Machine Learning

Anno accademico 2022/2023

A. Davitti (857132), N. Mauri (856896), M. Rivolta (856863), A. Suardi (846627)

Indice

1 Introduzione: target e covariate

Le osservazioni utilizzate nel modello rappresentano le inserzioni presenti su Airbnb nella città di Amsterdam il 7 e 8 dicembre 2018.
L'obiettivo è quello di classificare le inserzioni che non hanno un voto complessivo delle recensioni, utilizzando le informazioni fornite dall'host della struttura. Tale classificazione permetterebbe alla piattaforma di segnalare ad un potenziale cliente le inserzioni che, nonostante siano ancora senza recensioni, possano essere considerate ottime.
Per la costruzione del modello sono state considerate le seguenti 40 covariate, di cui 20 quantitative *(Q)*, 13 dummy *(D)* e 7 factor con più di due livelli *(F)*.

- **host_is_superhost** *(D)*: host è/non è superhost
- **host_response_rate** *(Q)*: tasso di risposta dell'host ai messaggi
- **host_response_time** *(F)*: tempo di risposta dell'host ai messaggi
- **host_acceptace_rate** *(Q)*: tasso di accettazione delle richieste di prenotazione
- **host_neighbourhood** *(F)*: quartiere di residenza dell'host
- **host_listings_count** *(Q)*: numero di inserzioni totali dell'host presenti su Airbnb
- **host_has_profile_pic** *(D)*: host ha/non ha foto profilo
- **host_identity_verified** *(D)*: host ha/non ha identità verificata
- **neighbourhood** *(F)*: quartiere indicato dall'host nell'inserzione
- **neighbourhood_cleansed** *(F)*: quartiere identificato tramite le coordinate della posizione geografica
- **latitude** *(Q)*: latitudine
- **longitude** *(Q)*: longitudine
- **property_type** *(F)*: tipo di struttura
- **room_type** *(F)*: tipo di alloggio
- **accommodates** *(Q)*: capienza massima
- **bathrooms** *(Q)*: numero di bagni
- **bedrooms** *(Q)*: numero di stanze da letto
- **beds** *(Q)*: numero di letti
- **bed_type** *(F)*: tipo di letto
- **square_feet** *(Q)*: superficie in piedi quadri della struttura
- **security_deposit** *(Q)*: costo cauzione
- **cleaning_fee** *(Q)*: spese di pulizia
- **guests_included** *(Q)*: numero massimo di ospiti inclusi nel prezzo
- **extra_people** *(Q)*: numero massimo di ospiti extra

- **minimum_nights** *(D)*: numero minimo di notti richieste agli ospiti per soggiornare
- **maximum_nights** *(Q)*: numero massimo di notti che gli ospiti possono prenotare
- **has_availability** *(D)*: disponibile/non disponibile nel momento dello scraping dal sito
- **availability_30** *(D)*: numero di giorni di disponibilità nel successivo mese
- **availability_60** *(D)*: numero di giorni di disponibilità nei successivi due mesi
- **availability_90** *(D)*: numero di giorni di disponibilità nei successivi tre mesi
- **availability_365** *(D)*: numero di giorni di disponibilità nel successivo anno
- **price** *(Q)*: prezzo giornaliero
- **instant_bookable** *(D)*: la struttura può/non può essere prenotata automaticamente
- **cancellation_policy** *(F)*: politica di cancellazione della prenotazione
- **require_guest_profile_picture** *(D)*: necessità/non necessità di avere foto profilo per prenotare
- **require_guest_phone_verification** *(D)*: necessità/non necessità di avere numero di telefono verificato per prenotare
- **calculated_host_listings_coun** *(Q)*: numero di inserzioni totali dell'host presenti su Airbnb ad Amsterdam nel momento dello scraping
- **host_for_date** *(Q)*: numero di giorni passati dalla pubblicazione dell'inserzione
- **host_verifications_num** *(Q)*: numero di contatti social dell'host
- **amenities_num** *(Q)*: numero di servizi disponibili

Le ultime cinque variabili esplicative sono state da noi costruite a partire da altri dati resi disponibili da Airbnb. Infine, sono state escluse dal dataset di partenza le variabili contenenti i voti delle recensioni relative ad alcune caratteristiche specifiche della struttura poichè è stato scelto di utilizzare solo le informazioni non fornite da altri clienti.

