Università degli Studi di Padova

Strumenti statistici per l'analisi di dati aziendali

Google Play Store: il mercato Android

Gruppo
Girolimetto - Negro - Pozza

Maggio 2020

Introduzione

Il **mercato digitale negli ultimi anni è cresciuto in maniera esponenziale**, in particolare nel settore delle applicazioni digitali per la maggior parte legate a piattaforme come Android ed iOS. Questo forte incremento ha portato all'entrata di sempre più sviluppatori che vogliono rendere la loro app "di tendenza" comportando una maggiore competizione e la necessità per una società di muoversi in maniera mirata sul mercato.

Come si fa? **Capendo in anticipo** se sia opportuno continuare la progettazione di una determinata app oppure se convenga abbandonarla.

Domanda di business

(1) **Siamo in grado di prevedere se un'app sarà di successo?**

(2) **Quali sono i fattori che incidono di più?**

(3) **Esistono delle categorie privilegiate di app?**

Assunzione: ***successo*** $\rightarrow$ app più scaricata

Un mondo eterogeneo

Esiste un mercato "unico" delle app? **NO**
I due ecosistemi più grandi e forti sono il mondo Android guidato dal Google Play Store e quello iOS con Apple Play Store:

(1) **Android** ha attualmente la più grande quota globale e ha un'utenza eterogenea;

(2) **iOS**, invece, presenta degli utenti, in genere, con un reddito più elevato e più propensi alla fidelizzazione (pro per chi è già entrato, contro per chi vuole emergere).

Il dataset

L'assunzione iniziale che abbiamo fatto sulla definizione di successo, ci porta quindi a scegliere di concentrarci sul mercato più ampio possibile, cioè Android.

Per fare questo è stato utilizzato il dataset "Google Play Store Apps", disponibile su Kaggle, il quale contiene, per 9660 applicazioni, il numero di download ed altre variabili esplicative relative all'anno 2018.

Google Play Store Apps: variabile risposta

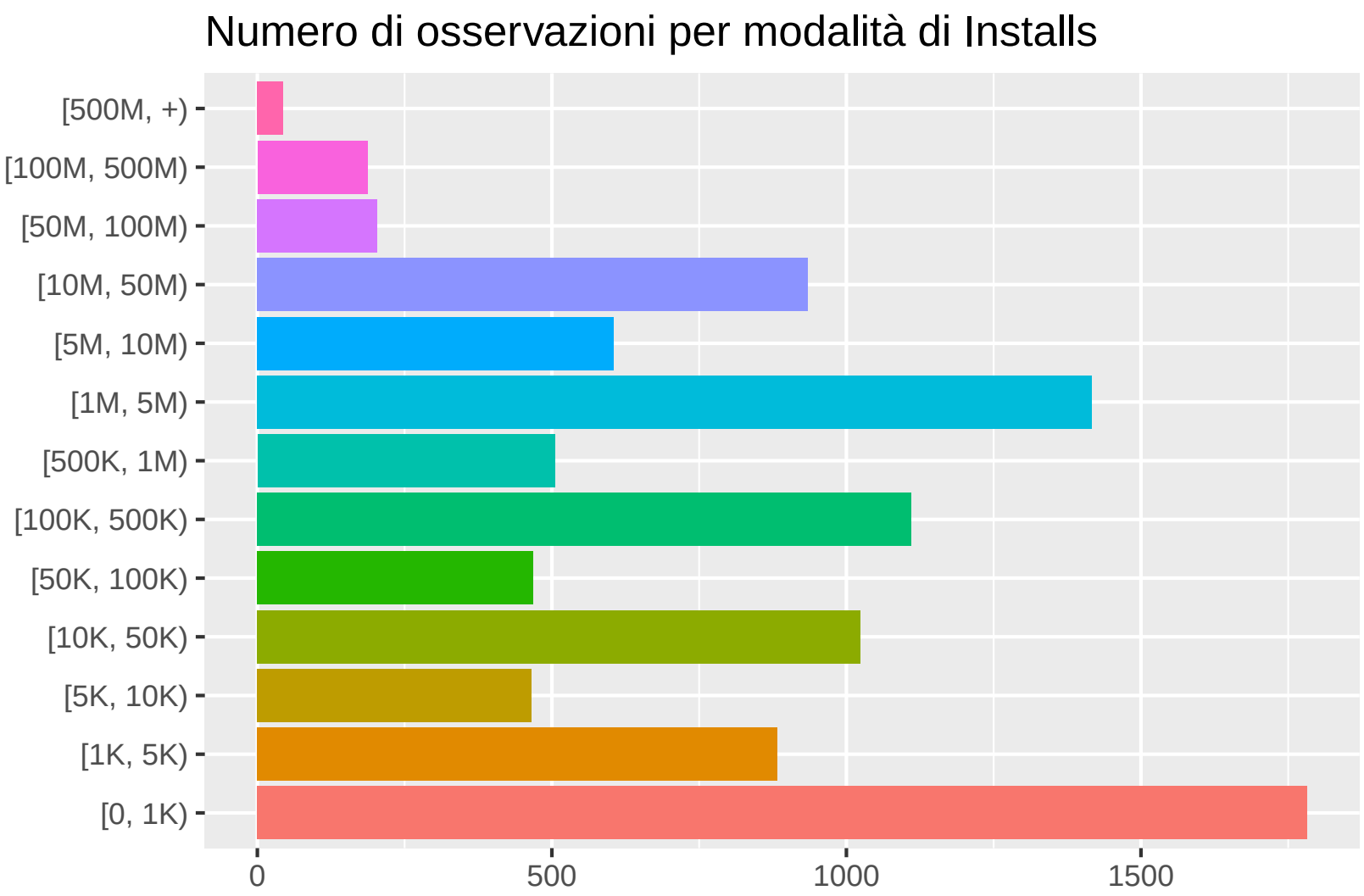
Purtroppo non abbiamo a disposizione il numero esatto di download della singola app (unità statistica).

La variabile risposta che abbiamo a disposizione, **Installs**, categorizza, con 22 modalità, il numero di utenti che hanno scaricato l'applicazione almeno su un dispositivo. Se l'applicazione viene più volte installata da un singolo utente, anche su altri dispositivi, il conteggio non subisce variazioni.

Google Play Store Apps: variabile risposta

Il numero di modalità è molto elevato: molte di queste presentano un andamento simile, ma soprattutto abbiamo categorie che presentano un numero esiguo di osservazioni e non hanno un vero scopo per esistere in modo singolo (come da 0 a 5 o da 5 a 10 download). Per questo motivo abbiamo deciso di raggrupparle, ottenendo un totale di 12 classi.

Google Play Store Apps: variabile risposta



Google Play Store Apps: le altre variabili

Sebbene il dataset sia composto da un numero relativamente piccolo di variabili (12), dal punto di vista statistico non si può considerare il nostro un problema a bassa dimensionalità. Sono,infatti, presenti una serie di variabili categoriali con un numero estremamente elevato di modalità. In particolare le variabili disponibili sono

- **Size**: variabile continua, inidica la quantità di memoria necessaria per installare l'applicazione.
- **Category**: categoria a cui appartiene l'applicazione. 34 modalità.
- **Type**: variabile dicotomica che indica se l'applicazione è gratutita oppure a pagamento.
- **Price**: variabile continua che indica il costo dell'applicazione ed è 0 se è gratis.

