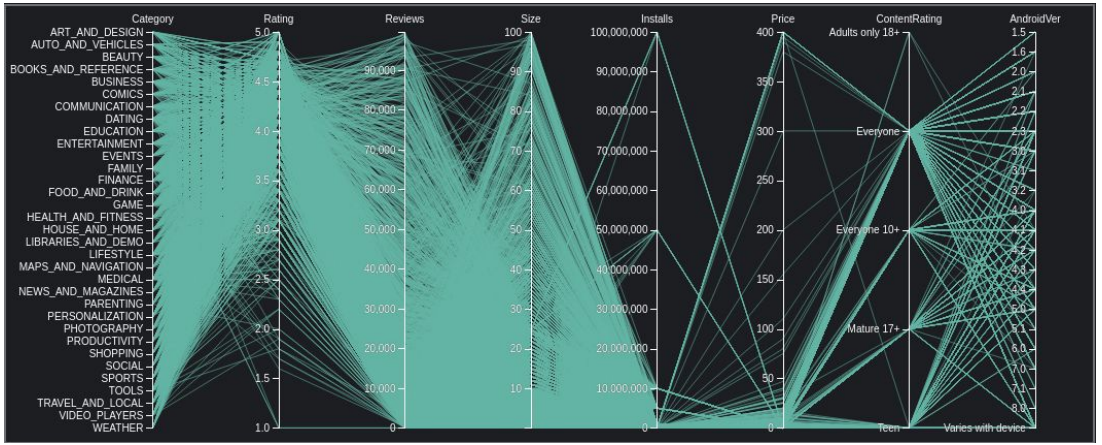
Visualizations

Views, from left to right, top to bottom:

1. Header
2. Scatterplot
3. Parallel Coordinates Plot
4. Box plots
5. Name list
6. Bar chart - Category
7. Bar chart - Content rating
8. Bar chart - Android version

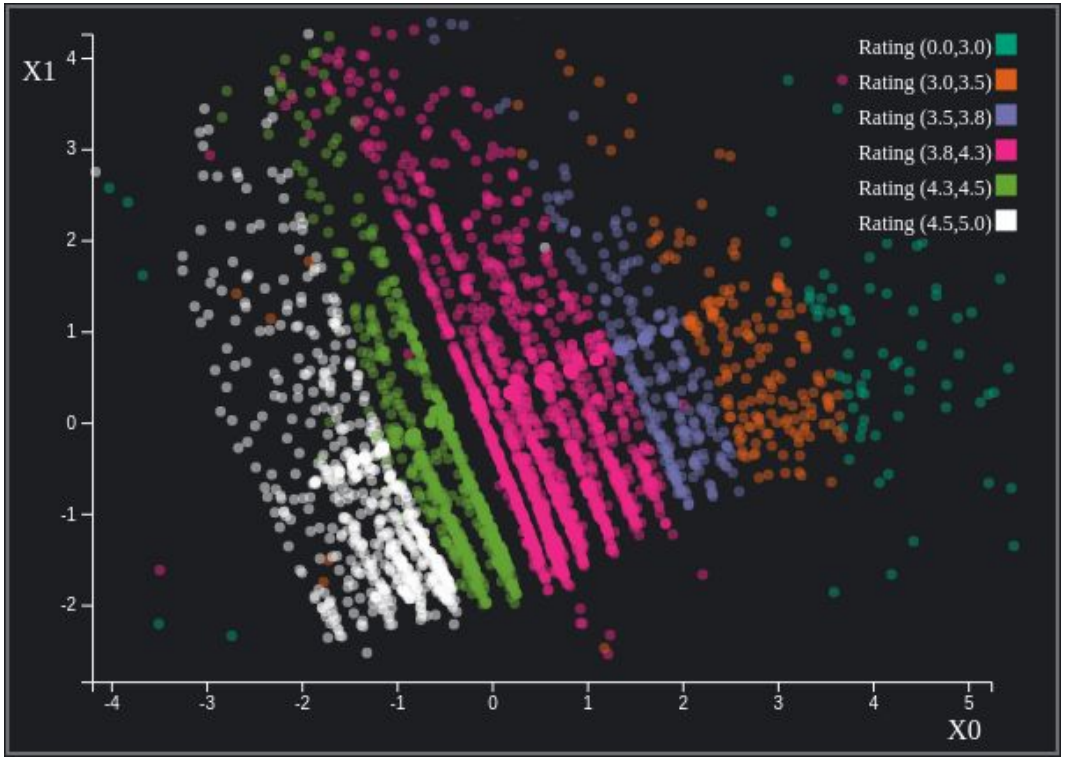
Parallel Coordinates Plot

- This is the main visualization of the tool.
- This graph contains most of the features of the dataset in a single plot.
- Brushing one axis or more, the user is able to constraint the dominion for all other visualizations.



