A Survey of Video-Editing Apps Reviews

20593 Innovation and Marketing Analytics

Gelsomino Ludovico[1†], Giancola Simone Maria[2†], Vaccarezza Gaia [3†], and Yao Yangxin[4†]

[†]Bocconi University, Milan

April 21, 2023

[1]3085366
[2]3074413
[3]3073561
[4]3215756

Abstract

In this report, we describe the foundations and results of our group project. The focus is on video-editing applications and the key factors determining their performance in terms of ratings. Our motivation lies in the recent rise of interest towards innovative types of online content, with the success of *reels* and *TikTok* short videos. Both formats potentially benefit from post processing creators' productions. In this context, video-editing apps are light smartphone tools that can support the finalization of a recording. We perform scraping of data from application stores of iOS and Android devices. Such raw information is then processed in different ways to formulate meaningful comments on whether certain features or properties of the application itself have an impact on the user-level score or the app-level score. Such analysis, after a thoughtful graphical exploration, is carried out systematically with classical models to inspect the magnitude of dependence. Given our results, to resume the discussion, we comment on potential points of focus if we were asked to devise a product that would be well-rated on the current market.

Contents

List of Figures

List of Tables

I Introduction

Improvements in hardware technology and broadband networks have helped make watching videos on mobile devices one of the most common activities for global users [Sta22a; NTI22]. In 2021, smartphones were the most widely used devices to watch digital video content by users in the United States, with 62 percent of respondents reporting to prefer these devices to laptops and smart TVs [Sta23a]. The volume of data created and consumed globally is also on the rise, with a projected growth to more than 180 zettabytes by 2025 [IS21]. The spark rise of online video consumption coupled with the exponential growth of social media has greatly impacted the way people consume content on the internet. As evidenced by recent statistics, social networking has become one of the most popular digital activities worldwide, with an estimated 4.59 billion users in 2022 and a projected increase to almost six billion in 2027 [Sta22b].

As social media usage continues to increase globally , and with platforms such as TikTok, Instagram, and Snapchat revolutionizing the way we consume and create video content, short video-editing apps are becoming increasingly popular. Now more than ever users engage with the social platforms by consuming, creating, and sharing short duration videos. Usually, this form of content does not require professional editing, and the majority of the creators rely on mobile apps for photo and video-editing. These apps are usually available on app stores or marketplaces, such as Google Play and the Apple Store. Although their download is usually free, most of them require the payment of a monthly fee to use their most advanced functionalities.

As the popularity of short-form video content is on the rise, it does not come as a surprise that the market for photo and short video-editing apps is also flourishing. Lensa AI, a photo editor app that uses AI to create avatars and artistic renderings of pictures and selfies, generated over 1.1 million downloads among U.S. users in late November and early December 2022 and grossed almost US$2.8 millions from users in the United States in the week ending December 4, 2022 [App22a; App22b]. These results made it the most downloaded and profitable photo and video-editor app during this period, followed by Canva, which is particularly popular for allowing users to create collages and photo content, with more than 1.6 million U.S. dollars in revenues and Facetune with around 1.03 million U.S. dollars in revenues [App22a].

Overall, the market for photo and video-editing apps is rapidly expanding. With around 30 million expected downloads in 2023, mobile photo and video apps are likely to maintain their position as second most downloaded apps [Sta21]. Furthermore, the revenue of photo & video-editing apps is projected to reach US$11.51 bn in 2023 (15.2% increase from 2022) and around US$16 bn in 2027 [Sta23b]. By far, the highest source of revenues for apps in this segment is given by in-app purchase (which is expected to reach US$6.44 bn in 2023), followed by advertising, with US$2.91 bn and US$2.16 bn respectively [Sta23b]. The number of downloads of these apps is also estimated to increase rapidly, going from 30 million downloads in 2023 to 37 million downloads in 2027 [Sta21]. By revenue, the largest market for these apps is China, followed by the United States, Japan, the United Kingdom and South Korea [Sta23b].

Given the rapidly expanding market for these types of applications we decided to devote our time and efforts into analyzing the characteristics that make a mobile video-editing app popular. The analysis will be conducted by means of data acquisition, data analysis and modeling. In the initial pages of this report you will find an outline of the research question and the motivation behind the analysis. Subsequently, we will include a section describing the data acquisition process together with a graphical and descriptive analysis of gathered data. Lastly, we will present a set of machine learning models that will be used to find patterns in the data and extract as many insights as possible.

II Motivation & Research Question

As mentioned in the previous section, the global user base of social media platforms has exhibited remarkable growth over the years and is projected to touch 5.85 billion by 2027 [Sta22b]. Meanwhile, applications such as Instagram, TikTok, and Snapchat have emerged as the most downloaded and widely used social media platforms worldwide.

The content that users upload on these leading social media platforms can be classified into several formats, including reels, videos, carousels, stories, and images. As per the latest data pertaining to the content reach of Instagram accounts with varying numbers of followers, reels have emerged as the most prevalent format that users utilize to publish their content on Instagram, irrespective of their follower count [Met22a; Met22b; Met22c; Met22d; Met22e]. As TikTok's content format resembles that of reels and given recent statistics on content reach, it is safe to say that reels and shorts are the most popular form of content posted on social networks.

The aforementioned upsurge of short-video content on the internet has increased both the popularity and the capabilities of mobile video-editing applications. When it comes to short-video-editing, the continual advancement in hardware and software development has made mobile applications a competitive substitute to professional computer software. Therefore, this type of application has become a popular choice among short-video content creators.

When observing applications on various stores and marketplaces, we noticed that apps matching the video-editing filter are categorized or ranked based on their rating scores and review numbers. This led us into thinking that there may exist a causal link between high ratings, large numbers of reviews and the success of an application in terms of adoption and retention rates. This simple reasoning sparked our sense of curiosity and prompted us to explore the reasons behind the differences in the volume and quality of reviews and ratings among these applications.

Thus, we formulated the following research question:

> **Research Question (RQ)**: *What are the key factors that make a video-editing app a success in terms of number of positive reviews and high rating score?*

and hypothesis:

> **Hypothesis 1 (H1)**: *Low cost, low amount of advertisement, no large size, user-friendliness and application features are the most impactful factors on review score.*

> **Hypothesis 2 (H2)**: *Presence (or not) of specific features is influential in determining the average review score.*

III Methods

As it is reasonable to assume that the volume and quality of reviews and ratings may be a reflection of the quantity and quality of the features offered by the application themselves, we decide to control for these factors by including in our analysis the features offered by every application subject to investigation.