Google Play Store Apps: le altre variabili

- **Content Rating**: variabile categoriale che indica la fascia d'età per la quale è stata progettata l'applicazione, 7 modalità.
- **Genres**: genere dell'applicazione, 120 modalità.
- **Android Version**: versione minima di android neccessaria per poter utilizzare l'applicazione, 35 modalità
- **Current Version**: ultima versione dell'applicazione disponibile sull'app store, 2834 modalità.

Google Play Store Apps: le altre variabili

- **Last Updated**: Data nella quale l'applicazione è stata aggiornata per l'ultima volta.
- **Rating**: Indica il rating assegnato dagli utenti all'applicazione.
- **Reviews**: Numero di reviews per l'applicazione.

Problematiche e assunzioni sulle variabili esplicative

E' importante sottolineare come le variabili **Current Version**, **Last Updated**, **Rating** e **Reviews** non siano disponibili nel momento del lancio di una applicazione. Per questo motivo, per essere utilizzate, necessitano di una analisi più approfondita.

Rating, Last Updated, Current Version

In particolare, a nostro avviso, la variabile **Rating** può essere utilizzata assumendo che una stima di essa è ottenibile facendo testare, in fase di sviluppo, l'applicazione a dei tester.

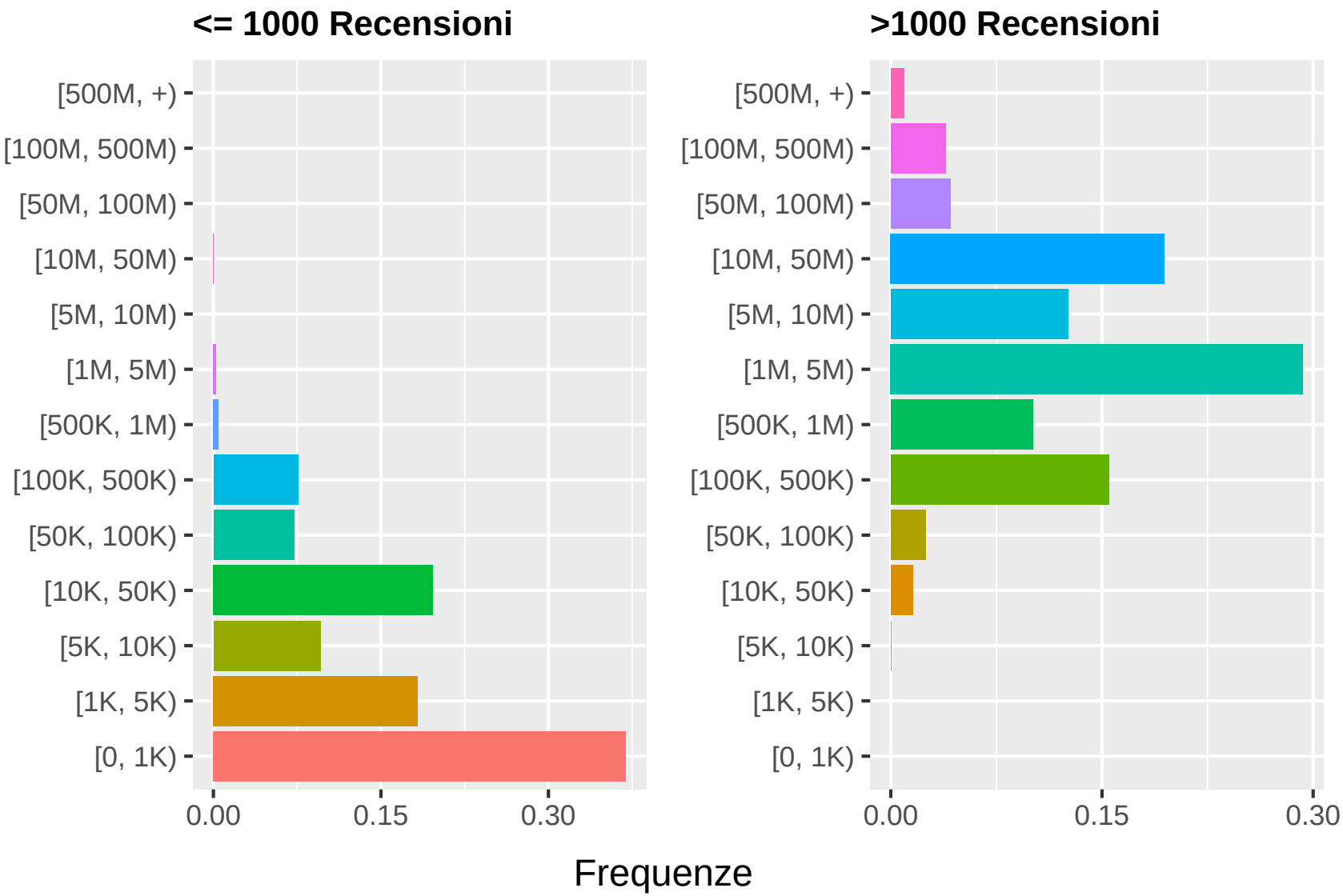
La variabile **Last Updated** può essere un buon indicatore di come bisogna comportarsi una volta che l'app viene lanciata: è conveniente continuare ad aggiornare l'app anche successivamente?

Current Version, invece, presenta modalità che non sono standard, ma differiscono a seconda dello sviluppatore. In questo senso non può venire utilizzata per quest'analisi.

Reviews

Diverso è il caso del numero di reviews. Infatti, è possibile immaginare che una politica di marketing atta a far aumentare il numero di recensioni di una applicazione possa aumentarne la popolarità. Allo stesso tempo, però, i dati "fotografano" il numero di istallazioni in un determinato istante temporale rendendo praticamente impossibile l'utilizzo diretto di questa variabile. E'infatti abbastanza ovvio che le applicazioni più scaricate abbiano anche un numero maggiore di recensioni, indipendentemente dalla politica utilizzata.

Reviews


<= 1000 Recensioni
>1000 Recensioni
[500M, +)
[100M, 500M)
[50M, 100M)
[10M, 50M)
[5M, 10M)
[1M, 5M)
[500K, 1M)
[100K, 500K)
[50K, 100K)
[10K, 50K)
[5K, 10K)
[1K, 5K)
[0, 1K)
0.00
0.15
0.30
Frequenze

Reviews rate

Ci siamo quindi trovati di fronte a due possibili scelte

- **Non utilizzare la variabile Reviews**: In questo modo avremmo eliminato completamente il rischio di sovradattamento. Allo stesso tempo però sarebbe potenzialmente andata perduta dell'informazione importante per il nostro problema di business.

- **Lavorare su una opportuna traformata di Reviews**: Presenta un rischio di sovradattamento molto elevato ma allo stesso tempo, se valutato con spirito critico, potrebbe fornire delle indicazioni interessanti sul fenomeno da noi considerato.

Reviews rate

Dato che il dataset di partenza non risulta particolarmente dettagliato abbiamo optato per la seconda opzione, scegliendo di creare una nuova variabile, **reviews.rate**, definita come

$$\text{reviews.rate} = \frac{\text{Reviews}}{\text{Installs}^*}.$$

* *non avendo a disposizione il vero numero di download, abbiamo considerato l'estremo superiore della classe.*

Reviews rate: alcune considerazioni

A questo punto è opportuno sottolineare alcune caratteristiche della variabile creata

(1) E' interpretabile come un indice "di apertura" dell'applicazione al farsi valutare dai propri clienti. E' quindi naturale chiedersi se ad un maggior livello di questo indice corrisponda un maggior successo per l'applicazione.

