# Dataset

We decided to use as a dataset: Google Play Store Apps.

The information in the dataset is extracted from the Google Play Store and includes data such as app name, category, rating, number of reviews, size, number of installs, type of app (paid or free), price, target age group and genres.

AS= 10842 \* 8 = 86 736 , we expect that after the preprocessing we will have approx: AS= 7000 \* 8 = 56 000

# General Idea

The project intends to offer an interface for viewing the data of the applications present on the Google Play store in such a way as to be able to analyze the trends of the various applications. In fact, we intend to analyze the main components of the data, through the PCA technique in order to simplify the understanding of the data and identify hidden patterns. From a visual point of view, graphs consistent with the data will be presented which will allow the user to analyze the data concerned.

# Intended user

The target users of this project include app developers, advertisers, market analysts and anyone interested in better understanding the mobile application landscape on the Google Play Store. To do this, the user can analyze the data based on the category by analyzing the various characteristics of the apps searched for.

# Used analytics

The PCA technique and Parallel coordinates representation will be used for the project. Furthermore, for the selected data the interface will interactively modify the values in the boxplots and histograms. You will also have the possibility of filtering the data based on specific filters.

# Visual Analytics cycle

As regards the Visual analytics cycle, the dataset will be processed to make the data homogeneous and usable within the analytical processes. The data visualization will be made interactive such that selected items or filters applied within a chart modify the surrounding data. This will allow the user to gain knowledge and hypothesize relationships in the data.

# Mockup user interface

A screenshot of a graph

Description automatically generated