To conduct the analysis, we recovered data of 101 apps (53 from Google Play Store and 46 from Apple App Store) with scrapers from online applications stores of common mobile operating systems. For reasons of width of use, both Google Play Store and the App Store were scraped. Below, we will briefly explain the main method and differences between the two extraction processes. The common thread is the exploitation of `Python` libraries designed to scrape app reviews and app information. Regarding this expansion of features (which are eventually all those mentioned in Table 2), we restricted our analysis only to free apps and free-of-charge functionalities to ensure equitability and comparability of the research. Among all the freely available apps, we randomly selected a subset of 84.

As the research is based on user-generated content, we collected data by means of web-scraping techniques. The gathered data includes a diverse set of information that will be comprehensively explained and explored in the subsequent section. After conducting an exhaustive examination of the collected information, we utilized machine learning methods to discover connections and patterns among the variables.

Reference Choices For both, we restricted ourselves to the US market, which is often used for general purpose analysis of digital data. The underlying assumption would be that *a video-editing application that is successful in the United Stated is (supposedly) popular in similar terms in the rest of the world.* This reasoning can be ignored if we focus on drawing conclusions for the US market only. To do so, a VPN connection was established.
In terms of identification, both stores use unique IDs to codify different items, with the difference that the App Store includes the name of the product in the ID itself.

The Google Play Store, largely used for Smartphones supporting the Android OS, is accessible with functions coming from the library `google play scraper`[1]. The construction is clever enough to allow for all of the data to be recovered with two function calls. On the other hand, App Store data is retrieved with `app store scraper`[2], which works in a more complicated manner. As previously said, the identification is performed by concatenating a numeric ID and the name of the app. Additionally, access to the reviews was manageable only one at a time, and secondary data (e.g. size) was not obtained as a result of the function call. For this reason, we coded a process based on the `BeautifulSoup` [3] library to complete the extraction. Eventually, we recover from metadata some useful app features that will populate further the columns of our main dataframe.

[1]Link to the API documentation
[2]Link to the API documentation
[3]Link to documentation

Name	Content	type
`time`	review time	`datetime`
`replycontent`	app support answer	`str`
`content`	review content by user	`str`
`score_x`	single review score	`int` 1 to 5
`username`	name of the user	`str`
`app_name`	name of the app	`str`
`google_play`	dummy for Google / App Store	`int` binary
`ratings`	number of ratings	`int`
`score_y`	average app review	`float` 1 to 5
`price`	price of the app	`float`, often zero
`size`	size of the app	`int`

Table 1: Description of Features

Preliminary Dataset Construction In order to build the primordial version of our dataset, we perform some basic feature alignment of the three sources of information. To recall, we have:

- API data from Android apps
- API data from the App Store
- scraped data from the `.html` webpage of the apps

Clearly, the three have a different aspect. To perform a merge that presents the same unit measures and enough consistency we decided to:

- convert all bytesize variables into continuous numbers of bytes
- adjust all prices to dollars (assumed adimensional by homogeneity)
- add all app features to the columns appropriately

Text Analysis When performing analysis on features which have a string datatype, as for the content of reviews, we make some tailored operations to ease the computing process. The reasons are twofold: some are empty, thus not strings, others contain emojis or non-ASCII characters (e.g. arab alphabet). The former problem is solved by adding empty strings, i.e. ''. The latter is managed through a purposeful library to process emojis, and by removing substrings with non ASCII characters. After these basic operations, what we have in hand is a dataset that we can operate on.

III.I Features, cleaning and engineering

The result of scraping could be summarized in the set of features for the dataset of reviews of Tables 1, 2. As a second step, we clean the dataset and perform different explorations of the information we retrieved with two perspectives. The former is the one presented by having *one review per row.* The latter is the result of a reduction of our information to focus on features that are app-specific, using *one app per row.*

Name	**type**	**Description** (*the possibility of ...*)
trim_video	int	cutting off either a part of the beginning or the end of a video file to reduce its total length and leave the desired content.
remove_middle_part	int	removing the middle section of a video. Usually done by cutting the file in two places and lifting out the middle section and combining the leftover video back together.
split_video	int	cutting the video file in two or more places.
merge_clips	int	merging two or more separate clips into a unique video file.
adjust_speed	int	adjusting the speed of the video (0.5x, 2x, etc.)
filter	int	altering or enhancing the visual appearance of a video. These filters are applied to adjust factors such as brightness, contrast, color saturation, hue, and sharpness. They can also be used to reduce visual noise, such as motion blur or chromatic aberration.
effects	int	manipulating or altering a video image. Examples of video effects include green screen effects, transitions and overlay effects.
animation	int	including 2D or 3D animated objects or effects to the video. Examples of video animation include Motion Graphics and Animated Characters.
remove_background	int	altering or removing the background of the video file.
retouch_face_body	int	altering or enhancing the aesthetic look of an individual in the video.
stabilizer	int	reducing unwanted shakes and jitter in the video.
free_vlog_music	int	adding copyright free music to the video file.
add_own_music	int	adding a user's own music to the video file.
sound_effects	int	altering or enhancing the audio of the video file.
voice_overs	int	adding a vocal commentary to the video file.
animated_stickers_and_texts	int	adding a sticker or a text that is animated as a 2D or 3D object.
sync_stickers_with_video	int	synchronizing the appearance or disappearance of a Sticker object according to a specific timeline.
sync_text_with_video	int	synchronizing the appearance or disappearance of a Text object according to a specific timeline.
auto_caption	int	generate a transcription or translation of spoken language in video content and to display it as a text overlay on the lower part of the screen.
character_effects	int	applying a special visual to a person's face or body to create a specific look or mood. Examples of character effects include adding makeup and adding props.
templates	int	using a pre-designed layout as a starting point for editing the video file.
canvas	int	using a pre-designed format for editing the video file. Examples of canvas include Instagram Story size format and Tik Tok size format.