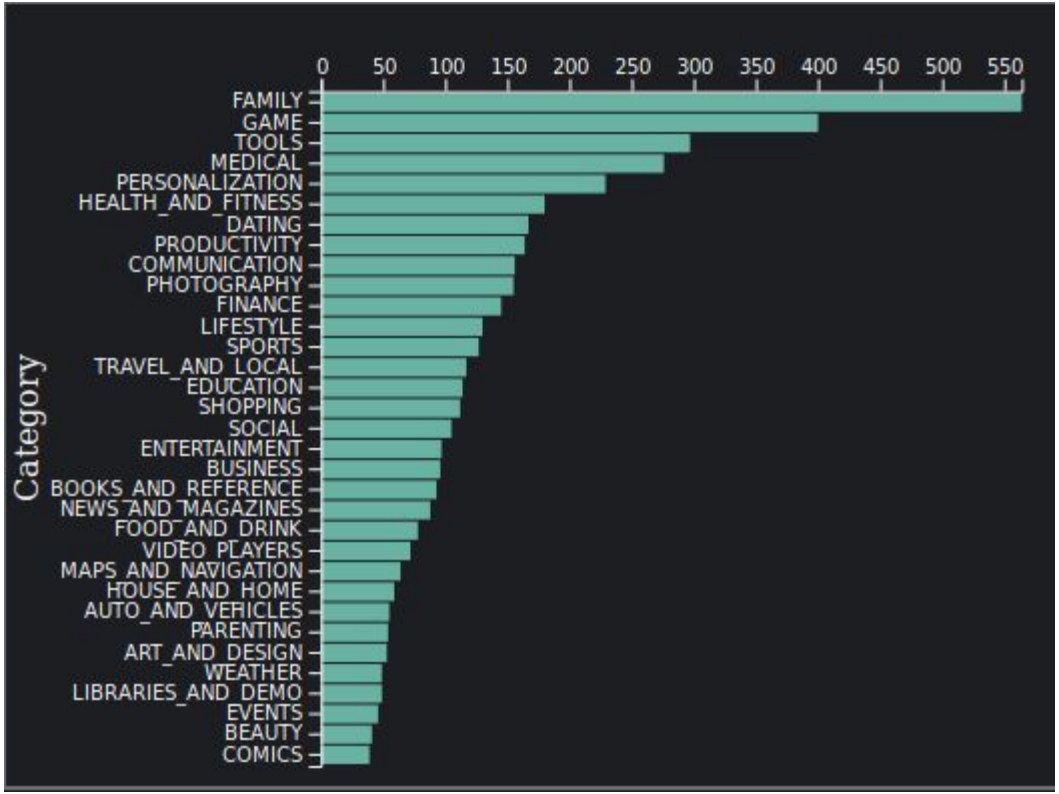
Scatterplot

- The result of MDS over the two principal component of the data is shown in the scatterplot.
- We used different colors for some ranges of rating values to show clusters.
- It supports brushing to highlight paths in parallel coordinates plot.