La variabile target scelta è ottenuta binarizzando il voto complessivo delle recensioni (review_score_rating) secondo la seguente regola (scelta avvenuta data la forte asimmetria negativa della variabile dipendente, come mostrato in *Fig.1*):

- $y = 1$ se $review\_score\_rating = 100$
- $y = 0$ se $review\_score\_rating < 100$

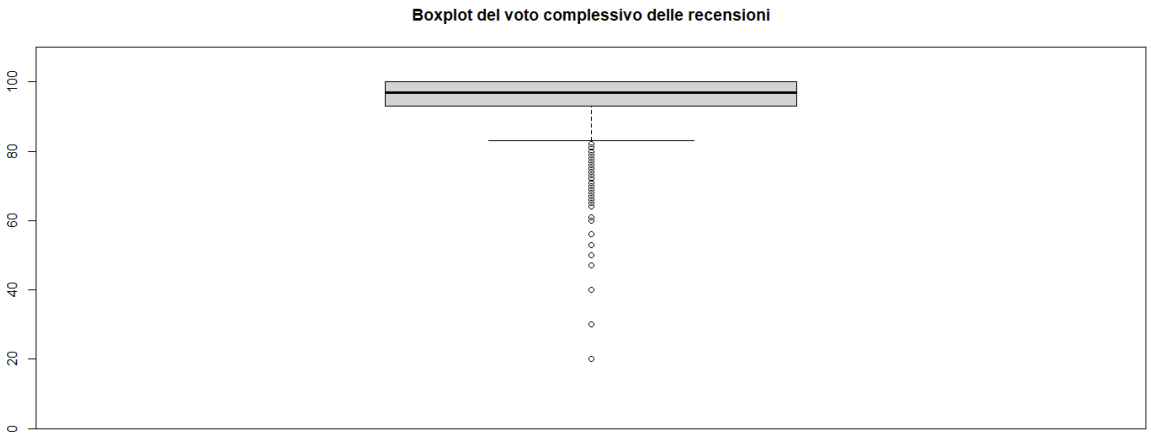

Fig.1 - Boxplot del voto complessivo delle recensioni

Ci si vuole quindi concentrare sulle inserzioni che hanno un voto complessivo pari al massimo possible (100/100).
Utilizzando questa regola si ottiene un dataset bilanciato (27.3% classificato come y=1 e 72.7% come y=0), che quindi non ha bisogno di essere corretto.

2 Gestione dei valori mancanti

La situazione iniziale dei dati mancanti è rappresentata in *Fig.2*.

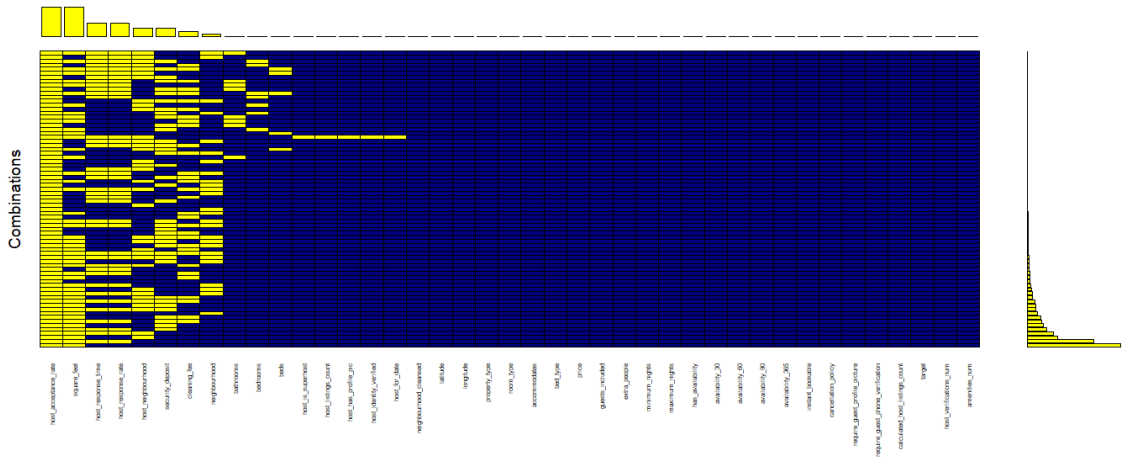

Fig.2 - Profili dei dati mancanti

La gestione dei missing values è stata svolta nei seguenti due passaggi:

