PRESENTATION
by

Barberis Pietro – **3142272**
Bordoni Matteo – **3133592**
Necchi Marco Vittorio Maria – **3120288**
Porfidia Elena – **3127703**
Ungarelli Federico – **3123212**
Ventura Antonio Roberto – **3127698**

HOME

01
02
03
04
05
06
07
08

Google Play Store

30316 – BIG DATA AND DATABASES

VIEW MORE

SCROLL

OUTLINE OF THE REPORT

MENU

01

INTRODUCTION

Group Project Overview

VIEW MORE

SCROLL

Overview

The dataset used for this project contains observations scraped from the internet focusing on the results of Google Play Store app downloads, users' reviews, ratings, and detailed information about the app and its author. The data frame contains 2,312,944 entries, composed of 24 columns.

The objective is to exploit the dataset to analyze the sales and downloads of the Google Play Store market from the point of view of new potential investors interested in entering the industry. The target variable used is "Rating," which is present in the dataset as continuous and required categorical transformation to be utilized in the algorithms as a binary variable. From a managerial point of view, having the correct prediction and characteristics of a publicly appreciated application will facilitate investing in or developing a potential successful app. Furthermore, analyzing the thriving components of an excellent or terrible review will enable managers to be guided by the path of already performing apps on the market.

To complete the task described above, the algorithms of Regularized Logistic Regression, Random Forest, and Gradient Boosting Machine are implemented after a first raw data description and tailored data preparation.

The dataset

The data frame contains

2,312,944 entries

24 Features divided in:

- 19 Categorical
- 5 Numerical

MENU

02

DATA DESCRIPTION

Univariate Analysis

VIEW MORE

SCROLL

RATING

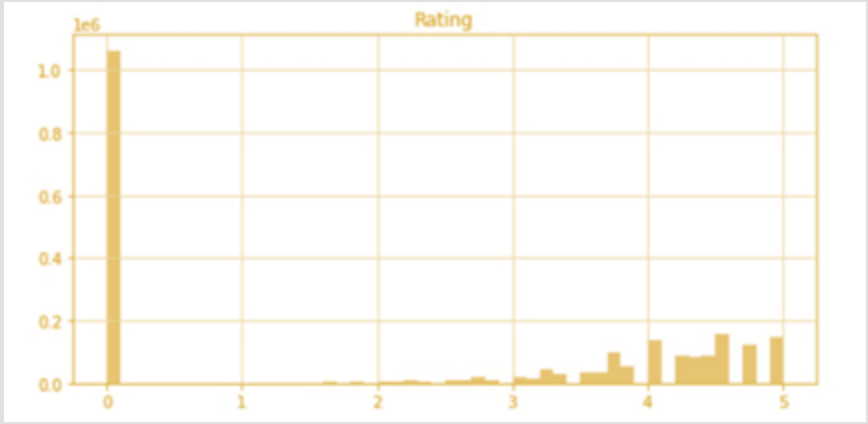
Target Variable

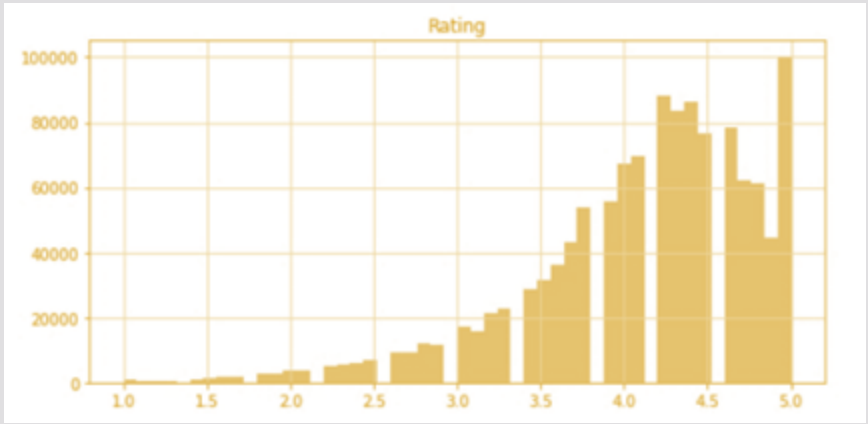
Nature: Numerical, Continuous

Description: evaluation on a scale from 1 to 5 based on users' experience

Range: [1, 5]; 0 where rating is not present

Insights: The absence of ratings (evaluated as 0) heavily left-skews the histogram and creates an unbalanced concentration of data, with mean close to 0. The graphs displays the rating distribution after the removal of rows (1'058'567) with rating equal to 0. Starting from this point, the dataset taken in consideration has these rows removed.





CATEGORY

Row ID	count
Education	129547
Tools	87045
Entertainment	84262
Music & Audio	80493
Books & Reference	67270
Personalization	57962
Lifestyle	55783
Business	52114
Finance	39752
Productivity	39208
Health & Fitness	34283
Shopping	33316
Puzzle	32833
Travel & Local	32358
Arcade	31397
Casual	30521
Sports	27218
Social	25917
News & Magazines	25726
Communication	25269
Food & Drink	23366
Photography	23006
Simulation	19687
Action	18840

Row ID	count
Action	18840
Adventure	16149
Maps & Navigation	15407
Medical	14809
Educational	12659
Video Players & Editors	10683
Auto & Vehicles	9787
Art & Design	9168
Role Playing	8614
Racing	7720
Trivia	7308
Board	7287
Card	6295
Strategy	6254
Word	5964
House & Home	5738
Weather	5543
Beauty	4205
Dating	4202
Casino	4044
Events	3923
Music	2891
Libraries & Demo	2724
Parenting	2313
Comics	2098

Nature: Categorical, Nominal

Description: classification based on app function and usage.

Categories: dataset contains 48 categories

Insights: The top 3 categories are Education (10% of total apps), Tools, and Entertainment. The category least present is Comics (2098).

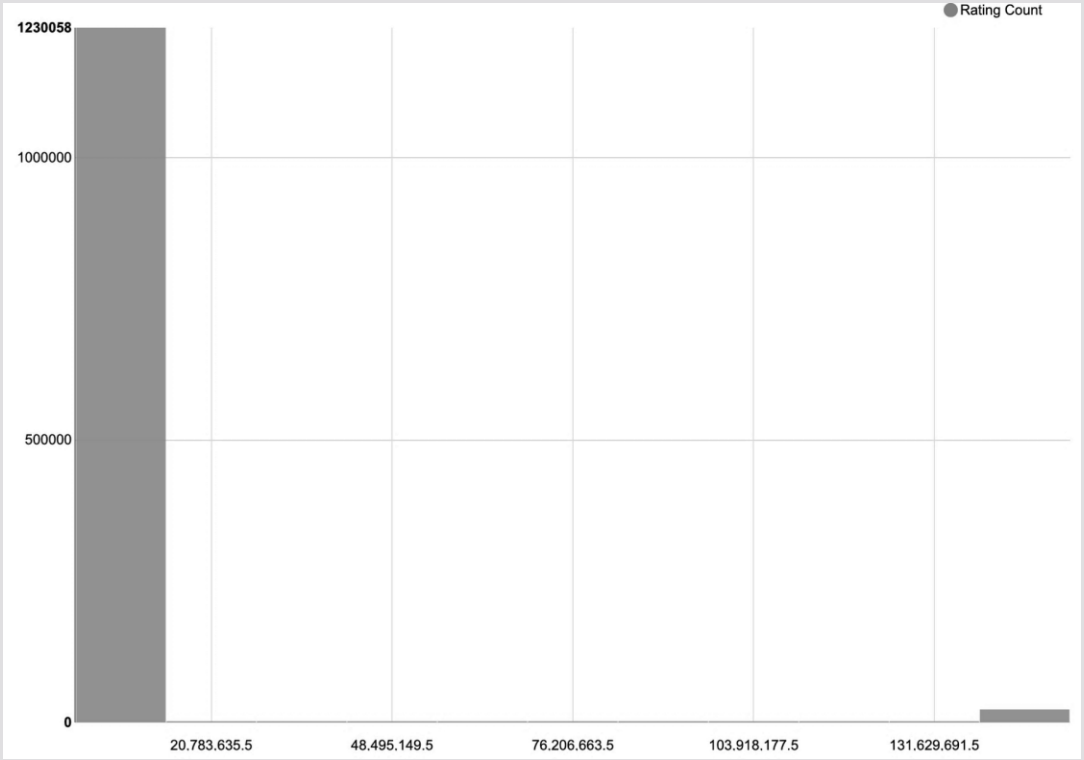
RATING COUNT

Nature: Numerical, Continuous

Description: the number of evaluations per app

Range: [5, 138'557'570]

Insights: The number of ratings is highly concentrated below 1 million.
Mean: 5'333
Standard deviation: 289'460.



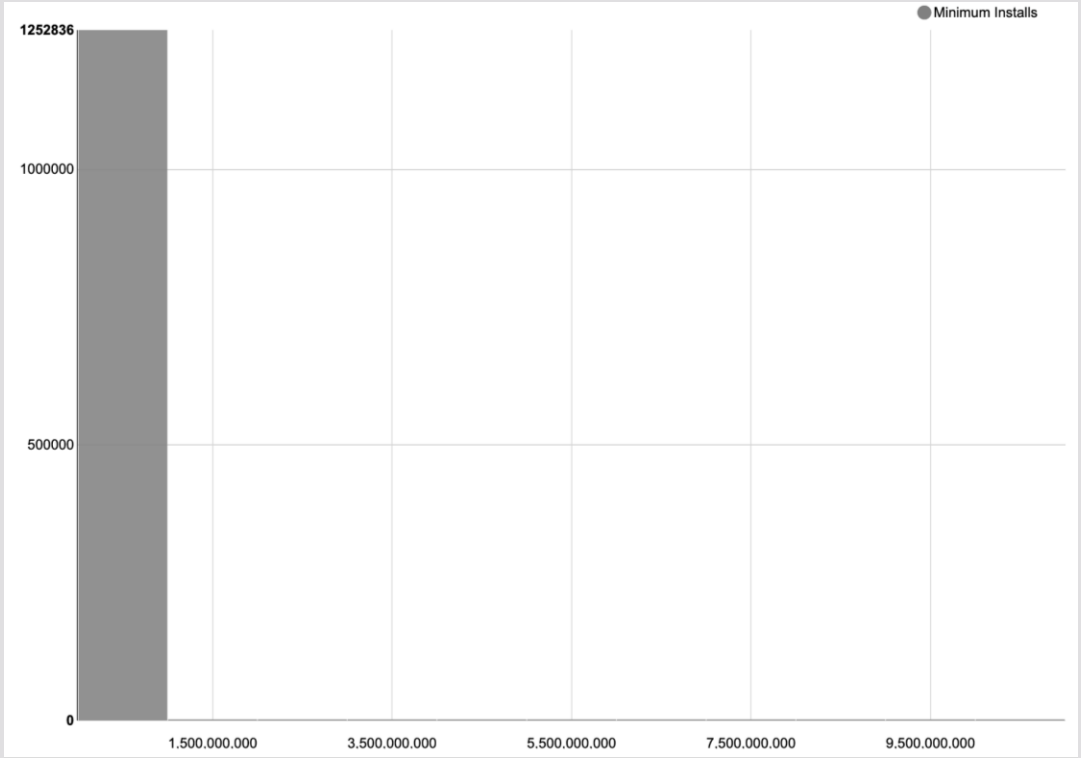
MINIMUM INSTALLS

Nature: Numerical, Discrete

Description: the lower bound of the downloads per application

Range: [0, 10'000'000'000]

Insights: The vast majority of minimum installs is less than one million, left skewing the distribution of the data, with mean equals to 338,119.



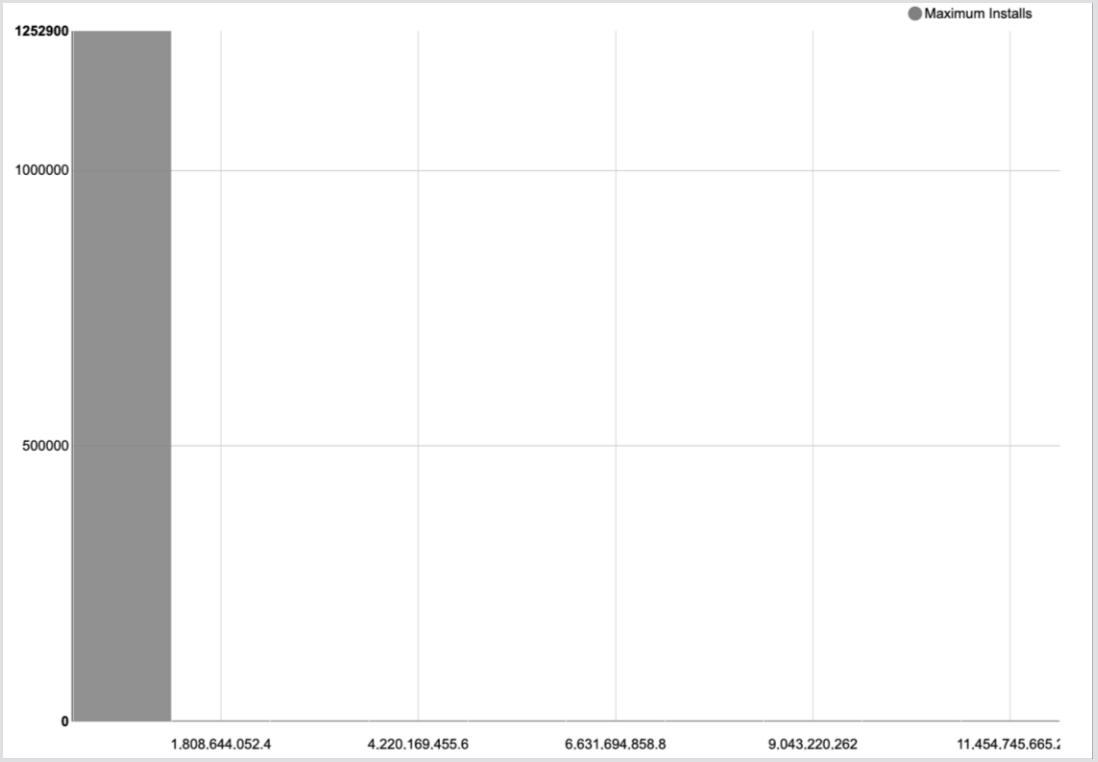
MAXIMUM INSTALLS

Nature: Numerical, Continuous

Description: the number of downloads received by each application

Range: [0, 12'057'627'016]

Insights: the majority of the applications were downloaded less than a million times, with mean equals to 590,163.



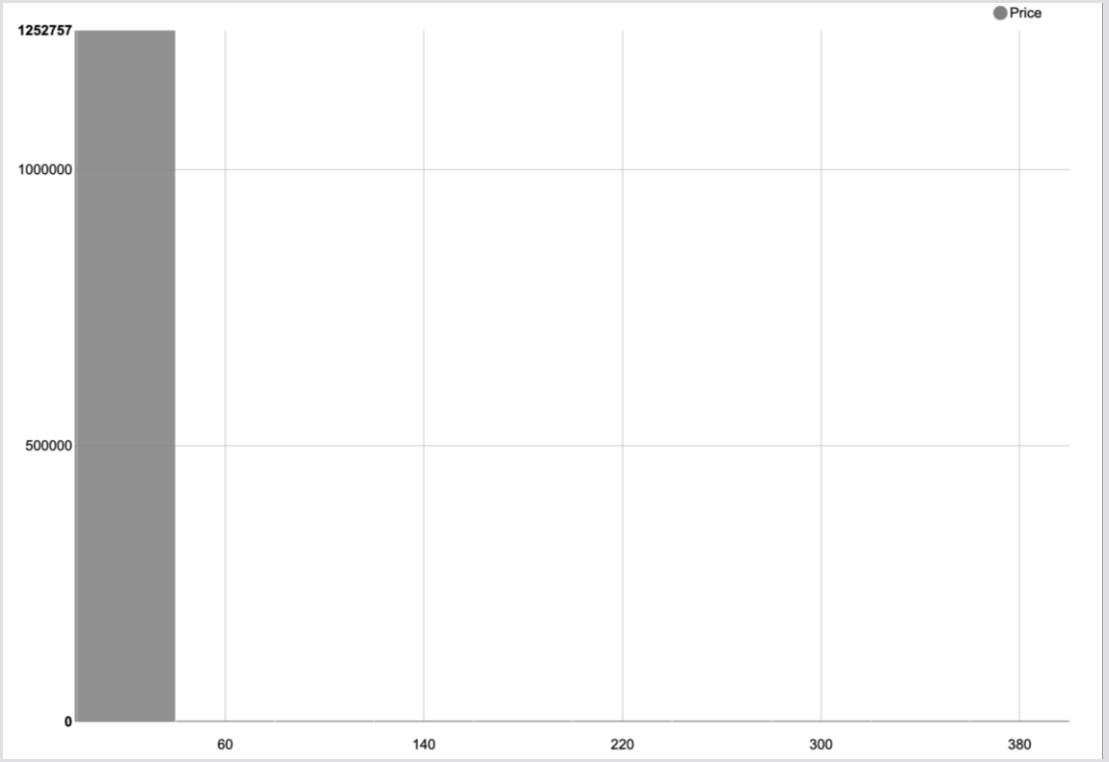
PRICE

Nature: Numerical, Continuous

Description: monetary value required to be paid in order to download an app

Range: [0.0, 399.99]

Insights: The most popular app type is free; hence, the majority of the apps present in the dataset do not have any download price, completely left-skewing the graph, with mean equals to 0.0926.



SIZE

Example of App Sizes

Size
11M
3.7M
6.4M
27M
492k
6.4M
46M
12M
5.8M
44k
26M
10M
7.2M

Nature: Numerical, Continuous

Description: the space occupied in the computer or mobile memory

Range: [3kB, 1.5GB]

Insights: The data are scaled differently varying in kilobytes (k), megabytes (M) and gigabytes (G). The majority of apps are in megabytes and only 11 of them are in gigabytes.

SCROLL

RELEASED

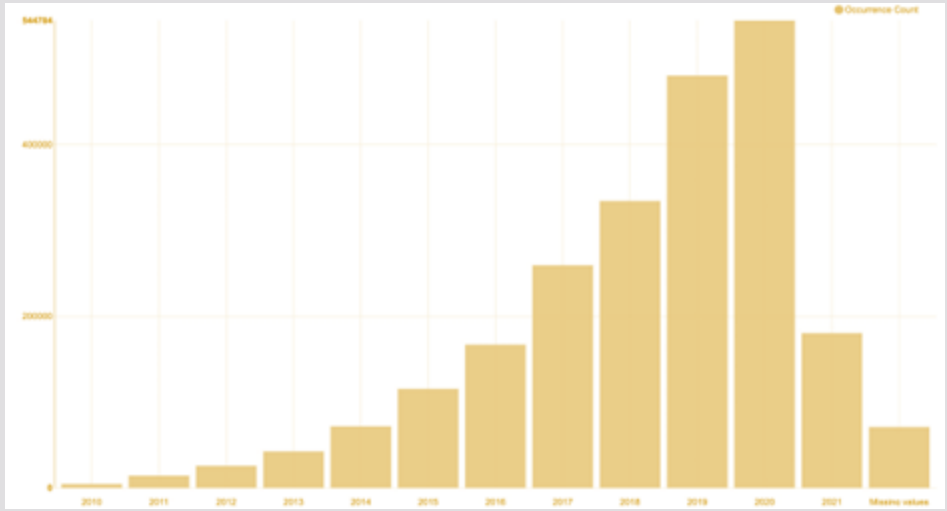
Row ID	count
?	71025
2010	4641
2011	14361
2012	25747
2013	42471
2014	71805
2015	115416
2016	167085
2017	259634
2018	334499
2019	480573
2020	544784
2021	180484

Nature: Numerical, Discrete

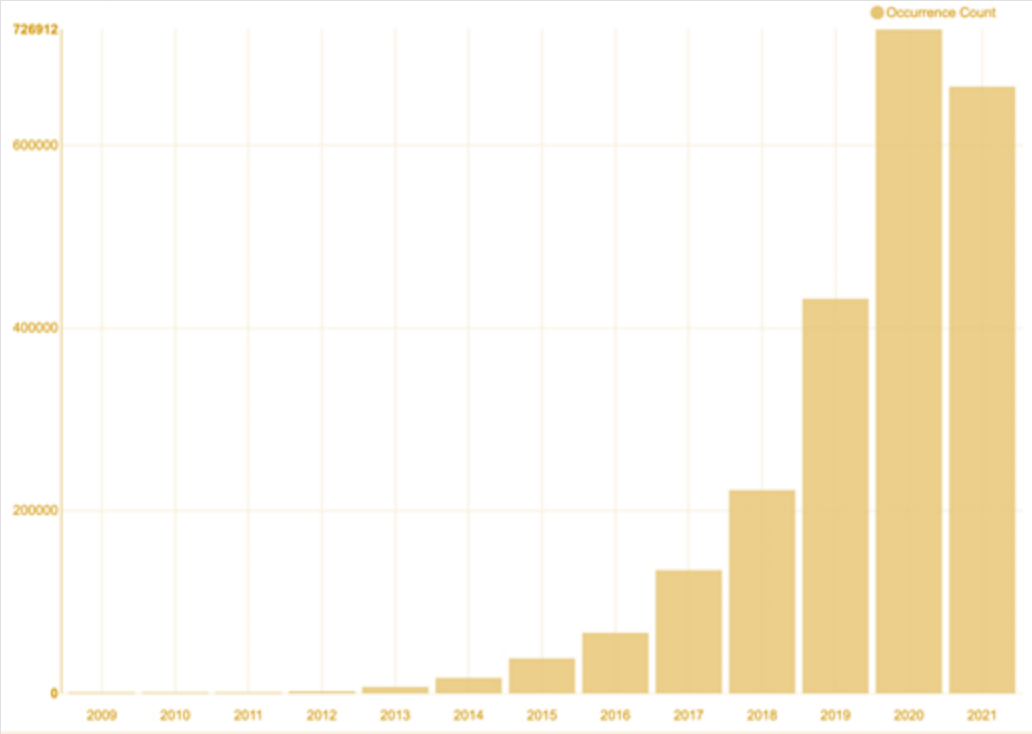
Description: launch date on Google Play Store

Range: [28/01/2010 - 16/06/2021]

Insights: The number of apps released on Play Store is increasing with time; as for 2021, the dataset considers only the period up to 16/06, which explains the drop clearly visible from the histogram.



LAST UPDATED



Year	Count
2009	13
2010	222
2011	898
2012	2327
2013	7008
2014	16950
2015	38290
2016	66301
2017	134839
2018	222523
2019	431963
2020	726912
2021	664279

Nature: Numerical, Discrete

Description: the date of the last update

Range: [31/12/2011 - 16/06/2021]

Insights: As years goes by, there is a tendency to release more and more updates to keep up with new system updates and inserting new functions. The updates are concentrated mainly in the last 3 years.

CURRENCY

Row ID	count
?	128
BRL	1
EUR	3
GBP	2
INR	1
KRW	1
PKR	1
RUB	1
SGD	1
TRY	1
USD	1252366
VND	1
XXX	451

Nature: Categorical, Nominal

Description: monetary currency

Categories: USD, EUR, GBP, ...

Insights: In the dataset are present only 13 apps sold in a different currency from USD. There are 579 entries that don't specify a currency.

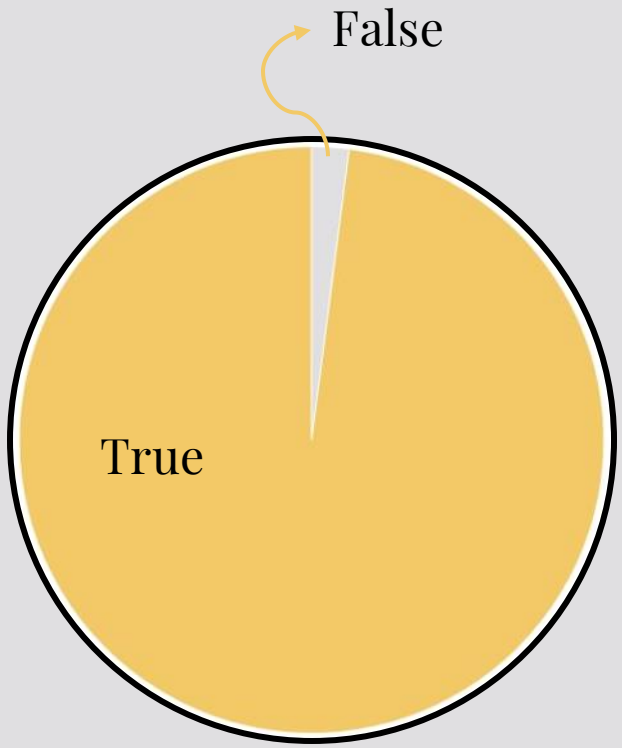
FREE

Nature: Categorical, Boolean

Description: whether the application is free or not

Categories: True, False

Insights: As displayed in the graph, the vast majority (98%) of apps is free. This is also evident from the distribution of the variable "Price" described above.



