**Google Play Store — App Performance Analytics (Final Report)**

**Author:** Manasa Shetty

**1. Introduction**

For this project I analyzed Google Play Store app data to understand app quality and market dynamics. My approach was three-stage:

1. **Data cleaning in Python** — I cleaned the raw CSV, standardized columns (Installs, Size, Price), handled missing values, converted types, and saved a cleaned CSV for reporting.
2. **Import into Power BI** — I imported the cleaned CSV into Power BI to build visuals and interactive measures.
3. **Calculated columns & measures (DAX)** — I created numeric and categorical features in Power BI (e.g., Install\_Count\_Numeric, Performance\_Category, Days\_Since\_Update, Size\_Category) and measures to summarize performance (Avg\_Rating, Total\_Installs, Top\_Performers, Market\_Share\_Top).

The final deliverable is a Power BI dashboard summarizing app counts, installs, ratings, category insights and performance segmentation.

**2. Dataset & cleaning**

* **Raw data loaded:** googleplaystore.csv (shape of dataset: **10,841 rows × 13 columns**).
* **Missing values:** I found missing ratings (1,465 rows), a few missing Type, Content Rating, Current Ver, and Android Ver. I filled missing Rating values with the dataset median and then dropped remaining rows with nulls so the final cleaned set was consistent for reporting.
* **Column cleaning & conversions:**
  + Installs: removed + and , then converted to integer.
  + Price: removed $ and converted to float.
  + Size: converted strings like 19M/2.8M to kilobytes (KB) numeric values; set Varies with device to NaN and later dropped or handled as needed.
  + Reviews: cast to integer.
  + Last Updated: parsed to datetime.
* **Saved outputs:** I exported the cleaned dataset as googleplaystore\_cleaned.csv and used that file in Power BI.

**3. Power BI modelling — calculated columns & measures created**

Below are the exact DAX expressions implemented in Power BI to standardize metrics and categorize app performance.

**Calculated columns**

**Install\_Count\_Numeric**

Install\_Count\_Numeric =

SWITCH(TRUE(),

SEARCH("M", [Installs], 1, 0) > 0, VALUE(SUBSTITUTE([Installs], "M", "")) \* 1000000,

SEARCH("K", [Installs], 1, 0) > 0, VALUE(SUBSTITUTE([Installs], "K", "")) \* 1000,

VALUE([Installs]))

**Performance\_Category**

Performance\_Category =

SWITCH(TRUE(),

[Rating] >= 4.5 && [Install\_Count\_Numeric] >= 1000000, "Star Performers",

[Rating] >= 4.5 && [Install\_Count\_Numeric] < 1000000, "High Quality",

[Rating] < 4.0 && [Install\_Count\_Numeric] >= 1000000, "Popular but Poor",

"Needs Improvement)

**Days\_Since\_Update**

Days\_Since\_Update = DATEDIFF([Last Updated], TODAY(), DAY)

**Size\_Category**

Size\_Category =

SWITCH(TRUE(),

[Size] <= 10, "Small (≤10MB)",

[Size] <= 50, "Medium (10-50MB)",

[Size] <= 100, "Large (50-100MB)",

"Very Large (>100MB))

**Measures**

**Avg\_Rating**

Avg\_Rating = AVERAGE('Apps'[Rating])

**Total\_Installs**

Total\_Installs = SUM('Apps'[Install\_Count\_Numeric])

**Apps\_Count**

Apps\_Count = COUNT('Apps'[App])

**Top\_Performers**

Top\_Performers =

CALCULATE(COUNT('Apps'[App]),

'Apps'[Rating] >= 4.5,

'Apps'[Install\_Count\_Numeric] >= 1000000)

**Market\_Share\_Top**

Market\_Share\_Top =

DIVIDE(

CALCULATE(SUM('Apps'[Install\_Count\_Numeric]), 'Apps'[Performance\_Category] = "Star Performers"),

SUM('Apps'[Install\_Count\_Numeric])

)

**Avg\_Days\_Since\_Update**

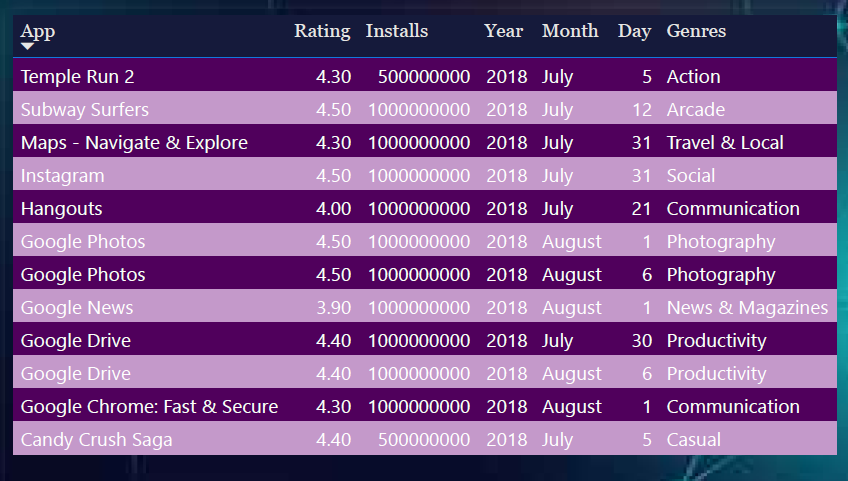
Avg\_Days\_Since\_Update = AVERAGE('Apps'[Days\_Since\_Update])