(2) E' un numero puro, ossia senza unità di misura, compreso tra 0 e 1. Nonostante questo, è innegabile che vi possa essere ancora una forte dipendenza con la variabile "Installs"

(3) Può essere vista come la stima di massima verosimiglianza del parametro di una variabile binomiale.

Reviews rate: stima bayesiana empirica

Utilizzando l'osservazione **3)** possiamo pensare di ridurre la dipendenza di reviews.rate dalla variabile "Installs" utilizzando un **approccio bayesiano empirico**. Questo si traduce in pratica indicando con r_i, il numero di reviews, con n_i, il numero di installazioni e con θ_i il valore della variabile reviews.rate per l'applicazione i-esima. Dato che il nostro obiettivo è ottenere una stima per θ_i possiamo specificare il seguente modello bayesiano

$$y_i \sim Bi(n_i, \theta_i) \quad \text{con} \quad \theta_i \sim Be(\alpha, \beta),$$

dove con $Be(\alpha, \beta)$ assumiamo che la distribuzione a priori di θ_i sia una beta con iper-parametri α e β da identificare.

Reviews rate: stima bayesiana empirica

- Una stima bayesiana "classica" implicherebbe che gli iper-parametri α e β fossero fissati utilizzando una qualche sorta di **conoscenza a priori** del fenomeno di interesse. Visto, però, che stiamo cercando di stimare un numero molto grande di parametri θ_j, possiamo pensare di ottenere un valore sensato per α e β direttamente dai dati.
- L'approccio bayesiano empirico raggiunge questo risultato ponendo α e β pari alle relative stime di massima verosimiglianza $\hat{\alpha}$ e $\hat{\beta}$. I valori per la variabile reviews.rate vengono quindi ottenuti calcolando la media a posteriori di θ_i utilizzando come iper-parametri $(\hat{\alpha}, \hat{\beta})$

Presenza di NA nelle esplicative

Dove il numero di NA era ridotto abbiamo deciso di eliminare l'intera osservazione, tuttavia per due variabili questo non è stato possibile. Infatti Size e Ratings presentavano, rispettivamente 1224 (12.71%) e 1455 (15.11 %) valori mancanti, per un totale di 2622 (27.24%) osservazioni che avevano NA in almeno una delle due.

Si è deciso di inputarle utilizzando il valore mediano all'interno dello stesso genere.

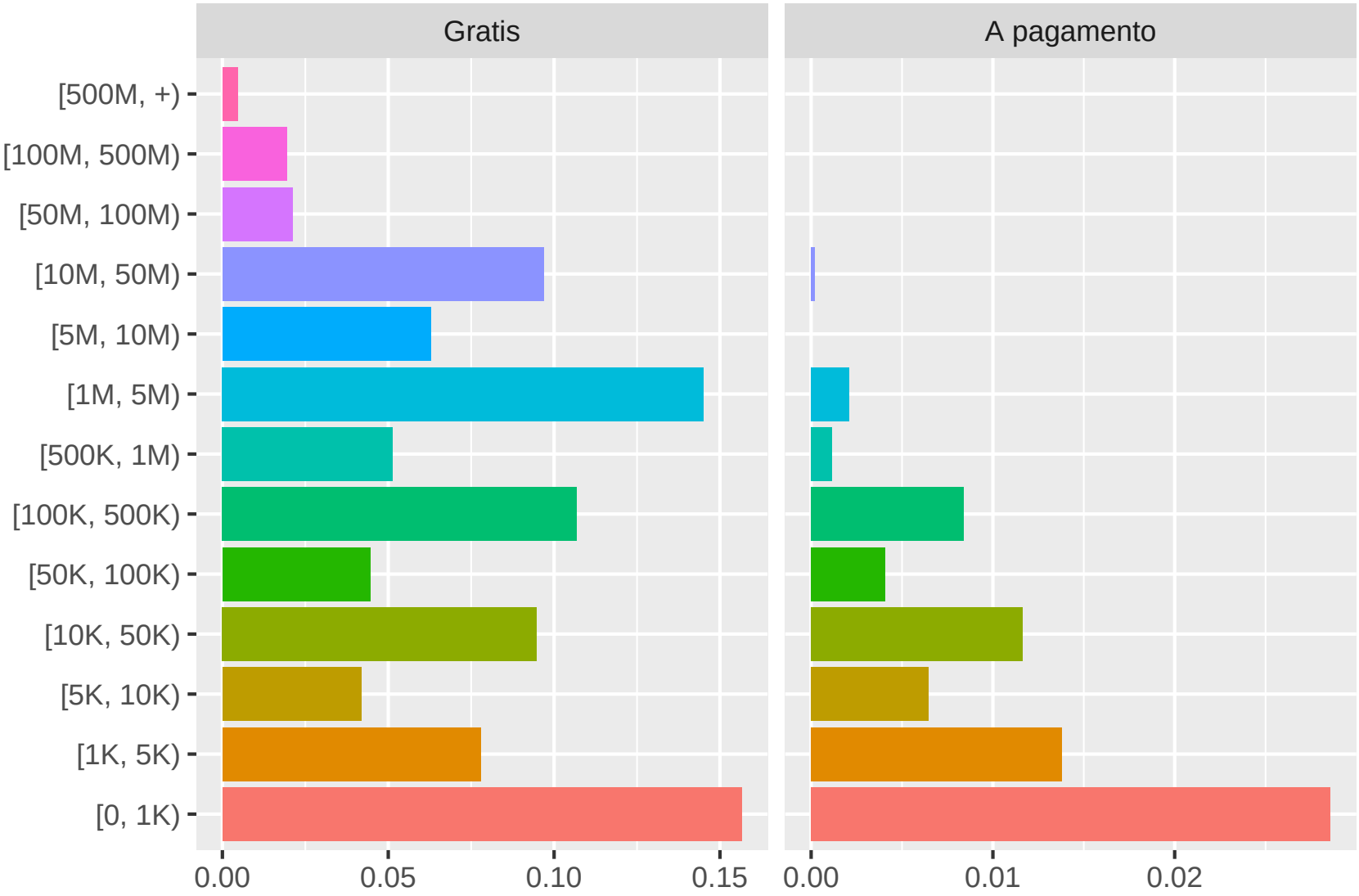
Riassumendo

- **Size**: variabile continua, dimensione in byte.
- **Category**: variabile categoriale con 33 modalità.
- **Price**: variabile continua, da 0 a 100 dollari.
- **Content Rating**: variabile categoriale con 3 modalità.
- **Genres**: variabile categoriale con 52 modalità.
- **Android Ver**: variabile categoriale con 8 modalità.
- **Last Updated**: data.
- **Rating**: variabile continua da 0 a 5.
- **Reviews rate**: variabile continua da 0 a 1.

Nel dataset vi è un totale di 9627 osservazioni.