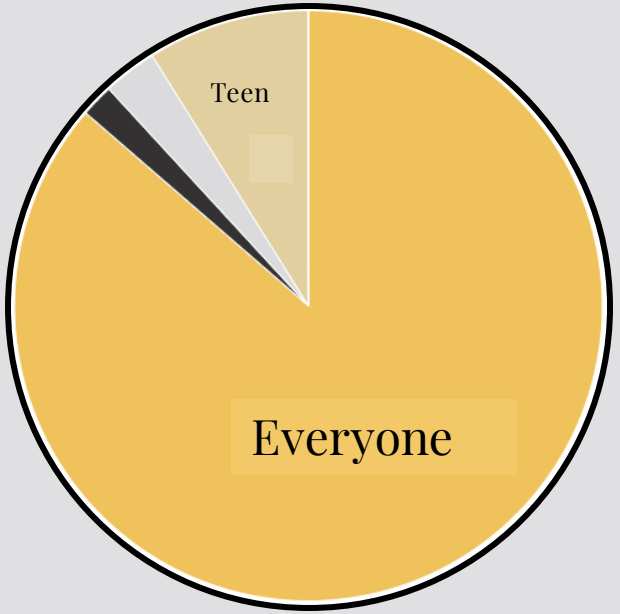
CONTENT RATING

Nature: Categorical, Nominal

Description: type of content displayed

Categories: Adults only 18+, Everyone, Everyone 10+, Mature 17+, Teen, Unrated

Insights: The majority (86%) of applications has no restriction ("Everyone"). There are present 125 apps listed as "unrated".



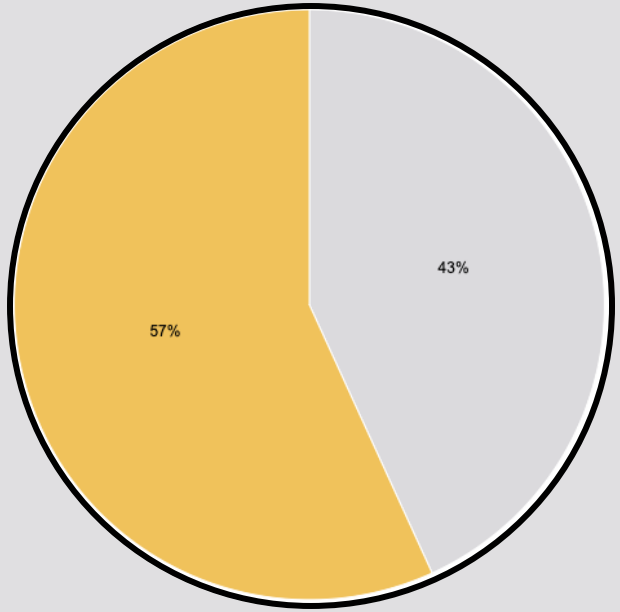
AD SUPPORTED

Nature: Categorical, Boolean

Description: whether there is the support for advertising or not

Categories: True, False

Insights: The graph shows that the sample is slightly unbalanced in favor of apps that support advertising (57% "True" - 43% "False")



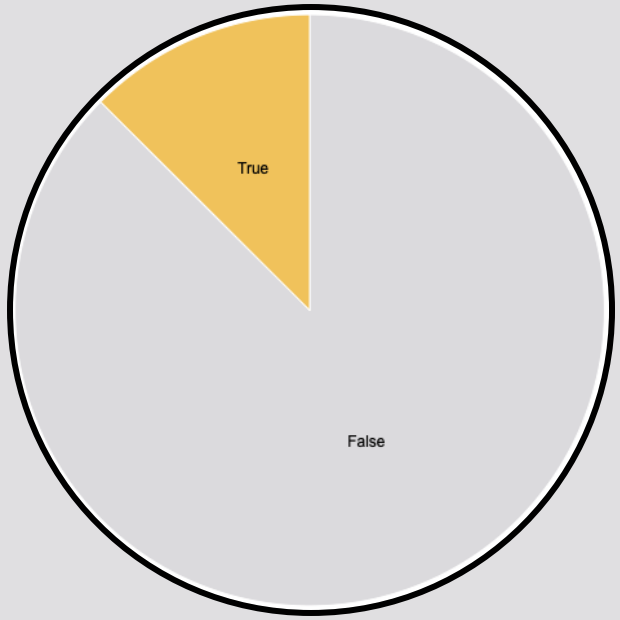
IN APP PURCHASES

Nature: Categorical, Boolean

Description: whether the app has the possibility of having purchases inside it

Categories: True, False

Insights: There is a noticeable majority (87%) of apps that does not offer the possibility of in app purchases



EDITORS CHOICE

Row ID	count
False	1164348
True	649

Nature: Categorical, Boolean

Description: whether the app was recommended by Google Play Store

Categories: True, False

Insights: The vast majority of apps were not recommended by Google Play Store

MENU

DATA PREPARATION

03

Data Cleaning
Missing Values

Data
Preparation
Feature
Engineering

Variable
Encoding

Outliers

VIEW MORE

SCROLL

Overview of Data Preparation

01

DATA CLEANING

In this section, data cleaning is executed.

Irrelevant variables are eliminated, and missing values are correctly handled.

02

DATA PREPARATION

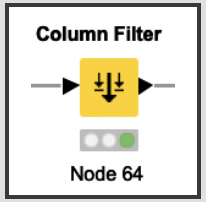
The preliminary data preparation steps for the Logistic Regression, Random Forest, and Gradient Boosting models are shown in this section.

03

OUTLIERS

Outliers are observed and studied over in this section

SCROLL



Irrelevant Variables

The dataset has a categorical feature called “Free” that indicates whether each application on the Google Play Store App is indeed free to download or not.
Therefore, for the scope of the analysis, the two variables “Price” and “Currency” can be eliminated, while Free, another variable that also falls into the “price category”, is kept. This represents a reasonable choice because the distributions of both variables are skewed to the left, with more than half of the apps costing 0.0 and most of the apps having as their currency US dollars. Moreover, the information that the two variables are conveying is beyond the scope of this study, irrelevant.

Price

Row ID	count
0.0	1227342
0.99	6039
1.99	3596
2.99	2296
1.49	2087
4.99	1637
3.99	1549
2.49	1287
3.49	771
5.99	542
9.99	539
4.49	467
6.99	447
5.49	347
7.99	321
8.99	173
19.99	162

Currency

Row ID	count
?	128
BRL	1
EUR	3
GBP	2
INR	1
KRW	1
PKR	1
RUB	1
SGD	1
TRY	1
USD	1252366
VND	1
XXX	451

SCROLL

Editor's Choice

File Edit Hilite Navigation View

Table "default" – Rows: 2 | Spec – Column: 1 | Properties | Flow Variables

Row ID	count
False	1164348
True	649

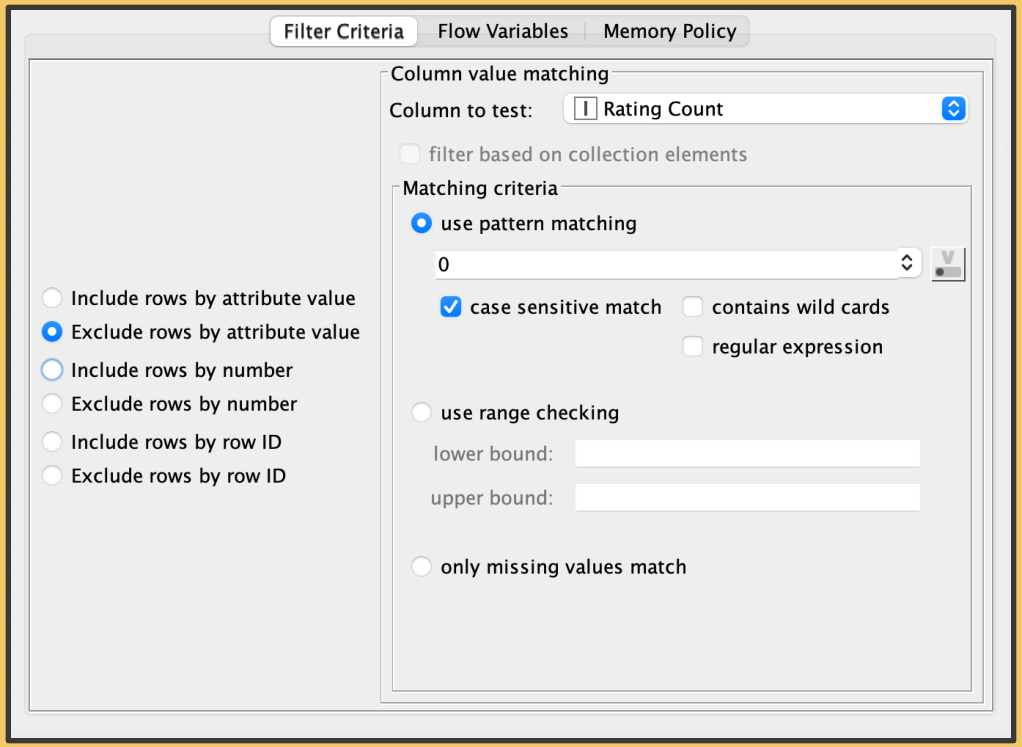
"Editors' Choice refers to the apps that introduce users to the best innovative, creative, and designer apps on Android. Google released an improved Editors' Choice section featuring app reviews organized by the editorial team of the Play Store." [1]

Nevertheless, the feature "Editor's Choice" is removed from the dataset as it is unnecessary for the construction of the models. A reason for said choice is that the distribution of this variable is particularly skewed, with most values (1,164,348) being "False", while only 649 apps were chosen by the editorial team.

[1] "Editors' Choice Android Apps and Games on Google Play." Google, Google, https://play.google.com/store/apps/editors_choice?hl=en_US&gl=US&pli=1

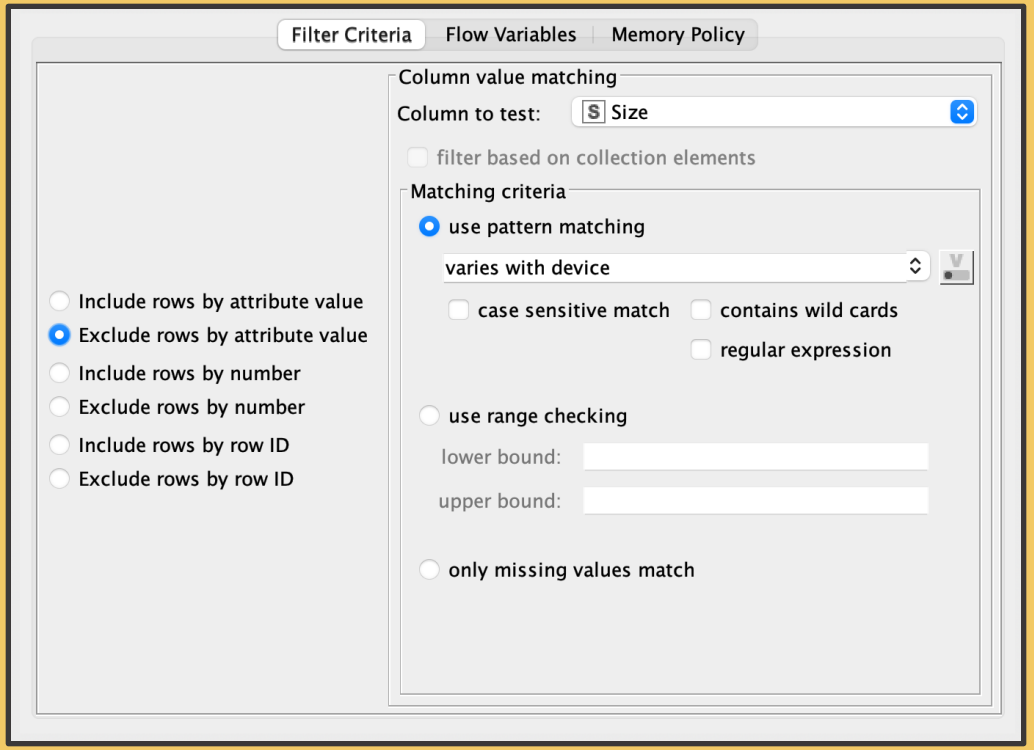
Rating Count

The values equating to zero from the feature “Rating Count” are excluded from the study using the node “Row Filter".



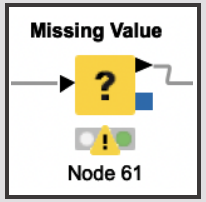
Size

In a similar way, the variable size had misplaced values that were not indicative of the actual measure of the apps but rather only provided information that the apps’ sizes varied depending on each device. Thus, this information was not satisfactory and needed to be removed.



DATA PREPARATION

Missing Values



String	Remove Row*
Number (double)	Remove Row*
Number (integer)	Remove Row*
Number (long)	Remove Row*

Row ID	Category	Rating	Rating Count	Minimum Installs	Maximum Installs	Free	Size	Released	Last Updated	Content Rating	Ad Su...	In App Purchases
Row1	ools	4.4	64	5000	7662	True	2.9M	May 21, 2020	May 06, 2021	Everyone	True	False
Row3	ommunication	5	5	10	19	True	1.8M	Sep 10, 2018	Oct 13, 2018	Everyone	True	False
Row6	oraries & Demo	4.5	12	1000	2567	True	2.5M	Sep 23, 2019	Sep 27, 2019	Everyone	True	False
Row7	estyle	2	39	500	702	True	16M	Jun 21, 2019	Jun 21, 2019	Everyone	False	False
Row9	rsonalization	4.7	820	50000	62433	True	3.5M	Sep 22, 2019	Oct 07, 2020	Everyone	True	False
Row10	icing	4.9	55	100	329	True	51M	Jul 30, 2020	Jul 30, 2020	Everyone	False	False
Row12	avel & Local	3.9	118	10000	37763	True	7.6M	Apr 3, 2018	Jun 11, 2021	Everyone	True	False
Row14	avel & Local	3.7	1572	10000	42856	True	29M	Sep 5, 2018	May 30, 2020	Everyone	False	False
Row15	od & Drink	4.2	16	1000	4313	True	12M	Apr 5, 2020	Mar 23, 2021	Everyone	False	False
Row16	avel & Local	3.4	5	500	949	True	2.9M	Nov 28, 2016	Oct 30, 2019	Everyone	False	False
Row17	ools	5	6	50	62	True	2.0M	Apr 24, 2019	May 05, 2019	Everyone	False	False
Row18	estyle	3.7	328	10000	31235	True	70M	Jul 1, 2020	May 26, 2021	Everyone	False	False
Row22	izzle	4.4	211	10000	15471	True	16M	Mar 13, 2020	May 11, 2020	Everyone	True	True
Row23	ntertainment	3.8	736	500000	646456	True	20M	Apr 21, 2019	Nov 13, 2020	Everyone	True	False
Row24	ools	4.2	35	10000	21124	True	2.4M	May 28, 2017	May 30, 2020	Everyone	True	False
Row25	cade	4.6	11	100	152	True	25M	Mar 5, 2018	Mar 26, 2018	Everyone	True	True
Row28	aps & Navigation	4.1	27	1000	2414	True	11M	Dec 15, 2013	Dec 15, 2013	Everyone	False	False
Row29	to & Vehicles	3.4	16	1000	3784	True	6.2M	Dec 4, 2019	May 11, 2020	Everyone	False	False
Row31	avel & Local	3.4	67	10000	30921	True	7.6M	Oct 13, 2016	Jun 29, 2020	Everyone	False	False
Row32	otography	2.5	21	5000	8984	True	6.2M	Nov 13, 2020	Nov 14, 2020	Everyone	True	False
Row34	aps & Navigation	2.3	420	50000	74510	True	2.6M	Nov 6, 2013	Jun 06, 2017	Everyone	True	False
Row35	avel & Local	3.7	143	10000	43422	True	11M	Feb 14, 2017	Dec 28, 2020	Everyone	False	False
Row36	alth & Fitness	2.1	33	10000	19889	True	3.7M	Sep 27, 2017	May 29, 2020	Everyone	True	False
Row40	opping	4.9	102	1000	4103	True	3.9M	Aug 29, 2019	Sep 01, 2019	Everyone	False	False
Row42	rsonalization	5	11	500	825	True	21M	May 30, 2020	May 30, 2020	Teen	True	False
Row43	cade	5	8	1000	3427	True	26M	Jul 8, 2019	May 01, 2021	Everyone	True	True
Row44	ools	2.7	420	50000	59427	True	4.9M	Jun 28, 2013	Jan 14, 2020	Everyone	True	False
Row45	ools	4.3	20	1000	1396	True	4.3M	Sep 2, 2019	Aug 16, 2020	Everyone	True	False
Row48	oks & Reference	3.8	13	1000	1079	True	2.8M	Jun 19, 2015	Apr 22, 2020	Everyone	True	False
Row49	orts	4.4	2781	100000	340980	True	38M	Apr 27, 2013	Mar 21, 2015	Everyone	True	True
Row55	ance	3.1	17	5000	7119	True	3.7M	Jul 5, 2017	Apr 11, 2020	Teen	False	False
Row56	ools	3.8	31	10000	13524	True	38M	Jul 3, 2019	Sep 06, 2019	Everyone	True	False
Row57	oks & Reference	4.7	30	10000	10958	True	16M	Jun 15, 2017	Jul 24, 2020	Everyone	True	False
Row60	estyle	4.3	266	10000	43836	True	7.1M	Oct 24, 2015	Apr 16, 2016	Everyone	False	True
Row61	oks & Reference	4.1	337	100000	107229	True	24M	Apr 14, 2012	May 16, 2020	Everyone	True	True
Row62	usic & Audio	4.7	12	500	682	True	11M	Sep 10, 2018	Mar 28, 2019	Teen	True	False
Row63	ws & Magazines	3.1	84	10000	27270	True	27M	Nov 23, 2015	Dec 21, 2020	Everyone	True	False
Row64	to & Vehicles	3.7	6	500	784	True	18M	Aug 16, 2017	Nov 14, 2019	Everyone	False	False
Row69	sual	4.9	8	100	205	True	39M	May 14, 2021	May 24, 2021	Everyone	True	False
Row71	oductivity	4.3	6	100	207	True	1.2M	Feb 7, 2017	Feb 08, 2017	Everyone	False	False
Row72	lucation	5	5	10	18	True	38M	Oct 15, 2020	Jun 13, 2021	Everyone	False	False
Row76	t & Design	3.9	9	500	787	True	22M	Nov 8, 2020	Nov 08, 2020	Everyone	True	False
Row77	ance	4.8	2663	10000	49190	True	4.1M	May 3, 2016	Apr 16, 2021	Everyone	True	False
Row78	cade	3.3	7	5000	5554	True	50M	Mar 19, 2021	Mar 19, 2021	Everyone	True	False
Row81	avel & Local	4.9	7	1000	1391	True	127k	Apr 30, 2014	Mar 12, 2019	Everyone	False	False
Row82	ouse & Home	2.7	205	50000	53898	True	8.4M	Jan 7, 2019	Jun 15, 2021	Everyone	False	False

This dataset has more than one million values; therefore, there is no need to concern about directly removing all the missing values.

To gain insights from the data, the dataset is fed to statistics node 61.

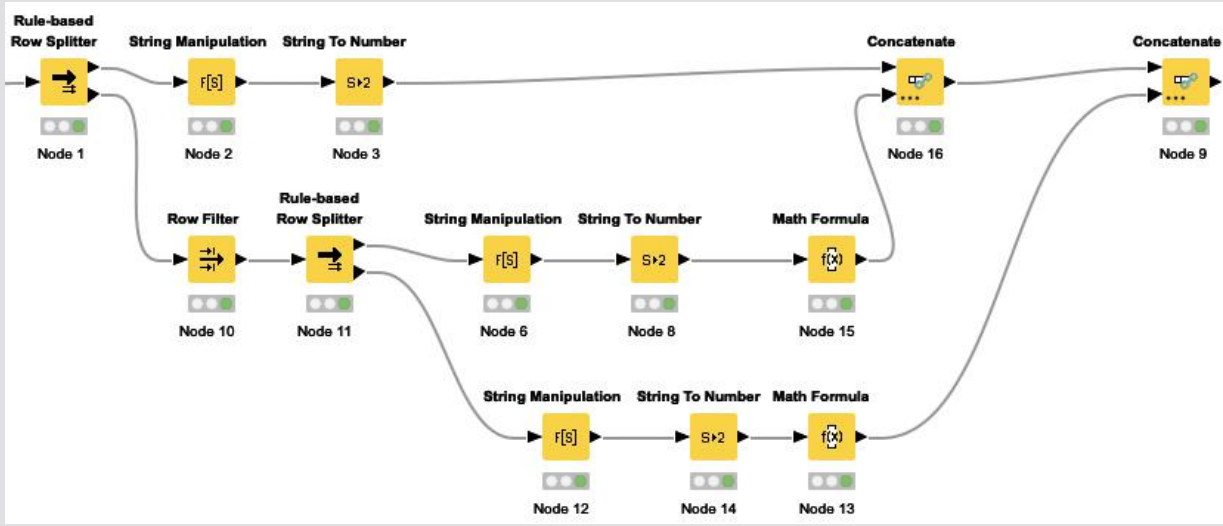
This reveals a total of 49,803 missing values.

SCROLL

Fix "size" column
Node 27

Size

Initially, for the feature of size, values were partitioned into megabytes, kilobytes, and gigabytes, so to achieve a common measure, the values reported in kilobytes and gigabytes needed to be transformed into **megabytes (MB)**. To begin, the **Rule-Based Row Splitter** separates the data into those that are MB and those that are not. Then, there is a further separation between KB and GB, and according to the conversion, they are both **adapted** to megabytes utilizing the math formulas below:



GB → MB

Expression

```
1 $Size$ * 1000
```

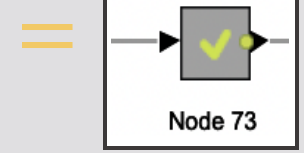
KB → MB

Expression

```
1 $Size$ / 1000
```

Finally, the "MB string" is removed, and what is left is the singular numerical value indicating the size for each application.

SCROLL



EVALUATION OF THE CONTENTS

As visible from the window below, the distribution of the "*Content Rating*" feature is unbalanced since most of the applications (1,007,114) are considered for "Everyone". The "Unrated" values are removed.
Whereas, for the sake of simplicity and clarity, all the other Ratings are merged into one renamed as "Restriction".

Row ID	count
Adults only 18+	81
Everyone	1007114
Everyone 10+	20340
Mature 17+	34731
Teen	102610
Unrated	121

Row ID	count
Everyone	1007114
Restriction	157762

DATA PREPARATION

Feature Engineering

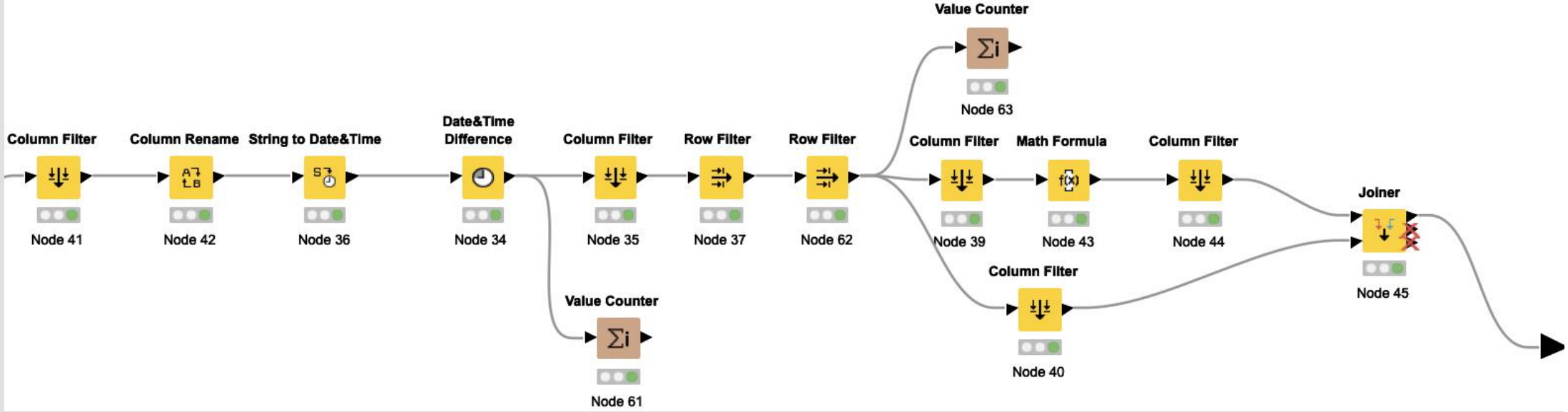
The dataset provided both the maximum and minimum numbers of installs, two continuous variables. The minimum number is dropped from the dataset, and only the maximum number of installs is considered the reference value for installs since it allows for a better interpretation.