Table 2: Bonus **binary** app-specific features

Name	Description	Notes
`score_y_times_log_ratings`	custom feature	`int`
`score_level`	quantile distribution of `score_y_times_log_ratings` $[(0\%, 44\%), (44\%, 77\%), (77\%, 100\%)]$	`category`

Table 3: Added Features

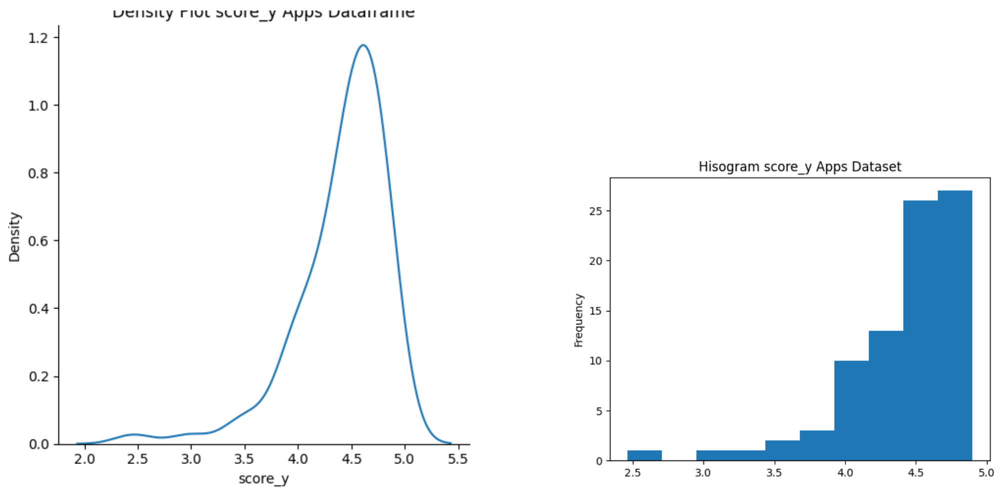

Figure 1: Density and Histogram of `score_y` distribution

It can be easily seen from the distribution of review ratings and of average review per app that the two features are highly unbalanced. For the purpose of doing a meaningful analysis, we need to overcome this difficulty, by adding some features to enhance the results. Comments on those are found in Table 3. In particular, the first one is an attempt to improve the highly skewed distributions of `score_y`, the average review, and `ratings`, the number of reviews. The former is often high, sometimes 1 and empty in the middle. The latter is either very high or very low. This can be noticed with a quick glance at Figures 1 for the average scores and Figure 2 for the ratings.

Despite being crucial for our analysis, the shape of their distribution is difficult to manage. For this reason, we resort to a monotone transformation that preserves their properties but *smooths* their distribution. Our new feature is obtained as:

$$\texttt{score\_y\_times\_log\_ratings} = \texttt{score\_y} \log(\texttt{ratings})$$

Where the log is used to tame the difference in magnitude of the two quantities[4]. The result is satisfactory, as it can be inferred from Figure 3. Our additional layer of transformation into unbalanced (but **not** too unbalanced) categories is also promising (again, see Figure 3). For free, we also obtain a more reasonable proxy for **popularity** of an application. An app could have a low number of high scoring reviews, thus making it successful but for a limited sample (e.g. it could be that it was just launched). Our feature weighs the average rating by multiplying it by a factor that depends on the number of reviews. It is also worth stressing out that this does not mean that we resize the dataset as to be balanced, which would mean modeling a system that is not a loyal representative of the real scenario. Our reviews are still unbalanced, as can be seen by the histogram of single user reviews of

[4] i.e. ratings are often in the thousands, while the average review is a number between 1 and 5