1. eliminazione delle variabili che presentano una percentuale di dati mancanti superiore al 20% delle osservazioni totali;
2. imputazione dei valori mancanti rimanenti utilizzando la metodologia *cart* nella procedura mice, ipotizzando che si tratti di Missing At Random *(MAR)*.

3 Step 1

Si scelgono 11 diverse tipologie di modelli da costruire per l'obiettivo classificativo affrontato, effettuando per ogni modello sviluppato tutte le operazioni di pre-processing necessarie.
Per il modello logistico, ad esempio, vengono svolti tutti i passaggi che possano renderlo robusto (prerequisito necessario per poter essere un buon classificatore) e le procedure di best practice per migliorarlo ulteriormente.
Per evitare problemi di overfitting si utilizzano le seguenti tre strategie, variandole in base alla tipologia del modello considerato:

1. *Metodo Holdout*: data l'elevata numerosità di osservazioni (20030), si decide di dividerle in un dataset di training e in uno di validation; il primo, composto dal 70% delle osservazioni (14021), viene utilizzato usato per costuire i modelli; il secondo (6009 osservazioni), invece, funge per il controllo della possibile presenza di overfitting nei modelli.

2. *Model selection* (5 tipologie di modelli): si effettuano le seguenti cinque strategie per scegliere le covariate da usare nei modelli:

   (a) Random Forest: usare solo le covariate con un importanza superiore al 10% in una random forest costruita con tutte le covariate;

   (b) Logistico: usare solo le covariate significative in un modello logistico;

   (c) Componenti Principali: usare le componenti principali estratte da tutte le covariate;

   (d) R. Forest e C. Principali: usare le componenti principali estratte dalle covariate con un importanza superiore al 10% in una random forest costruita con tutte le covariate;

   (e) C. Principali e R. Forest: usare solo le componenti principali con un importanza superiore al 10% in una random forest costruita con tutte le componenti principali estratte da tutte le covariate.

3. *Tuning cross-validato* (6 tipologie di modelli): i modelli che hanno dei parametri che regolano la loro complessità vengono tunati cercando di massimizzare l'AUC cross-validato utilizzando 10 folds (*Fig.3*). Si considera questa metrica di valutazione della performance classificativa poichè è indipendente dalla soglia scelta in seguito.

