Piccardi Giovanni, matricola 817712

Lucini Paioni Andrea, matricola 826578

Izzo Massimiliano, matricola 833221

Data mining

MOBILE APP STORE

Dataset

Il dataset “App Store mobile” (https://www.kaggle.com/ramamet4/app-store-apple-data-set-10k-apps) contiene 7197 osservazioni relative ad applicazioni per smartphone presenti all’interno dell’App Store di Apple.

Variabili

Di seguito vengono presentate le 17 variabili contenute nel dataset:

Nome	**Descrizione**
X	Numero progressivo dell’app
id	Codice identificativo app
track_name	Nome dell'app
size_bytes	Dimensione (in byte)
currency	Valuta di acquisto
price	Prezzo
rating_count_tot	Numero di valutazioni degli utenti (per tutte le versioni)
rating_count_ver	Numero di valutazioni degli utenti (per la versione corrente)
user_rating	Valore medio della valutazione degli utenti (per tutte le versioni)
user_rating_ver	Valore medio della valutazione degli utenti (per la versione corrente)
ver	Codice della versione più recente
cont_rating	Età utenti consigliata
prime_genre	Categoria (ambito applicativo)
sup_devices.num	Numero di dispositivi supportati
ipadSc_urls.num	Numero di screenshot mostrati per la visualizzazione
lang.num	Numero di lingue supportate
vpp_lic	Licenza basata su dispositivo Vpp abilitata

Obiettivo

Lo scopo dello studio consiste nella messa a punto di un classificatore che sia capace di predire la valutazione degli utenti sulla base delle caratteristiche di un'applicazione.

Si sceglie dunque *user_rating* come variabile target; di seguito le sue caratteristiche.

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.000	3.500	4.000	3.527	4.500	5.000

Preprocessing

Inizialmente si rimuovono dal dataset le variabili ritenute superflue per l'analisi (*X, id, currency, rating count ver, user rating ver, ver, vpp_lic*) e le osservazioni relative alle app che presentano un numero di recensioni inferiori a 50 o zero lingue supportate.

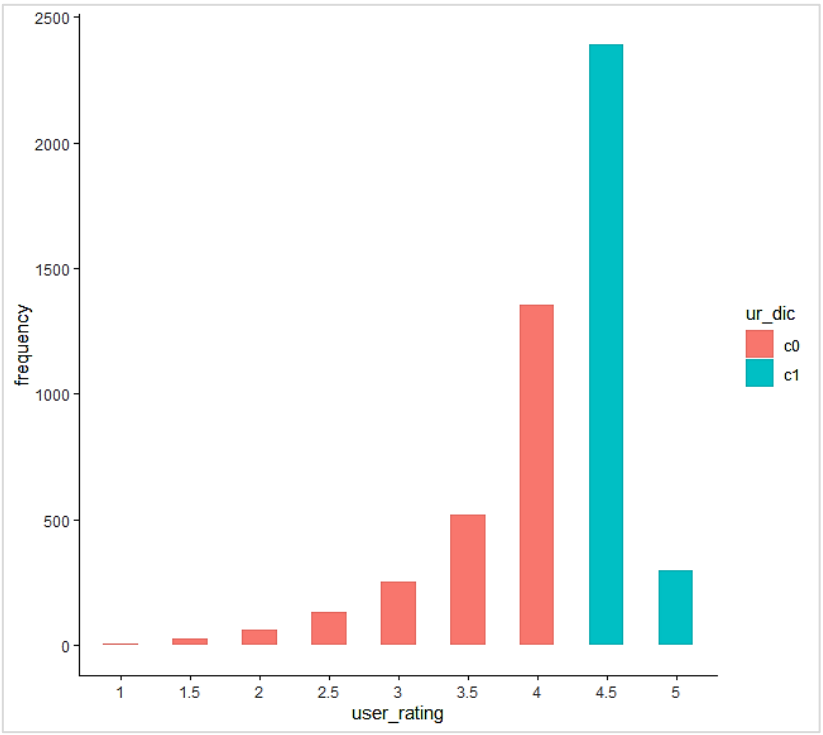
Rimangono quindi 5042 osservazioni e 10 variabili.