Analisi esplorativa

Price


Gratis
A pagamento
[500M, +)
[100M, 500M)
[50M, 100M)
[10M, 50M)
[5M, 10M)
[1M, 5M)
[500K, 1M)
[100K, 500K)
[50K, 100K)
[10K, 50K)
[5K, 10K)
[1K, 5K)
[0, 1K)
0.00
0.05
0.10
0.15
0.00
0.01
0.02

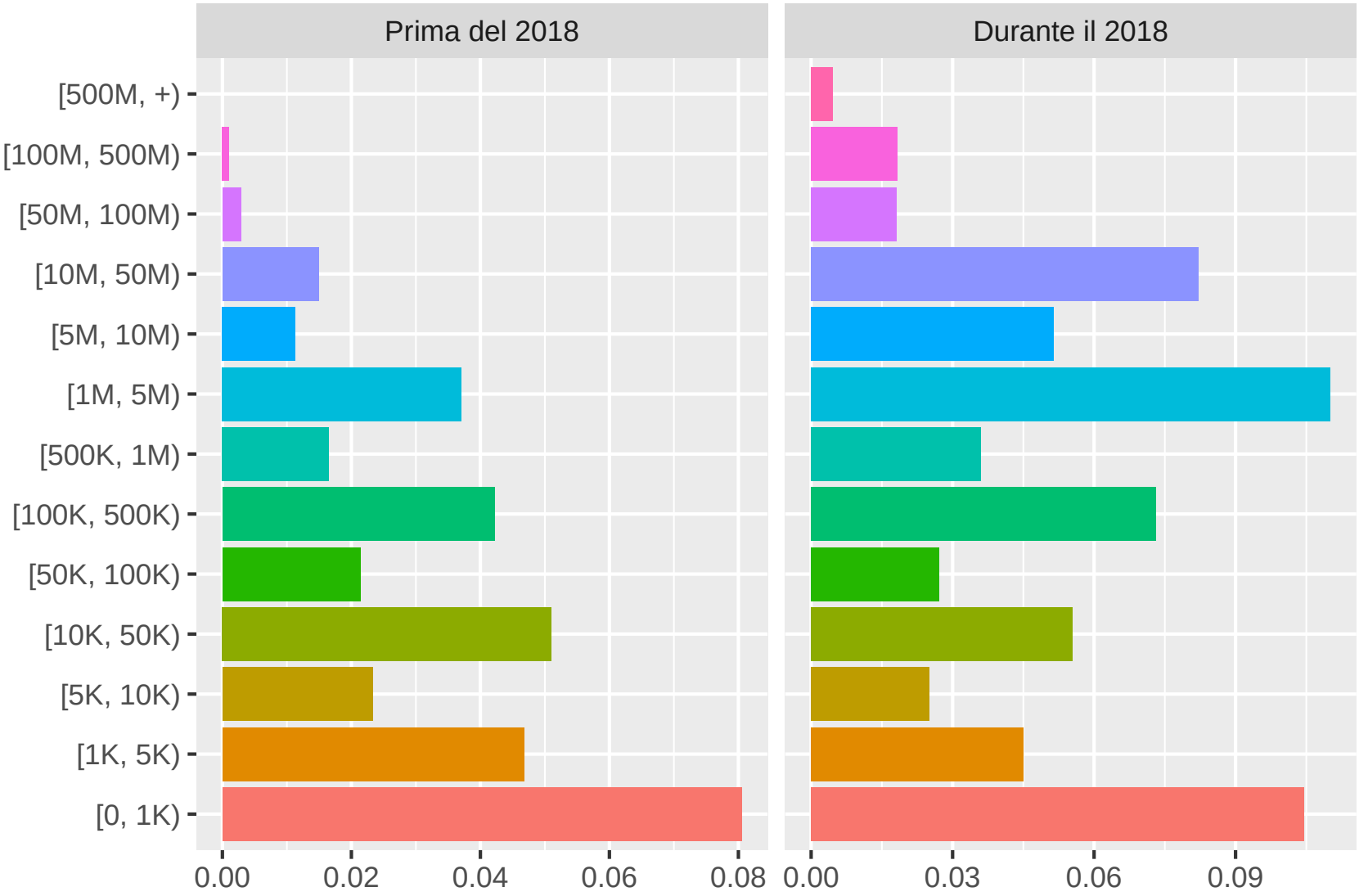
Price

Non si deve cadere nel tranello di considerare un app gratis meno redditizia.

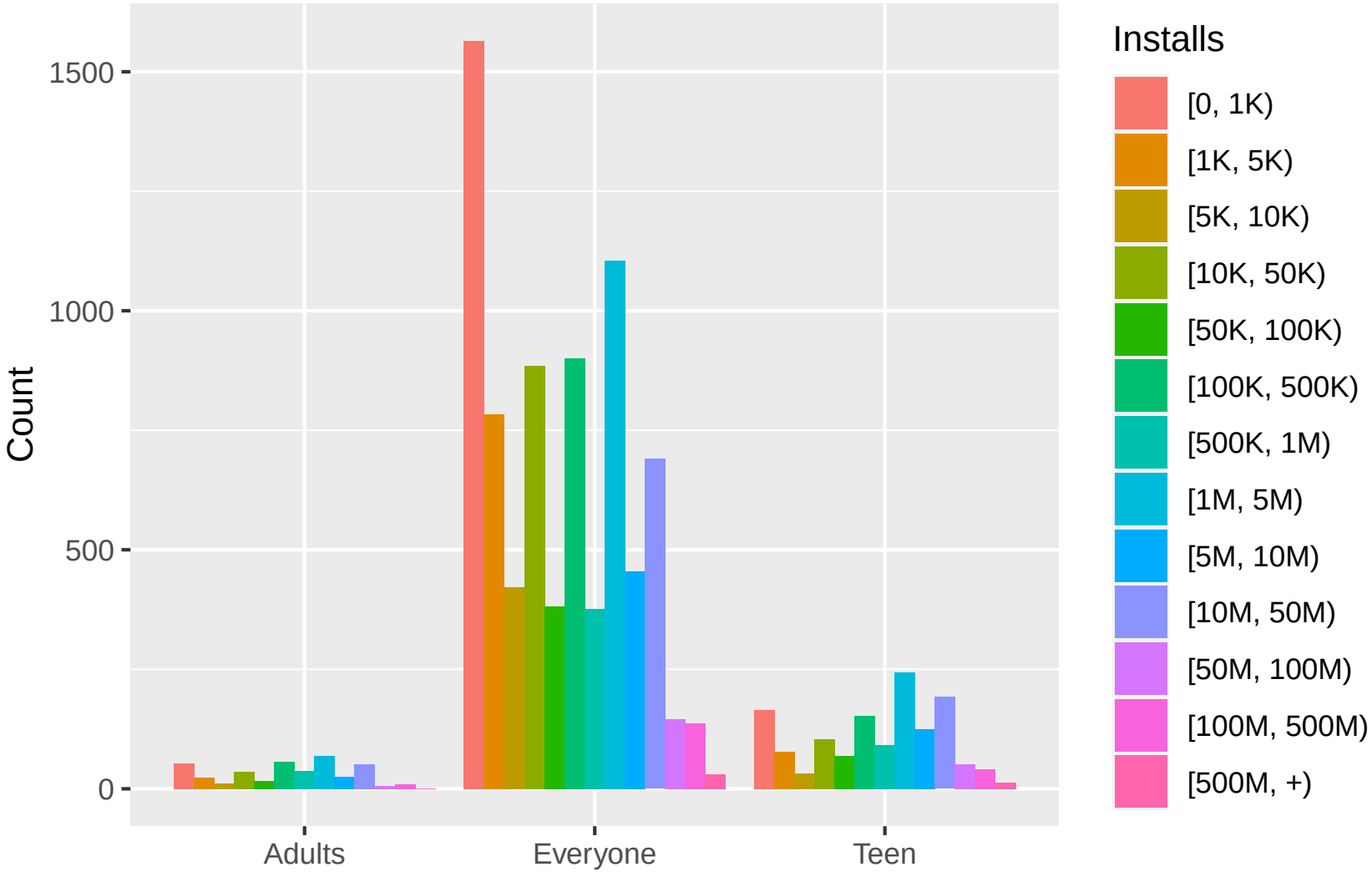
Infatti l'app viene considerata gratis nel momento dell'acquisto: molte di queste prevedono la possibilità di spendere denaro all'interno, come ad esempio per app di gioco d'azzardo oppure giochi in cui vengono venduti oggetti estetici.