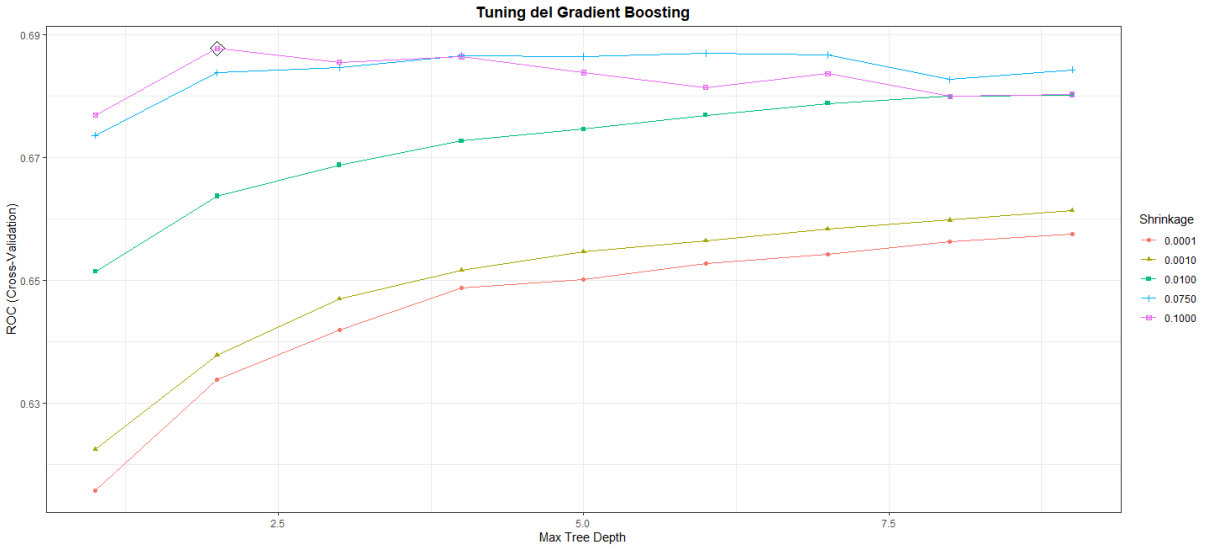

Fig.3 - Tuning del Gradient Boosting

Combinando le diverse strategie vengono costruite le seugeuenti tabelle, una relativa ai modelli senza tuning (*Tab.1*) e una con i modelli tunati (*Tab.2*).

Modello	ID	Preprocessing	Model Selection	Best Practice
Logistico	1	Separation Collinearità Near Zero Var.	R. Forest	
	2			Trasf. covariate (1)
	3			Oss. influenti (2)
	4			(1) - (2)
Naïve bayes		Separation Collinearità Near Zero Var.		
LDA	1	Collinearità	Logistico	
	2		C. Principali (3)	
	3		R. Forest (4)	
	4		(3) - (4)	
	5		(4) - (3)	
QDA	1	Collinearità	Logistico	
	2		C. Principali (3)	
	3		R. Forest (4)	
	4		(3) - (4)	
	5		(4) - (3)	
Bagging				

Tab.1 - Modelli senza tuning

Modello	ID	Preprocessing	Model Selection	P. Tuning	AUC-cv
KNN	1	Scale	Logistico	k = 13	0.565
	2		C. Principali (3)	k = 11	0.576
	3		R. Forest (4)	k = 13	0.578
	4		(3) - (4)	k = 13	0.575
	5		(4) - (3)	k = 13	0.567
Tree				cp = 0.0015	0.611
G. Boost				int. depth = 4 shrinkage = 0.1	0.687
R. Forest				m = 15	0.675
PLS		Std.		ncomp = 3	0.62
MLP		Collinearità Near Zero Var. Normalizzazione	R. Forest	size = 4 decay = 0.01	0.658

Tab.2 - Modelli con tuning

3.1 Overfitting

Viene calcolato per tutti i modelli l'AUC sul dataset di validation e si controlla che l'AUC di training non diminuisca più del 10%. Si osserva che 7 modelli overfittano e quindi si decide di non considerarli più per la scelta del modello migliore (*Tab.3*).

Modello	ID	AUC Training	AUC Validation	Differenza Percentuale
Logistico	1	0.616	0.62	-0.743
	2	0.645	0.652	-1.168
	3	0.621	0.63	-1.494
	4	0.639	0.649	-1.585
Naïve bayes		0.616	0.634	-2.872
LDA	1	0.614	0.618	-0.689
	2	0.617	0.628	-1.718
	3	0.596	0.605	-1.426
	4	0.626	0.63	-0.556
	5	0.591	0.603	-1.946
QDA	1	0.607	0.6	1.449
	2	0.612	0.611	0.317
	3	0.598	0.592	0.916
	4	0.618	0.612	0.958
	5	0.593	0.589	0.697
Bagging		1	0.651	34.947
KNN	1	0.711	0.575	19.088
	2	0.72	0.591	17.822
	3	0.72	0.571	27.77
	4	0.719	0.498	30.676
	5	0.715	0.571	20.043
Tree		0.612	0.615	-0.592
G. Boost		0.767	0.7	8.718
R. Forest		1	0.684	31.6
PLS		0.624	0.632	-1.243
MLP		0.684	0.681	0.366

Tab.3 - Controllo overfitting

4 Step 2

Per scegliere il modello migliore tra i 19 che non presentano overfitting, si confrontano le curve ROC calcolate sul dataset di validation.
I modelli che hanno una curva maggiore (superiore) delle altre sono MLP e Gradient Boosting, in particolare quest'ultimo sembra avere la curva ROC massima tra tutti i modelli considerati (*Fig.4*).