In addition, the market for the tech industry's applications has undergone constant change in the last six years, since 2015. Thus, the decision to only include applications from 2015 stems from a deeper understanding of this market.
The release date of each application is used to calculate the number of years since the app was first released on the Google Play Store. Once the “age” of each app is calculated, applications with less than 1 year and more than 6 years are excluded. The ratio “Installs per year” is built and inserted.

* The date 06/16/2015 was used as the lower bound for the creation of this variable.
Whereas the date used as the upper bound for both this variable and the one described in the next slide is: *16/06/2021*.

SCROLL

I N S T A L L S



The variable “Installs” on its own does not accurately serve as a true representation for the purpose of the analysis proposed in this report. Undoubtedly, older applications have been on the Google Play Store for a longer time, and therefore, it would not be fair to compare their number of installs with that of an application that has only been created for a few years. For this reason, a new variable is created:

INSTALLS PER YEAR =

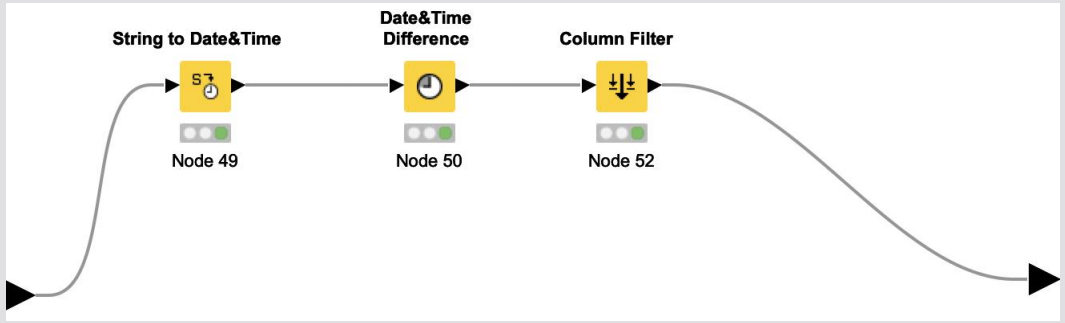
Expression

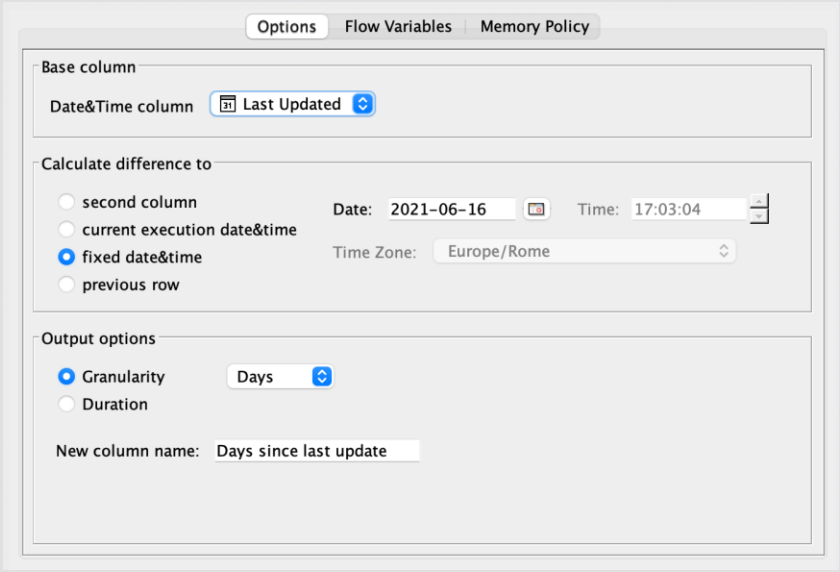
```
1 $Installs$/$Years since release$
```

SCROLL

I N S T A L L S

DAYS SINCE LAST UPDATE

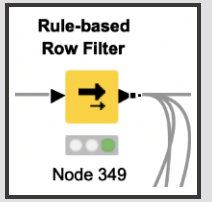




It follows from a logical standpoint that applications with the recent updates would be more likely to receive positive feedback. As a result, the date "06/16/2021" is used as the reference for the most recent "Last Update," and a new variable, "Days Since Last Update," is created. This variable constructed is used to compare more easily the applications thanks to this index that is equal to:

reference date chosen – date of the last update

The higher the value of this index, the less often the app is updated.



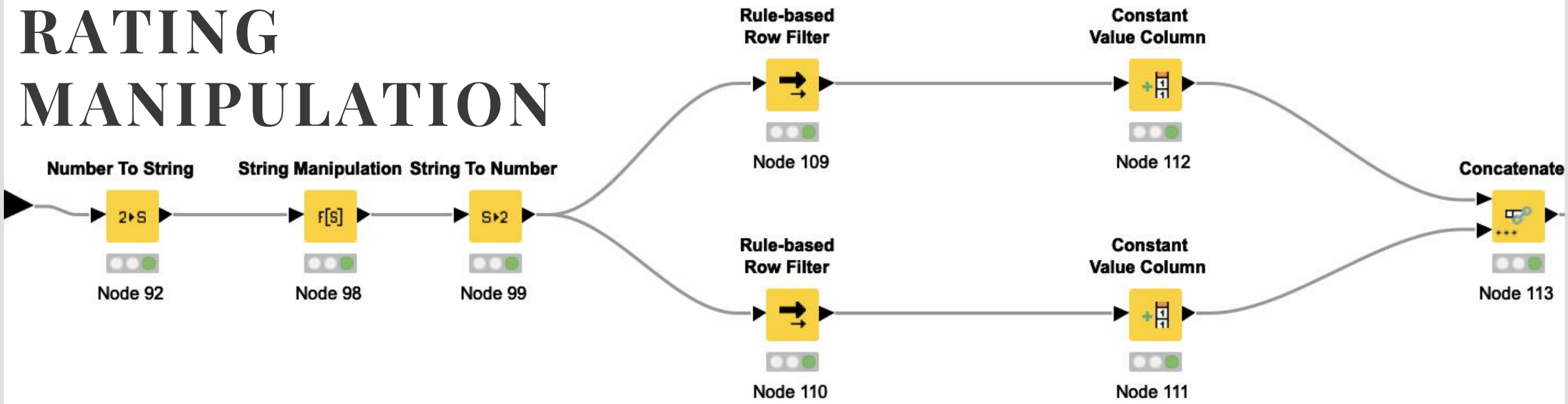
Row ID	count
Education	91826
Music & Audio	59070
Entertainment	54363
Tools	52626
Books & Reference	47793
Personalization	38766
Lifestyle	38267
Business	34726
Productivity	24066
Finance	23742
Health & Fitness	23116
Shopping	21123
Travel & Local	20566
Arcade	19364
Puzzle	19312
Food & Drink	17514
Photography	17273
Casual	17035
Social	16781
Communication	16465
Sports	16290
News & Magazines	15295
Simulation	12392
Action	11498
Adventure	10337
Maps & Navigation	9551
Medical	9216

Category

In total, there are 48 categories, but only the first 8 are relevant enough in terms of the number of values they each have. Accordingly, only eight "*Categories*" are kept:

Education, Music & Audio, Entertainment, Tools, Books & References, Personalization, Lifestyle, Business

RATING MANIPULATION



Rating is the target variable of the analysis. This variable is numeric initially, and it is comprised of all the values from 0 to 5 (star rating). However, even from a managerial angle, the most captivating concept would be to observe whether an application is successful or not rather than analyzing each value, e.g., there is little to no added value in comparing an application that has a rating of 3.2 with one of 3.3; such a study would result in unsatisfactory outcomes. Therefore, for efficiency purposes, the applications are categorized into two different groups: A apps and B apps.

A apps

Class A applications are apps that received a rating greater than or equal to 4.

Expression

```
// enter ordered set of rules, e.g.:
// $double column name$ > 5.0 => FALSE
// $string column name$ LIKE "*blue*" => FALSE
// TRUE => TRUE
$Rating$ >=40 => TRUE
```

B apps

Whereas B applications are those apps that are less successful with a rating that is lower or equal to 2.

Expression

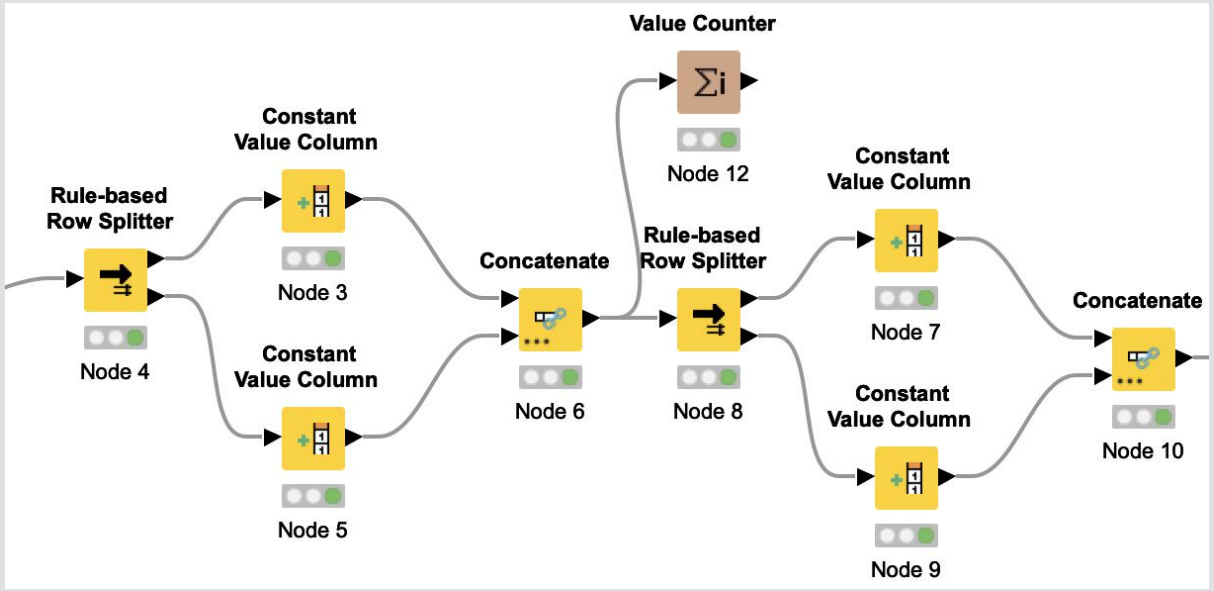
```
// enter ordered set of rules, e.g.:
// $double column name$ > 5.0 => FALSE
// $string column name$ LIKE "*blue*" => FALSE
// TRUE => TRUE
$Rating$ <= 20 => TRUE
```

* Instead of using 2.0 and 4.0, the Rating values are transformed to 20 and 40 respectively for simplicity.

Row ID	count
A	294590
B	5587

Finally, the apps that were found to be in the middle, with ratings strictly ranging between 2 and 4, are removed from the dataset so as to create a better gap and achieve a clearer differentiation.

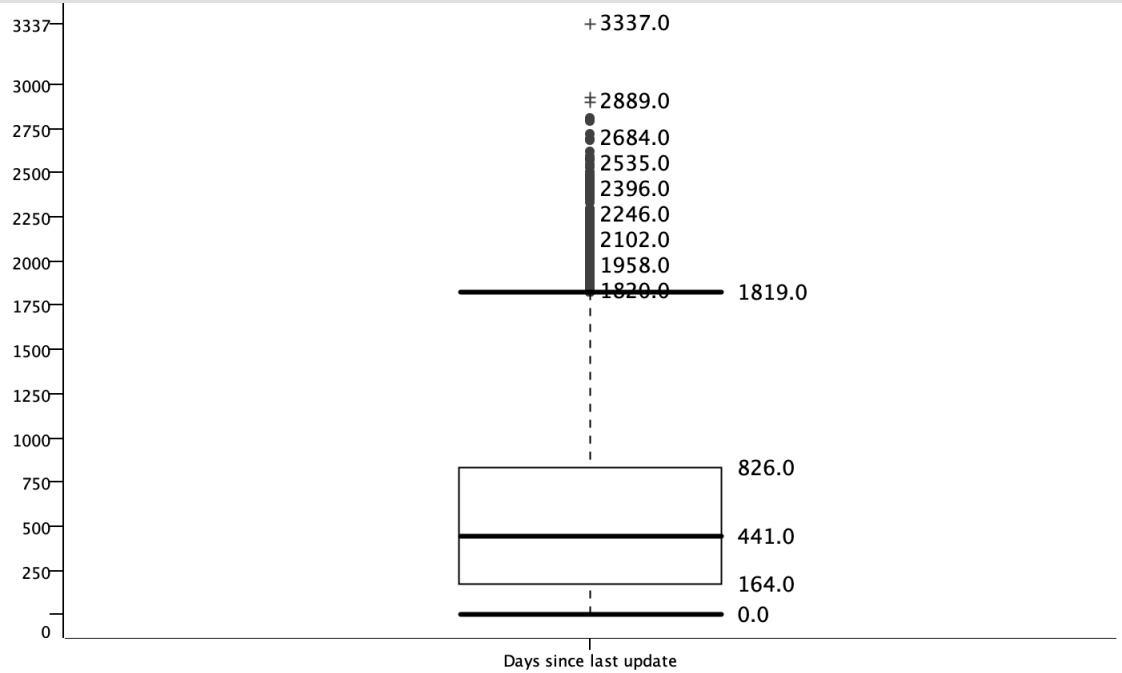
Variable Encoding



One portion of the Data Preparation section is the transformation of categorical dummy variables into integers: 0s and 1s. All the values from the variables “Supported Ad”, “Purchases in Apps”, and “Free” are switched from True to 1 and from False to 0. In addition, the values in “Content Rating” is replaced from Everyone to 1 and from Restriction to 0.

Outliers Detection

Row ID	Outlier column	Member count	Outlier count	Lower bound	Upper bound
Row0	Installs per year	300177	44032	-14,371.5	25,368.5
Row1	Rating Count	300177	45773	-155.5	288.5
Row2	Size	300177	28103	-18.75	43.25
Row3	Days since last update	300177	7504	-829	1,819



To gain insights from the data, the dataset is entered into the Outliers Detection node. This reveals that all the continuous variables have outliers:

- "Installs per year" has 14.6% of its values as outliers,
- "Rating Count" with 15.2% being outliers,
- "Size" with 9.4%,
- and "Days since last update" with 2.5%.

* The Box Plot is an example to show the numeric outliers for the "Days since last update" variable.

MENU

DATA DESCRIPTION

Bivariate Analysis

04

VIEW MORE

SCROLL

Overview of the Bivariate Analysis

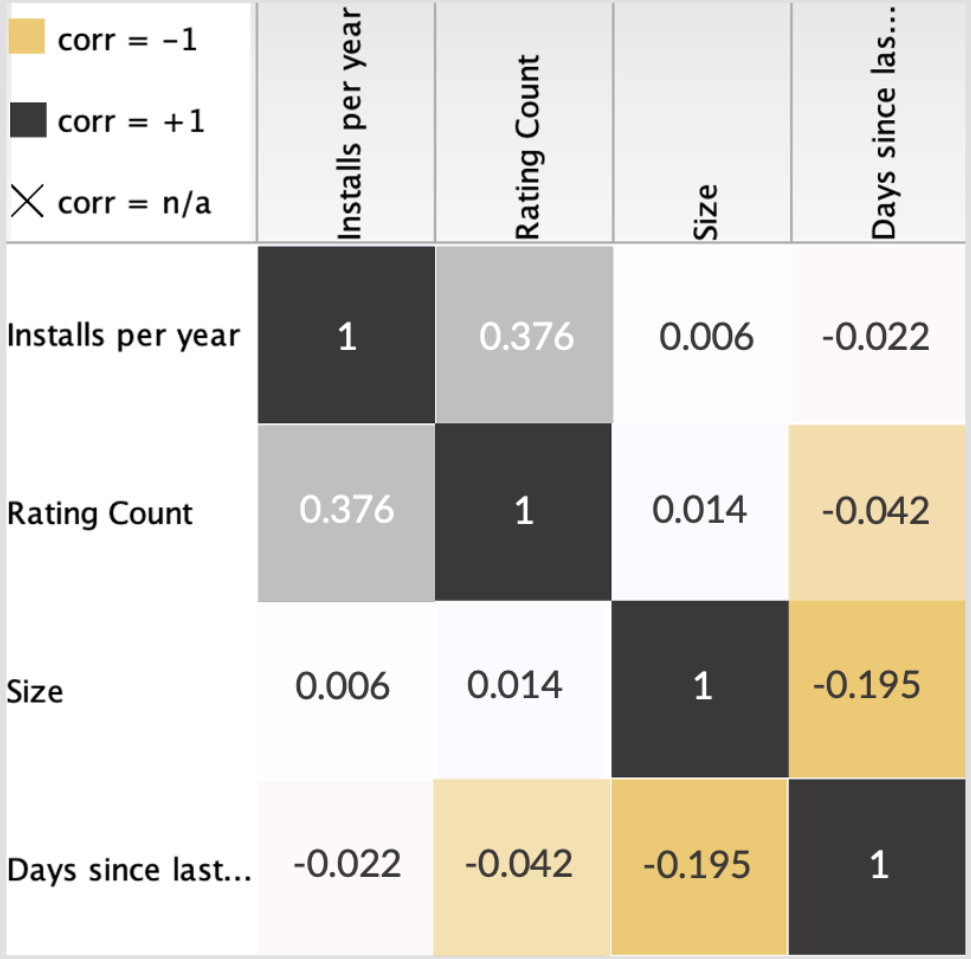
A modest p-value is produced when dealing with a vast dataset. Several additional tests are carried out to assess the magnitude of the relationship between different variable categories. The correlation matrix is used to assess the strength of the relationship between continuous variables

The unique elevated value of linear correlation derives from the relationship present between rating count and install; this outcome is expected as, logically, the more downloads an app achieves, the more ratings it will receive. The remaining variables are weakly correlated, with the relationship between installs and rating count standing out as the sole exception.

Furthermore, to evaluate the intensity of relations among categorical variables, the Cramer's V test is implemented, resulting in a value representing the correlation; the visual interpretation, instead, is displayed in the cross-tabulation section. None of the variables exhibit signs of high correlation to the dependent variable in the model: rating. An example of emerging correlated variables are ad supported and category, which could be explained by the fact that only certain category admit advertising.

Difference in means is computed to examine the correlation in between the categorical and continuous variables, resorting to One Way Anova. In addition, the difference in variance is implemented though the Levene test; the graphical representation of the relation between these two class of variables could be found in the boxplots, which could be considered as the equivalent of the cross tabulation for different categories.

Bivariate Analysis of continuous variables



Row ID	First column name	Second column name	p value
Row0	Installs per year	Rating Count	0.0
Row1	Installs per year	Size	6.859364032041348E-4
Row2	Installs per year	Days since last update	9.14072515603111E-34
Row3	Rating Count	Size	3.219646771412954E-14
Row4	Rating Count	Days since last update	3.1220247155667363E-116
Row5	Size	Days since last update	0.0

Correlation Matrix

Bivariate Analysis of categorical variables

Bivariate Analysis of Categorical Variables

Categorical Variables	Cramer's V
Rating – Ad Supported	0.088 low
Rating – Content Rating	0.017 low

Cross tabulation

Frequency Row Percent	A	B	Total
0	107,483	3,794	111,277
	96.5905%	3.4095%	
1	187,107	1,793	188,900
	99.0508%	0.9492%	
Total	294,590	5,587	300,177

Frequency Row Percent	A	B	Total
0	32,128	390	32,518
	98.8007%	1.1993%	
1	262,462	5,197	267,659
	98.0584%	1.9416%	
Total	294,590	5,587	300,177

SCROLL

Bivariate Analysis of Categorical Variables

Categorical Variables	Cramer's V	Cross tabulation
Rating – Category	0.096 low	

Frequency Row Percent	A	B	Total
Books & Reference	38,693	259	38,952
	99.3351%	0.6649%	
Business	22,099	797	22,896
	96.519%	3.481%	
Education	67,047	1,132	68,179
	98.3397%	1.6603%	
Entertainment	33,617	784	34,401
	97.721%	2.279%	
Lifestyle	26,262	768	27,030
	97.1587%	2.8413%	
Music & Audio	47,072	340	47,412
	99.2829%	0.7171%	
Personalization	30,821	110	30,931
	99.6444%	0.3556%	
Tools	28,979	1,397	30,376
	95.401%	4.599%	
Total	294,590	5,587	300,177

SCROLL

Bivariate Analysis of Categorical Variables

Categorical Variables	Cramer's V	Cross tabulation
Rating – Free	0.003 low	
Rating – In App Purchases	0.005 low	

Frequency Row Percent	A	B	Total
0	4,927	77	5,004
	98.4612%	1.5388%	
1	289,663	5,510	295,173
	98.1333%	1.8667%	
Total	294,590	5,587	300,177

Frequency Row Percent	A	B	Total
0	273,966	5,254	279,220
	98.1183%	1.8817%	
1	20,624	333	20,957
	98.411%	1.589%	
Total	294,590	5,587	300,177

SCROLL

Bivariate Analysis of Categorical Variables

Categorical Variables	Cramer's V
Ad Supported – Free	0.15 medium
Ad Supported – Content rating	0.12 medium

Cross tabulation

Frequency Row Percent	0	1	Total
0	4.823	181	5.004
	96,3829%	3,6171%	
1	106.454	188.719	295.173
	36,065%	63,935%	
Total	111.277	188.900	300.177

Frequency Row Percent	0	1	Total
0	6.615	25.903	32.518
	20,3426%	79,6574%	
1	104.662	162.997	267.659
	39,1027%	60,8973%	
Total	111.277	188.900	300.177

SCROLL

Bivariate Analysis of Categorical Variables

Categorical Variables	Cramer's V	Cross tabulation
Ad Supported – Category	0.405 strong	

Frequency Row Percent	0	1	Total
Books & Reference	7.907	31.045	38.952
	20,2993%	79,7007%	
Business	20.057	2.839	22.896
	87,6005%	12,3995%	
Education	33.513	34.666	68.179
	49,1544%	50,8456%	
Entertainment	8.957	25.444	34.401
	26,037%	73,963%	
Lifestyle	12.611	14.419	27.030
	46,6556%	53,3444%	
Music & Audio	9.582	37.830	47.412
	20,2101%	79,7899%	
Personalization	5.039	25.892	30.931
	16,2911%	83,7089%	
Tools	13.611	16.765	30.376
	44,8084%	55,1916%	
Total	111.277	188.900	300.177

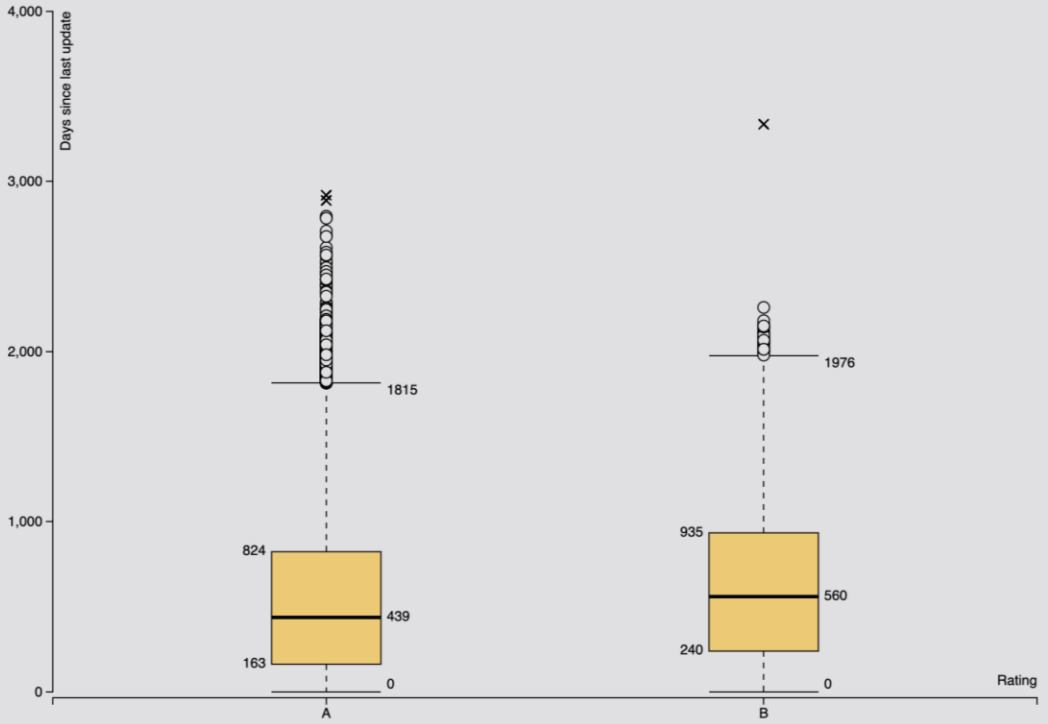
SCROLL

Bivariate analysis of continuous and categorical variables

Variable

Rating
–
Days since last update

Boxplot



SCROLL

Test Statistics

Descriptive Statistics

	Group	N	Missing	Missing Group	Mean	Std. Deviation
Days since last update	B	5587	0	0	635.9454	485.026
Days since last update	A	294590	0	0	562.6473	493.0564
Days since last update	Total	300177	0	0	564.0116	493.0069