Tra il 2016-2019, per la prima volta, è scoppiato uno scandalo che ha coinvolto la struttura delle *loot boxes* che ha portato molti dirigenti di società videoludiche (anche nel settore del mobile, come EA) a doversi difendere dall'accusa di aver introdotto minorenni del gioco d'azzardo.

LastUpdate


Prima del 2018
Durante il 2018
[500M, +)
[100M, 500M)
[50M, 100M)
[10M, 50M)
[5M, 10M)
[1M, 5M)
[500K, 1M)
[100K, 500K)
[50K, 100K)
[10K, 50K)
[5K, 10K)
[1K, 5K)
[0, 1K)
0.00
0.02
0.04
0.06
0.08
0.00
0.03
0.06
0.09

Content rating

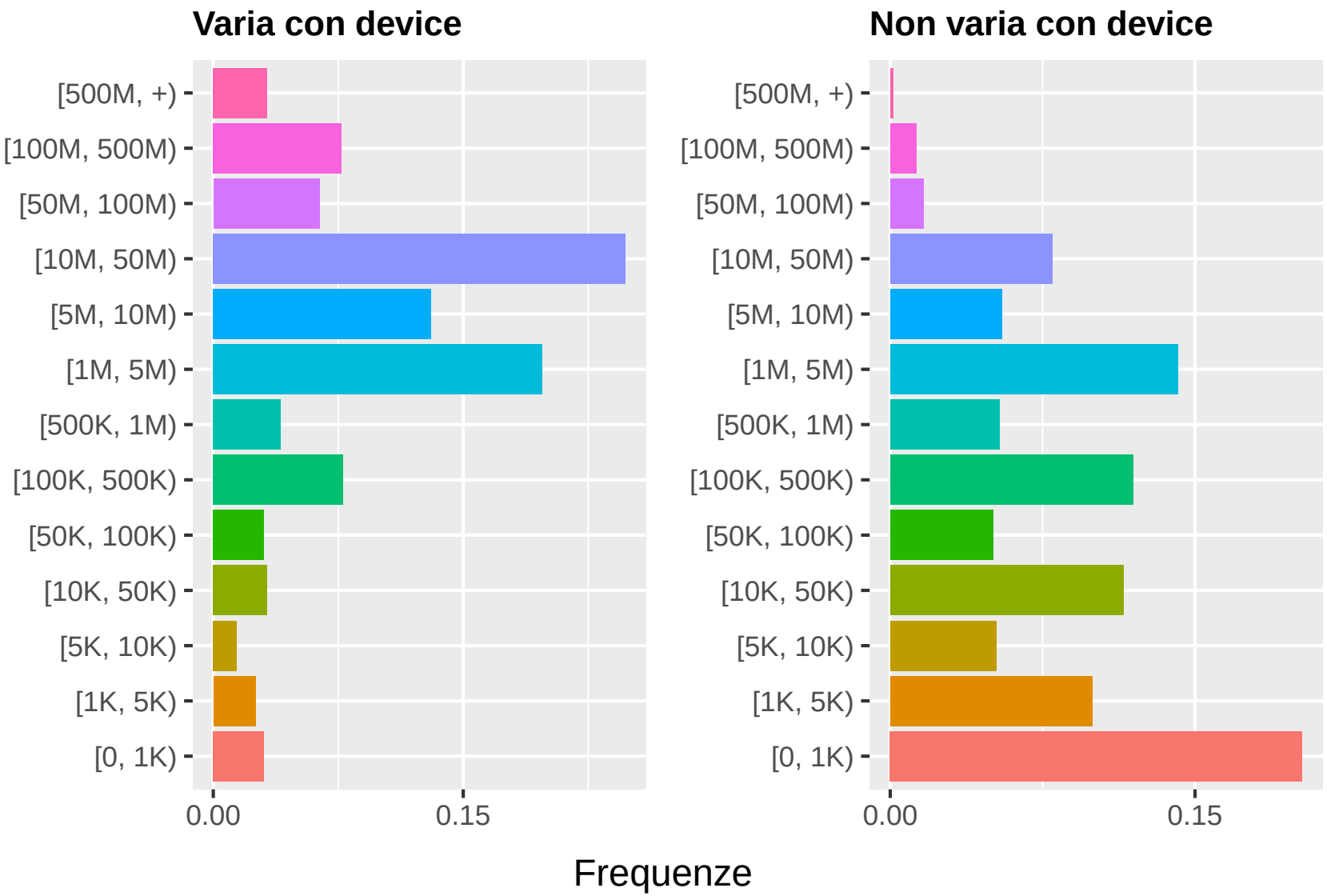


Content rating: commento

Si può subito osservare come la maggior parte delle applicazioni sia dedicata ad utenti senza un particolare limite di età. La distribuzione di download sembra essere abbastanza costante nelle tre classi e concentrata attorno a valori medio-bassi.

Interessante è il caso della categoria "teen". Per questa classe di applicazioni, infatti, la percentuale di unità con più di 500 milioni di download sembra essere leggermente più alta.