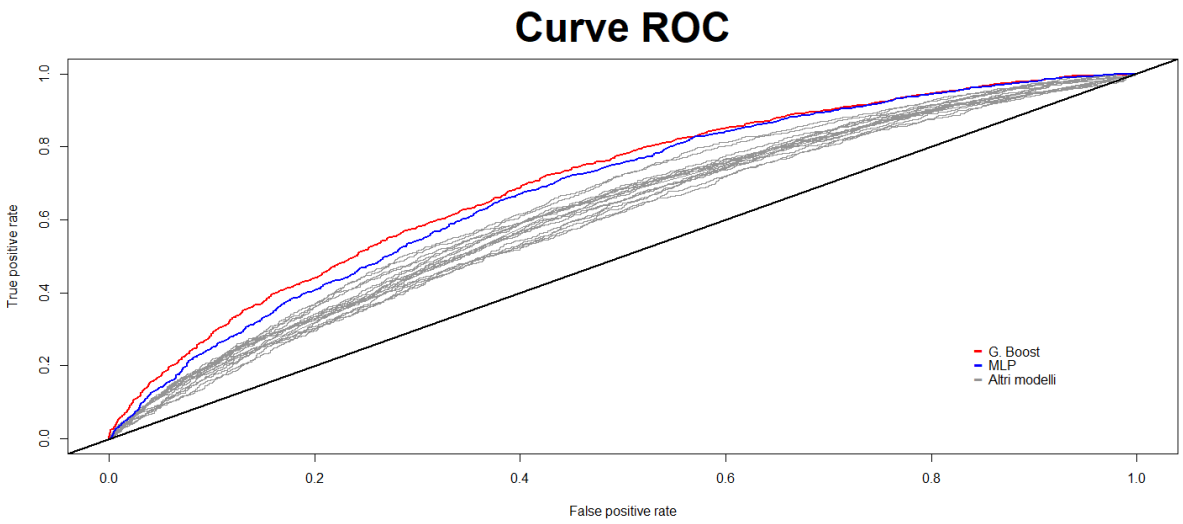

Fig.4 - Curve ROC dei 19 modelli concorrenti

Per avere un'ulteriore certezza, vengono calcolate le curve LIFT per i due modelli migliori (*Fig.5 e Fig.6*). Da questi grafici emerge che il migliore è il Gradient Boosting.