Levene Test

	F	df 1	df 2	p-Value
Days since last update	0.1491	1	300175	0.6994

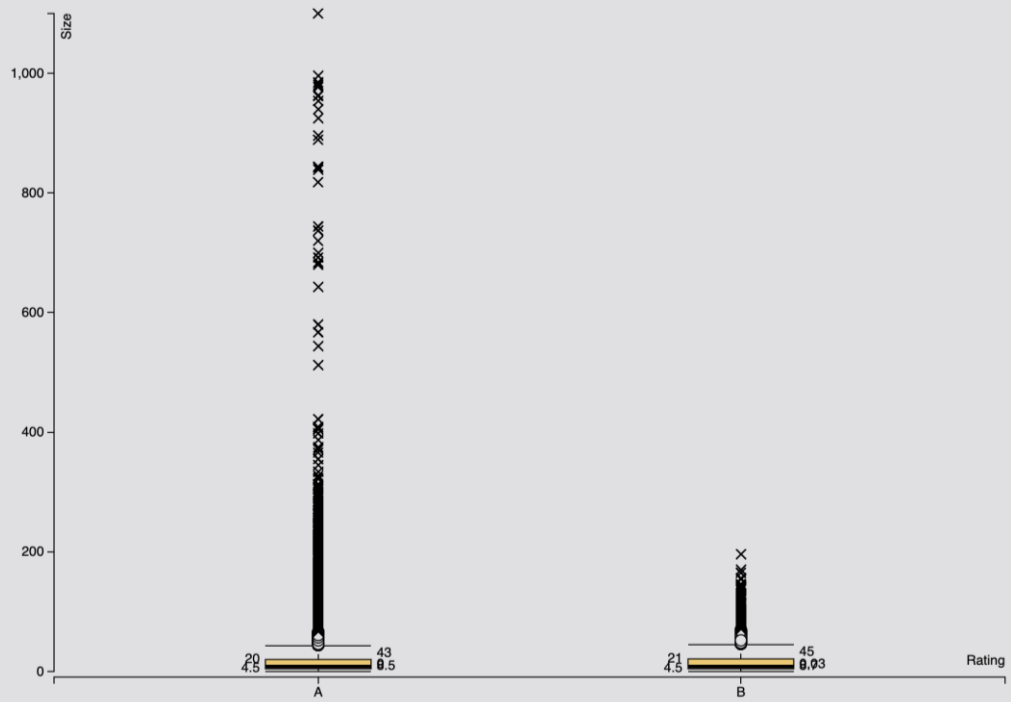
Anova

	Source	Sum of Squares	df	Mean Square	F	p-value
Days since last update	Between Groups	29,458,066.6811	1	29,458,066.6811	121.2473	0.0
Days since last update	Within Groups	7.29E10	300175	242,958.4607		
Days since last update	Total	7.30E10	300176			

Variable

Rating
–
Size

Boxplot



SCROLL

Test Statistics

Descriptive Statistics

	Group	N	Missing	Missing Group	Mean	Std. Deviation
Size	B	5587	0	0	16.7992	20.4365
Size	A	294590	0	0	16.9301	22.8489
Size	Total	300177	0	0	16.9276	22.8063

Levene Test

	F	df 1	df 2	p-Value
Size	2.247	1	300175	0.1339

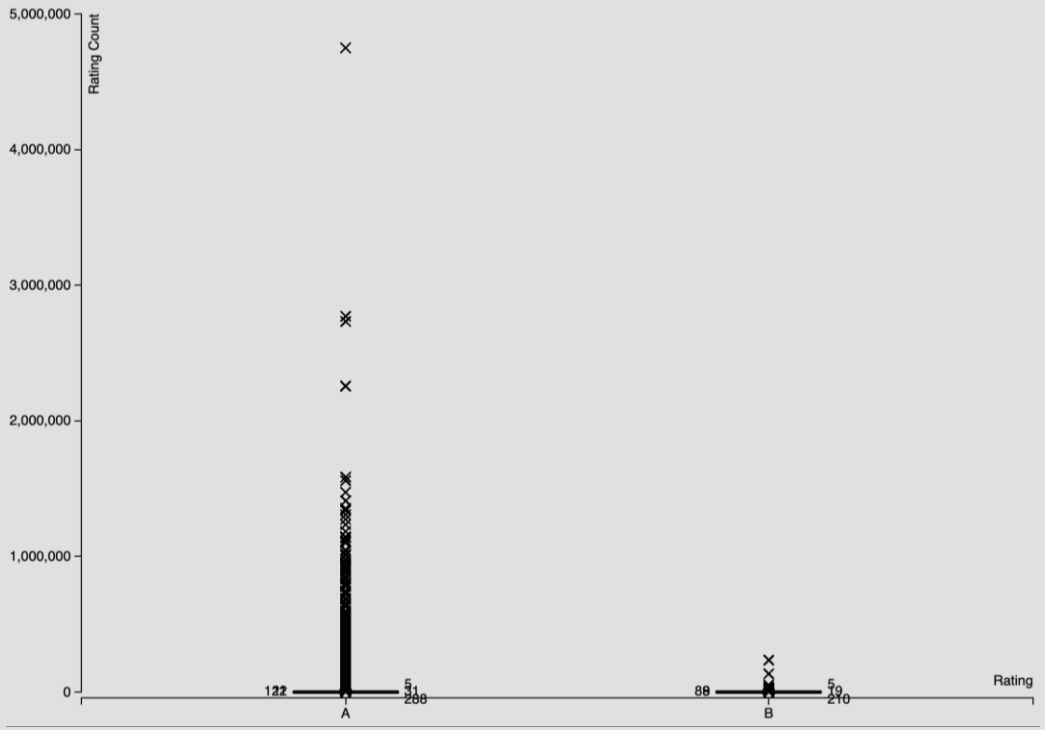
Anova

	Source	Sum of Squares	df	Mean Square	F	p-value
Size	Between Groups	93.9412	1	93.9412	0.1806	0.6708
Size	Within Groups	1.56E8	300175	520.1304		
Size	Total	1.56E8	300176			

Variable

Rating
–
Rating Count

Boxplot



SCROLL

Test Statistics

Descriptive Statistics

	Group	N	Missing	Missing Group	Mean	Std. Deviation
Rating Count	B	5587	0	0	321.9672	5,010.7522
Rating Count	A	294590	0	0	1,248.5821	21,793.0585
Rating Count	Total	300177	0	0	1,231.3356	21,600.4764

Levene Test

	F	df 1	df 2	p-Value
Rating Count	32.3936	1	300175	1.26E-8

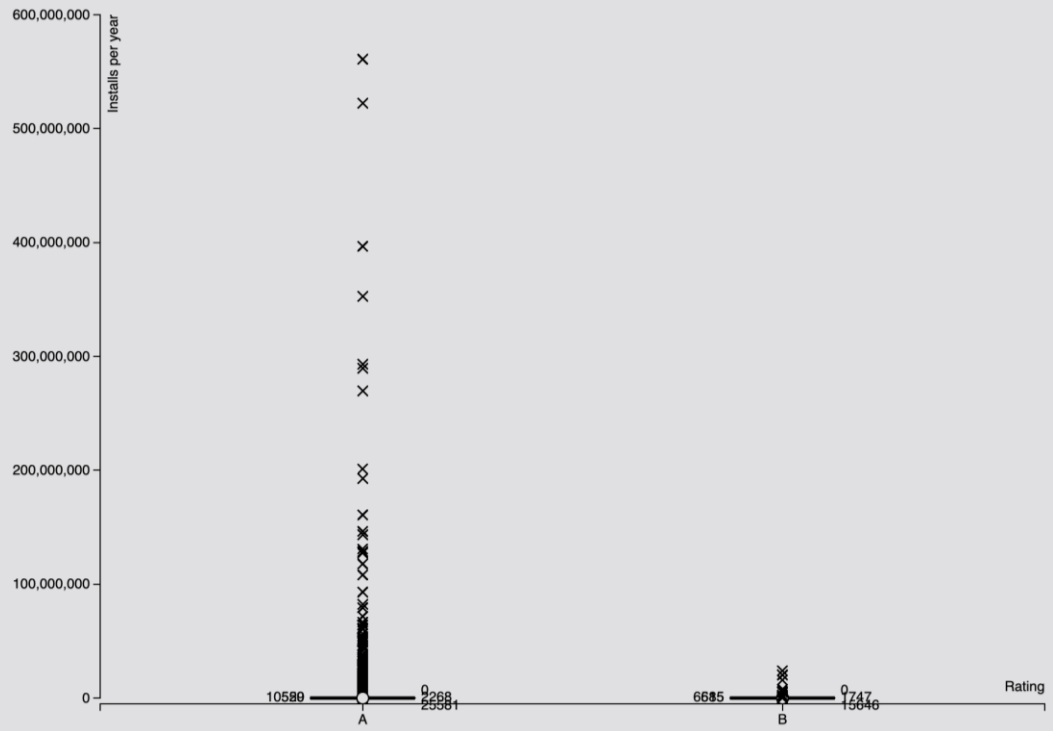
Anova

	Source	Sum of Squares	df	Mean Square	F	p-value
Rating Count	Between Groups	4.71E9	1	4.71E9	10.0903	0.0015
Rating Count	Within Groups	1.40E14	300175	4.67E8		
Rating Count	Total	1.40E14	300176			

Variable

Rating
–
Installs
per year

Boxplot



SCROLL

Test Statistics

Descriptive Statistics

	Group	N	Missing	Missing Group	Mean	Std. Deviation
Installs per year	B	5587	0	0	35,387.1498	575,687.3355
Installs per year	A	294590	0	0	73,142.4232	2,267,680.4526
Installs per year	Total	300177	0	0	72,439.7088	2,247,855.8701

Levene Test

	F	df 1	df 2	p-Value
Installs per year	4.6306	1	300175	0.0314

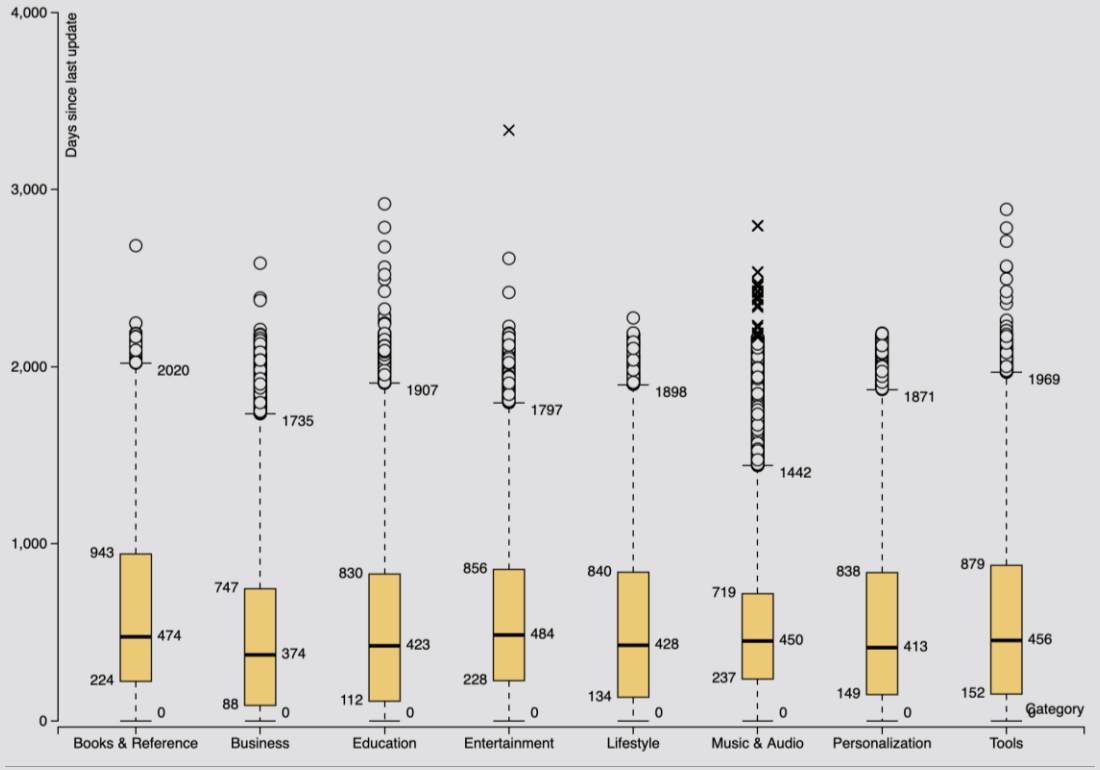
Anova

	Source	Sum of Squares	df	Mean Square	F	p-value
Installs per year	Between Groups	7.82E12	1	7.82E12	1.5468	0.2136
Installs per year	Within Groups	1.52E18	300175	5.05E12		
Installs per year	Total	1.52E18	300176			

Variable

Category
–
Days

Boxplot



SCROLL

Test Statistics

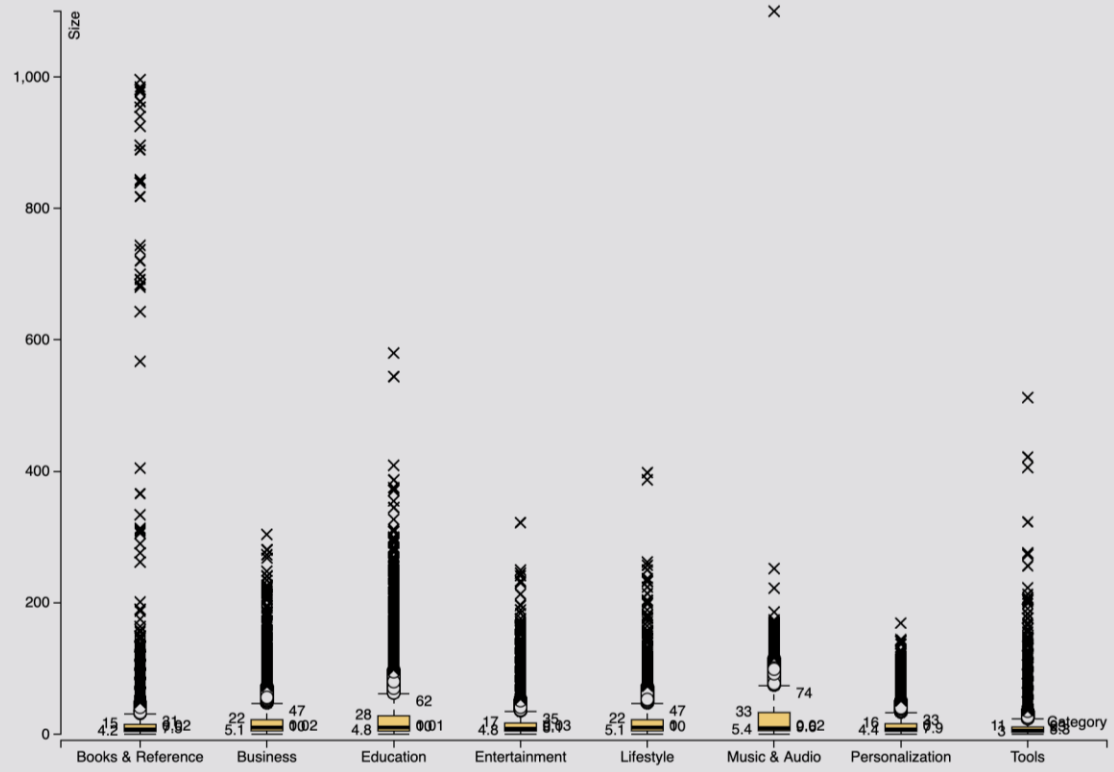
Descriptive Statistics

	Group	N	Missing	Missing Group	Mean	Std. Deviation
Days since last update	Entertainment	34401	0	0	599,1521	481,0856
Days since last update	Lifestyle	27030	0	0	560,6836	504,3938
Days since last update	Tools	30376	0	0	587,5581	519,6804
Days since last update	Personalization	30931	0	0	566,4819	511,5444
Days since last update	Music & Audio	47412	0	0	544,8925	434,5992
Days since last update	Education	68179	0	0	544,8139	504,4424
Days since last update	Books & Reference	38952	0	0	615,0212	511,7107
Days since last update	Business	22896	0	0	490,5428	467,0562
Days since last update	Total	300177	0	0	564,0116	493,0069

Variable

Category
–
Size

Boxplot



SCROLL

Test Statistics

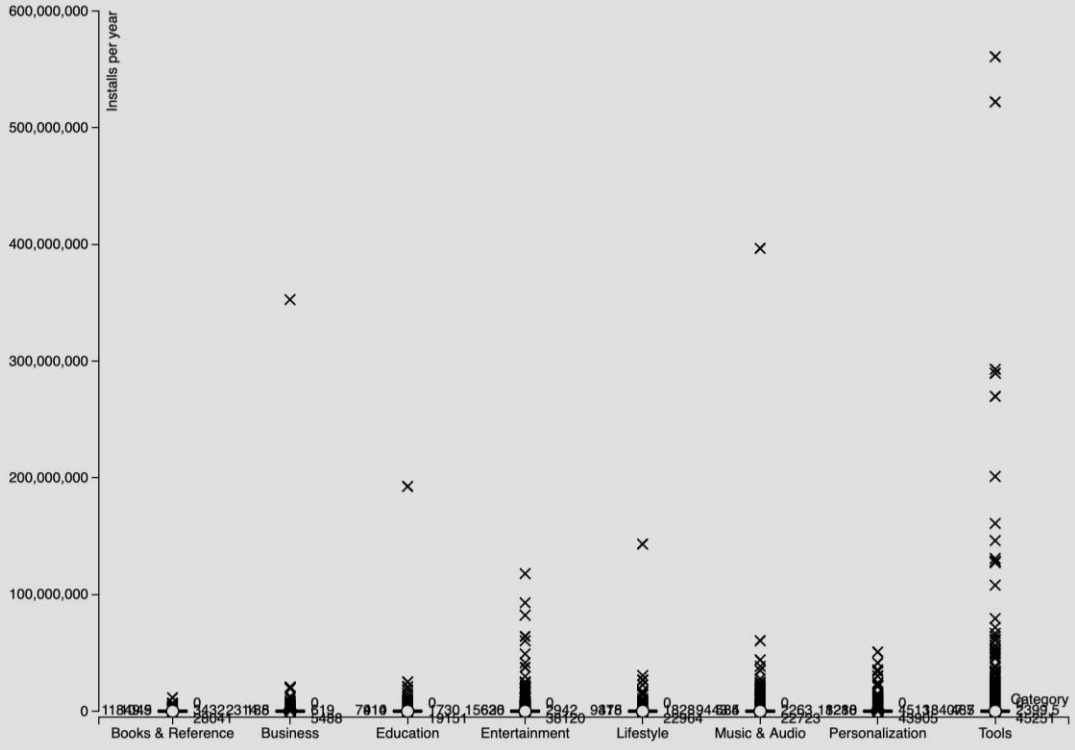
Descriptive Statistics

	Group	N	Missing	Missing Group	Mean	Std. Deviation
Size	Entertainment	34401	0	0	15,1151	18,696
Size	Lifestyle	27030	0	0	18,094	20,6925
Size	Tools	30376	0	0	10,0615	15,0957
Size	Personalization	30931	0	0	12,5033	13,7627
Size	Music & Audio	47412	0	0	23,3483	27,7214
Size	Education	68179	0	0	19,7345	24,5209
Size	Books & Reference	38952	0	0	13,5354	26,2183
Size	Business	22896	0	0	17,4771	21,3292
Size	Total	300177	0	0	16,9276	22,8063

Variable

Category – Installs per year

Boxplot



SCROLL

Test Statistics

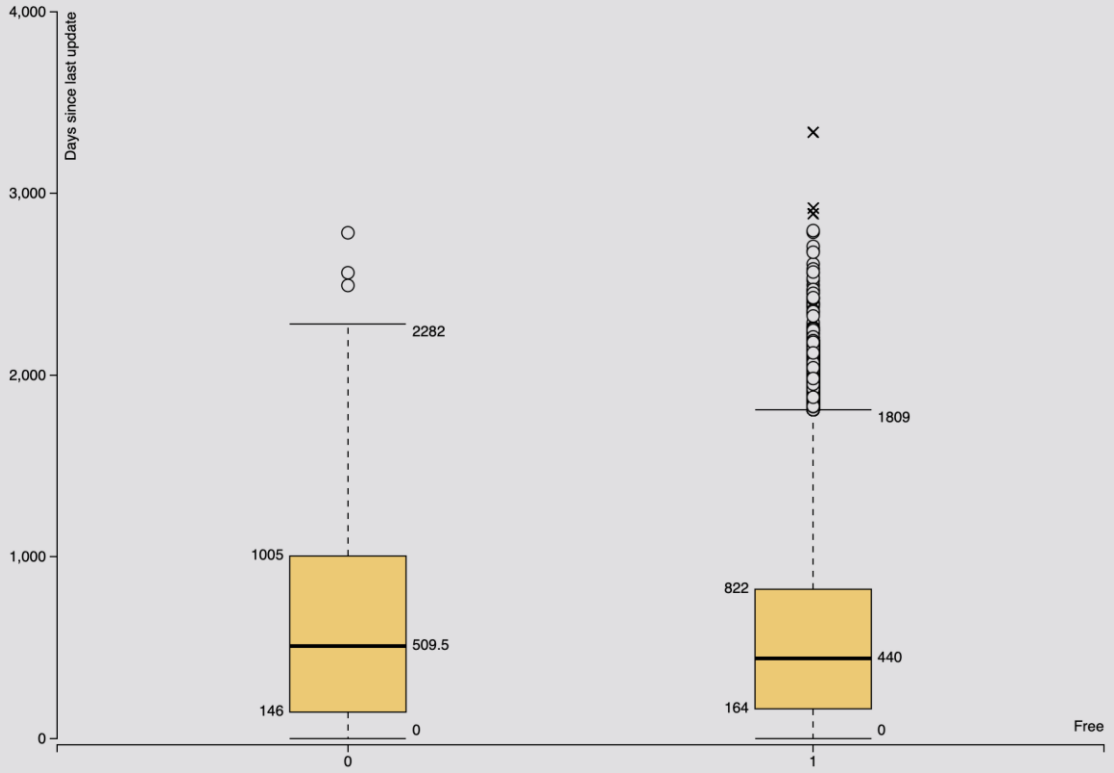
Descriptive Statistics

	Group	N	Missing	Missing Group	Mean	Std. Deviation
Installs per year	Entertainment	34401	0	0	95,071.3947	1,283,894.4131
Installs per year	Lifestyle	27030	0	0	46,206.7738	984,009.4538
Installs per year	Tools	30376	0	0	295,894.8196	5,940,792.3333
Installs per year	Personalization	30931	0	0	79,978.8637	726,038.6327
Installs per year	Music & Audio	47412	0	0	50,130.051	1,921,540.0277
Installs per year	Education	68179	0	0	25,714.2543	773,246.234
Installs per year	Books & Reference	38952	0	0	20,875.0175	137,734.007
Installs per year	Business	22896	0	0	35,823.9534	2,356,329.4015
Installs per year	Total	300177	0	0	72,439.7088	2,247,855.8701