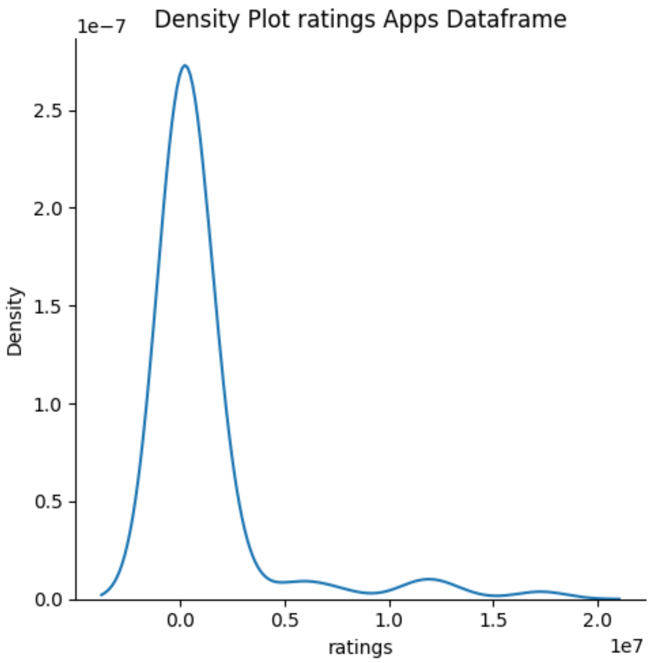

Figure 2: Density of `ratings` distribution, notice the 10^7 scale of both axes

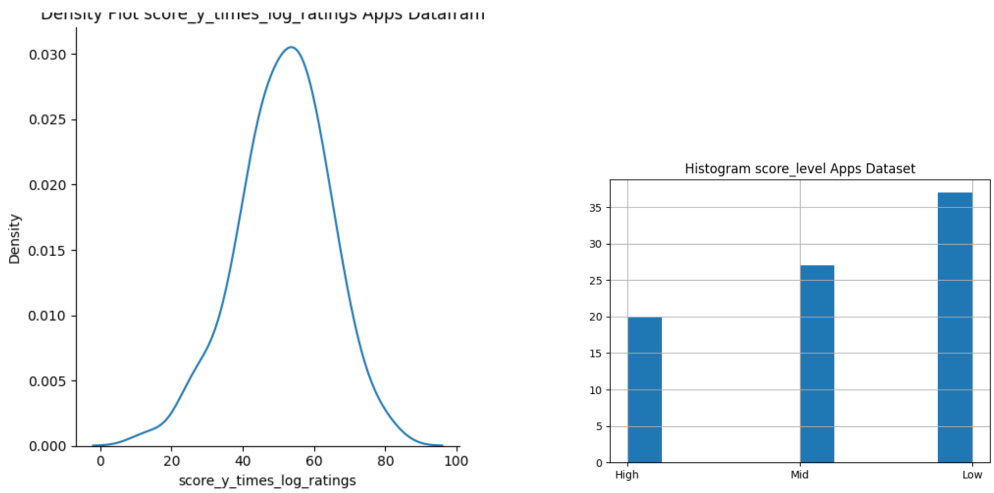

Figure 3: Density `score_y_times_log_ratings` and histogram of `score_level`

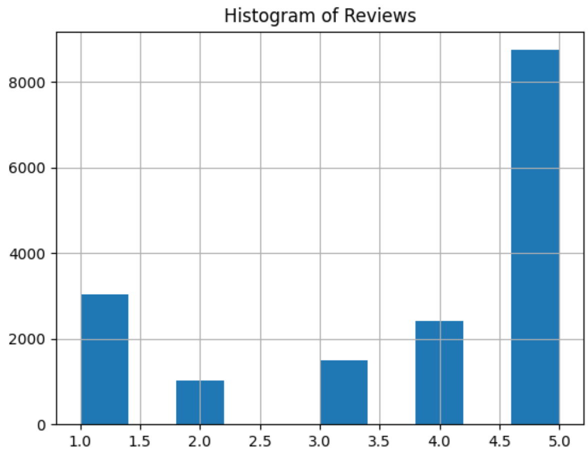

Figure 4: Histogram of single users reviews

Figure 4. The purposely engineered feature is just a tool to help us understand better data without compromising it.

IV Data Exploration

Analysis Principles to perform meaningful comments, we refrain from using features that are non algebraic. Specifically, we ignore throughout strings (e.g. review texts) and the time stamp column. Texts will be commented on on a separate note.

A general purpose collection of function calls is used to make a rough tour around the potential relationships of features available. In principle, the analysis could be carried out for any combination. Having around 20 columns, this is unfeasible. We present here a subset of plots with interesting comments.

Reviews Dataset The bigger collection of rows, where each item is a review, is the result of our scraping. Its granularity allows for assessing potential relationships between individual ratings and app characteristics. In Figure 5 we inspect the different distributions of ratings across the iOS based store and the Android OS store. The two exhibit an immediate difference: App Store reviews are more *honest*, in the sense that their distribution is more spread across ratings. This is easily noticeable by the more homogeneous number of tilts of the violin plot on the right part of Figure 5.

Apps Dataset We also extracted a smaller dataframe with app-specific features. This weighs out the number of reviews, and is more reliable for assessing behaviors at the global level of phenomenology. In this case, App Store average reviews appear to be more *optimistic*. A depiction is Figure 6. Compared with Google Store, they are centered on higher ratings and present a slightly more bumped and considerably shorter tail. In terms of size of the applications, we verify graphically that the distribution is sufficiently well-shaped, apart from two outliers in the right tail (see Figure 7).