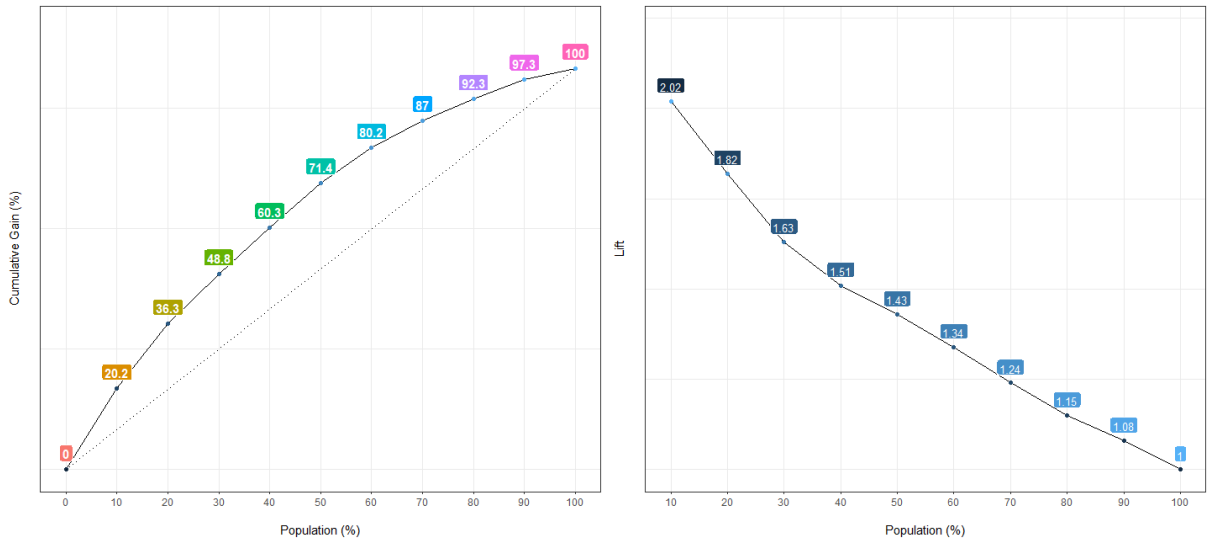

Fig.5 - Curva della percentuale di risposte catturate cumulata (sinistra) e curva lift (destra) del G. Boosting

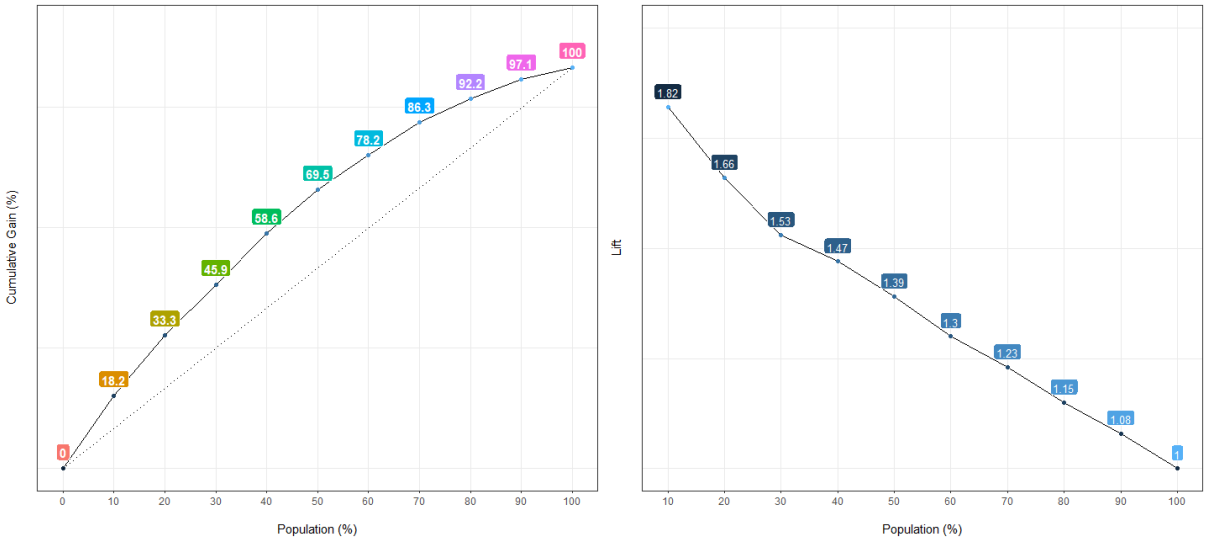

Fig.6 - Curva della percentuale cumulata di risposte catturate (sinistra) e curva lift (destra) del MLP

5 Step 3

Per scegliere la soglia classificativa si calcolano diverse metriche di valutazione del modello al variare del cut-off (*Fig.7*).