Variable

Free
–
Days since last update

Boxplot



SCROLL

Test Statistics

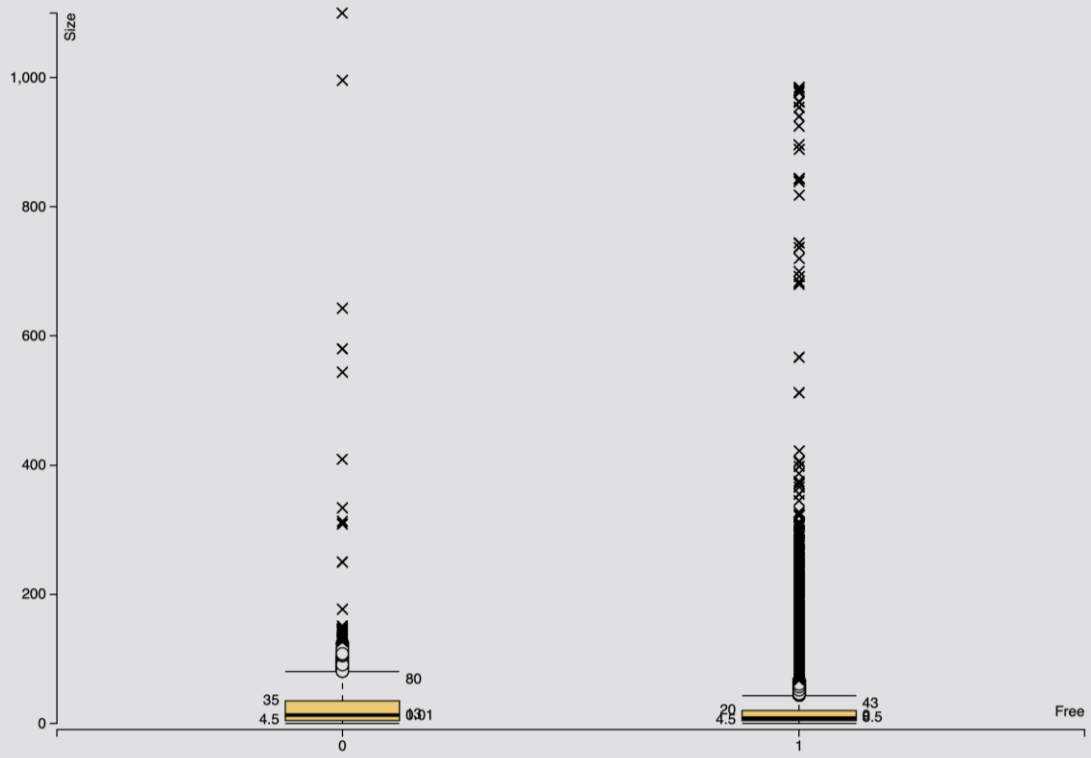
Descriptive Statistics

	Group	N	Missing	Missing Group	Mean	Std. Deviation
Days since last update	1	295173	0	0	562,6917	491,6768
Days since last update	0	5004	0	0	641,8715	560,5421
Days since last update	Total	300177	0	0	564,0116	493,0069

Variable

Free
–
Size

Boxplot



SCROLL

Test Statistics

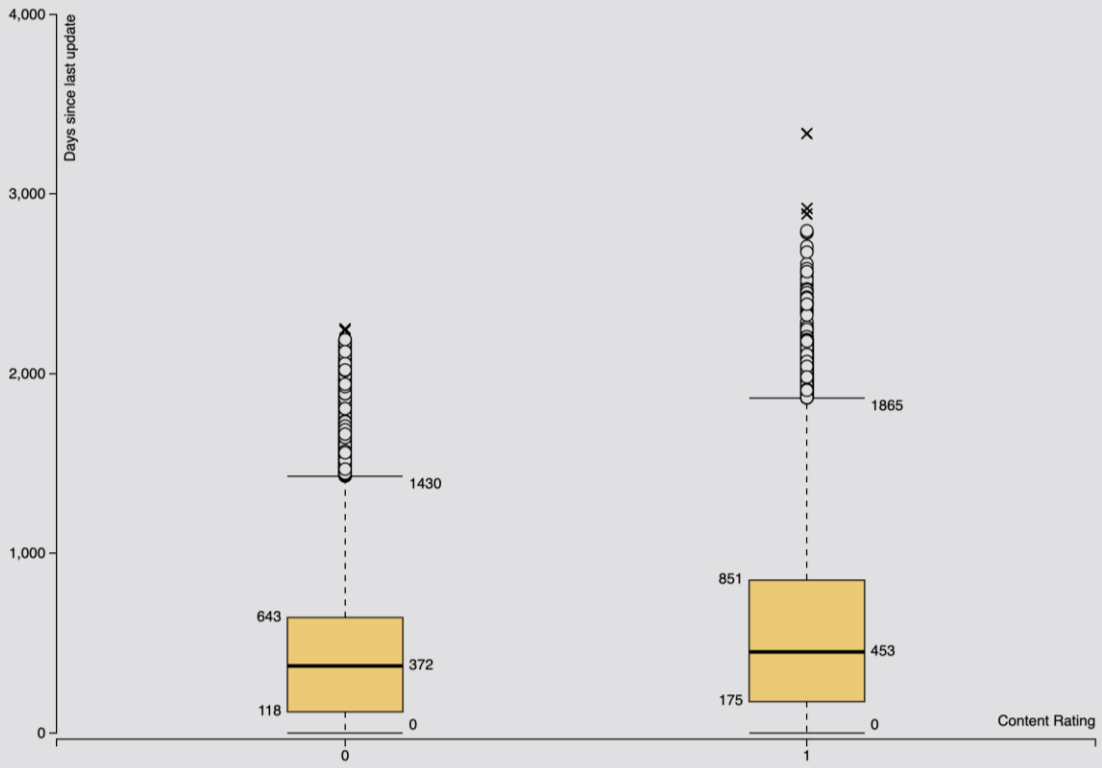
Descriptive Statistics

	Group	N	Missing	Missing Group	Mean	Std. Deviation
Size	1	295173	0	0	16,7871	22,4399
Size	0	5004	0	0	25,2171	37,7973
Size	Total	300177	0	0	16,9276	22,8063

Variable

Content Rating – Days since last update

Boxplot



SCROLL

Test Statistics

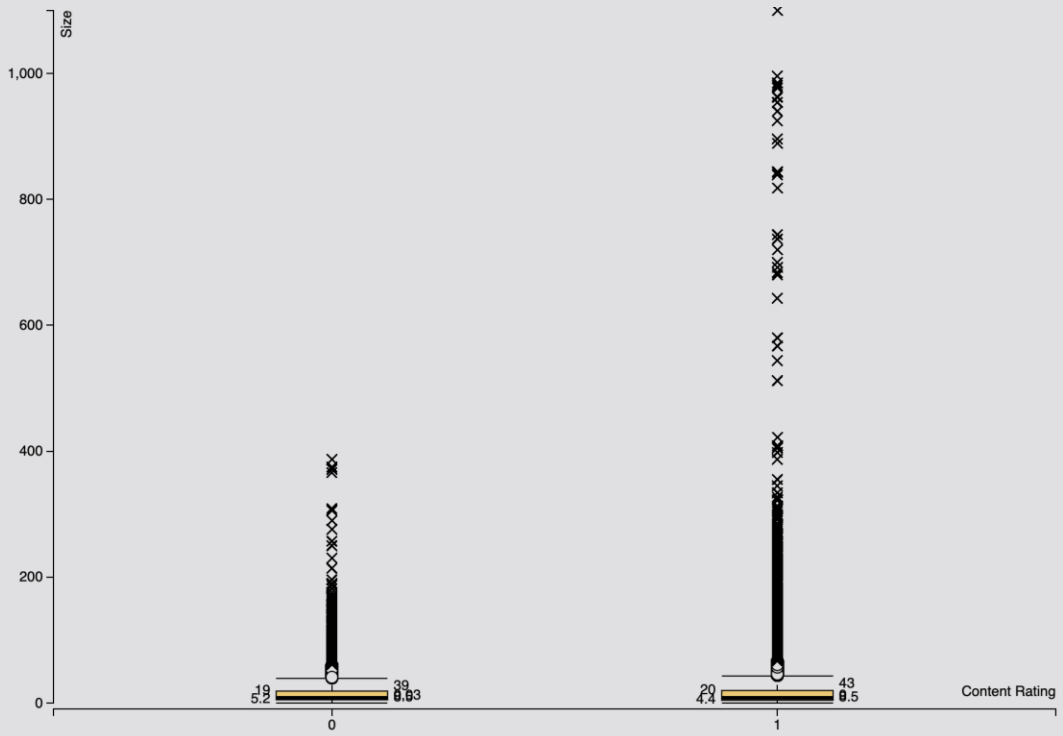
Descriptive Statistics

	Group	N	Missing	Missing Group	Mean	Std. Deviation
Days since last update	1	267659	0	0	577,5982	500,6749
Days since last update	0	32518	0	0	452,1788	407,8211
Days since last update	Total	300177	0	0	564,0116	493,0069

Variable

Content Rating – Size

Boxplot



SCROLL

Test Statistics

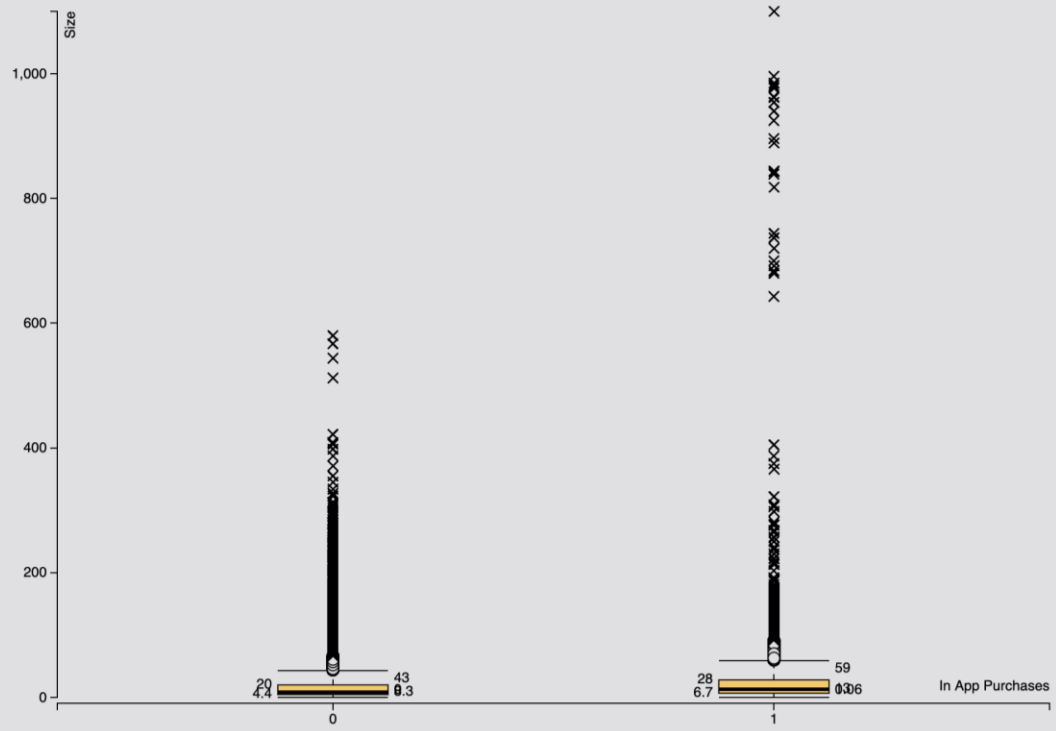
Descriptive Statistics

	Group	N	Missing	Missing Group	Mean	Std. Deviation
Size	1	267659	0	0	16,9589	23,0706
Size	0	32518	0	0	16,6699	20,5003
Size	Total	300177	0	0	16,9276	22,8063

Variable

In App Purchases – Size

Boxplot



SCROLL

Test Statistics

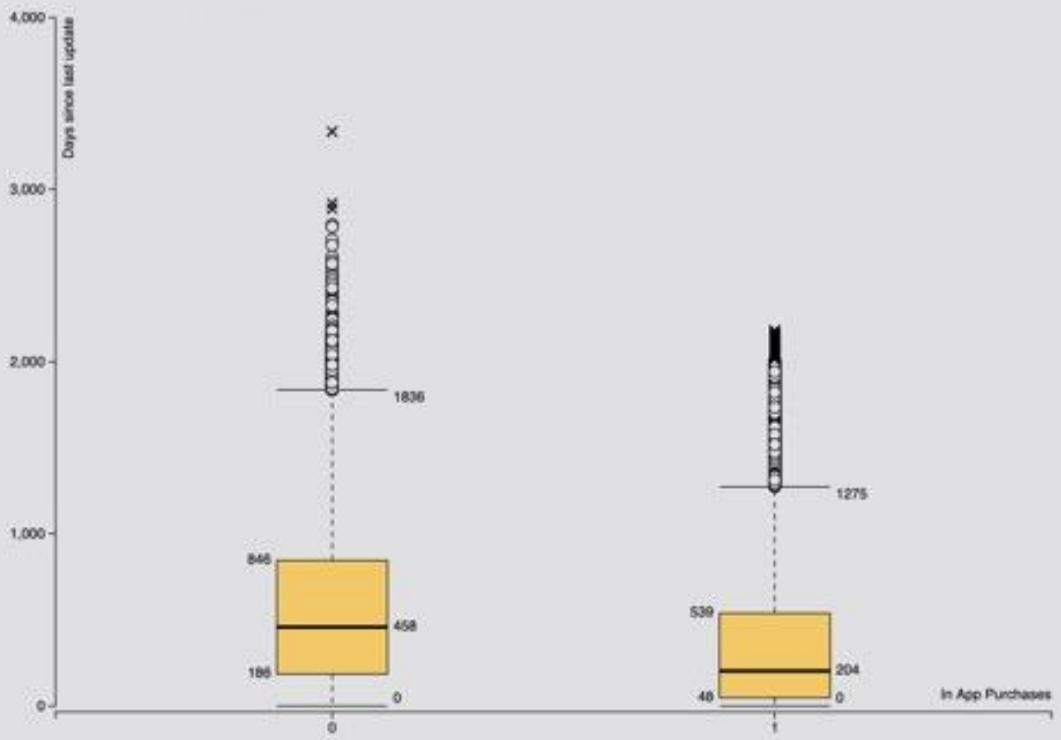
Descriptive Statistics

	Group	N	Missing	Missing Group	Mean	Std. Deviation
Size	1	20957	0	0	22,94	37,9567
Size	0	279220	0	0	16,4764	21,1689
Size	Total	300177	0	0	16,9276	22,8063

Variable

In App Purchases – Days since last update

Boxplot



SCROLL

Test Statistics

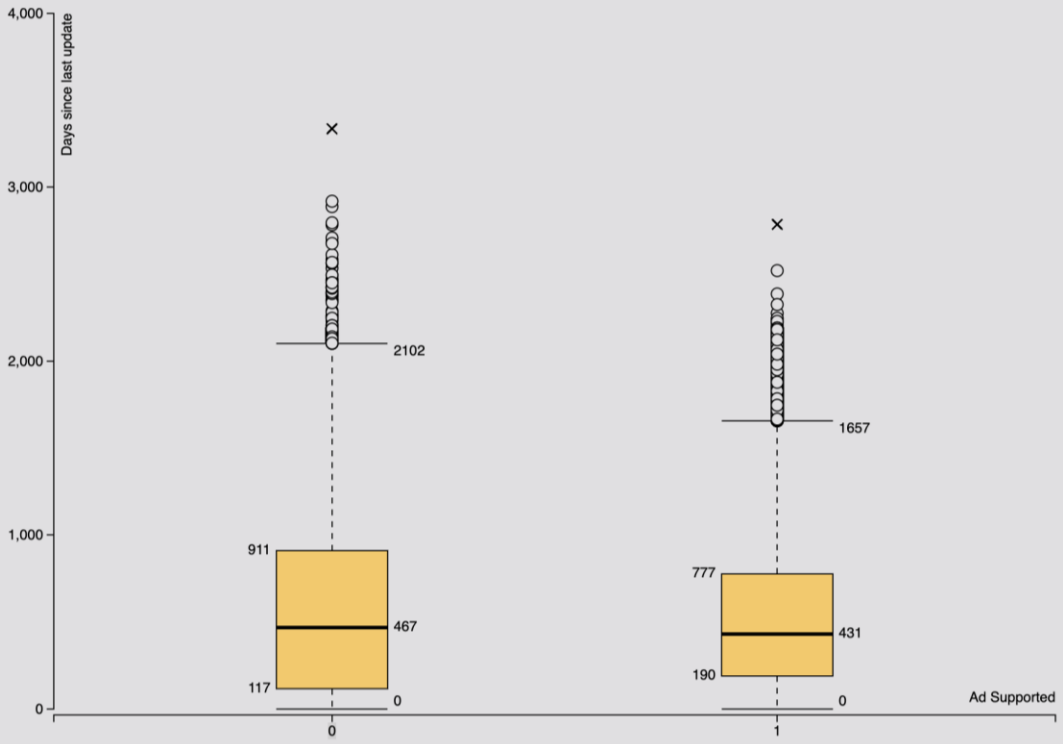
Descriptive Statistics

	Group	N	Missing	Missing Group	Mean	Std. Deviation
Days since last update	1	20957	0	0	357,5966	406,8468
Days since last update	0	279220	0	0	579,5042	495,4166
Days since last update	Total	300177	0	0	564,0116	493,0069

Variable

Ad Supported – Days since last update

Boxplot



SCROLL

Test Statistics

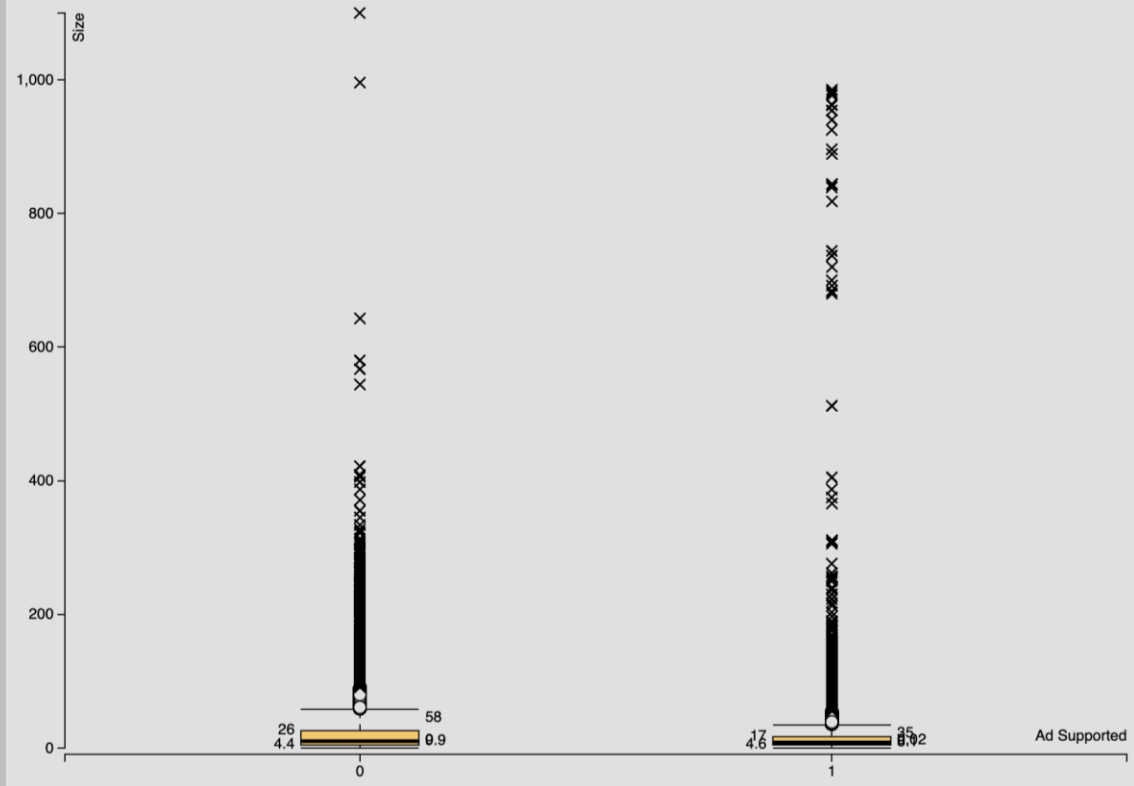
Descriptive Statistics

	Group	N	Missing	Missing Group	Mean	Std. Deviation
Days since last update	1	188900	0	0	548,148	465,3263
Days since last update	0	111277	0	0	590,9411	535,6655
Days since last update	Total	300177	0	0	564,0116	493,0069

Variable

Ad Supported – Size

Boxplot



SCROLL

Test Statistics

Descriptive Statistics

	Group	N	Missing	Missing Group	Mean	Std. Deviation
Size	1	188900	0	0	15,5991	21,5567
Size	0	111277	0	0	19,1828	24,6203
Size	Total	300177	0	0	16,9276	22,8063

MENU

DATA ANALYSIS

05

VIEW MORE

SCROLL

Overview of Data Analysis

01

LOGISTIC REGRESSION

FIRST MODEL

02

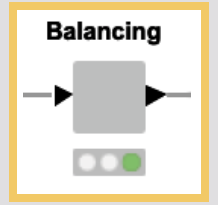
RANDOM FOREST

SECOND MODEL

03

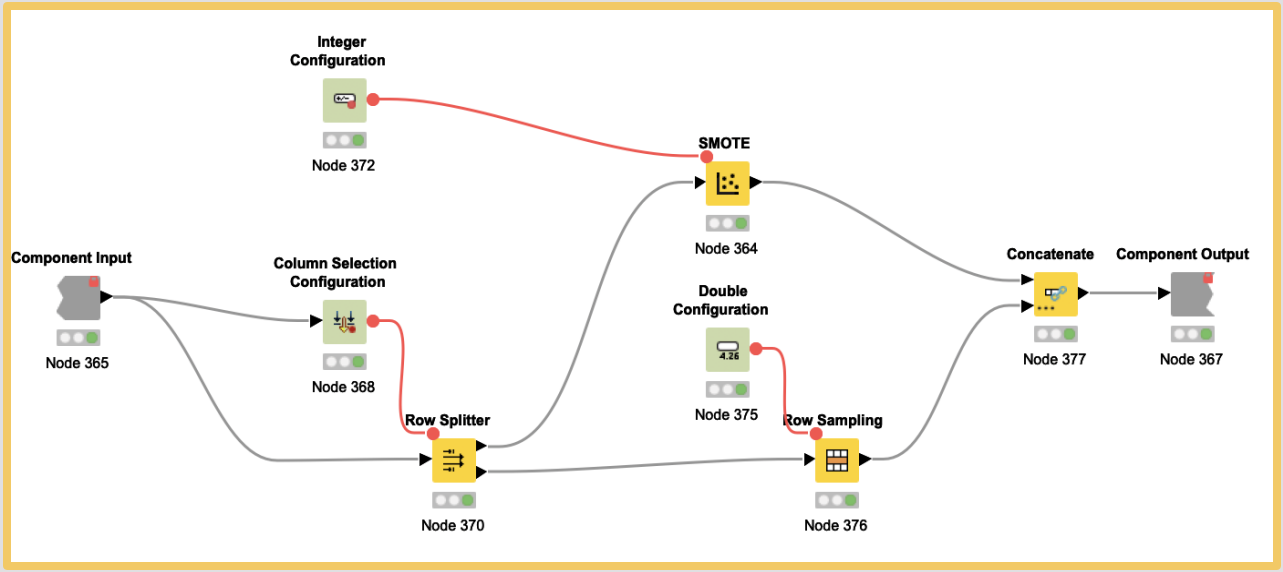
GRADIENT BOOST

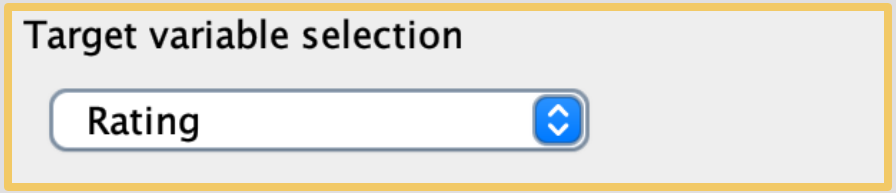
THIRD MODEL



BALANCING

Since the target variable is initially unbalanced (only 2% of data is B-rated), a balancing is performed before the application of algorithms. To do so, a combination of over-sampling and under-sampling is used on the dataset. Particularly, over-sampling of B-rated apps is achieved through the SMOTE node (Synthetic Minority Over-sampling Technique) with a rate of 10. As regards under-sampling, the rate used in the Row sampling node is 0.2. Eventually, the new dataset is balanced between A and B (49% of A vs. 51% of B).