Varies with device

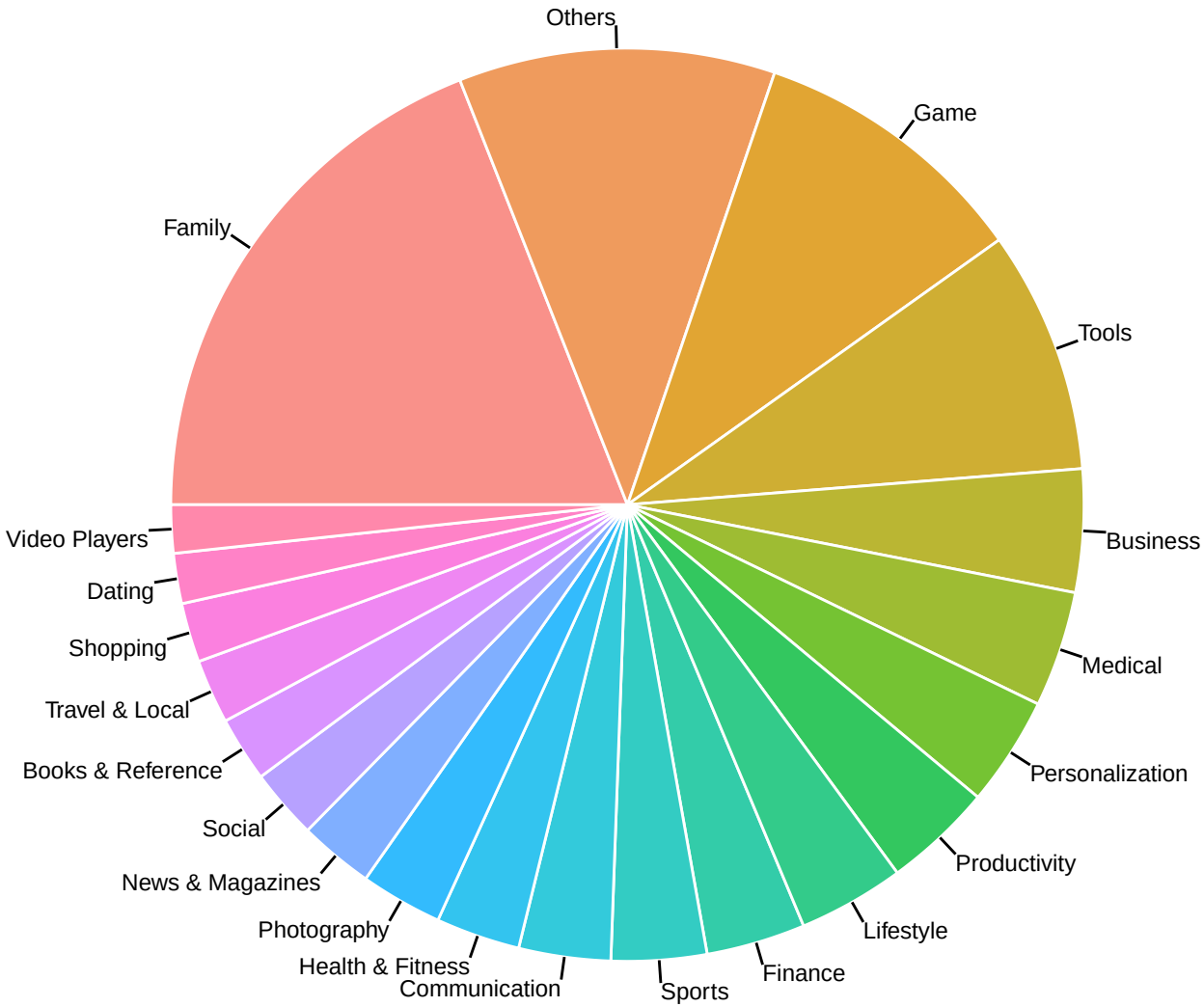


Varies with device

Le applicazioni che prevedono diversi requisiti,in base al modello di cellulare, per quanto riguarda il sistema operativo, sono in generale più scaricate.

Questo fenomeno ci sembra abbastanza ragionevole. Infatti è probabile che questo genere di applicazioni possa funzionare in un numero maggiore di dispositivi. Di conseguenza, anche il numero di clienti potenzialmente raggiungibili è maggiore.

Distribuzione della variabile Category



Distribuzione della variabile Category

Osservando il grafico a torta in Figura 31 risulta evidente come le categorie di applicazioni più diffuse siano di gran lunga **Family**, **Games** e **Tools**. Nella fase dell'inferenza statistica sarà quindi interessante valutare se a questo tipo di applicazioni corrisponda una maggiore probabilità di successo.

Analisi e modellistica

Introduzione

Quanto osservato in fase di analisi esplorativa può sicuramente migliorare la nostra comprensione riguardo i fattori che caratterizzano un'applicazione di successo. Allo stesso tempo, per trarre conclusioni più accurate, è necessario testare queste ipotesi attraverso degli opportuni modelli statistici.

Per fare questo riteniamo che la variabile Installs con 12 classi sia eccessivamente troppo ampia, quindi l'abbiamo ridotta a categoriale con 4 classi: [0, 10K), [10k, 1M), [1M, 10M) e [10M, +).

Introduzione

Prima di tutto abbiamo diviso il dataset in due parti:

- un'insieme di **stima** con il 70% delle osservazioni;
- e quello di **verifica** con il restante 30%.

Oltre i classici modelli visti a lezione abbiamo anche considerato modelli per variabili categoriali con penalizzazione lasso.

Modelli per variabili categoriali con penalizzazione lasso

Come è ben noto, il modello lineare con penalizzazione lasso può risultare molto efficace nella gestione di modelli con un alto numero di variabili esplicative, migliorando allo stesso tempo sia la capacità previsiva che l'interpretabilità. Per variabili categoriali, due possibili estensioni si basano sul modello **multinomiale** e sul modello a **logit cumulati.** Nel modello multinomiale viene minimizzata la quantità

$$-\frac{1}{n}\sum_{i=1}^{n}\log P\left(Y=y_i|x_i;\{\beta_{0k},\beta_k\}_{k=1}^{K}\right)+\lambda\sum_{k=1}^{K}\parallel\beta_k\parallel_1,$$

Modelli per variabili categoriali con penalizzazione lasso

Nel modello a logit cumulati con penalizzazione lasso ad essere minimizzata è invece la quantità

$$-\frac{1}{n}\sum_{i=1}^{n}\log P\left(Y = y_i|x_i;\beta_0,\beta\right) + \lambda \parallel \beta \parallel_1 .$$

In entrambi i casi il valore ottimale per il parametro λ verrà scelto attraverso una convalida incrociata sull'insieme di stima. Infine è importante sottolineare che, in entrambi i modelli, la stima dei coefficienti non è invariante rispetto alla standardizzazione delle variabili di partenza. Per favorire l'interpretabilità, nel seguito, verranno quindi utilizzate come esplicative le variabili standardizzate.

Metodi di classificazione

Nella scelta del modello più adeguato abbiamo deciso di tenere conto sia dell'interpretabilità che della capacità previsiva. Per quest'ultimo aspetto abbiamo utilizzato i seguenti indici:

- tasso di errata classificazione;
- percentuale dei falsi positivi (osservazioni erratamente classificate [1M, 10M) e [10M, +);
- percentuale dei falsi negativi (osservazioni erratamente classificate [0, 10K), [10k, 1M).