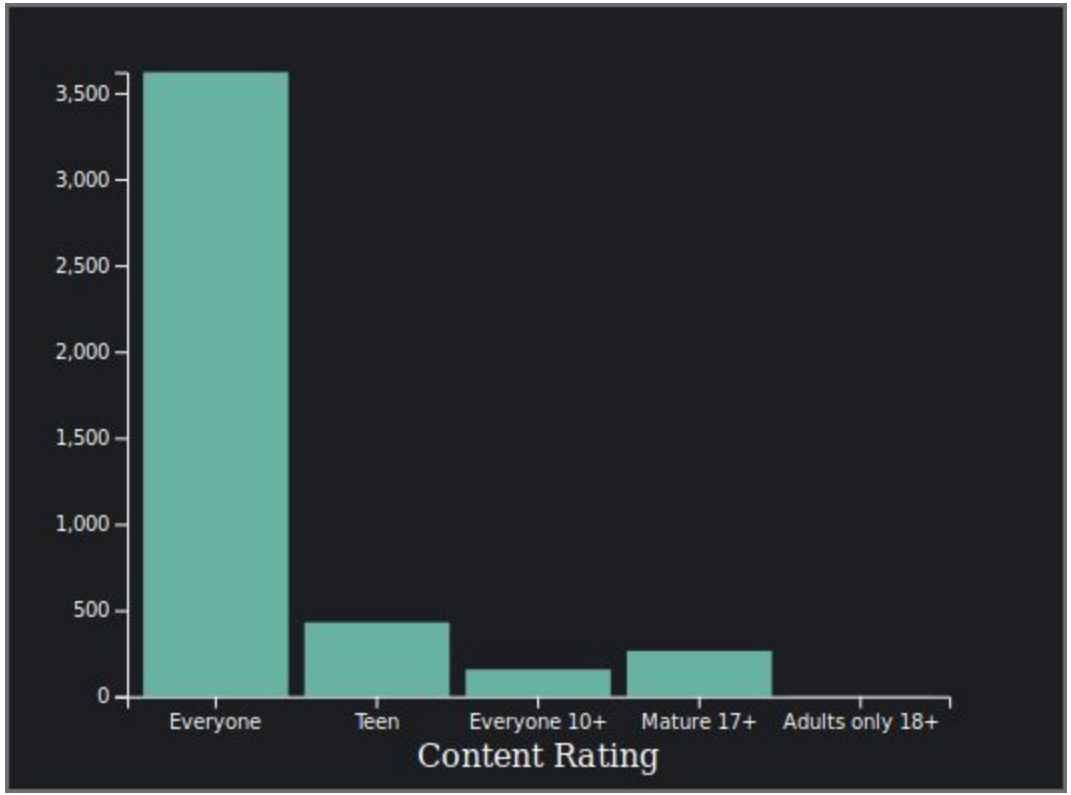
Category Bar Chart

- It shows how many apps there are in the dataset for each category.
- The bar is sorted and shows only entries not empty.



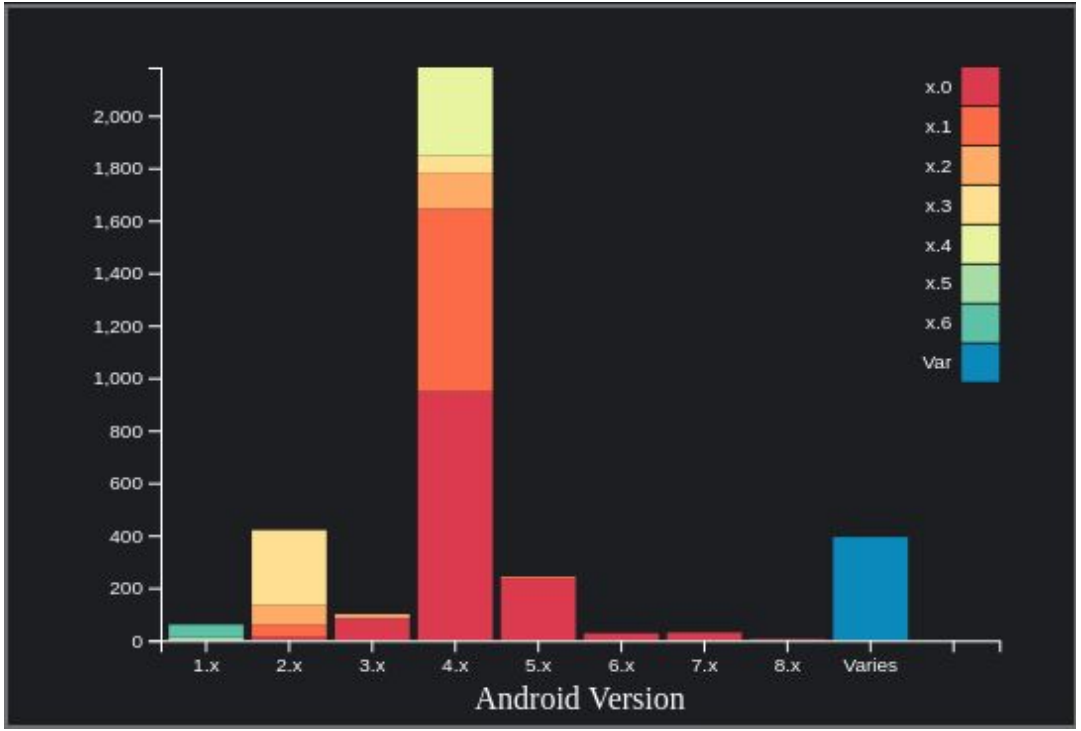
Content Rating Bar Chart

- This bar chart shows the user how many apps there are for each content rating keeping even empty entries.