Si procede con la ricodifica della variabile d'interesse (*user_rating)* in una variabile dicotoma, nel seguente modo:

0 se *user_rating* <= 4 (punteggio basso), 2350 applicazioni

1 se *user_rating* > 4 (punteggio alto), 2692 applicazioni

Nella figura seguente si può osservare la distribuzione di frequenze corrispondente alla variabile *user_rating* (in rosso i punteggi considerati bassi, in azzurro quelli alti).



Successivamente si divide il dataset in due parti, in maniera randomica:

- Dataset di score (10% del dataset iniziale, 504 osservazioni e 9 variabili, poiché non viene considerata la variabile target)

- Il restante 90% si divide a sua volta in due parti:

    1) Dataset di training (70%, 3178 osservazioni e 10 variabili)

    2) Dataset di test (30%, 1360 osservazioni e 10 variabili)

Type variabili

```
$ track_name      : chr  "PAC-MAN Premium" "Evernote - stay organized" "WeatherBug - Local Weather, Radar,
Maps, Alerts" "eBay: Best App to Buy, Sell, Save! Online Shopping" ...
$ size_bytes      : num  1.01e+08 1.59e+08 1.01e+08 1.29e+08 9.28e+07 ...
$ price           : num  3.99 0 0 0 0 0.99 0 0 9.99 3.99 ...
$ rating_count_tot: num  21292 161065 188583 262241 985920 ...
$ user_rating     : num  4 4 3.5 4 4.5 4 4 4 4.5 4 ...
$ cont_rating     : Factor w/ 4 levels "12+","17+","4+",..: 3 3 3 1 3 3 3 1 3 3 ...
$ prime_genre     : Factor w/ 23 levels "Book","Business",..: 8 16 23 18 17 8 6 12 22 8 ...
$ sup_devices.num : num  38 37 37 37 37 47 37 37 37 38 ...
$ ipadSc_urls.num : num  5 5 5 5 5 5 0 4 5 0 ...
$ lang.num        : num  10 23 3 9 45 1 19 1 1 10 ...
```

Preprocessing generale sul dataset di training

Missing values

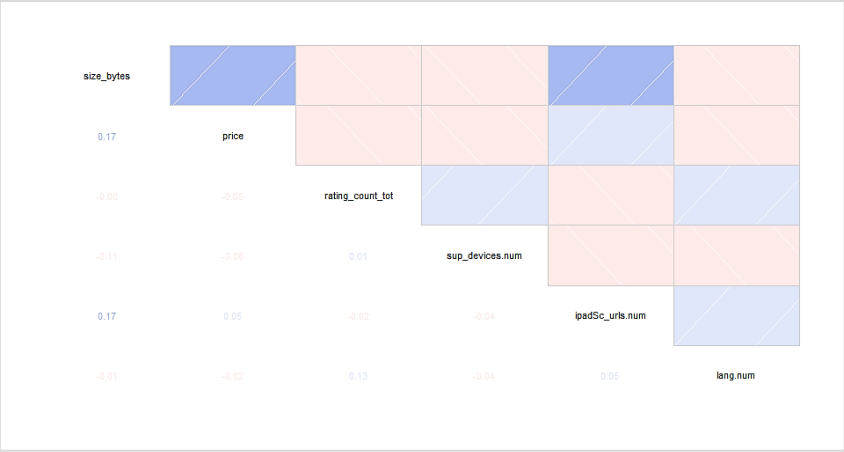
Si verifica l'eventuale presenza di valori mancanti all'interno del dataset:

track_name	size_bytes	price	rating_count_tot
0	0	0	0
user_rating	**cont_rating**	**prime_genre**	**sup_devices.num**
0	0	0	0
ipadSc_urls.num	**lang.num**		
0	0		

Non vengono rilevati "missing values".

Collinearità

Non è presente collinearità importante tra le variabili, come si evince anche dalla seguente matrice di correlazioni tra covariate numeriche (massima correlazione in modulo pari a 0.17).