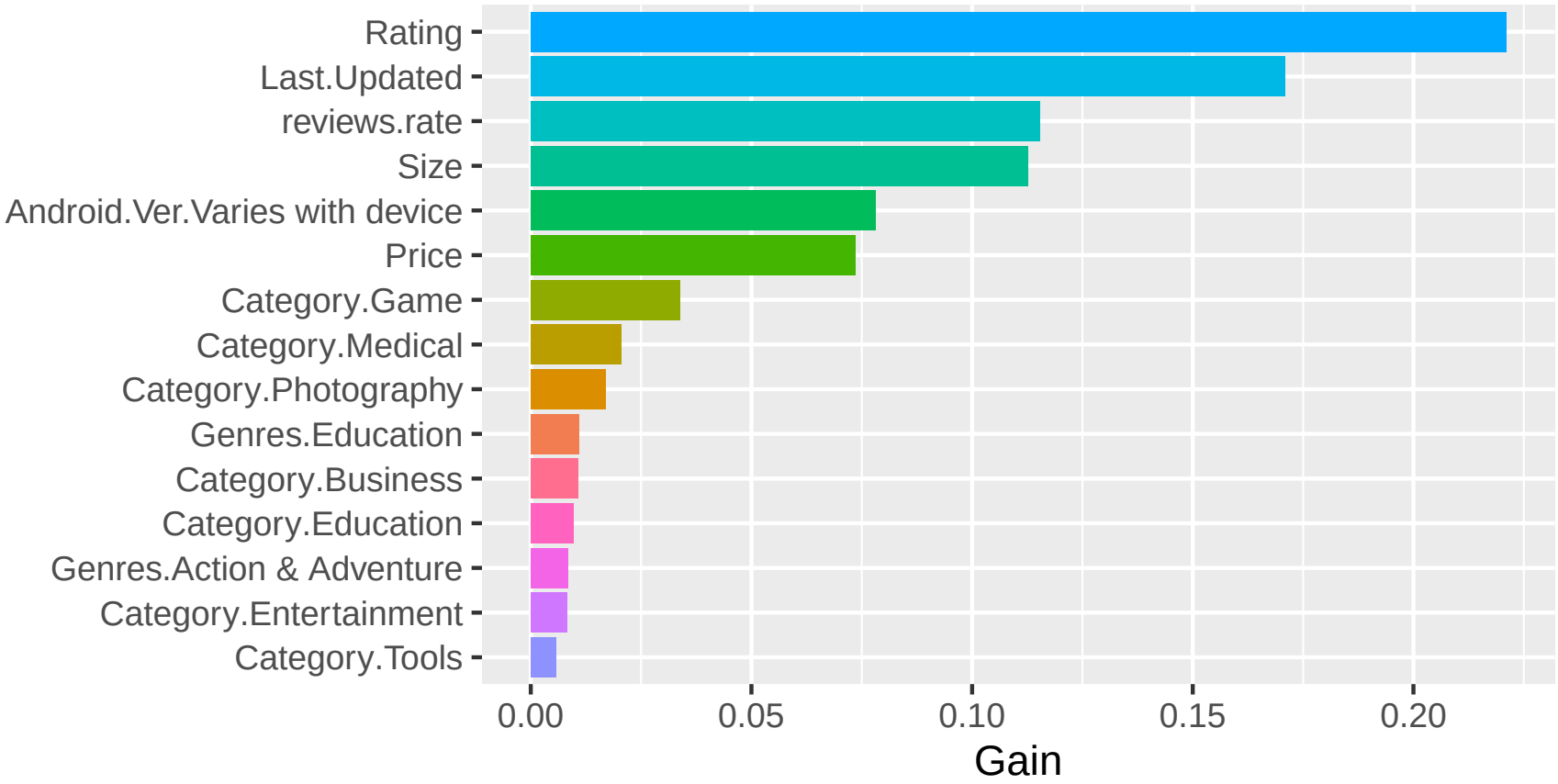
Metodi di classificazione

Modello	**tasso di errata classificazione**	**% falsi negativi**	**% falsi positivi**
Modello lineare	0.5335	10.33%	52.47%
Analisi discriminante	0.5443	12.83%	50.69%
MARS	0.4847	13.31%	46.23%
Alberi di classificazione	0.5349	12.19%	54.65%
Random Forest	0.4577	11.92%	38.71%
Multinomiale lasso	0.5275	9.73%	49.80%
KNN (k=5)	0.5973	25.32%	51.02%
Logit cumulati	0.5685	9.04%	54.35%
Logit cumulati lasso	0.564	8.78 %	54.25%
Gradient boosting	0.442	12.13 %	34.85%

Metodi di classificazione

Modello	**tasso di errata classificazione**	**% falsi negativi**	**% falsi positivi**
Modello lineare	0.5335	10.33%	52.47%
Analisi discriminante	0.5443	12.83%	50.69%
MARS	0.4847	13.31%	46.23%
Alberi di classificazione	0.5349	12.19%	54.65%
Random Forest	0.4577	11.92%	38.71%
Multinomiale lasso	0.5275	9.73%	49.80%
KNN (k=5)	0.5973	25.32%	51.02%
Logit cumulati	0.5685	9.04%	54.35%
Logit cumulati lasso	0.564	8.78 %	54.25%
Gradient boosting	**0.442**	12.13 %	34.85%

Gradient boosting: Importanza variabili



Gradient boosting: considerazioni

Ovviamente il Gradient Boosting non fornisce nessuna indicazione sul tipo di influenza che le variabili esercitano sul numero atteso di download, ma solo il loro grado d'importanza.

Significativa è la presenza di **Ratings** e **Last.Update** nei primi posti.

Pure la dimensione dell'App sembra avere una qualche rilevanza come la capacità dell'app di adattarsi alla versione Android del device dell'utente.

Nelle prime posizioni troviamo, in aggiunta, **Android.Ver.Varies with device** e **Price**.

Logit cumulati con penalizzazione lasso

Variabili con coefficiente positivo					
Android.Ver.Varies with device	1.87	Category.Shopping	0.47	Genres.Communication	0.23
Category.Entertainment	1.58	Category.Weather	0.40	Category.Communication	0.18
Category.Education	1.33	Genres.Role Playing	0.40	Genres.Weather	0.16
Genres.Action & Adventure	1.31	Genres.Casual	0.39	Android.Ver.2.0 and up	0.16
Category.Photography	1.13	Genres.Racing	0.32	Genres.Photography	0.13
Category.Game	0.91	Size	0.32	Category.House & Home	0.09
Genres.Brain Games	0.60	Genres.Simulation	0.32	Content.Rating.Adults	0.08
Genres.Pretend Play	0.56	Genres.Puzzle	0.28	Android.Ver.1.0 and up	0.07
Category.Video Players	0.55	Category.Tools	0.28	reviews.rate	0.06
Genres.Strategy	0.52	Genres.Creativity	0.28	Genres.Shopping	0.04

Logit cumulati con penalizzazione lasso

Variabili con coefficiente negativo					
(Intercept):4	-14.99	Genres.Card	-0.50	Content.Rating.Everyone	-0.22
(Intercept):3	-13.51	Category.Dating	-0.47	Genres.Casino	-0.22
(Intercept):2	-11.83	Category.Auto & Vehicles	-0.44	Android.Ver.6.0 and up	-0.21
Genres.Trivia	-1.52	Category.Lifestyle	-0.34	Genres.Adventure	-0.19
Category.Events	-1.19	Genres.Educational	-0.32	Category.Health & Fitness	-0.15
Category.Medical	-1.14	Genres.Entertainment	-0.29	Category.Family	-0.11
Category.Business	-0.93	Genres.Music	-0.27	Android.Ver.4.0 and up	-0.10
Genres.Education	-0.69	Category.News & Magazines	-0.27	Category.Books & Reference	-0.08
Android.Ver.7.0 and up	-0.53	Genres.Board	-0.25	Genres.Dating	-0.08
Price	-0.52	Category.Finance	-0.23	Category.Personalization	-0.08

Logit cumulati con penalizzazione lasso: considerazioni

Le tabelle nelle due slide precedenti ci permettono di trarre una serie di interessanti considerazioni

- La categoria **Game** sembra essere particolarmente diffusa tra applicazioni "di successo". Infatti, tra le variabili con coefficienti più alti, sono presenti numerose sue sotto categorie.
- Produrre una applicazione compatibile con vasta gamma di dispositivi android sembra essere una condizione fondamentale per favorirne un'ampia diffusione.
- Dal nostro punto di vista, il coefficiente positivo per la variabile **Size**, sta ad indicare che gli utenti Android tendono a prediligere applicazioni ben curate e quindi necessariamente più pesanti in termini di memoria occupata sul dispositivo.

Logit cumulati con penalizzazione lasso: considerazioni

- L'avere una community di utenti attiva in termini di recensioni può, in generale, migliorare la diffusione dell'applicazione.
- Le applicazioni a pagamento generalmente ottengono una diffusione più limitata.

Conclusioni analisi

Dal nostro punto di vista, un'azienda interessata a sviluppare un nuovo prodotto da rendere disponibile su Play Store, dovrebbe focalizzarsi sulla produzione di una applicazione

- Gratuita.
- Appartente alla categoria Game
- Ben sviluppata (anche a discapito della memoria occupata sui dispositivi).
- Ottimizzata per il maggior numero possibile di versioni Android.

Nelle fasi successive al lancio del prodotto gli sforzi andrebbero focalizzati nel mantenimento di una community di clienti attiva. Questa potrebbe, infatti, fornire interessanti suggerimenti per migliorare l'applicazione nei successivi aggiornamenti, i quali devono essere abbastanza frequenti.

Text Mining

Text Mining

Oltre al dataset con i nomi e le caratteristiche delle App, avevamo a disposizione i testi di alcune recensioni.