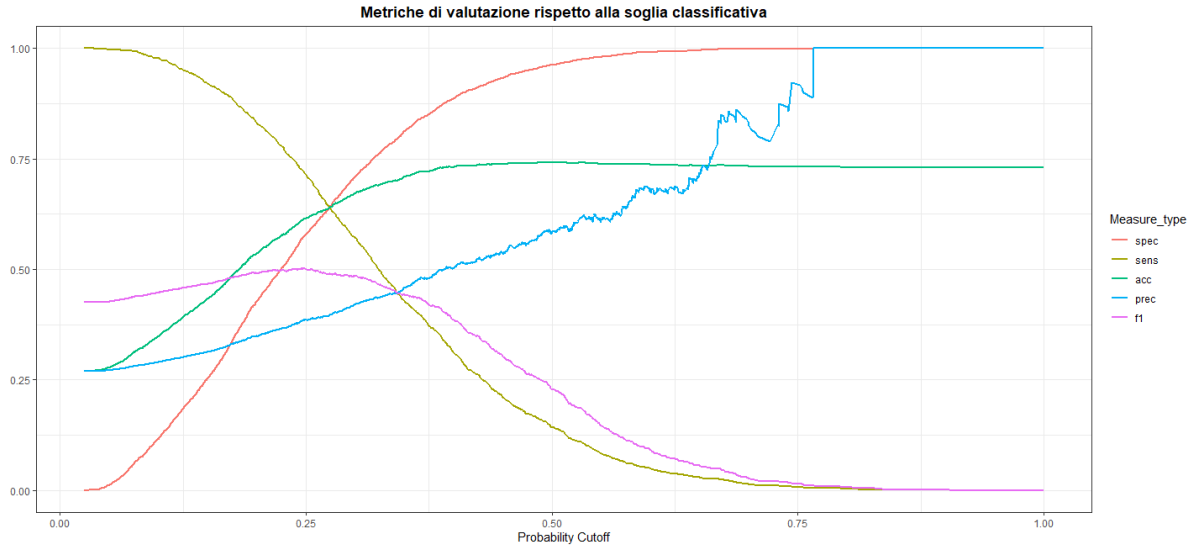

Fig.7 - Principali metriche di valutazione del modello Gradient Boosting al variare della soglia classificativa

Si sceglie come metrica di interesse per il nostro obiettivo classificativo la precision, perchè si vogliono massimizzare i casi True Positive e al contempo minimizzare i False Positive. Tale scelta rappresenta una tutela verso il cliente, evitando che possa scgliere un'inserzione da noi segnalata come ottima (y=1) quando invece non lo è.
La soglia classificativa che viene scelta corrisponde al valore massimo di precision ed è pari 0.766.

La confusion matrix e le principali metriche di valutazione della bontà classificativa del modello relative a questa soglia e al dataset di Validation sono presentate nelle *Tab.4* e *Tab.5*.

Osservato \ Previsto	y=1	y=0
y=1	8	1399
y=0	0	3809

Tab.4 - Confusion Matrix del dataset di Validation relativa alla soglia scelta per il modello G. Boost

Specificity	Sensitivity	Accuracy	Precision	F1
1	0.006	0.732	1	0.011

Tab.5 - Metriche di valutazione del modello G. Boosting relative alla soglia classificativa scelta per il dataset di Validation

Le performance del modello rispecchiano la scelta di massimizzare la precision, ma indicano una scarsa propensione a classificare come y=1 molte inserzioni; questo dipende dalla scelta iniziale della regola per binarizzare il target. Probabilmente il modello potrebbe aumentare il numero di True Positive nel caso si scegliesse una soglia inferiore a 100/100 come voto.

6 Step 4

Il modello classificativo finale viene testato su alcune delle inserzioni che inizialmente erano state escluse in quanto non recensite (*Tab.6*). In particolare, vengono scelte le osservazioni che non presentano dati mancanti in nessuna delle covariate inserite nel modello (1548 osservazioni totali).

y=1	y=0
49	1499

Tab.6 - Classificazione delle osservazioni del dataset di Scoring

7 Interpretazione modello

Il modello Gradient Boosting non fornisce strumenti interpretativi accurati per comprendere meglio il suo algoritmo classificativo. Si costruiscono quindi i Partial Dependence Profile per le covariate presenti nel modello, per vedere come varia la probabilità prevista media in funzione di una variabile indipendente alla volta (*Fig.8*).