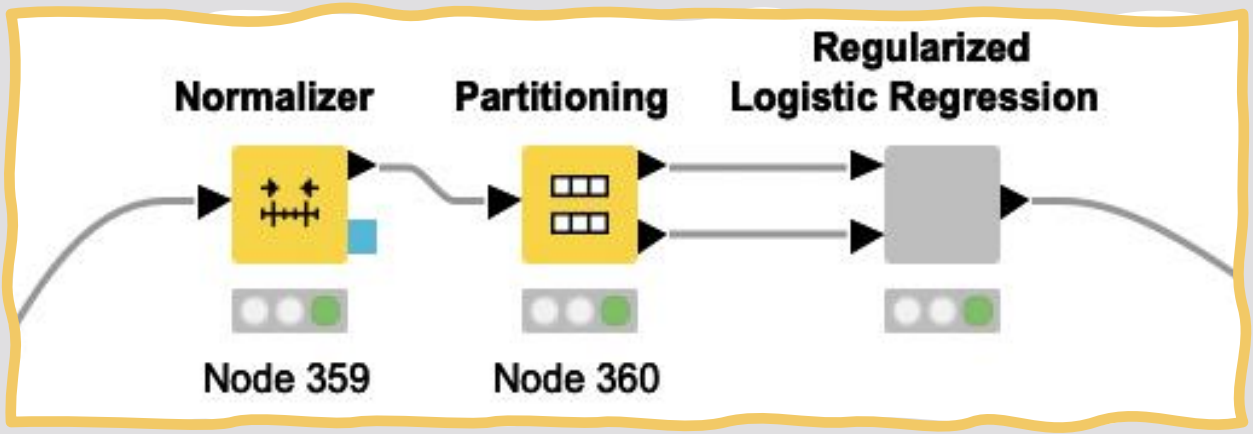
Row ID	count
A	294590
B	5587

Row ID	count
A	58918
B	61457

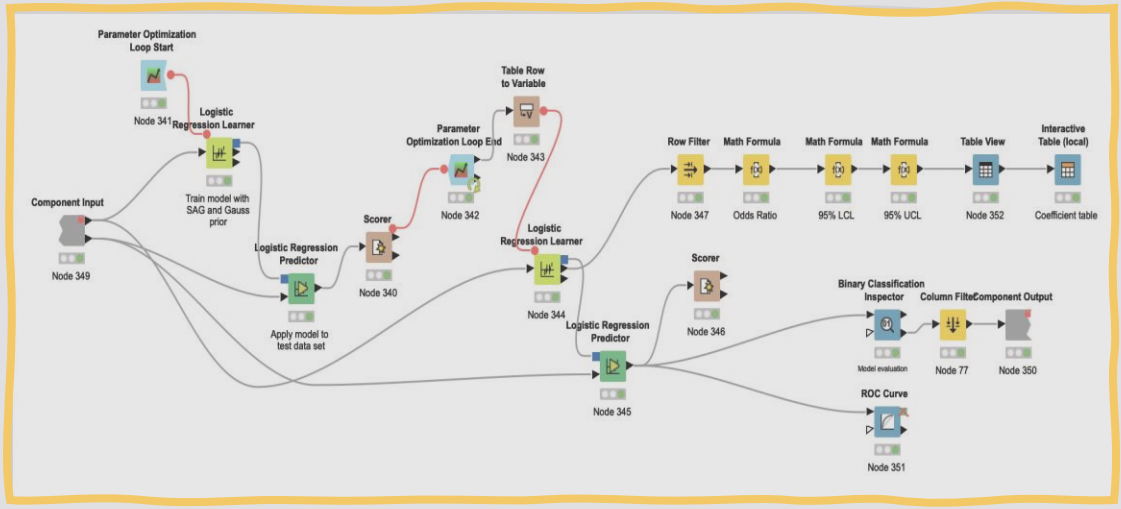
LOGISTIC REGRESSION

Since the goal is to classify the different apps and the target variable is binary, the first model proposed is a Logistic Regression. At first, a standard Logistic Regression is performed, but the model does not have good performance, and the results are not interpretable.
To account for this problem, the best decision was to opt for Regularized Logistic Regression, as this model reduces overfitting and the complexity of the prediction function. In particular, the **Ridge Regularization** is chosen to be coherent with the normalization method chosen.

Before the application of the model, a *z-score normalization* is applied to all variables that are not dummies (which were previously encoded), and the dataset is partitioned into **80% training** and **20% test** sets through a *Partitioning node*.



LOGISTIC REGRESSION



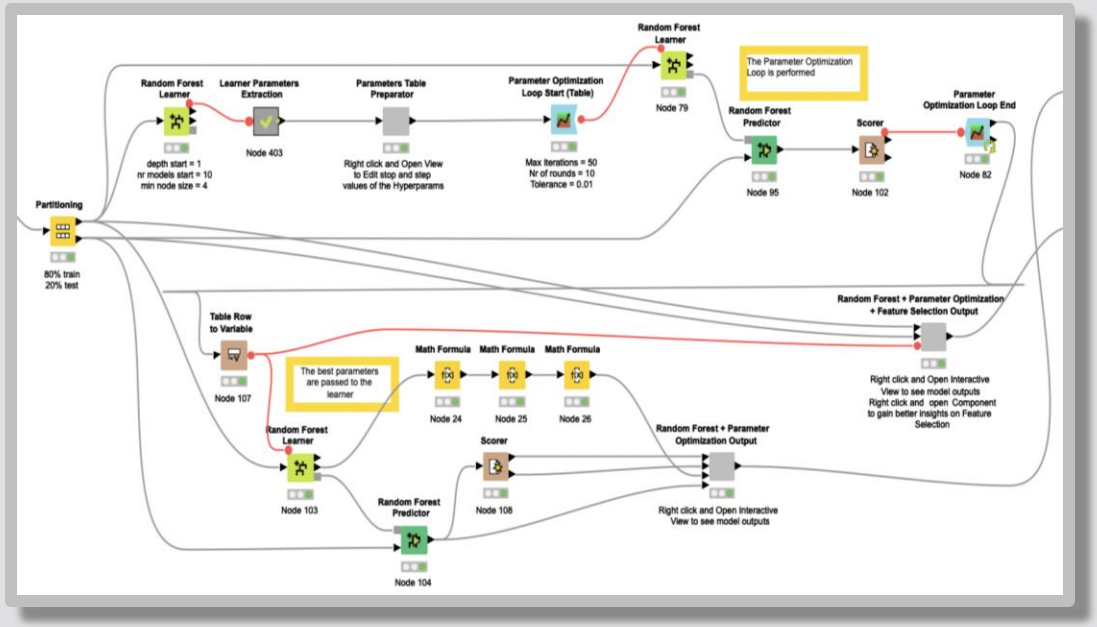
To ensure a better classification of the dependent variable and maximize accuracy, an optimization of the variance of the prior is performed through a Parameter Optimization Loop. Values from 0.5 to 15 with a step of 0.5 are tried with **'Brute Force'**. From the results of the loop, it seems that the accuracy of the model is not affected by different prior variance values (any value leads to same accuracy), and the parameter optimization loop outputs as best parameter **variance** = 0.5, which is then passed as a flow variable to the Logistic Regression Learner. To apply regularization the **'Stochastic Average Gradient'** *solver* is used, the *max number of epochs* is set to 500 and the *epsilon* to 1.0E-5. Furthermore, as *learning rate strategy it is chosen* **'Line Search'**, an experimental learning rate strategy that tries to find the optimal learning rate for the SAG solver.

In conclusion, three math formulas were added to the model for computing two relevant statistics:

- Odds ratio
- Extreme values of a 95% confidence interval (95% LCL and 95% UCL)

The performance of this model is analyzed in the **'Logistic Regression Outcomes'** Section.

RANDOM FOREST



The second model implemented is Random Forest, an ensemble model consisting of many decision trees. Before the optimization, the dataset is split into **80% training** and **20% test** sets through a Partitioning node with the same seed as the other models to ensure consistency. To tune the parameters, an optimization loop is implemented. Because there are so many possible parameter combinations to tune, **Random search** is used, with *number of iterations* set to 50, *tolerance* set to 0.01 and *number of rounds* set to 10

Parameters to tune in the optimization loop:

Depth

SPECIFIES THE MAXIMUM NUMBER OF LEVELS OF THE TREE

START = 1
STOP = 10
STEP = 2

Nmodels

SPECIFIES THE NUMBER OF MODELS

START = 10
STOP = 100
STEP = 10

MinNodeSize

MINIMUM NUMBER OF OBSERVATIONS IN A TERMINAL NODE

START = 1
STOP = 15
STEP = 1

RANDOM FOREST

The Random Forest is implemented using the combination of parameters that yields the highest accuracy:

- **Depth**: 24
- **Nmodels:** 67
- **MinNodeSize:** 2

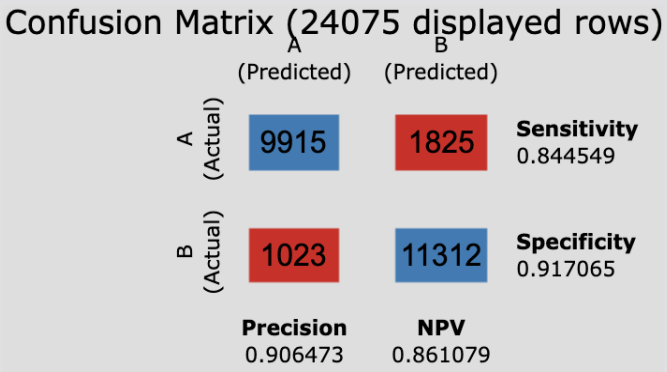
To get a balanced confusion matrix, Youden's index[1] has been maximized, as it balances sensitivity and specificity. This procedure yields a threshold probability of **0.598**

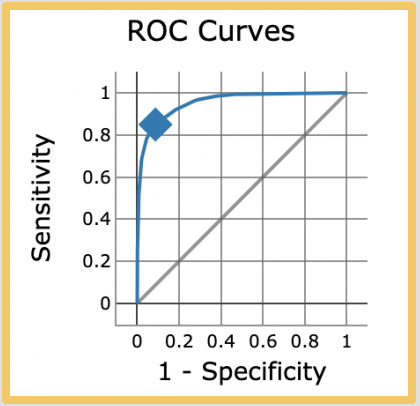
Model Statistics

ACCURACY: 0.882
AUC: 0.955
PRECISION: 0.906
SPECIFICITY: 0.917
SENSITIVITY: 0.845

FEATURE IMPORTANCE

1. **Ad Supported**
2. **Content Rating**
3. **Free**
4. **In App Purchase**
5. **Category**

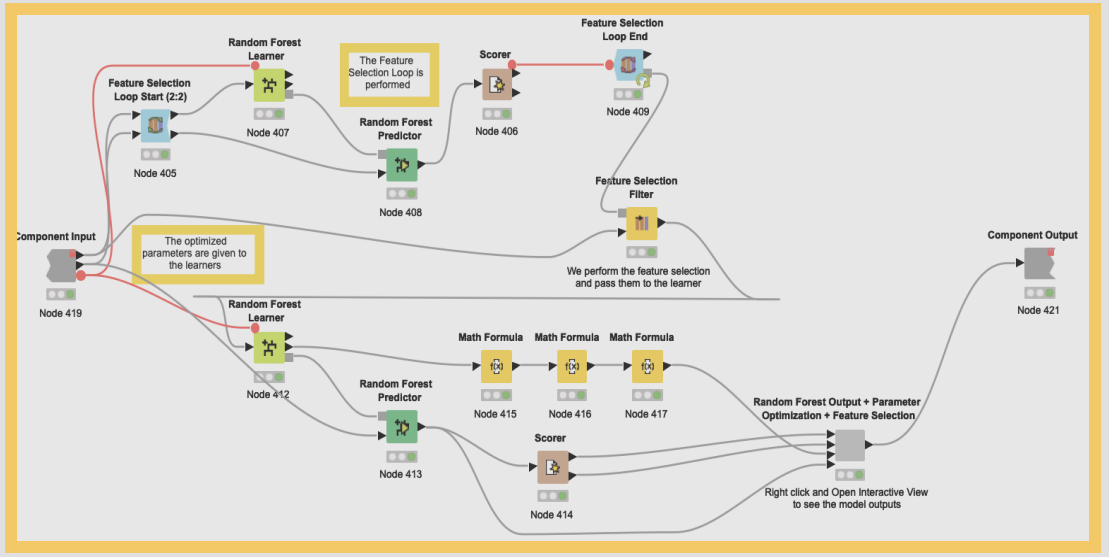




[1]Youden's index = Sensitivity + Specificity - 1

SCROLL

RANDOM FOREST



At the end, a ‘Feature Selection Loop’ is performed to check whether satisfactory performance can be reached through a **simplified model** (the goal is to keep accuracy at a similar level). The parameters used are the same as those found in the ‘Parameter Optimization Loop’.

Backward Feature Elimination, an iterative approach that starts with having all features selected, is used. The feature whose removal has the least impact on the model's performance is removed in each iteration. The *threshold* for the minimum features for the loop is set to 5.

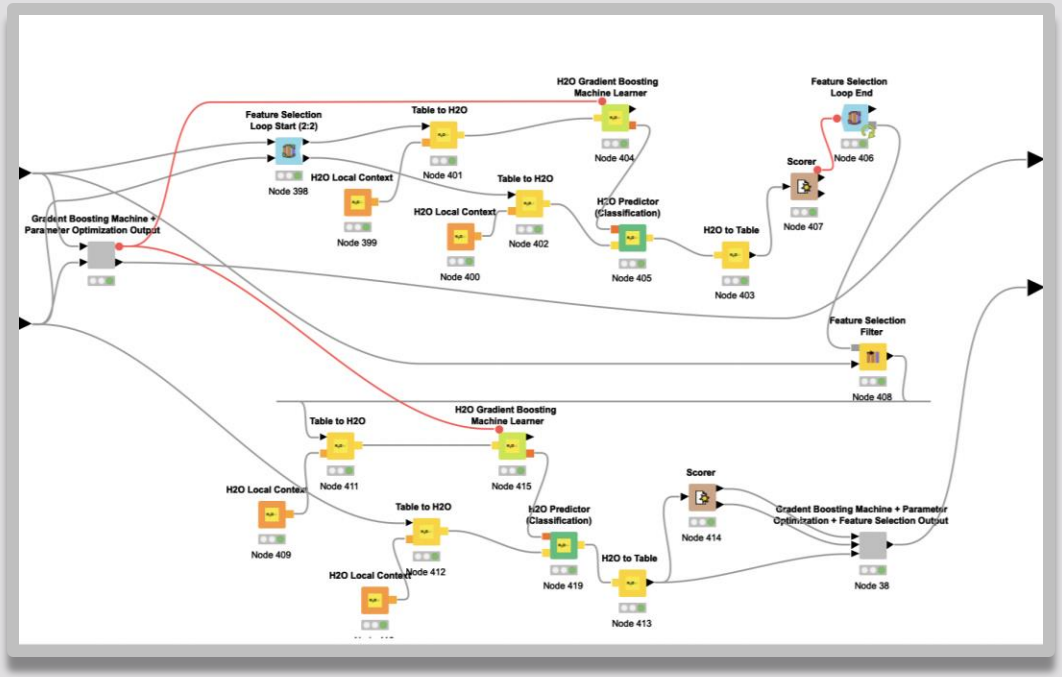
RANDOM FOREST

The model with 5 features is chosen instead of the one with all 9 features because, as it stands, it leads to the same accuracy while being simpler. Therefore, this means that the excluded variables (i.e., *'Category'*, *'Free'*, *'Content Rating'*, *'In App Purchases'*) are those that have no significant impact on the model's performance and are weakly correlated with the target variable. In this way, the computational cost is reduced, ensuring a better interpretation of the model, maximization of relevance, and minimization of redundancy. The performance of this model is analyzed in the **'Random Forest Outcomes'** Section.

Accuracy	Nr. of features
0,881	9
0,881	5
0,879	6
0,871	7
0,868	8

D	Installs per year
S	Category
S	Rating
D	Rating Count
D	Free
D	Size
D	Content Rating
D	Ad Supported
D	In App Purchases
D	Days since last update

GRADIENT BOOSTING MACHINE



The third model proposed is the H2O Gradient Boosting Machine, a machine learning technique that predicts using an ensemble of weak prediction models (random trees).
Again, the dataset is split into **80% training** and **20% test** sets, keeping the same seed as the other models to ensure consistency. In order to tune the parameters, an optimization loop is implemented.
Because there are so many possible parameter combinations to tune, **Random search** is used, with *number of iterations* set to 50, *tolerance* set to 0.01 and *number of rounds* set to 10.

Parameters to tune in the optimization loop:

Minobs

MINIMUM NUMBER OF OBSERVATIONS

START = 5
STOP = 30
STEP = 1

Maxdepth

MAXIMUM TREE DEPTH

START = 6
STOP = 15
STEP = 2

ColSamPerTree

COLUMN SAMPLE RATE PER TREE

START = 0.1
STOP = 1
STEP = 0.1

RowSamPerTree

ROW SAMPLE RATE PER TREE

START = 0.1
STOP = 1
STEP = 0.1

GRADIENT BOOSTING MACHINE

The Parameter Optimization Loop yields the following results, which maximize accuracy:

Minobs: 12 **ColSamPerTree:** 0.155
Max Depth: 13 **RowSamPerTree:** 0.192

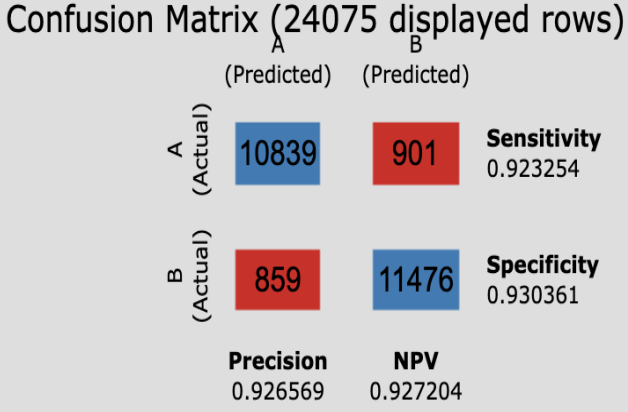
All the optimal parameters found are passed to the H2O Gradient Boosting Learner, and the performance is evaluated through *accuracy statistics*, a *confusion matrix*, and a *ROC curve* that can be viewed in the "Gradient Boosting Machine + Parameter Optimization Output" node. The threshold probability is then manually set to **0.7** in order to reduce the number of false positives, since initially there were too many.

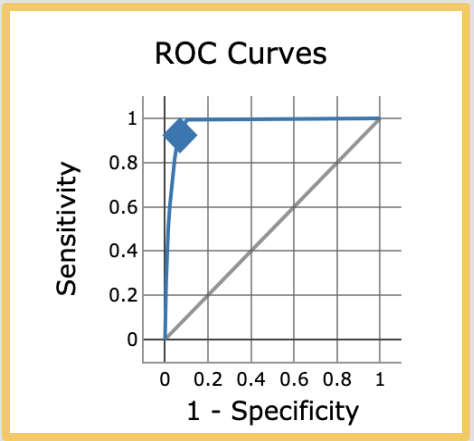
FEATURE IMPORTANCE

1. Size
2. Category
3. Rating Count
4. Ad Support
5. In App Purchase

Model Statistics

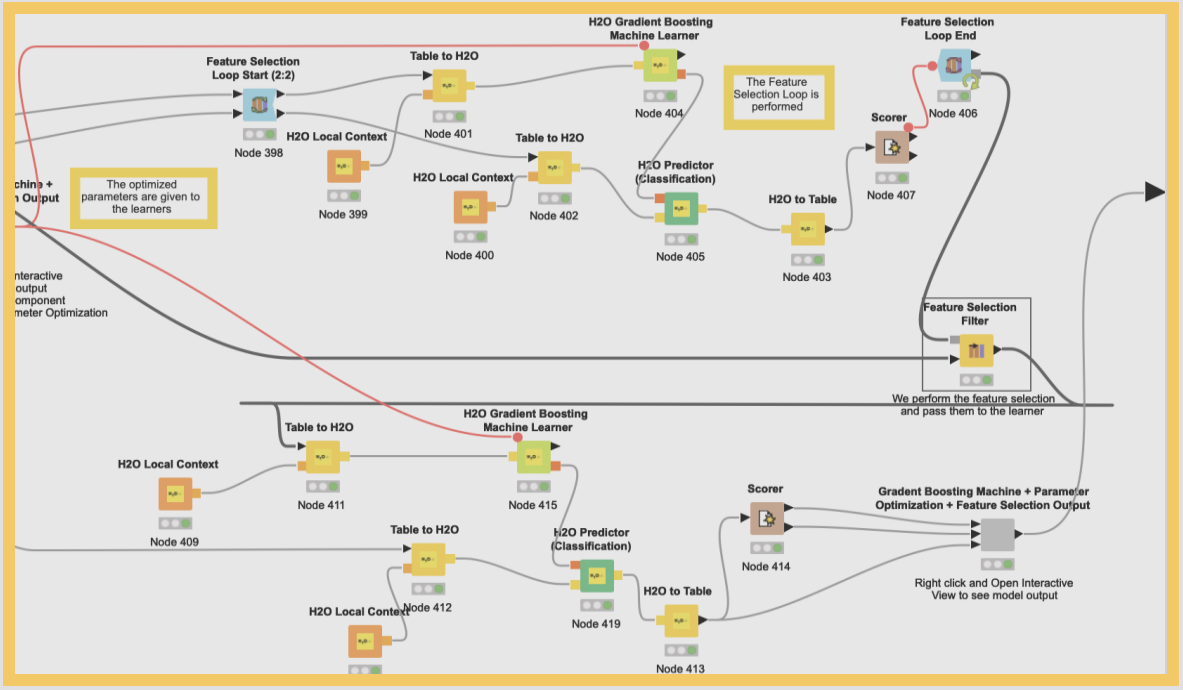
ACCURACY: 0.927 **SPECIFICITY:** 0.930
AUC: 0.976 **SENSITIVITY:** 0.923
PRECISION: 0.926





SCROLL

GRADIENT BOOSTING MACHINE



Finally, a 'Feature Selection Loop' is run to see if satisfactory performance can be obtained with a **simplified model** (the goal is to maintain accuracy at a similar level).

The parameters used are the same as those found in the 'Parameter Optimization Loop'.
Again, **Backward Feature Elimination** method is selected, and the *threshold* is set to 5**.**

GRADIENT BOOSTING MACHINE

The model with 5 features is chosen instead of the one with all 9 features because, as it stands, it leads to better accuracy while being simpler.
Therefore, this means that the excluded variables (i.e., *'Installs per year'*, *'Category'*, *'Free'*, *'Days since last update')* are those that do not have a significant impact on the model performance and are weakly correlated with the target variable. In this way, the computational cost is reduced, ensuring a better interpretation of the model, maximization of relevance, and minimization of redundancy.
The performance of this model is analyzed in the **'Gradient Boosting Machine Outcomes'** section.

Accuracy	Nr. of features
0,947	5
0,946	6
0,945	7
0,944	9
0,942	8

- D Installs per year
- S Category
- S Rating
- D Rating Count
- D Free
- D Size
- D Content Rating
- D Ad Supported
- D In App Purchases
- D Days since last update

MENU

MODEL OUTCOMES

06

VIEW MORE

SCROLL

LOGISTIC REGRESSION OUTCOMES

Variable	Coeff.	Std. Err.	z-score	P>|z|	odds_ratio	low_95%	upp_95%
Installs per year	0.001	0.01	0.143	0.886	1.001	-0.018	0.021
Category=Books & Reference	0.792	0.251	3.155	0.002	2.209	0.3	1.285
Category=Business	-0.476	0.251	-1.898	0.058	0.621	-0.968	0.016
Category=Education	0.118	0.25	0.473	0.637	1.126	-0.373	0.609
Category=Entertainment	-0.416	0.251	-1.659	0.097	0.66	-0.907	0.076
Category=Lifestyle	-0.451	0.251	-1.799	0.072	0.637	-0.943	0.04
Category=Music & Audio	0.616	0.251	2.456	0.014	1.852	0.125	1.108
Category=Personalization	1.243	0.252	4.927	0	3.465	0.748	1.737
Category=Tools	-0.926	0.251	-3.696	0	0.396	-1.418	-0.435
Rating Count	0.541	0.052	10.457	0	1.718	0.44	0.643
Free	-0.899	0.066	-13.677	0	0.407	-1.028	-0.771
Size	0.03	0.008	3.939	0	1.03	0.015	0.045
Content Rating	-0.094	0.029	-3.236	0.001	0.911	-0.15	-0.037
Ad Supported	1.121	0.017	65.315	0	3.069	1.088	1.155
In App Purchases	-1.218	0.029	-42.247	0	0.296	-1.274	-1.161
Days since last update	-0.34	0.008	-43.885	0	0.711	-0.356	-0.325
Constant	0.591	0.26	2.277	0.023	1.806	0.082	1.1

SCROLL

LOGISTIC REGRESSION OUTCOMES

All the explanatory variables are reported in the previous slide. The model presented has some significant variables at a 5% level, and none of the coefficient is equal to 0. According to the model accuracy, results are not perfectly interpretable, but some interest insights can still be noticed:

- The categories that are found to have more odds of having good ratings are respectively:
  - Personalization (odds ratio[1]: 3.465)
  - Books and Reference (odds ratio: 2.209)
  - Music & Audio (odds ratio: 1.852)
- The odds for an app of having a good rating are 3 times higher for apps that include advertising.
- Having a consistent number of rating (high rating count) positively impacts good rating (odds ratio: 1.7)
- Apps that have age restrictions have less probability of having a good rating (odds ratio: 0.911)
- The odds of an app being A-rated increases by about 1.03 times for every unit increase in Megabytes of size.

[1] An odds ratio (OR) calculates the relationship between a variable and the likelihood of an event occurring.