Abbiamo deciso di sfruttare anche le informazioni di questo insieme di dati per poter ricavare alcune informazioni utili su che cosa funzioni o non funzioni, e quali sono le cause delle maggiori lamentele da parte degli utenti.

Text Mining

Per ogni applicazione avevamo a disposizione il nome della stessa e il testo della recensione in lingua inglese.

Da un totale di 64295 recensioni rimuovendo i valori mancanti ne abbiamo tenute 37432.

Text Mining: operazioni preliminari

Come in ogni analisi di text mining sono state effettuate alcune operazioni preliminari:

- le parole scritte in modo sbagliato più frequentemente sono state corrette e sostituite, in modo tale da uniformarle tra le varie recensioni e renderle interpretabili (*es. angri -> angry*);
- sono state rimosse le *stop words*, utilizzando il dizionario ontologico *stopwords-iso* con l'aggiunta di alcune parole non significative che ho notato tra le più frequenti.

Sentiment Analysis

La Sentiment Analysis è una procedura che assegna un punteggio, o un coefficiente, in base alla positività o negatività di un determinato testo.

In quella che abbiamo utilizzato, per ogni recensione, viene assegnato -1 se la recensione viene interpretata come negativa, 0 come neutrale, 1 come positiva.

Uno sguardo generale

Apps Top vs Apps Flop

Dopo esserci fatti un'idea generale, abbiamo deciso di selezionare le migliori e le peggiori 5 App a livello di Sentiment delle recensioni per valutarne le differenze.

Apps Top

Le migliori 5 App:

- Cookbook Recipes
- Goldstar: Live Event Tickets
- Hipmunk Hotels & Flights
- Sports News Digest
- C++ Tutorials

Apps Flop

Le peggiori 5 App:

- Guns of Glory
- Anthem BC Anywhere
- All Social Networks
- AirAsia
- Be A Legend: Soccer

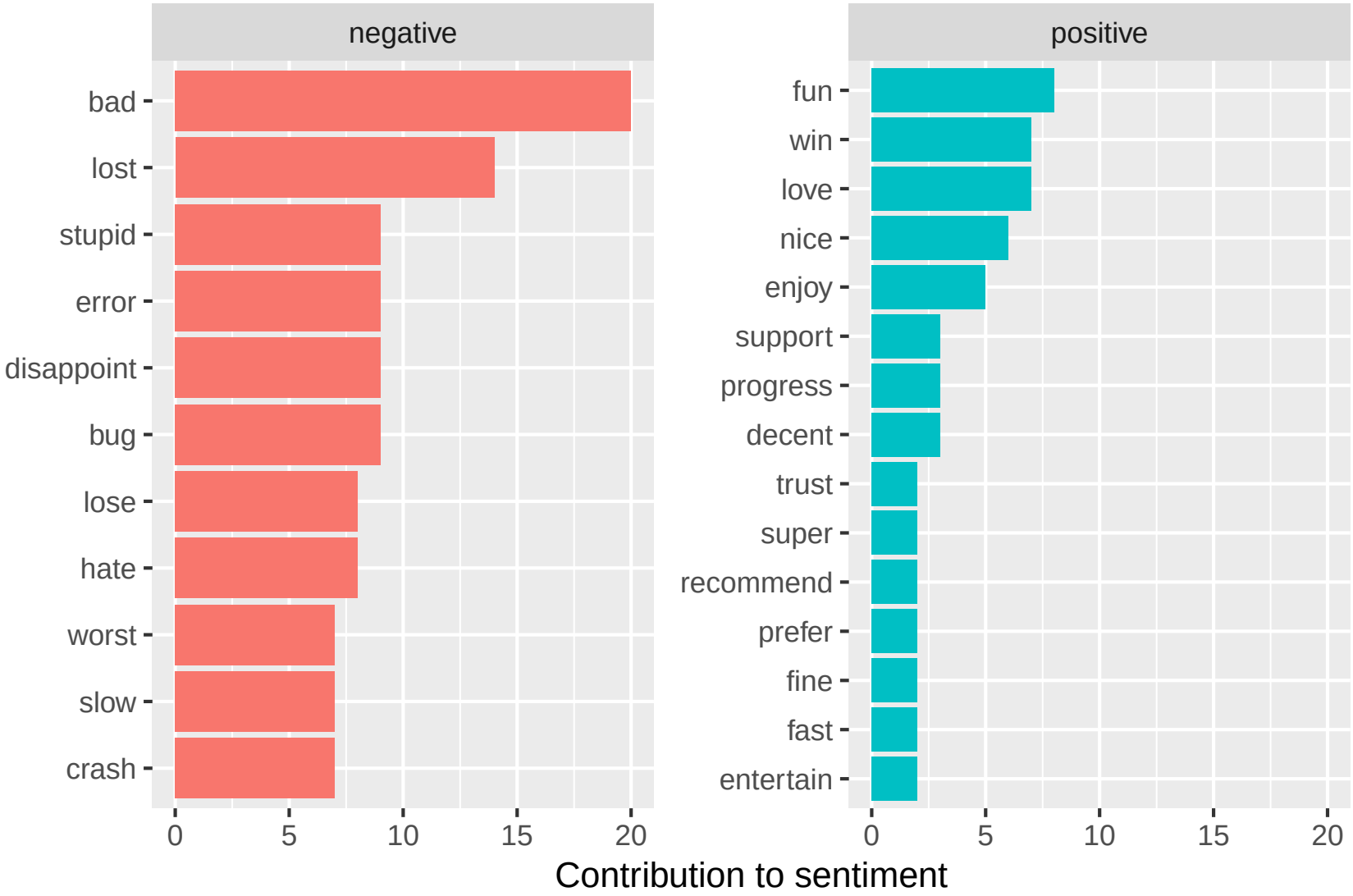
Apps Top


useless
pleas
suck
dead
crash
hard
spoil
bombard
bad
addict
stuck
stupid
love
hate
enjoy
nice
honest
cool
superb
recommend
afford
lover
excel
fast
honor
fun
wonder
glad
good
super
splendid

Apps Flop


joke
addict
annoy
worst
fall
rumor
broken
fist
bother
stuck
wrong
plot
fail
stupid
useless
disgust
hate
bleed
greed
retard
loss
fake
difficult
rubbish
grind
crash
disappoint
frozen
foolish
poor
glitch
lag
damn
slow
disturb
complaint
lie
kill
suck
delay
bug
lost
crap
bore
pleas
froze
crush
lose
error
attack
die
scam
sad
hang
enjoy
bad
nice
dumb
correct
decent
fun
win
wise
cool
entertain
gain
trust
fast
support
recommend
heal
love
favour
fine
wonder
skill
cheer
gem
prefer
progress
super
impress
golden
proper
comfort
defeat
faster
gold
smooth

Apps Flop


negative
bad
lost
stupid
error
disappoint
bug
lose
hate
worst
slow
crash
0
5
10
15
20
positive
fun
win
love
nice
enjoy
support
progress
decent
trust
super
recommend
prefer
fine
fast
entertain
0
5
10
15
20
Contribution to sentiment

Conclusioni analisi

Dal nostro punto di vista, un'azienda interessata a sviluppare un nuovo prodotto da rendere disponibile su Play Store, dovrebbe focalizzarsi sulla produzione di una applicazione

- Gratuita.
- Appartente alla categoria Game
- Ben sviluppata (anche a discapito della memoria occupata sui dispositivi).
- Ottimizzata per il maggior numero possibile di versioni Android.
- Con un numero di bug ridotti al minimo possibile già al momento del lancio.
- Più fluida possibile per quanto elaborata.

Grazie per l'attenzione