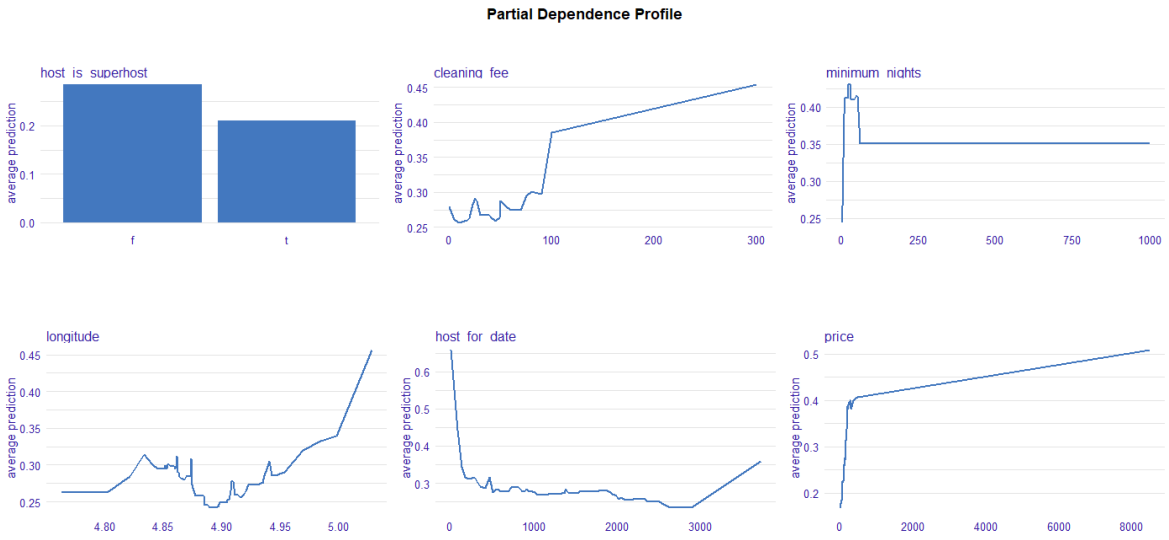

Fig.8 - Partial Dependence Profile di alcune covariate usate nel modello G. Boost

Si creano inoltre i breakdown plot di due osservazioni del dataset di Scoring che vengono classificate in modo diverso in base ai differenti valori osservati nelle covariate (*Fig.9*).

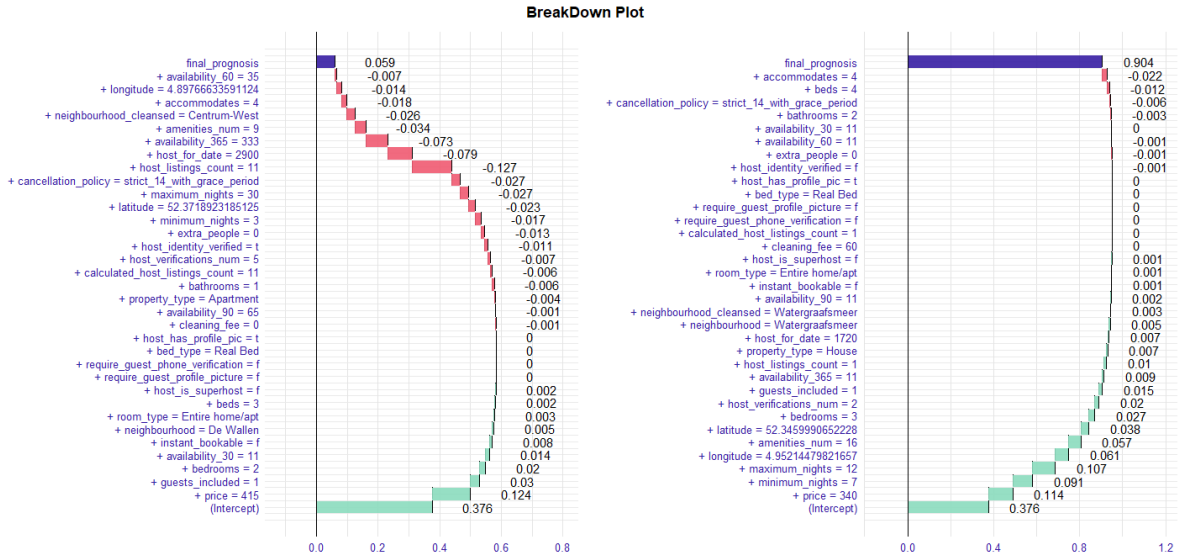

Fig.9 - Breakdown plot di due osservazioni del dataset di Scoring classificate come y=0 (a sinistra) e y=1 (a destra)