Per quanto riguarda le due variabili di tipo factor, la loro indipendenza viene confermata tramite il test Chi-Quadro, di cui si riportano nel seguito i risultati:

	X1	Row	Column	Chi.Square	df	p.value	n	u1	u2	nMinu1u2	Chi.Square.norm
1	1	cont_rating	prime_genre	1024.924	66	0	5042	3	22	15126	0.06775908

Near Zero Variance

Non è presente, come si può osservare dalla tabella seguente.

	freqRatio	percentUnique	zeroVar	nzv
track_name	1.000000	100.0000000	FALSE	FALSE
size_bytes	1.000000	99.4021397	FALSE	FALSE
price	5.862295	0.8495909	FALSE	FALSE
rating_count_tot	1.076923	69.0685966	FALSE	FALSE
user_rating	1.765808	0.2831970	FALSE	FALSE
cont_rating	3.628731	0.1258653	FALSE	FALSE
prime_genre	7.329218	0.7237256	FALSE	FALSE
sup_devices.num	1.909206	0.5663940	FALSE	FALSE
ipadSc_urls.num	4.794702	0.1887980	FALSE	FALSE
lang.num	9.233918	1.6991819	FALSE	FALSE

Creazione dei dataset

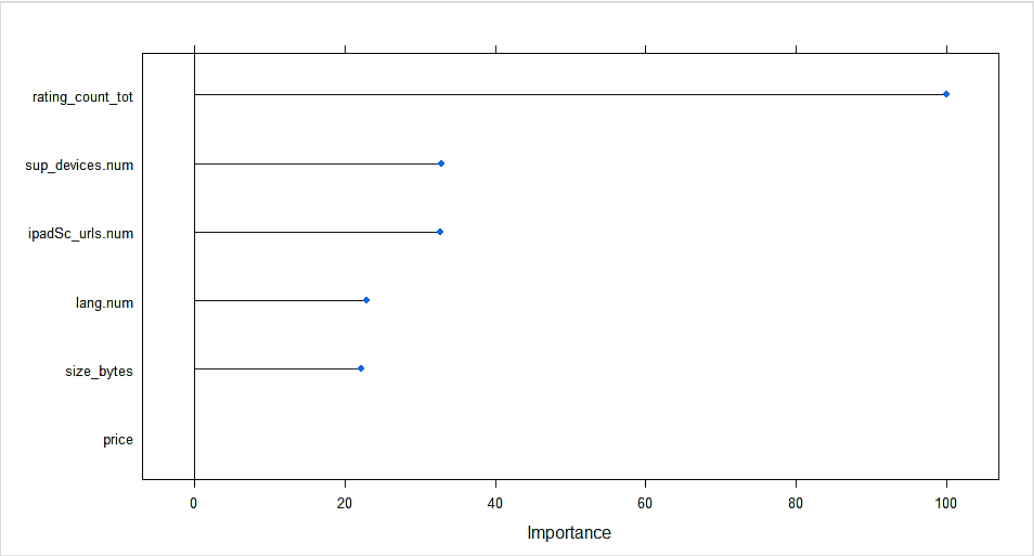
Vengono dunque creati molteplici dataset (tutti a partire dal dataset Train) per le analisi e le considerazioni sui modelli che verranno svolte successivamente:

1) *Train*: dataset originale.

2) *Std*: dataset in cui sono state standardizzate le variabili quantitative.

3) *Norm*: dataset in cui sono state normalizzate le variabili quantitative.