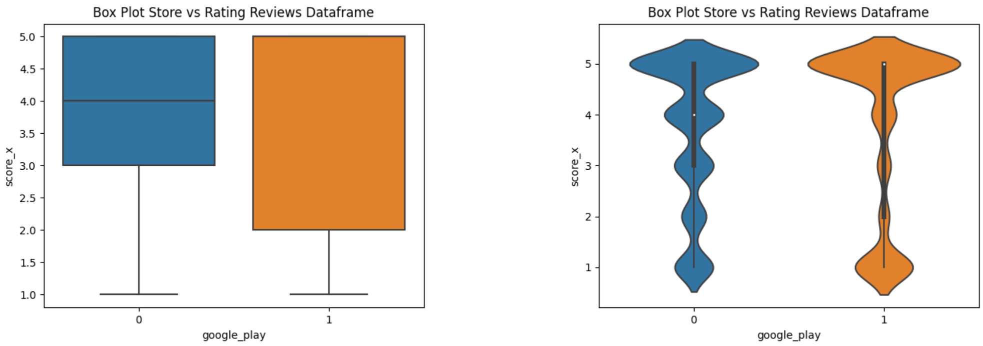

Figure 5: Distribution plots to compare Store and User ratings

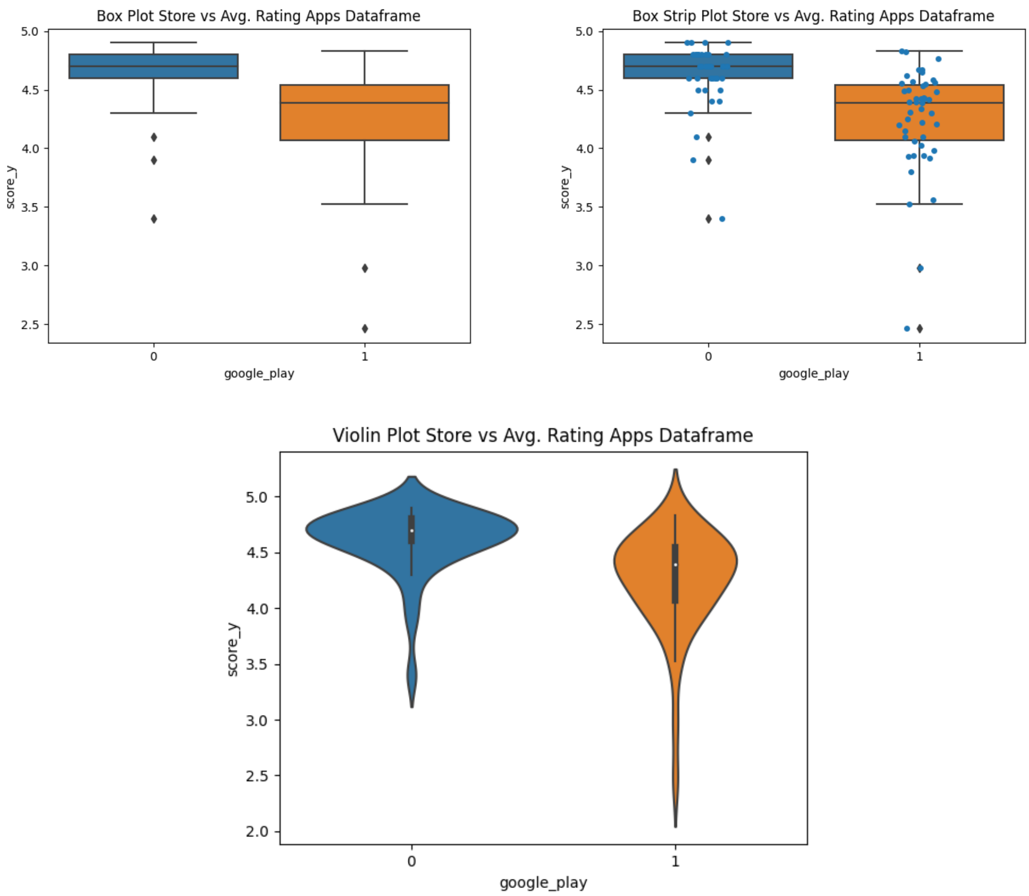

Figure 6: Distribution plots to compare Store and average rating

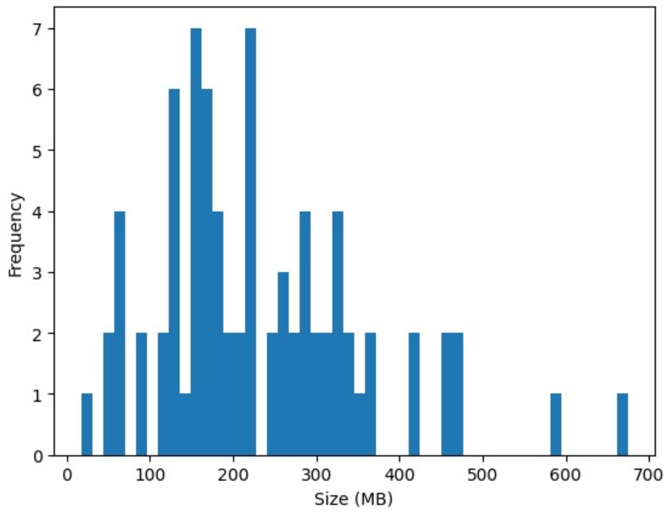

Figure 7: Histogram of sizes in MB of the applications

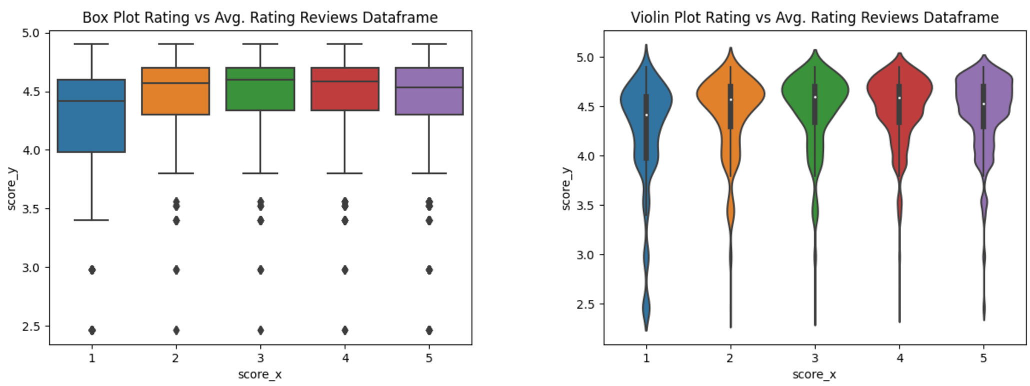

Figure 8: Distribution plots of single rating and average rating

Collective Behavior In the dataframe of reviews, it is possible to compare the relationship between average score and individual score, since they are both available. We find, as shown in Figure 8, that there are 3 macro behaviors, highlighted by the violin plots shapes on the right. One star reviews (blue) have a long tail and a slimmer spread around average ratings. This justifies the heuristics that there are always unsatisfied customers. Middle reviews (2,3 stars, respectively orange and green), start to concentrate towards the higher part of the distribution, with one significant mode. High reviews (4, 5 stars, red and purple) have progressively a more evident bimodal distribution and appear to be more tilted in average ratings above 4. Both mid and high individual ratings explore the lower part of the distribution, especially for apps that on average have around 3.5 stars.

An additional classical piece of analysis is correlation. In Figure 9, we present two heatmaps. On the left, only numerical features for the apps dataframe[5] are inspected. On the right, also categorical features are included. It is important to point out that the latter is often meaningless since it depends on the magnitude of interactions of categorical variables. We find no red flags in terms of highly correlated features, apart from the ones we engineered from others.

[5]NB: it has all the numerical features of the reviews dataframe, and some additional ones such as `score_y_times_log_ratings`