**4. Key dashboard figures & what they mean**

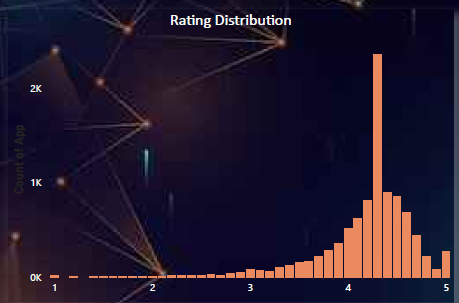
* **Total apps in dashboard:** **~10.3K**
* **Average Rating (overall):** **4.2**
* **Total Installs (sum):** **~146,630M**
* **Total categories:** **33**

These topline metrics tell me that the universe of apps I analyzed is large, ratings are generally high (average ≈ 4.2), and installs are dominated by a relatively small number of high-install apps.

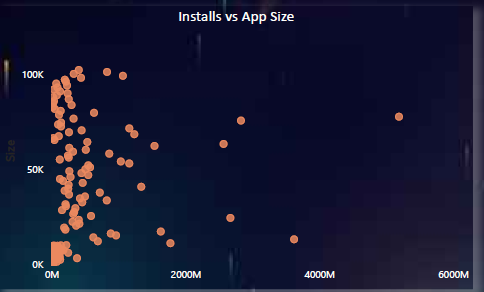
**5. Exploratory findings & visuals**

****This Top 10 Apps table Shows high-install/high-rating apps such as **Temple Run 2, Subway Surfers, Instagram, Google Photos, Google Drive, Candy Crush Saga** with their Ratings and Installs.

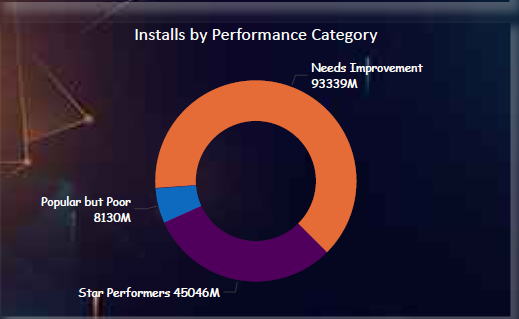
**Observation:** Top global 10 apps have both extremely high installs and strong ratings (≥ 4.3).

****

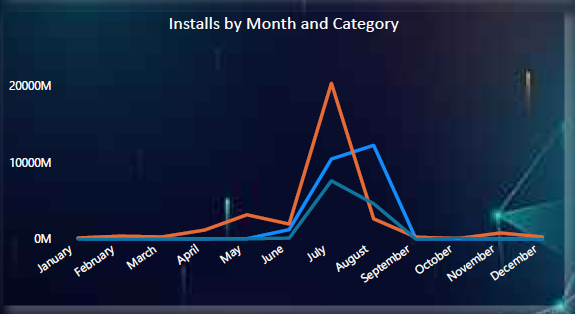
**Observation:** Ratings cluster around 4.0–4.5 with fewer low-rated apps; this supports the average rating ≈ 4.2.