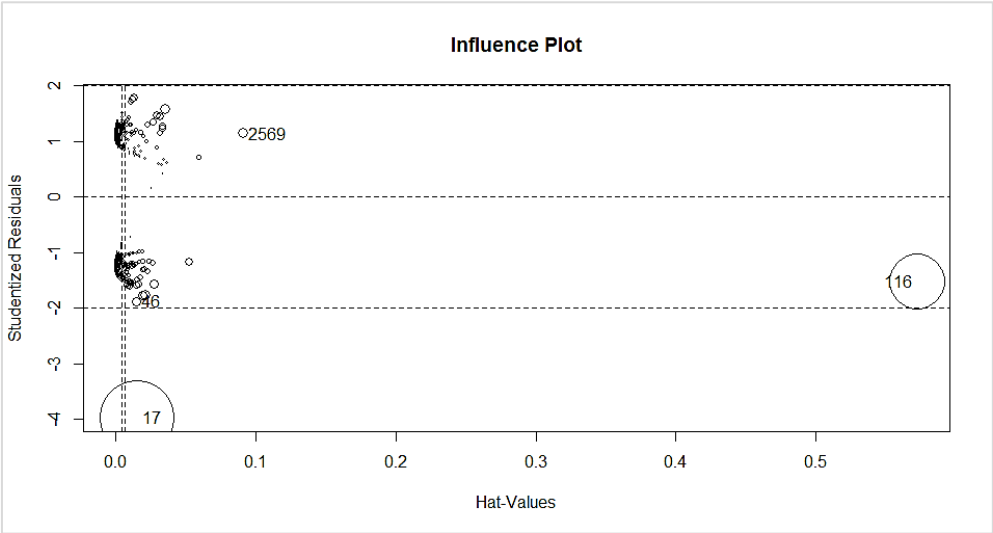
4) *Train + var imp*: dataset con solo le variabili più importanti (visualizzabili nel grafico seguente, individuate tramite Random Forest). Fissando un livello di cutoff pari al 10%, viene eliminata la variabile *price*.



5) *Std + var imp*: dataset con variabili quantitative standardizzate, con solo le variabili più importanti.

6) *Norm + var imp*: dataset con variabili quantitative normalizzate, con solo le variabili più importanti.

7) *Dataset senza outliers*: ottenuto eliminando, tramite il modello logistico, i valori che presentano Distanza di Cook > 4/(n-k-1). Si eliminano 92 osservazioni. (Nel grafico è possibile osservare i punti con Distanza di Cook elevata, con il diametro del cerchio più grande).



Creazione dei modelli

Viene visualizzata una tabella teorica che mostra la possibilità o meno di applicare un determinato modello ad uno dei dataset creati in precedenza.

	Train	**Std**	**Norm**	**Train var imp**	**Std var imp**	**Norm var imp**	**Senza outliers**
Logistico	NO	NO	NO	SÌ	SÌ	SÌ	SÌ
Naive Bayes	SÌ	SÌ	SÌ	SÌ	SÌ	SÌ	SÌ
Albero	SÌ	SÌ	SÌ	SÌ	SÌ	SÌ	SÌ
Random Forest	SÌ	SÌ	SÌ	SÌ	SÌ	SÌ	SÌ
Neural Network	NO	NO	SÌ	NO	NO	SÌ	NO
Gradient Boosting	SÌ	SÌ	SÌ	SÌ	SÌ	SÌ	SÌ
KNN	NO	SÌ	NO	NO	SÌ	NO	SÌ
Lasso	SÌ	SÌ	SÌ	NO	NO	NO	NO