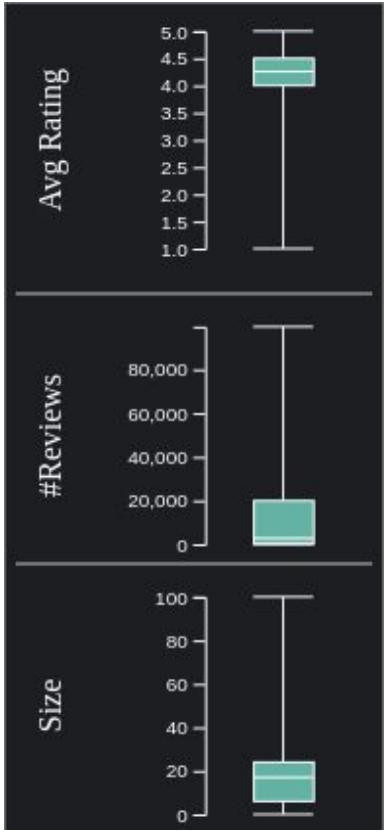
Android Version Bar Chart

- Shows the android versions required to run the apps.
- Each subversion is represented using a different color.
- This view is useful both for developers and users.



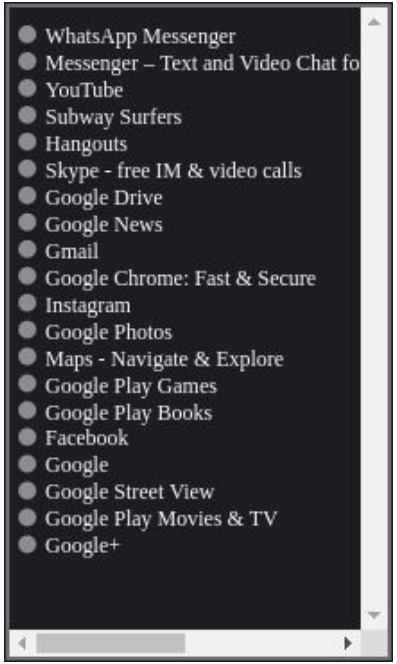
Box Plots

- We implemented three boxplot, respectively for Average Rating, Size and number of Reviews.
- They show the user the distribution of those features.
- The latter is particularly useful when we select a portion of the data.



Name List

- It lists 20 applications respecting the brushing made in the parallel coordinates plot, sorted by most installs.



User Interactions

- **Parallel Plot Brushing**, this tool allows the user to give new constraints on each axis through brushing and selecting an area on the axis.
- **Scatterplot brushing,** the user can brush over the points on the scatterplot selecting an area.
- **Paid/Free buttons,** the user can check to focus on paid, free apps or both.
- **Time range selector,** the user is able to select a time frame for the last updated feature.

Coordinated interactions

Parallel Brushing => All

- Brushing one of the axis trigger an update in all the views (except for the parallel itself).
- The parallel highlight the paths that respect the constraint.

Scatterplot brushing => Parallel, Scatterplot

- Brushing an area on the scatterplot will highlight using a different color the paths on the parallel plot that correspond to the points in the area.
- The point in the area change layout in the scatterplot as well (opacity change).

Paid/Free buttons => All

- Selecting the buttons all the view are updated in order to show only paid/free entry.

Time range selector => All

- Changing the time range trigger an update in which the views show only the entries that have a last updated value respecting the time rage.

Future work

Possible implementations in next versions:

- Add history: by periodically scraping data it could be possible to have a selection over the time changes of the applications.
- Add review content: with an auxiliary dataset containing the actual reviews and applying a preprocessing to classify them and give another useful visualization to the tool.
- Add more data: the current dataset holds a fragment of the total number of apps on the Play Store, getting the data from all of them would give more comprehensive views at the expense of enormous computational power.

Link and Resources

- Repo link: https://github.com/silviodeigiudici/VA_Project
- Dataset link: https://www.kaggle.com/lava18/google-play-store-apps
- Detail report: https://github.com/silviodeigiudici/VA_Project/tree/master/docs

Demo