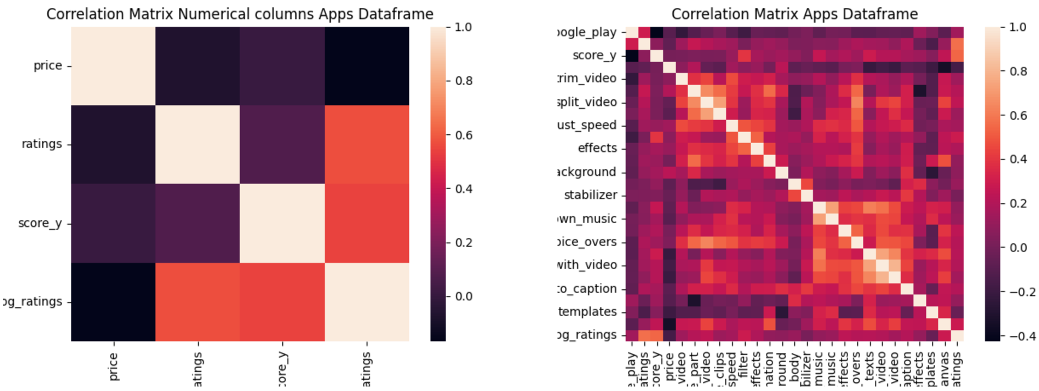

Figure 9: Correlations, numerical features and all features

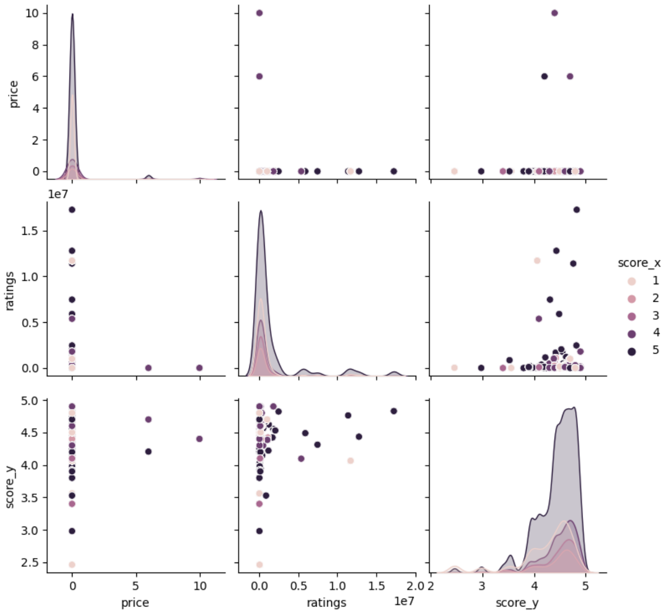

Figure 10: Hue of points is user rating, coupled plots of numerical columns

On the potential goodness of engineering features Lastly, we point out another interesting fact arising from a transformation of the average scores times logarithm of ratings. We focus on the diagonal of Figure 10. It is noticeable that at the reviews level, despite being not well behaved in distribution, our `score_y` and `ratings` are *not well*-behaved **homogeneously** across individual ratings `score_x`. This suggests that, once transforming as we did, at the apps level (thus ignoring individual ratings), the problem will only be a matter of distribution of the two numerical features. Its histogram distribution is shown in the lower right of Figure 11. Notice that the hue is no longer available since at an app level we *have lost* the individual ratings and can only look at aggregates. Nevertheless, this simple argument shows that we might not need to explore single behaviors but rather should focus on the global ones present in `score_y` and `ratings`.

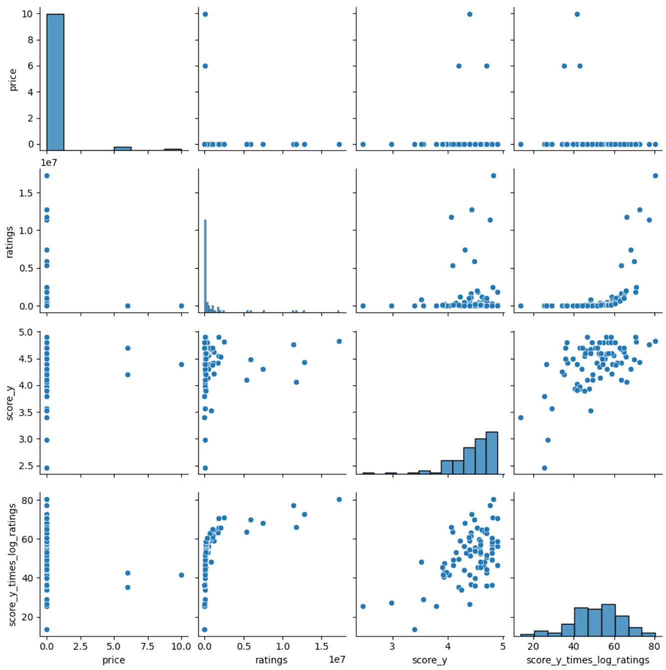

Figure 11: Coupled plots for numerical columns

V Data Analysis

The goal of the analysis is to understand the impact of several variables on the popularity of a video-editing app. It is important to define popularity for this specific case. An application is popular when it has a high rating score and has been downloaded many times. Therefore we are interested in studying these two phenomena.
Firstly, we focus only on understanding what makes a customer write a high-score review, which means a 5-star review with a range of 1 to 5.
Secondly, we combine the phenomena and focus both on the overall score of the app and the number of downloads. However, the number of downloads is not available for the Play Store app so we use the number of reviews as a proxy for it.

In order to study what affects the score of a single review we use two datasets. The first has 20000 reviews with the score of the single review as the dependent variable and the following independent variables:

- google play (the dummy variable that takes value 1 if it is Google Play store app and 0 if it is from Apple App store)
- price (in dollar)
- size (in MB)

The second, of 16723 rows, has the same dependent variable but the independent variables are the features of the single app mentioned in Table 2. The reason for having two datasets is that we can map the features only for free apps while to study the effect of price on the score we have to keep reviews related to paid apps.
We model the two datasets with *linear regression*, a *logistic regression* with dependent variable a

dummy that takes the value 1 for 4 and 5-star reviews and 0 otherwise, and an *ordered logit* which outputs a probability for each of the five classes. We first use all the reviews for every model and then split the dataset between Google Play App and Apple App Store to check how the impact of the variables change between the two platforms.

$$\text{score of the review} = \beta_0 + \beta_1 \cdot \text{price} + \beta_2 \cdot \text{size} + \epsilon$$

The linear regression with both the Google and Apple reviews shows that, all the rest constant, the Google Play Score reviews are 0.149 points higher than the Apple ones. Therefore, it is important to study the reviews of the two platforms on separate models. There could be various reasons for this discrepancy between the average ratings of the Google Play Store and Apple App Store reviews. A possible explanation is Demographics of the users. The user base of Android and iOS devices may have different demographics, which could affect their ratings. For example, Android devices tend to be more popular in developing countries where users may have less exposure to high-quality apps, and hence may give higher ratings to apps that are relatively basic. Another could be that the policies for reviewing apps on the two stores may differ, which could influence the ratings given by users. For instance, the Apple App Store has stricter guidelines for app developers, which could result in lower ratings for apps that do not meet these guidelines.
Using only Google reviews the linear regression predicts that costly apps have, *ceteris paribus*, higher reviews with 0.04 stars more for every dollar. Contrarily, in the Apple Store costly apps have 0.01 stars less for every dollar. For what concerns size, we get similar results on both the platform: heavier apps tend to have lower reviews on both platforms. It is important to note that while the coefficient of the price on the Google Play Store dataset is statistically significant at 2.5 percent, the one on Apple Store is not. Also the coefficients of the size are not significant. Therefore, their information has to be taken with a grain of salt.