Valutazione dei modelli

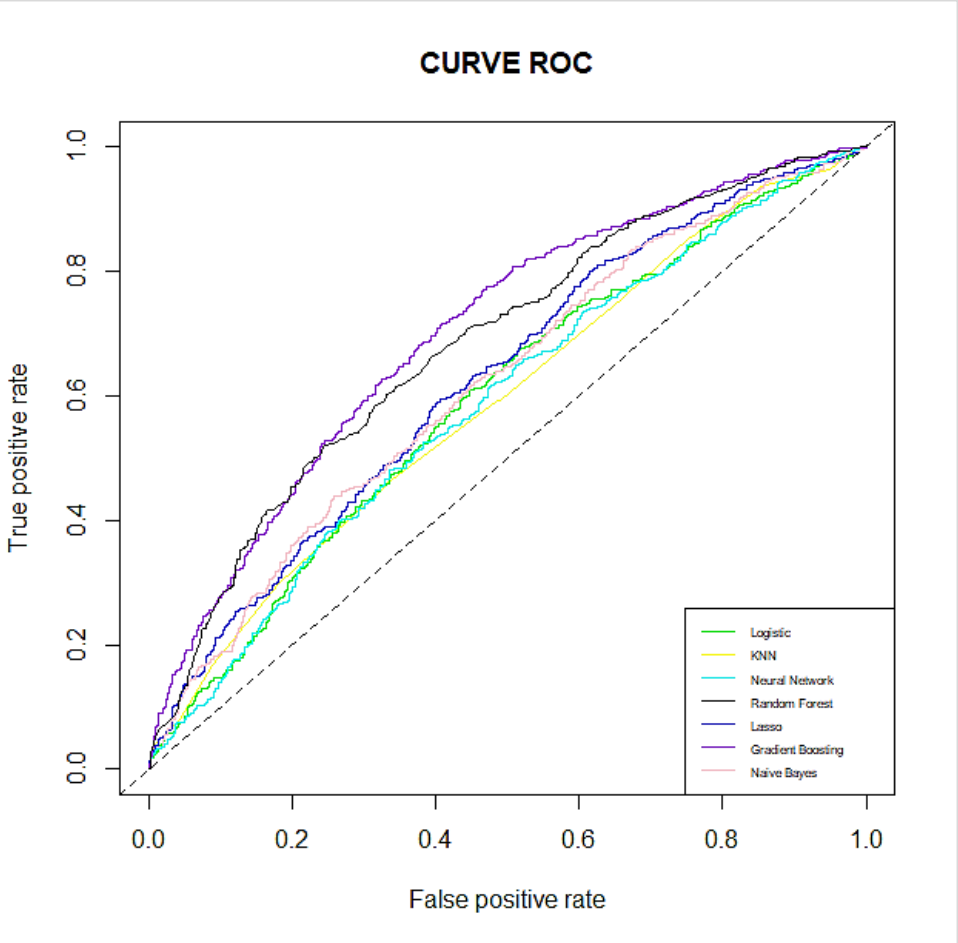
La seguente tabella riepiloga i risultati di ciascun modello stimato, laddove è possibile, ai diversi dataset. Si riporta il valore dell'area sottostante alla curva ROC e si evidenzia il più elevato per ciascun classificatore.

	Train	**Std**	**Norm**	**Train var imp**	**Std var imp**	**Norm var imp**	**Senza outliers**
Logistico				0.6001	0.5990	0.6004	0.6278
Naive Bayes	0.6103	0.5890	0.5855	0.5951	0.5985	0.5931	0.6038
Albero	0.5507	0.5569	0.5577	0.5283	0.5634	0.5492	0.5739
Random Forest	0.7039	0.6731	0.6672	0.6659	0.6576	0.6663	0.6824
Neural Network			0.5965			0.6014	
Gradient Boosting	0.7052	0.6879	0.6838	0.6810	0.6826	0.6784	0.6949
KNN		0.6122			0.6026		0.6152
Lasso	0.6409	0.6007	0.6008				