LOGISTIC REGRESSION OUTCOMES

Initially, the probability threshold is set by default at 0.5, leading to an *accuracy* of 0.693 and a *sensitivity* of 0.650. The main concern is that if an app with a negative rating is misclassified, this would lead to a counterproductive business strategy; on the other hand, if an app with a positive rating misclassified, thus not considered, there would be no monetary loss, and this makes this scenario the least dangerous.
This is the reason why it is chosen to maximize the **Youden's Index**, resulting in the probability threshold to be **0.536**, and the model to have an increased accuracy and a reduced number of false positives.

Model Statistics

ACCURACY: 0.699

AUC: 0.755

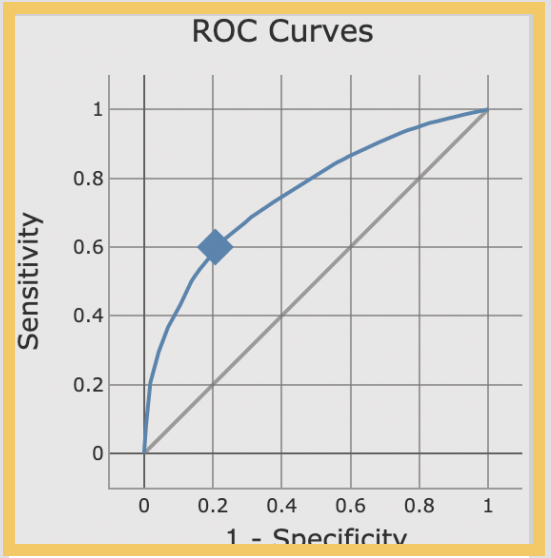
PRECISION: 0.734

SPECIFICITY: 0.793

SENSITIVITY: 0.600

Confusion Matrix (24075 displayed rows)

	A (Predicted)	B (Predicted)	
A (Actual)	7045	4695	**Sensitivity** 0.600085
B (Actual)	2555	9780	**Specificity** 0.792866
	Precision 0.733854	**NPV** 0.675648	



SCROLL

RANDOM FOREST OUTCOMES

In this section, the outputs from the Random Forest model are analyzed after the Feature Selection is performed as described previously. In order to gain insights on which features are more important, the table below illustrates the measures of importance of the variables. The model suggests that the most important features for predicting the rating are: '*Ad Supported*' , with an importance of 0.66, '*Days since last update*' (0.54), '*Installs per year*' (0.44), '*Rating Count*' (0.43) and '*Size*' (0.37).

RowID	#splits (level 0)	#splits (level 1)	#splits (level 2)	#candidates (level 0)	#candidates (level 1)	#candidates (level 2)	splits	candidates	importanza
Ad Supported	31	40	57	31	52	111	128	194	0.6597938144329897
Days since last update	8	37	61	29	56	111	106	196	0.5408163265306123
Installs per year	21	23	40	31	49	109	84	189	0.4444444444444444
Rating Count	7	20	48	20	58	96	75	174	0.43103448275862066
Size	0	14	54	23	53	109	68	185	0.3675675675675676

RANDOM FOREST OUTCOMES

To obtain the best trade-off between specificity maximization and accuracy optimization, it is decided to maximize the **Youden's index**. This maximization suggests a threshold probability **0.493**, and the following results are collected:

Model Statistics

ACCURACY: 0.881

AUC: 0.955

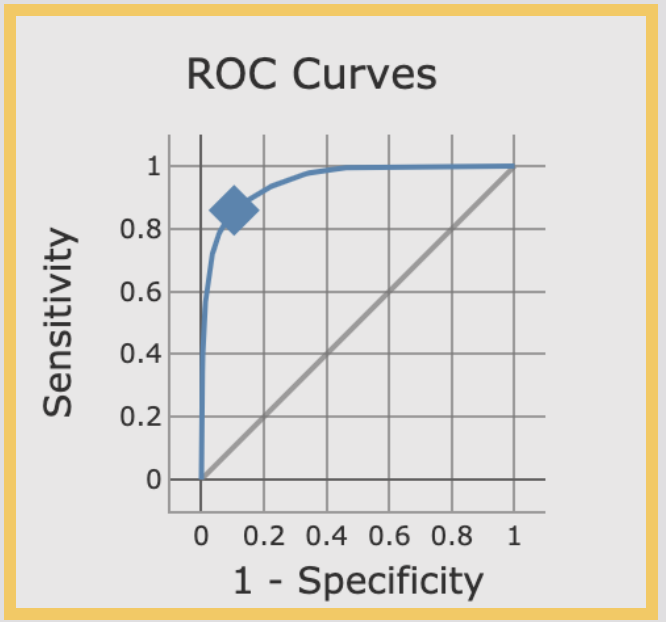
PRECISION: 0.89

SPECIFICITY: 0.899

SENSITIVITY: 0.862

Confusion Matrix (24075 displayed rows)

	A (Predicted)	B (Predicted)	
A (Actual)	10120	1620	**Sensitivity** 0.862010
B (Actual)	1250	11085	**Specificity** 0.898662
	Precision 0.890062	**NPV** 0.872491	



GRADIENT BOOSTING OUTCOMES

At last, the outcomes of the 'H2O Gradient Boosting Machine' are scrutinized.
Hence, let's scan through the output relative to the variable importance, reported in the table below in three different measures. The model suggests that the most important features for predicting the rating are: '*Ad Supported*' , with a percentage relative importance of 0.44, '*Days since last update*' (0.259), '*Installs per year*' (0.138), '*Rating Count*' (0.092) and '*Size*' (0.067).

Row ID	D Relative Importance	D Scaled Importance	D Percentage
Ad Supported	16,269.753	1	0.444
Size	9,501.639	0.584	0.259
Rating Count	5,061.925	0.311	0.138
Content Rating	3,372.487	0.207	0.092
In App Purchases	2,470.322	0.152	0.067

SCROLL

GRADIENT BOOSTING OUTCOMES

In the Gradient Boosting model, a slightly different approach is used with respect to the previous models, as initially there is a large number of false positives. Since a false positive (i.e., predicting a high rated app (A) when is low rated (B)) is worse than a false negative from a managerial point of view, this problem is accounted for through the manual setting of the probability threshold and not through Youden's index maximization process, as the results were not satisfactory. This results in the probability threshold being set to **0.750.**

Model Statistics

ACCURACY: 0.923

AUC: 0.969

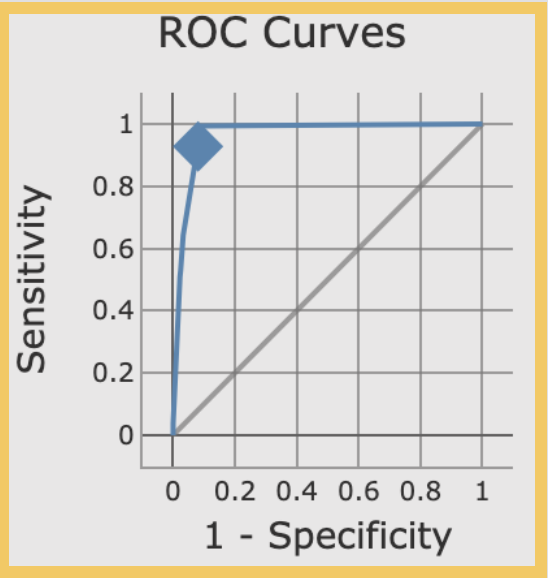
PRECISION: 0.916

SPECIFICITY: 0.919

SENSITIVITY: 0.927

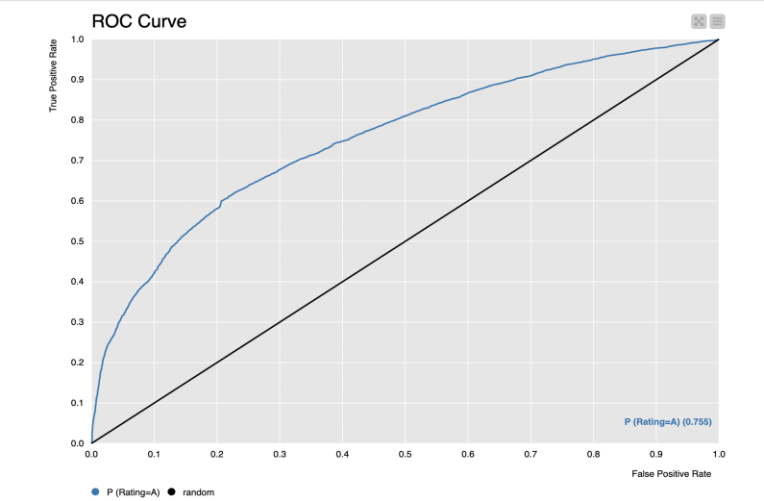
Confusion Matrix (24075 displayed rows)

	A (Predicted)	B (Predicted)	
A (Actual)	10889	851	**Sensitivity** 0.927513
B (Actual)	998	11337	**Specificity** 0.919092
	Precision 0.916043	**NPV** 0.930177	



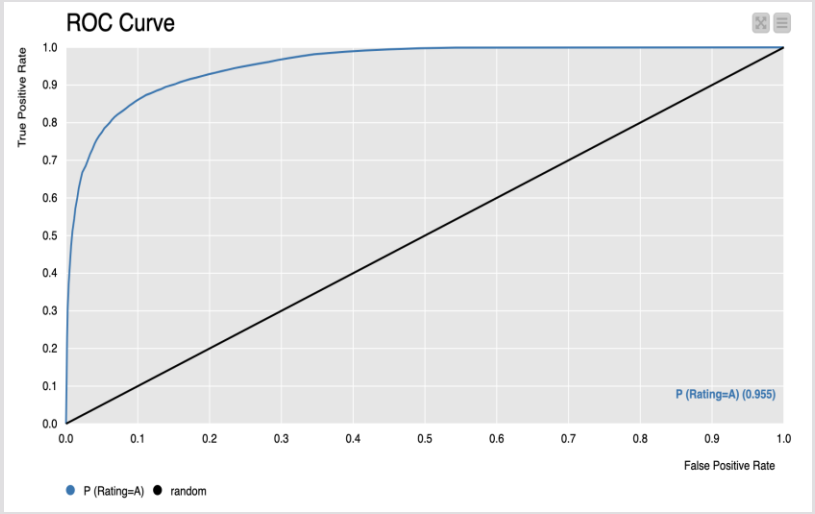
SCROLL

MODEL COMPARISON



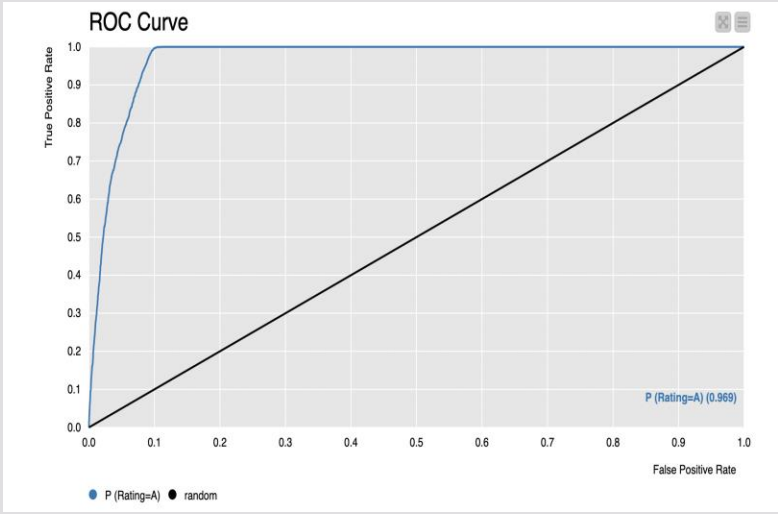
Regularized Logistic

ACCURACY: 0.699
AUC: 0.755
PRECISION: 0.734
SPECIFICITY: 0.793
SENSITIVITY: 0.600



Random Forest

ACCURACY: 0.881
AUC: 0.955
PRECISION: 0.89
SPECIFICITY: 0.899
SENSITIVITY: 0.862



Gradient Boosting

ACCURACY: 0.923
AUC: 0.969
PRECISION: 0.916
SPECIFICITY: 0.919
SENSITIVITY: 0.927

Model	Accuracy	Precision	Sensitivity	Specificity	AUC
Regularized Logistic Regression + Parameter Optimization	0.699	0.734	0.6	0.793	0.755
Random Forest + Parameter Optimization + Feature Selection	0.881	0.89	0.862	0.899	0.955
Gradient Boosting Machine + Parameter Optimization + Feature Selection	0.923	0.916	0.928	0.919	0.969

With an **accuracy** of **0.923** and an **AUC** of **0.969**, the Gradient Boosting Model is the best model to apps rating.

MENU

MANAGERIAL IMPLICATIONS

The Takeaways

07

VIEW MORE

SCROLL

MANAGERIAL IMPLICATIONS

The scope of the report is to conduct an analysis to attain further insights in order to enter the application market as investors and understand the most relevant characteristics that influence users' rating. The ultimate objective is to enter the industry with a performant app, able to stagger the market, in an attempt to maximize sales and revenue.

The models presented in the report render slightly different conclusions; according to the relevant statistics, the most accurate model is the Gradient Boosting: as a consequence, it is important to mostly rely on the latter. Unfortunately, due to different results in the Random Forest model, this section will focus on significant features that are consistent across models, succinctly explaining the less relevant ones first.

One of the least important variables in the Gradient Boosting with Feature Selection is **"In app purchases"**, which highlights the presence of additional options or features available for purchases inside the app that could have an impact on user ratings. The reason could be found by the fact that, even though app purchases increase the personalization and profitability of an app, in the end only a portion of the individuals truly consider the purchase of these features; hence, this feature does not significantly affect ratings. As a consequence, investing in this feature is not so relevant as the investment could be redirected towards other important factors that can enhance customers' satisfaction and increase the effectiveness of the relevant features in the models analyzed.

MANAGERIAL IMPLICATIONS

In addition, another less significant feature that could have a relative negative impact is the imposition of restrictions in the **app content**. In fact, this factor could be helpful only for a niche market of restricted apps; consequently, doesn't affect the overall rating. Even though these factors could help investors consider more paths of investment, they are not confirmed by the Random Forest model; therefore, the actual impact is probably less relevant than that the of other variables.

As result, the most important factor emphasized by the models is the presence of **in app advertising**. Indeed, in terms of odds (in the logistic regression), an app containing advertising is 3.069 times more likely to have a high rating than an app without. Despite the fact that this could be considered controversial, the result is interesting. Apps that support advertising have an additional source of revenue compared to the others, and consequently can increase their investments towards the improvement of app features and design, enhancing the overall quality. Through refinements, the user's satisfaction increases, improving the overall rating of the application. Another factor worth considering is the **size** of the app, which could be connected to the previous investment opportunity. This element has a significant impact in both models, making it worth to analyzing it. Even though the difference is relatively small, an app with larger size is more likely to have good rating with respect to others of a smaller size, as shown by the odd ratio.

MANAGERIAL IMPLICATIONS

This could be derived from the fact that an app with a considerable size has space for additional functionalities and features on top of improved aesthetics and graphics. Therefore, potential investors could concentrate a portion of their investments into the selection of requested features by users and a unique user interface with the objective of enhancing users' engagement and experience, without being concerned about the app's size. Additionally, the integration with third-party apps and with the operating system could further improve the functionality of the app and user satisfaction. Moreover, a high **rating count** reveals itself to be another important aspect to differentiate between a highly rated app and a poorly rated one. This evidence could be related to the fact that an individual is pushed to write a review or give feedback only when he or she is highly satisfied or vice versa. Therefore, it could be important to invest in an additional system (such as pop-ups, invitations, etc.) to increase the number of feedbacks collected and customers' reviews. This could also be implemented with a particular customer service that directly gets in contact with users, to provide them with answers to positive and negative feedback, as this could increase the customers' satisfaction and consequently the good ratings relative to the app.

MANAGERIAL IMPLICATIONS

To summarize, new potential investors interested in entering the app market can start with these deductions and develop an initial strategy for selecting the right investments toward critical factors that can positively affect the app's rating, leading to a better reputation, more sales, and revenue.

Most likely, further analysis is needed as this dataset encompasses a great variety of apps and is not focused only on a specific sector of the market. Indeed, investors should analyze the features mentioned above through the usage of the best model, "H20 Gradient Boosting," with an **accuracy** of 0.923 and an **AUC** of 0.969 aligned with text analysis (described in the next session of the report) conducted on the category of interest, proposing to gain further insights and more detailed deductions in terms of rating.

BONUS TASK:

TEXT ANALYSIS

From the text to the app

08

VIEW MORE

From now on, it is taken into consideration the workflow: **“Big_Data_TextAnalysis.knfw”** provided

Overview of the Additional Analysis

WEB SCRAPING

In this section is described the process of surfing the internet in the search of a dataset which could be used for text mining. The script is provided and described.

TEXT MINING

Here, the analysis is applied to the web scraped data frame. **Preprocessing** is applied to format the data for the procedure of creating plots through the computation of frequencies.

OUTCOMES

This illustrates the results produced by the analysis. Detailed description of the **plots** is provided with the main characteristics of a top performing app on the market in **comparison** to a worst ranked application.

Web Scraping

```
url='https://www.applyzer.com/?mmenu=worldcharts'


# configure webdriver  -- headless in order not to open the webpage everytime
options = Options()
options.headless = True  # hide GUI
options.add_argument("--window-size=1920,1080")  # set window size to native GUI size
options.add_argument("start-maximized")  # ensure window is full-screen


# configure chrome browser to not load images to not waste space and time
chrome_options = webdriver.ChromeOptions()
chrome_options.add_experimental_option(
    # this will disable image loading
    "prefs", {"profile.managed_default_content_settings.images": 2}
)


driver = webdriver.Chrome('/Users/MyUsername/Downloads/chromedriver',options = options, chrome_options=chrome_option
driver.get(url)


# wait for page to load to get the apps names to load
element = WebDriverWait(driver=driver, timeout=10).until(
   EC.presence_of_element_located((By.NAME, 'categoryid')))


select = Select(driver.find_element(By.NAME,"categoryid"))
select.select_by_value('21')  #setting the category to Games

#print(driver.page_source)
```

```
#creation of a dataframe
df = pd.DataFrame()
rev = {}

#for each name downloading the app id from play store
for x in top_list:
    name = search(x,
                lang="en",  # defaults to 'en'
                country="us",  # defaults to 'us'
                n_hits=1  # defaults to 30 (= Google's maximum)
    )

    #In the case the app is not found in play store, store a blank name
    if name != []:
        id_name = name[0]['appId']
    else:
        id_name = ""
        continue

    print(x,id_name)


    #For each app download 5,3 and 1 star reviews (150 each)
    for i in [5,3,1]:
        result, continuation_token = reviews(
        id_name,
        lang='en', # defaults to 'en'
        country='us', # defaults to 'us'
        sort=Sort.MOST_RELEVANT, # defaults to Sort.NEWEST
        count=150, # defaults to 100
        filter_score_with=i
            # defaults to None(means all score)
        )
        if result != []:
            for srev in result:
                df = df.append({"rating":i,"review":srev["content"]},ignore_index=True)


#store the reviews inside a csv called "game_rating.csv"
#df.to_csv("game_rating.csv", sep='\t', encoding='utf-8')
```

Web Scraping

From a managerial standpoint, it would be interesting to acquire more knowledge about the sector that might be more lucrative: **games**. To do this, a dataset is being **web scraped** using Python, and it is gathered from one of the most reliable websites for app store world charts, *APPlyzer* [2].

For the optimal evaluation of the success and failure parameters of an application, it was carefully selected to download only 1-star, 3-star, and 5-star ratings.

The created data frame contains 63'144 entries in total, divided into two columns for the **rating** and its **review**, with a balanced number of reviews for each rating.

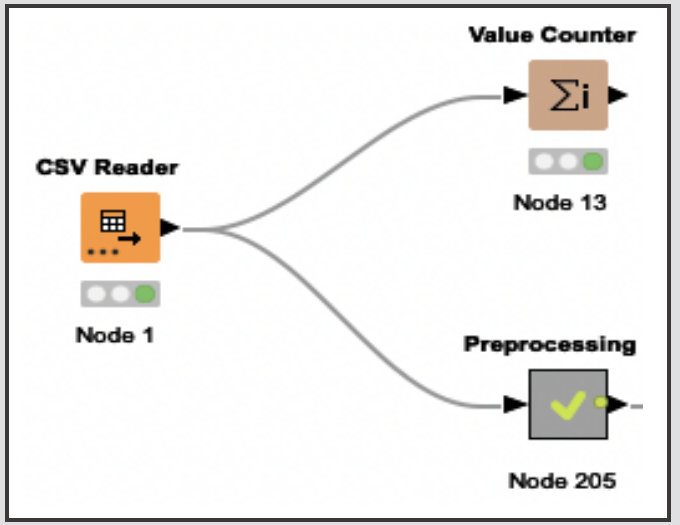
2 Dorn, Michael. "App Store World Charts." *APPlyzer*, https://www.applyzer.com/?mmenu=worldcharts

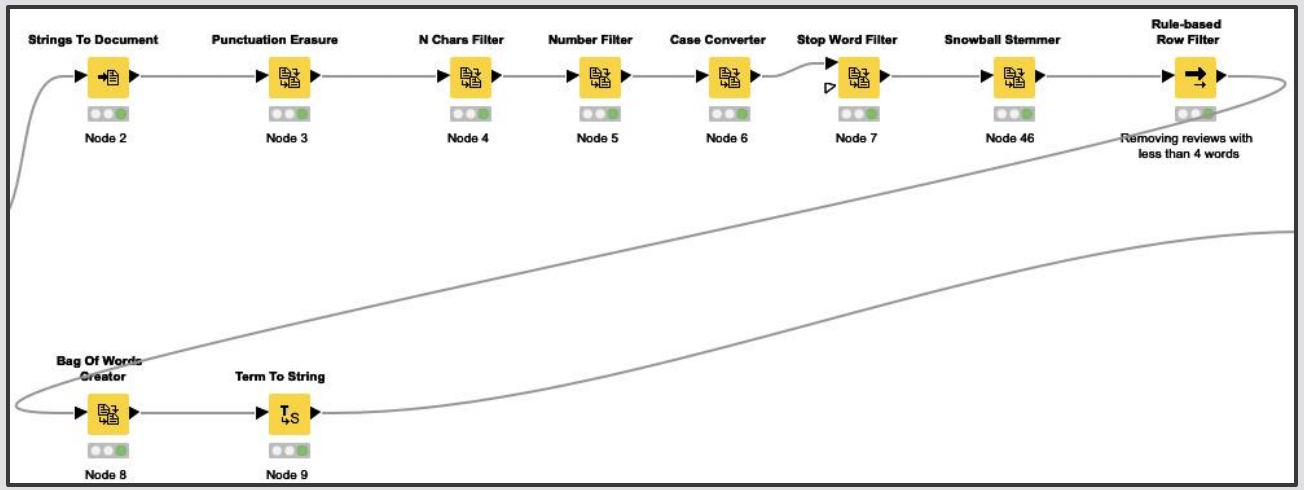
The Approach

Text mining is the process of examining large collections of text and converting the unstructured text data into structured data for further analysis, like visualization and model building. In terms of text mining approaches, there are two broad categories:

- **semantic parsing** where the word sequence, word usage as noun or verb, hierarchical word structure, etc. matter
- **bag of words** where all the words are analyzed as a single token and order does not matter.

The ultimate objective of any text mining process using the "bag-of-words" approach is to convert the text to be analyzed into a data frame that consists of the words used in the text and their frequencies. These are defined by the **document term matrix (DTM)** and the **term document matrix (TDM)**; to ensure that the DTM and TDM are cleaned up and represent the core set of relevant words, a set of pre-processing activities needs to be performed on the corpus. To proceed, a value counter and preprocessing are applied.