Variable	Coefficient	P-values
Price	0.04	0.025
Size	-0.0001	0.339

Table 4: Linear Regression on Google Play reviews

Variable	Coefficient	P-value
Price	-0.01	0.165
Size	-0.0001	0.198

Table 5: Linear Regression on Apple Play reviews

The logistic regression and the ordered logit model have similar results: Google apps tend to have higher reviews and costly apps have more probability to have higher reviews, while size is negatively correlated with the score on both platforms. In terms of size, it is expected that heavier applications slow down devices and reduce the memory for other applications or data, making customers less

satisfied.
Interpreting the bipolar effect of the cost is less obvious. The reason why costly apps have higher average ratings on the Google Play Store compared to free apps, while the opposite is observed on the Apple App Store, is not entirely clear. However, here are a few possible explanations:

- **User expectations:** users on different app stores may have different expectations of what constitutes a good app, depending on whether the app is free or paid. Users who pay for an app on the Google Play Store may have higher expectations and therefore be more likely to give a higher rating if the app meets or exceeds their expectations. On the other hand, users on the Apple App Store may be more critical of paid apps because they expect a higher level of quality.
- **App quality:** quality of free and paid apps may differ on different app stores. In the case of the Google Play Store, users may perceive that paid apps are of higher quality than free apps due to the prevalence of low-quality, ad-supported apps on the store. In contrast, the Apple App Store has strict quality standards, which may result in higher quality free apps and a perception that paid apps are of lower quality.
- **User demographics:** it is possible that the demographics of users on different app stores could play a role in the different rating patterns for paid and free apps. For example, users on the Google Play Store may be more price-sensitive and value-oriented, which could influence their ratings of paid and free apps.

It is important to note that these are just possible explanations, and further research would be needed to determine the exact reasons for the observed differences.

$$\text{score of the review} = \beta_0 + \beta_1 \cdot \text{dummy feature1} + \beta_2 \cdot \text{dummy feature2} + ... + \epsilon$$

To go further in the analysis, we replicate these models using the features on the apps as independent variables. All agree on the sign of the coefficient of the features, however, the magnitude changes slightly.
We present the features in order of importance, meaning from highest positive coefficient in the linear regression to lowest negative coefficient.

- Positive

  1. Stabilizer
  2. Remove Middle Part
  3. Free Vlog Music
  4. Adjust Speed
  5. Animated Stickers and texts
  6. Character Effects
  7. Templates
  8. Sync Text with video
  9. Filter
  10. Animation
  11. Merge Clips
  12. Remove Background
  13. Auto Caption

- Negative

  1. Effects
  2. Sound Effects
  3. Voice Overs
  4. Canvas
  5. Sync Stickers with video
  6. Split Video
  7. Retouch (Face/Body)
  8. Trim Video
  9. Add Own Music

The coefficients of the features are all statistically significant (at 5 percent) except for Animated Stickers and texts, Charachter Effects, Animation, Retouch (Face/Body).

$$\text{overall score} * \text{number of reviews} = \beta_0 + \beta_1\text{price} + \beta_2\text{size} + \epsilon$$

The second part of the study focuses on the apps' overall score and the number of reviews. To study them together we multiply average score by number of reviews application-wise, to obtain a continuous variable. We study the effect of size and price on this variable. It would be interesting to study also which features are crucial for the popularity of a video-editing app. However, the dataset has an app per row and being quite small (101 rows) it is not possible to estimate a model with so many dependent variables. Yet, we focus on this in the exploration part.

The linear regression on the complete dataset of apps (both Google Play store and Apple app store) suggests that Google apps have, *ceteris paribus*, much higher score times ratings (about 1.6 million more). Therefore, also here it is important to split the research.
By performing one linear regression per platform we get similar results for both platforms. Size is positively correlated with the dependent variable, in particular for Google Play Apps, while the price is negatively correlated.

Variable	Coefficient	P-values
Price	-1500000	0.559
Size	17323	0.367

Table 6: Linear Regression on Google Play reviews

Variable	Coefficient	P-values
Price	-110506	0.305
Size	291.93	0.866

Table 7: Linear Regression on Apple Play reviews

As these results are opposite to those we got in the other section about score of the review, we first tried to apply the log transformation to the number of reviews variable before multiplying it with the score. This was the starting idea from Section IV. As this did not change the results, we decided to study the overall score of the app and the number of ratings separately. Yet, it is important to highlight that the coefficients of the regression are not statistically significant so they are not very informative and we give more weight to the significant results of the previous part of the analysis.
What we found is that the results on the effect of size and price are mainly driven by their effect on the number of ratings. Indeed, for both platforms, their effect on the overall score is very close to 0. Thus, we focused on studying the relationship between price, size and number of reviews. It is clear that the negative effect of price on number of reviews is due to the fact that costly apps are

less downloaded than free apps and so less people decided to write a comment on them. It is more complex to understand why bigger apps have more reviews. Such observed modeling result might be due to the reason that we wrote before, meaning that heavier applications make the device slower and reduce free memory on devices, making the customers less satisfied. Yet, there may be also positive reasons. Applications with larger sizes may have more features, which could attract more users and encourage them to leave reviews. More features also mean more opportunities for users to encounter bugs or issues that they might want to report in their reviews. Differently, larger applications may be perceived as having higher quality because they have more features and require more effort from developers. Users may be more likely to leave positive reviews for applications that they perceive as high quality. Additionally, applications with larger sizes may receive more marketing and promotion from their developers, which could lead to more downloads and ultimately more reviews. Even app stores often promote apps with more reviews, so larger applications may be more likely to appear at the top of search results and therefore receive more downloads and reviews.

V.I Text Analysis

We finally focus on the comments related to each review. First, we split reviews between negative (score lower than 3) and positive (score higher than 3) ones, then we inspect monograms and bigrams. We looked for the 50^{th} most frequent monograms and bigrams in negative reviews and positive reviews, and group them to see the most crucial topics. Finally, we read the reviews that contain the informative bigrams.
Monograms are not quite informative but helped us ignore very frequent words that add no value to our research, and are present both in negative and positive reviews. Some examples are 'update', 'version', 'people', 'able'. Instead, the outcome of bigrams is very interesting.
We clean the reviews from frequent and not informative words such as:

- 'app', 'APP','ap', 'App', 'apps', 'application'
- 'browser', 'browse'
- 'website', 'websites'
- 'chrome'
- 'click'
- 'web'
- 'ip', 'address'
- 'files'
- 'android'
- 'service'
- 'use'
- 'one'
- 'download'
- 'email'
- 'video', 'video-editing', 'editing'
- 'photo', 'photos'

together with punctuation, stop words and words with less than 2 characters.
Whilst this cleaning is done inside the code to understand potential visualization of reviews texts, in the cover and in Figure 12 we report two examples of Word Clouds without polishing all of the non informative words. This is done to make the link with our context stronger for the reader.