Scelta del modello

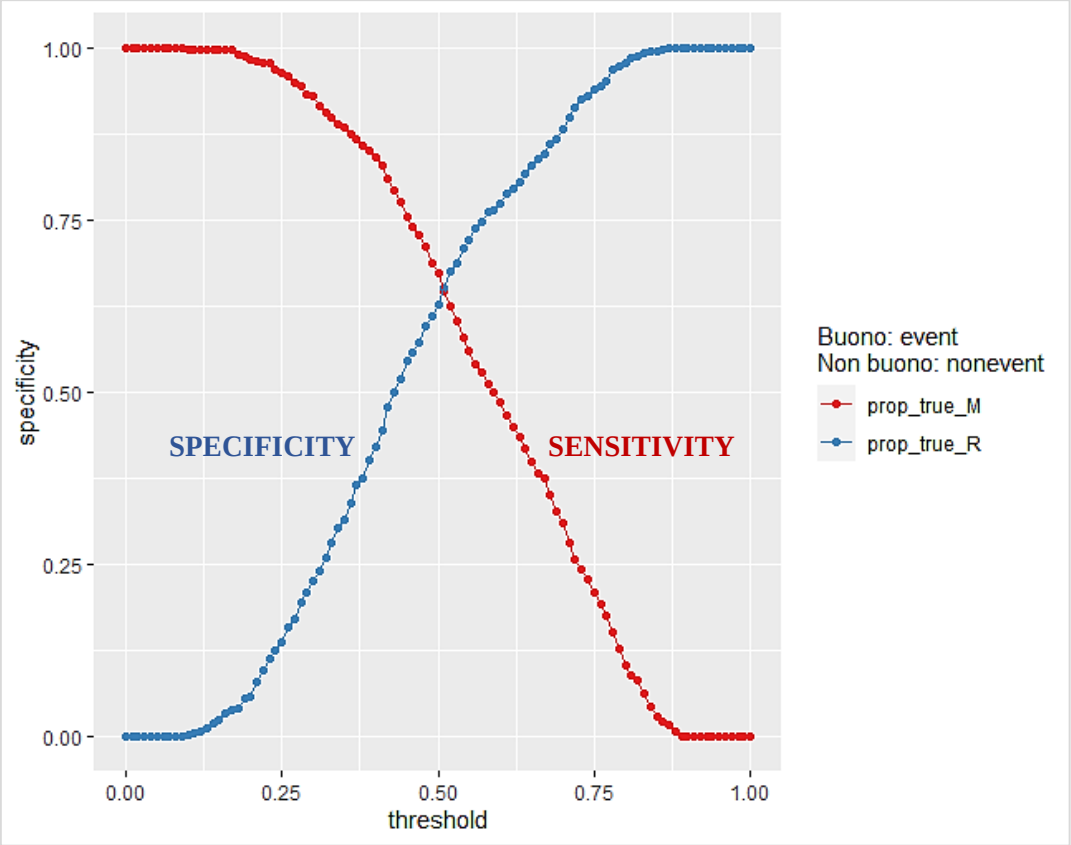
I risultati migliori, in termini di curva ROC, si riscontrano per i classificatori Gradient Boosting e Random Forest.

Come si intuiva dalla precedente tabella, e confermato dal seguente grafico che raffigura le curve ROC dei classificatori, la scelta del modello finale per la previsione di nuovi dati ricade sul Gradient Boosting (curva viola nella figura).



Soglia

Il grafico seguente mostra simultaneamente l'andamento di specificità e sensitività.



Il criterio di scelta della soglia corrispondente all'incrocio delle curve suggerisce un livello di soglia pari a circa 0.51, al quale però corrispondono sensitività e specificità piuttosto basse, poco sopra il valore 0.6.

Al fine di classificare a priori in maniera più efficace le app destinate ad avere punteggio alto e di conseguenza successo, è preferibile scegliere una soglia che garantisca una più elevata sensitivity.

Scegliamo dunque 0.4 come valore di soglia, a cui corrispondono i seguenti valori di sensitività, specificità e accuratezza:

Sensitivity	Specificity	Accuracy
0.8402	0.4196	0.6441

A livello di accuracy, tuttavia, il modello non è particolarmente performante.

Matrice di confusione

Visualizziamo ora la matrice di confusione relativa all'applicazione del modello scelto sul dataset Test (c0 = app con punteggio basso, c1= app con punteggio alto).

	predicted default		
actual default	c0	c1	Row Total
c0	266	368	634
	0.196	0.271	
c1	116	610	726
	0.085	0.449	
Column Total	382	978	1360

Come si nota dalla matrice, su 1360 app, 978 (71,9%) sono state classificate come app "buone" e 382 (28.1%) come app "scadenti".

È possibile osservare che delle 726 app con punteggio alto nella realtà, il modello ne ha classificate ben 610 correttamente e 116 invece come app dal punteggio basso.

Analogamente si può notare, però, come delle 634 app "scadenti", 368 sono state classificate in maniera errata, ossia come app considerate "buone".

Score

Con lo scopo di testare questo classificatore facciamo uso del dataset Score creato in precedenza (10% del dataset iniziale) e proviamo a predire il valore della variabile target *user_rating*.

Si ottiene il seguente risultato:

App con punteggio basso = 149 (29,6%)
App con punteggio alto = 355 (70,4%)