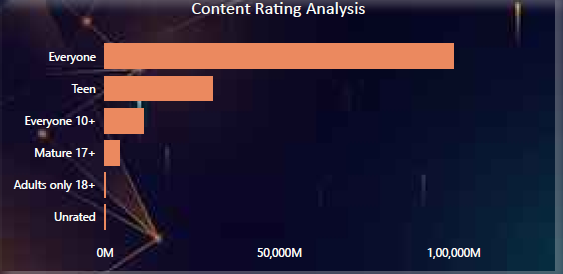
****

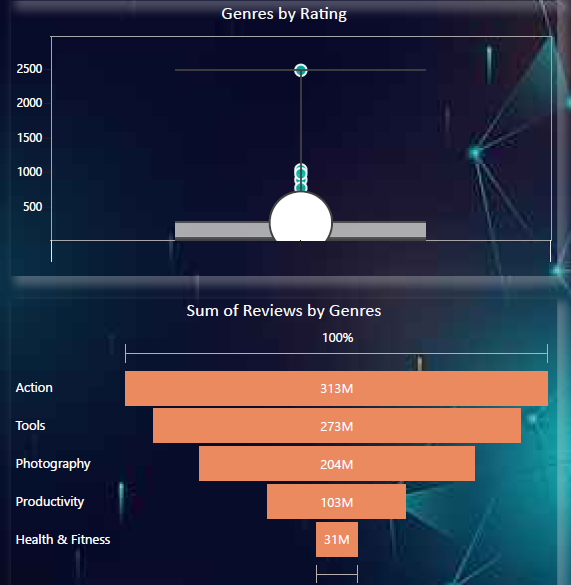
**Observation:** There is no strict linear relationship between app size and installs; some very large apps have massive installs, but many small apps also have wide reach.

****From

* **Star Performers:** ~93,339M installs
* **High Quality:** ~45,046M installs
* **Popular but Poor / Needs Improvement:** smaller shares (e.g., 8,130M shown for one segment)  
  **Observation:** Star Performers capture the large share of installs, indicating that apps that are both highly rated and widely installed dominate user attention.

**Observation:** Installs vary by month and category, with spikes corresponding to seasonal or release cycles in certain genres.

**Observation:** Most installs come from apps rated **Everyone** and **Teen**, while Mature/Adults-only categories contribute a small fraction of installs.

**Observation:** Genres such as **Action, Tools, Photography, Productivity, Health & Fitness** dominate review volumes and installs; Action and Tools show strong user engagement.

**6. Performance segmentation insights**

* **Star Performers dominate installs.** A small fraction of apps that are both highly rated and have very large install bases capture the majority of user attention — consistent with a strong long-tail market.
* **High average ratings but varied quality.** The median and average rating around 4.2 suggests overall positive sentiment, but there still exists a meaningful segment of apps in “Needs Improvement” or “Popular but Poor.” These are potential targets for quality improvement or market disruption.
* **Size is not the sole determinant of success.** App size alone does not predict installs — excellent design, marketing, and category fit matter heavily.
* **Content rating & genres matter.** Family/Everyone categories contribute heavily to install volume; genre-specific strategies (e.g., monetization in Games vs Tools) will differ.

**7. Recommendations**

**Prioritize quality improvements for “Popular but Poor” apps.** High-install but low-rating apps present a quick win: fixing UX/bugs or optimizing onboarding could improve retention and monetization.

1. **Invest in feature parity for mid-install high-quality apps.** Apps that rate highly but have fewer installs may benefit from marketing boosts or localization to scale.
2. **Use Size\_Category & Days\_Since\_Update as operational KPIs.** Monitor average days since update (Avg\_Days\_Since\_Update) to drive release cadence, and trim app size where practical to reduce friction for low-bandwidth users.
3. **Tailor strategies by genre and content rating.** Gaming apps require different retention tactics than Tools or Productivity apps; target improvements accordingly.
4. **Monitor Market\_Share\_Top regularly.** If Star Performers’ market share shrinks, that could indicate market fragmentation or the rise of new competitors.

**8. Limitations**

* **Dropped rows with remaining nulls** (after filling Rating median) — this simplified analysis but may remove edge cases such as apps with “Varies with device” size.
* **Size conversions** to KB approximate device download sizes but may not fully reflect storage behavior across devices.
* **Installs field** cleaned from strings (10,000+) — conversion assumes the reported bracket represents the lower bound; install counts may be coarse.
* **No time-series installs** (beyond Last Updated) — true trend analysis requires daily or weekly install telemetry.

**9. Conclusion**

In this project I combined Python data cleaning with Power BI modelling to produce a practical App Performance Analytics dashboard. Cleaning the dataset in Python allowed me to standardize crucial fields (Installs, Size, Price, Ratings) and to create a reliable base for Power BI. In Power BI I built the **Install\_Count\_Numeric** column and a clear **Performance\_Category** segmentation that helped me quickly spot which apps are both popular and highly rated (the Star Performers) versus those that are widely used but underperforming on quality.

My dashboard confirms a common marketplace pattern: a relatively small set of top apps capture the great majority of installs, while a very long tail of apps holds the rest of the market. It also reinforced that **high installs without quality** (popular but poor) represent actionable improvement opportunities; conversely, **high-quality, low-install apps** are opportunities for scaling.

Overall, I am confident this pipeline (clean in Python 🡪 model & visualize in Power BI 🡪 iterate with DAX) provides both the operational metrics and strategic insights needed for product, growth, and engineering teams to prioritize their work. If I continue this project, I will add time-series install trends, retention metrics, and A/B testing signals to move from diagnostics to prescriptive actions.

**Code & Dashboard**