Within the negative reviews, three main topics can be identified: in-app purchases, crucial features and bugs. Out of 5195 rows (score lower than 3 stars) around 650 (the 12.5$) are about angry customers that did not want to pay or had to pay to use specific features. In particular, because many apps required a costly long-term subscription while the customers downloaded the app for a one-time/very rare use. Moreover, many feedbacks claimed that the costly features were not worth the money, making causing dissatisfaction.
The second topic is about features, in particular, the reviews highlight the presence of crucial features

Figure 12: An example of a less polished WordCloud

that are missing or not well developed and so the producer should focus on them. These are: `camera roll, save video, editing software, share on social networks, export video` and `add music`. Finally, there are many comments complaining about bugs such as the app crashing often and showing long loading times.

Positive reviews are more general and many of them are *happy* comments that are not very informative on the reason for such satisfaction. However, we qualitatively cluster two main topics: features and design. Also here the comments highlight which are the crucial features of the app on which the development team should focus. In particular, they appear to be: `stop motion, share on social network, green screen, camera roll, add music, add text` and `sound effects`. It is interesting to note that some features are quoted both on negative and positive reviews, meaning that they are very important for the users. We also find among the 50 most frequent bigrams there are 'user friendly' and 'super easy'. This suggests that the design of the app is very appreciated by customers and should not be overlooked.

VI Conclusion

Given our analysis, it is advisable to create a video-editing app only for Google Play Store. In this platform the review scores are higher as well as the number of ratings. It is also reasonable to say that we would be able to charge a price for it, due to its positive effect on the popularity of the app. Heuristically, it would be 0.99€, meaning the lower possible. Such choice could be motivated by the willingness to spark interest in users, which are empirically eager to review well a paid application. At the same time, we would cover production costs without the need for in-app purchases or long-term subscriptions, which showed to be likely to cause dissatisfaction in customers.
The size constraint would be important, since it might slow the device of the users. This idea should not be extremized given evidence that dimension may be a proxy for good quality. Intuitively, applications with more features and better editing software occupy more memory, controlling for space-optimized development. A risk-less choice would be around 200 MB, the average of our mostly centered distribution (again, see Figure 7). Incidentally, a reasonable size would avoid frequent app crashes, provided that the there are no systematic bugs.

As a last note, design will be a crucial part of the app. Our analysis highlighted that this aspect is very much appreciated by customers, especially when efficient in terms of the keywords 'user-friendly' and 'clear'. Given our numerical results, a potential starting point in terms of specific binary features would be:

- Stabilizer
- Remove Middle Part
- Free Vlog Music
- Adjust Speed
- Templates
- Sync Text with video
- Filter
- Save video
- Share on social networks
- Export video
- Add music
- Stop Motion
- Green screen

It is important to stress that this is not an exclusive list but just a collection of factors that might turn out to be influential.

A combination of the above arguments, thanks to the generality of our scraped data, would be a principled starting point to think about how to devise a video-editing app that successfully answers our main research question. Our Hypotheses have been discussed throughout the text and proved to be at least interesting perspectives, which at the same time suggested to design a broader view of our set of information.

Bibliography

[Sta22a] Statista. *Most popular smartphone activities for global users 2022*. en. https://www.statista.com/statistics/1337895/top-smartphone-activities/?locale=en. Accessed: 2023-4-12. Sept. 13, 2022.

[NTI22] NTIA. *Leading U.S. internet activities 2021*. en. https://www.statista.com/statistics/183910/internet-activities-of-us-users/. Accessed: 2023-4-12. May 11, 2022.

[Sta23a] Statista. *Most used devices for digital videos in the U.S. 2022*. en. https://www.statista.com/forecasts/997109/most-used-devices-for-digital-videos-in-the-us. Accessed: 2023-4-12. Mar. 7, 2023.

[IS21] IDC and Statista. *Total data volume worldwide 2010-2025*. en. https://www.statista.com/statistics/871513/worldwide-data-created/?locale=en. Accessed: 2023-4-12. June 7, 2021.

[Sta22b] Statista. *Number of worldwide social network users 2027*. en. https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/. Accessed: 2023-4-12. June 15, 2022.

[App22a] AppMagic. *Most popular photo and video editor apps in the U.S. 2022, by revenues*. en. https://www.statista.com/statistics/1351014/top-grossing-fhoto-and-video-apps-us/. Accessed: 2023-4-12. Dec. 8, 2022.

[App22b] AppMagic. *U.S. top photo & video editor apps by weekly downloads 2022*. en. https://www.statista.com/statistics/1350996/top-photo-editor-apps-us-by-download/. Accessed: 2023-4-12. Dec. 8, 2022.

[Sta21] Statista. *Global: mobile app downloads by segment 2017-2025*. en. https://www.statista.com/forecasts/1262881/mobile-app-download-worldwide-by-segment. Accessed: 2023-4-12. Aug. 31, 2021.

[Sta23b] Statista. *Photo & Video - worldwide*. en. https://www.statista.com/outlook/dmo/app/photo-video/worldwide. Accessed: 2023-4-12. Mar. 2023.

[Met22a] Metricool. *Instagram accounts with 50k to 45m followers reach by content type 2022*. en. https://www.statista.com/statistics/1356216/instagram-content-reach-accounts-up-to-45-million-followers-by-type/. Accessed: 2023-4-12. Sept. 12, 2022.

[Met22b] Metricool. *Instagram: content reach for accounts with 500 to 2k followers 2022, by content type*. en. https://www.statista.com/statistics/1356220/instagram-content-reach-accounts-up-to-2k-followers-by-content-type/. Accessed: 2023-4-12. Sept. 12, 2022.

[Met22c] Metricool. *Instagram accounts with between 2k and 10k followers reach by content type 2022*. en. https://www.statista.com/statistics/1356213/instagram-content-reach-accounts-up-to-10k-followers-by-content-type/. Accessed: 2023-4-12. Sept. 12, 2022.

[Met22d] Metricool. *Instagram accounts with up to 500 followers reach by content type 2022*. en. https://www.statista.com/statistics/1356210/instagram-content-reach-accounts-up-to-500-followers-by-type/. Accessed: 2023-4-12. Sept. 12, 2022.

[Met22e] Metricool. *Reach of Instagram content for accounts with 10k to 50k followers as of June 2022, by format*. en. https://www.statista.com/statistics/1356221/instagram-content-reach-accounts-up-to-fifty-thousand-followers-by-type/. Accessed: 2023-4-12. Sept. 12, 2022.