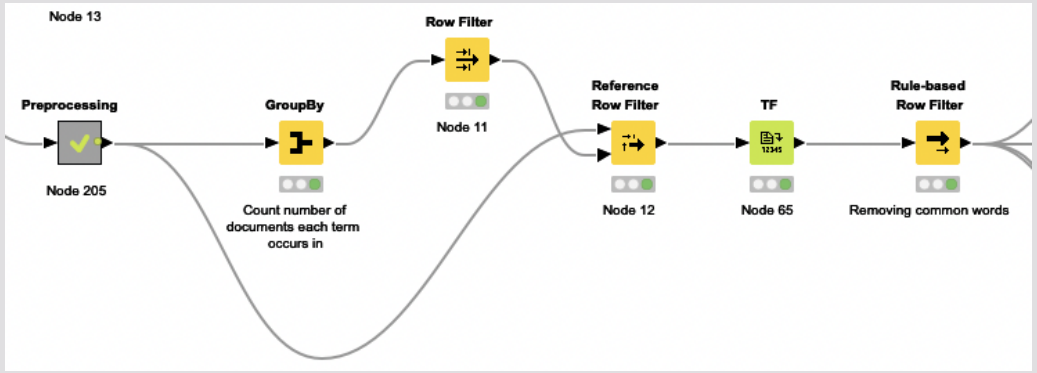
Pre-Processing

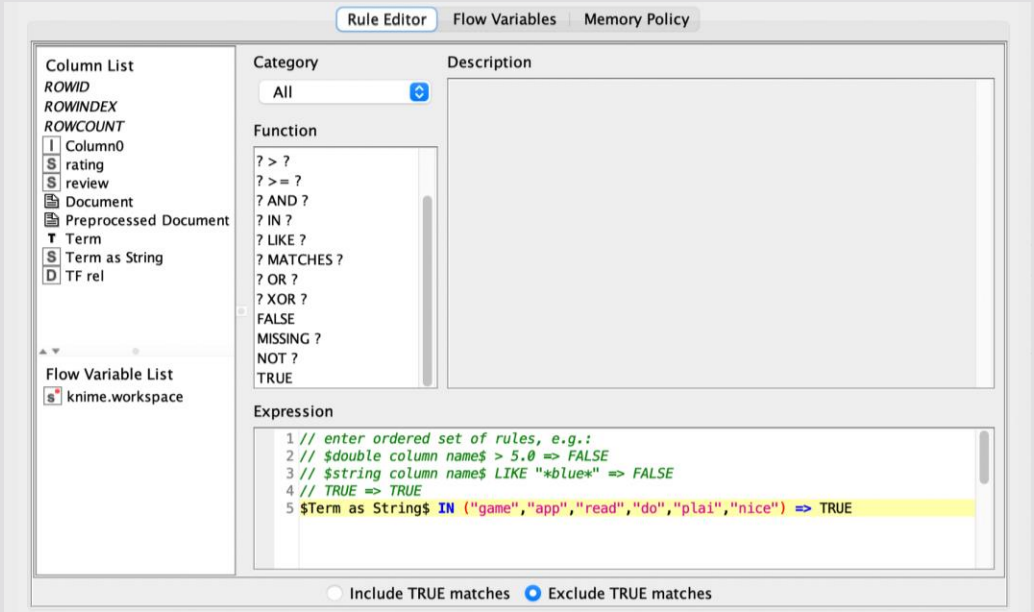
The first step is to convert the file from a csv format to a document- term matrix format. A **document-term matrix** is a mathematical matrix that describes the frequency of terms that occur in a collection of documents. In a document-term matrix, rows correspond to documents in the collection, and columns correspond to terms. The following step is to **remove punctuation** and **upper-case** characters. The text generally consists of many prepositions, pronouns, conjunctions, etc.; **removing** these **stop-words** is essential; otherwise, they will appear in all the frequently used words list and will not give the correct picture of the core words used in the text. **Words are** then **stemmed**: in linguistics, stemming is the process of reducing inflected (or derived) words to their word stem, base, or root form-generally a written word form.

To fully comprehend the genuine meaning of the users' assessment, each review **must contain more than four terms;** otherwise, it is eliminated after being filtered through the "Row Filter" node.

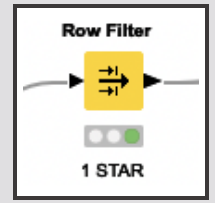
Text Mining

Before delving into the analysis of the desired categories of data, some manipulation is required to prepare the next stages. This procedure commences by **associating** each term with its **occurrence** in each document; for an optimal analysis, if this number is **less than three-hundred**, then the decision is to eliminate this term. Then, the "TF" node is used to **compute** the **relative term frequency** of each term according to each document and add a column containing the term frequency value. The value is computed by dividing the absolute frequency of a term according to a document by the number of all terms in that document. To maintain an accurate analysis, it's decided to **remove custom stop-words** based on the context of the text mining: these are words specific to the dataset that may not add value to the text -displayed in the picture. The last operation is to **divide the dataset** into **1-star** ratings, **3-star** ratings, and **5-star** ratings to fully comprehend what determines the users' ratings and reviews.

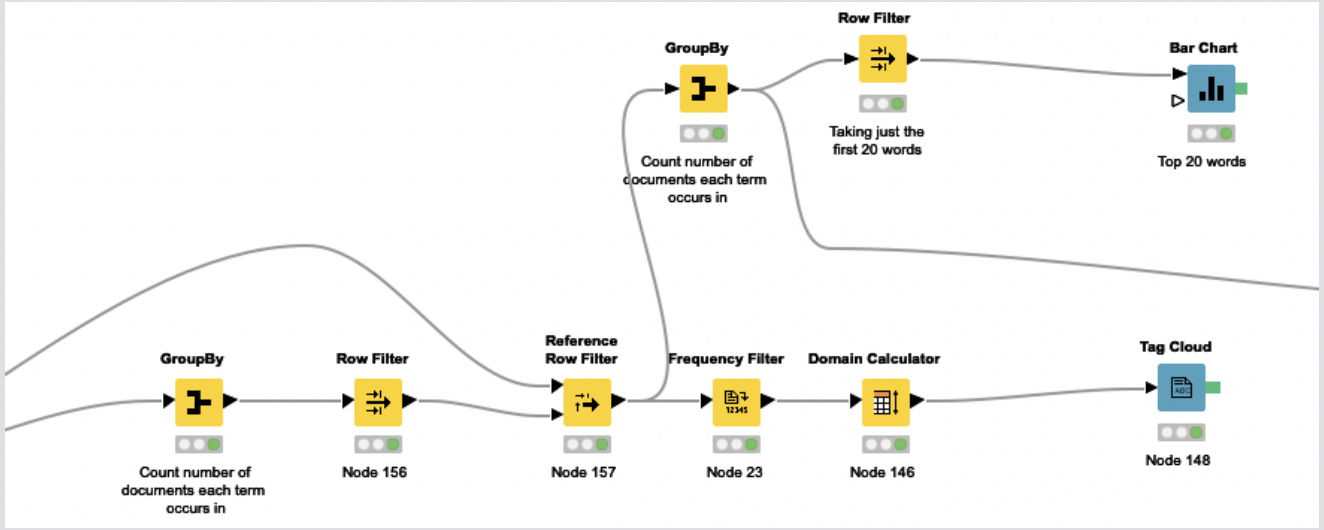




SCROLL



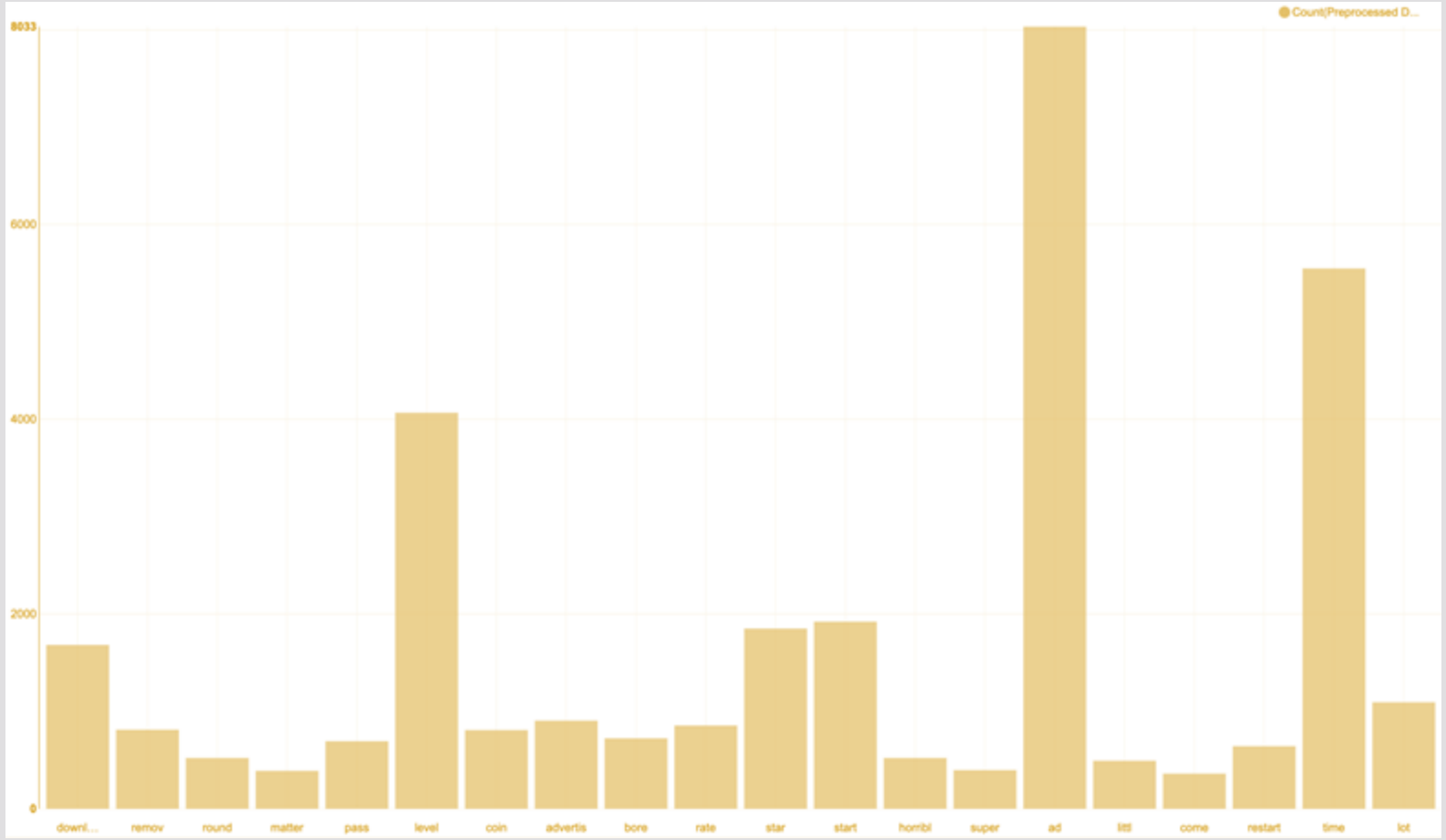
★ 1-Star Rating Analysis ★



After **filtering** for 1-star rating applications, preparation is essential to visualize the desired results. To commence this process, each term is associated with a number representing its occurrence in the document, and for an accurate analysis, it is removed if it appears fewer than three-hundred times. The arbitrary number allows for the elimination of niche terms found in user reviews. The first graphical representation, obtained through filtering the **twenty most recurrent** words, is the **histogram** displaying the value of their occurrence. Instead, to visualize the **word cloud,** a filter is applied to eliminate low **frequency terms** –below the value of 0.5 – prior to a "Domain Calculator" node, which assigns a value based on the "importance" of the term.

★ 1-Star Rating Outcomes ★

The twenty most frequent terms in the worst ranked apps reviews and their occurence



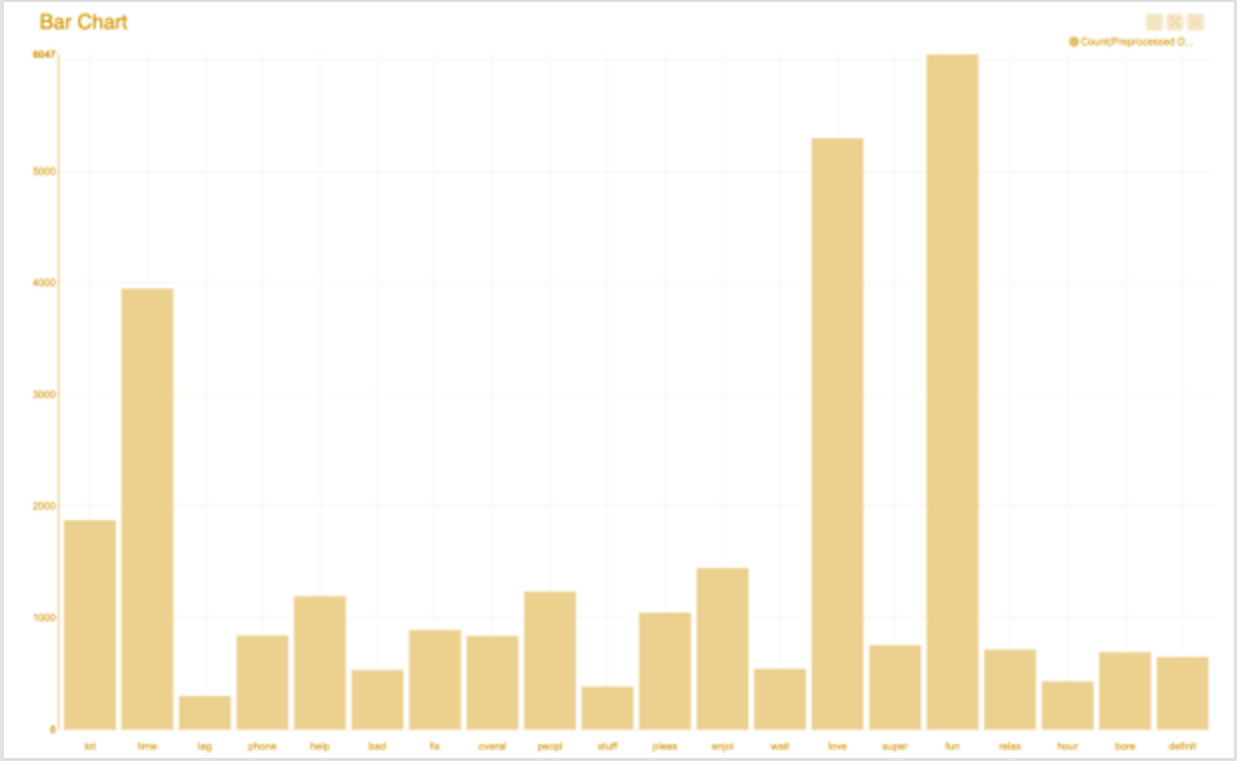
★ 1-Star Rating Outcomes ★

Word clouds are a common way of visualizing a text corpus to understand the most frequently used words. **The size and placement of the words** in the word cloud **vary according to their frequency**. The most relevant term in users' reviews for the worst ranked applications is **“ad”**, which is the word stem for **advertising**; hence, it is a consequence of the public manifestation of disapproval for the strategic positioning of advertising in the customer experience. Other relevant word stems visible from the output are:

- “add”
- “bad”
- “bore”
- “update”
- “level”

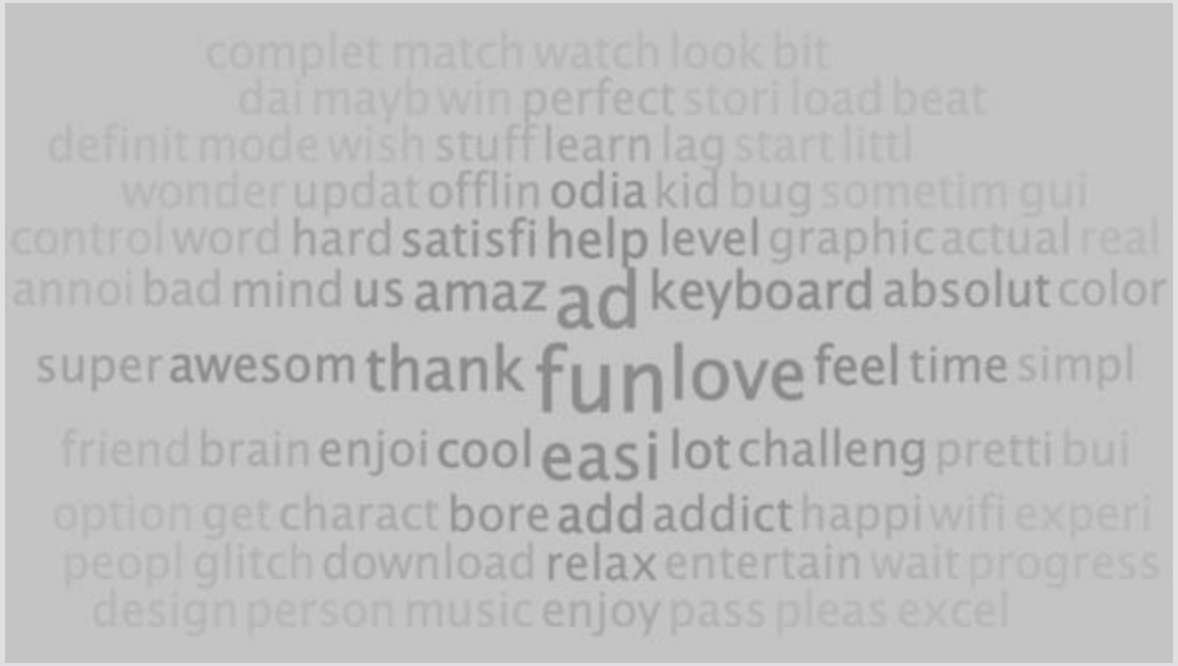
★★★★★ 5-Star Rating Outcomes ★★★★★

After applying the identical procedure to the 5-star rating apps, the twenty most frequent terms in the top ranked application reviews and their occurrence are presented



★★★★★ 5-Star Rating Outcomes ★★★★★

The word cloud procedure is applied for the 5-star rating in the same manner it was applied to the worst ranked applications. The most relevant term in user reviews for top-ranked applications is "fun," indicating a user's requirement. Moreover, other evident terms in the analysis are the word stems “ad” and “easi”, which could be determining factors in the success of an app. Other relevant word stems visible from the output are:

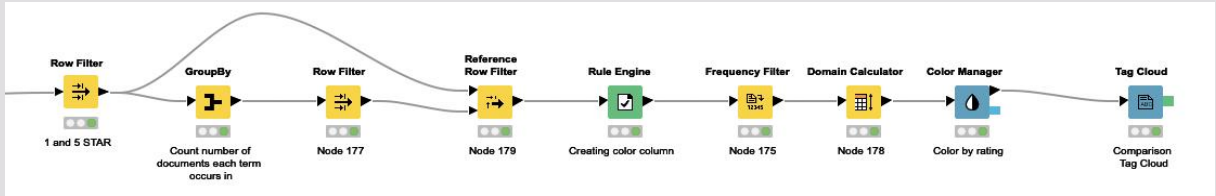


- “love”
- “amaz”
- “help”
- “keyboard”
- “cool”
- “lot”
- “add”

SCROLL

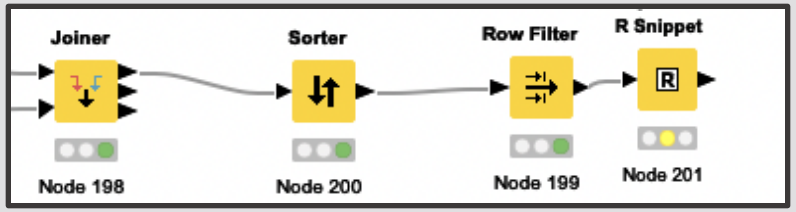
Comparated Analysis

One of the main objectives of this study is to analyze the **difference in keywords** between those who recommend and those who don't recommend the product in order to direct managers and developers toward a successful application. After filtering for the 1-star and 5-star apps, the above-described procedure is applied, with the main difference being the **color assignment**. This instrument functions as a visual aid for the comprehension of differences in keywords. Top ranked apps are associated with the **green** color, while the worst ranked ones are colored **red**. Based on the results of the single class analysis, not surprisingly, the most relevant word stem is **'ad'**. Advertising has a key role in the success of a mobile app.

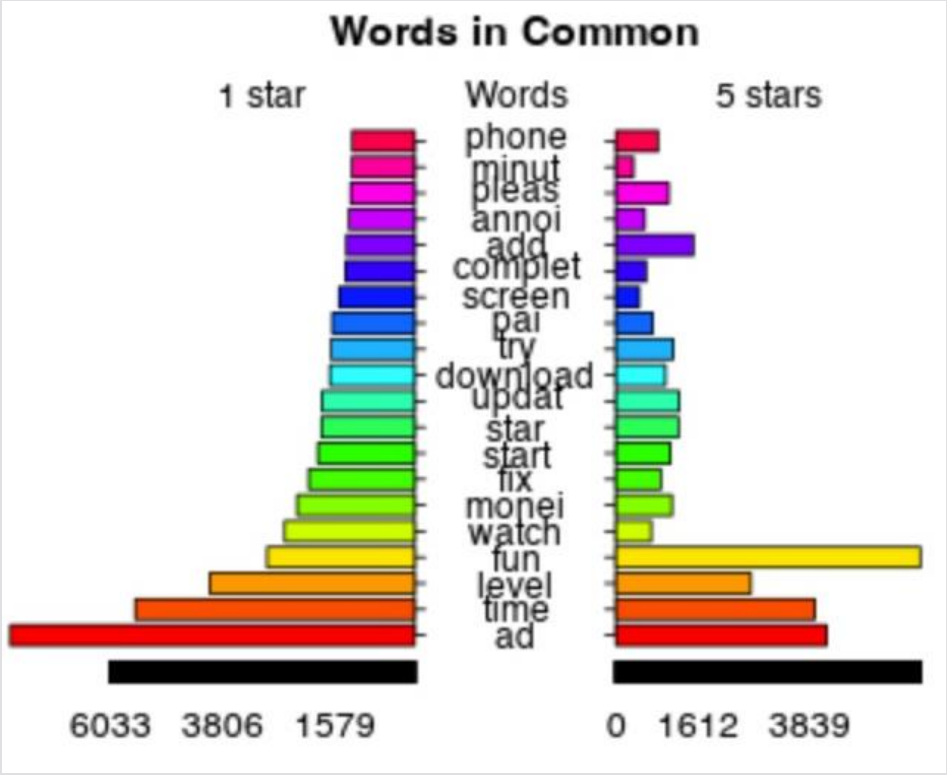


Other **negatively pertinent** word stems are:
"*add*", "*advertis*", "*minut*", and "*download*".

On the contrary, successful app word stems include:
"*fun*", "*help*", "*love*", "*thank you*", *and* "*easy*".



Comparative Analysis



A polarized tag plot is an improved version of the commonality cloud. It determines the **frequency** of a term used in both the corpora under comparison – worst rated against top rated apps. The **difference in frequencies** of common words might be insightful in many cases. First, a matrix is created with all the common words using a subset to ensure that it contains only words occurring in both classes. The matrix then has another column for the absolute difference between both the corpora for each word and the plot. The results emphasize the fact that **associated words are not captured** but only singularly: for example, 'fun' appears in reviews for 1-star apps preceded by 'not', and this is the reason why it is relevant for both classes. The notable insights emerging from the plot are the four relevant and commonly used words:

“ad”, “time”, “level”, “fun”

SCROLL

Outcomes

Intriguing results have emerged from text mining. In the first place, the 3-star rating analysis is not present in the report due to its lack of insightful and informative results obtained.

Second, the worst-rated apps analysis enables managers to gain awareness of what is and is not performing on the market. As an outcome of the analysis of 1-star rating apps, the most relevant keyword is the word stem 'ad'; **strategic positioning of advertising** has a crucial role in app reviews and is a frequently used term in the reviews of top tier apps as well. Hence, managers and developers carefully need to position advertising to engage users and efficiently subject ads to the public. Another urgent managerial decision is **timing**: awful apps tend to force users to **wait hours** in between games or levels, which, by the results, are generally too complicated to pass in this rating class. General complaints about the bad performance of the app on the device are referred to through the usage of an adjective such as **bad** or **garbage**.

Lastly, regarding successful apps, reviews tend to be more detailed and provide suggestions for improvements by the developers. On top of the positive adjectives used to describe the games, such as **fun, cool, and amazing,** the principal upshot given by users is regarding **advertising**. The public suggests that the ads within the mobile app be removed or better managed. According to the public, top performing games should be **easy** to play, with facile **levels**, and should provide users with **help** in the case of a struggle.

Managerial Implications

From a managerial perspective, the precedent's conclusions must be considered as an approach to the market for gaming applications. The text analysis yielded all of the major influential characteristics that a game should have in order to perform well on the test. In addition, the errors and avoidable inaccuracies are highlighted in the model as well, enhancing the negative components that construct the worst rated games in the app store.

This section of the report requires a fundamental declaration: **games** are chosen as the subject of the study, but the **analysis could be applied** to every single category in the dataset or be of interest to the managers. The decision to invest in any segment should be preceded by text mining of the reviews in order to present an outstanding product to the market. This section of the report **serves as illustration** of how the procedure should be carried out with a different category before investing in it. Different outcomes and insights will result i different influential factors and paths to follow when applied to different categories.

In conclusion, for the games segment, executives should invest in or develop an engaging application that is fun, ad-free, and easy to play; provides help to the users and avoids crashing. Following these market insights, **managers will succeed in their investments**.

PRESENTATION
by

Barberis Pietro – **3142272**
Bordoni Matteo – **3133592**
Necchi Marco Vittorio Maria – **3120288**
Porfidia Elena – **3127703**
Ungarelli Federico – **3123212**
Ventura Antonio Roberto – **3127698**

THANK YOU

SCROLL