POLITECNICO DI TORINO

Corso di Laurea Magistrale
Ingegneria Gestionale

Tesi di Laurea Magistrale

AI, Data Science e Modelli Econometrici applicati alle recensioni di Airbnb: analisi dell'impatto della gentilezza dell'host sulla sua performance

Relatore

Laura Rondi

Correlatrice

Laura Abrardi

Candidato

Giacobbe Edoardo

Anno Accademico 2019/2020

Indice

1. Introduzione

In poco più di 10 anni dalla sua fondazione la piattaforma Airbnb ha completamente rivoluzionato il mondo del turismo. Tramite una profonda innovazione del modello di business degli *Short Term Rent* ha permesso l'ingresso nel mercato di nuove tipologie di attori che, tramite il servizio della piattaforma, hanno avuto la possibilità di ottenere una rendita accessoria affittando una o più camere inutilizzate della propria abitazione (o addirittura l'intero alloggio).
Le grandi possibilità di guadagno lato offerente e la grande convenienza e semplicità per i clienti hanno portato ad una vera e propria esplosione della popolarità della piattaforma, che ha superato, alla fine del 2019, i 7 milioni di alloggi dislocati in più di centomila città nel mondo.
Se la grande abbondanza di offerta ha portato un forte beneficio per i clienti, allo stesso tempo ha innescato un atteggiamento di fortissima competizione tra gli alloggi, che ha reso indispensabile per gli host cercare sempre nuovi modi per distinguersi dalla massa.

Lo scopo di questo studio è quello di verificare se la "gentilezza" del padrone di casa sia un elemento di diversificazione che possa garantirgli un vantaggio competitivo. Nell'ambito di questa tesi, si utilizzerà il termine "gentilezza" per riferirsi alle caratteristiche del comportamento dell'host in tutte le fasi della transazione, a partire dalla prenotazione, in fase di check-in fino al momento di conclusione del soggiorno.
Cercare di quantificare la "gentilezza" di un host è un'operazione molto critica, in quanto si tratta di un aspetto implicito alla persona, che si manifesta solamente durante il contatto con soggetti terzi, in questo caso i clienti.

Dato che studi precedenti hanno mostrato alcuni limiti degli indicatori forniti direttamente da Airbnb, in quanto affetti da un forte *bias* inflativo, per estrarre informazioni sul comportamento del padrone di casa si è deciso di analizzare direttamente le recensioni pubblicate dai clienti, che sono l'unico vero momento di espressione di un giudizio completamente in forma libera. Le recensioni sono rese disponibili direttamente da Airbnb mediante la piattaforma InsideAirbnb, che presenta database aggiornati periodicamente per tutte le principali città. Per continuità con studi precedenti, ci si è concentrati sulla città di Barcellona, che garantisce un turismo molto diversificato, e sono state selezionate tutte le recensioni relative a soggiorni negli ultimi 18 mesi, da Gennaio 2019 a Aprile 2020.

Una volta isolato dalle recensioni complete il contenuto relativo al comportamento del padrone di casa, si è sviluppato un modello ad hoc di Machine Learning che permettesse di classificare le recensioni in: negative, neutre, positive e super. L'aggregazione delle recensioni appartenenti ad uno stesso alloggio ha permesso di elaborare un indicatore di gentilezza per ogni host. La misura così generata, denominata "rank" è stata poi inserita in modelli econometrici di tipo regressivo.

La prima relazione che si è voluta indagare è quella tra il prezzo dell'alloggio ed il livello di gentilezza del suo host, al fine di verificare se ad un miglioramento del comportamento del padrone di casa fosse associato un premio di prezzo. Si è inoltre effettuata una seconda modellizzazione per cercare di verificare se la gentilezza avesse un impatto positivo non solo sul prezzo dell'alloggio, ma anche sulla domanda che questo è in grado di catturare.

2. Analisi della letteratura

Lo scopo di questo capitolo è inquadrare nel suo contesto la domanda di ricerca che verrà sviluppata in questa tesi e fornire un punto di vista su come studi precedenti hanno provato ad approcciare tale problema, oppure problemi che si possano ritenere correlati.
Semplificando il più possibile ciò che si andrà a fare nei capitoli successivi, questa tesi si propone di andare ad analizzare quali siano i fattori di successo della piattaforma Airbnb.
Per fare ciò si vuole creare un modello di *Pricing* per andare a verificare come fattori diversi influenzino la formazione del prezzo di equilibrio.
Il primo paragrafo di questo capitolo mostrerà una panoramica sui modelli di *Pricing* più all'avanguardia che sono stati proposti per la piattaforma in analisi.
Un'analisi preliminare di questo tipo risulta molto utile per fornire linee guida per la modellizzazione che verrà presentata nel capitolo 6.

Per calare il punto di vista più nello specifico, l'obiettivo non è quello di esplorare la totalità dei fattori di successo della piattaforma, ma piuttosto capire se il comportamento del padrone di casa e la qualità della sua interazione con il cliente possano essere un fattore determinante rispetto al meccanismo di *Pricing*. Prima di ricercare di formalizzare una relazione tra il prezzo e ciò che in tutto lo studio verrà chiamata "gentilezza" del padrone di casa, è importante capire come avvenga questa interazione tra i due attori. Nel secondo paragrafo verranno presentati diversi lavori che si sono occupati dello studio delle modalità di interazione *guest-host* e della loro importanza nella formazione di un giudizio sul soggiorno.

La peculiarità di questo lavoro è la volontà di esprimere un giudizio sull'interazione tra padrone di casa e cliente a partire dal contenuto della recensione pubblicata dal cliente stesso.
È utile in questo senso capire come funzioni il meccanismo *double blinded* che è stato adottato dalla piattaforma.
Per concludere, verranno mostrati alcuni studi che spiegano le potenzialità di un sistema di recensioni per giudicare prodotti fisici e beni esperienza.

2.1. Modelli di Pricing

Il prezzo di un alloggio è sicuramente un elemento critico nel mondo dell'ospitalità, in quanto influenza la tipologia di clientela a cui si rivolge l'offerta, e di conseguenza i profitti del business.
Non stupisce per questi motivi che in letteratura si possano trovare svariati studi che si sono occupati della modellizzazione del meccanismo di *pricing* nel mondo degli hotel.
Uno dei lavori pioneristici è stato pubblicato da Adrian Bull nel 1994: lavorando su un campione di motel in Ballina, Australia, l'economista e giornalista ha indagato l'influenza della posizione dei motel sulla determinazione del loro prezzo.

Altri studi successivi, tra cui si vuole citare "*Pricing determinants in the hotel industry: Quantile regression analysis"* di Hung e al, hanno permesso di individuare tra i fattori di successo:

- Brand
- Numero di stelle
- Posizione
- Longevità
- Numero di stanze
- Tipologia dei servizi offerti

Dato che l'esplosione della *Sharing Economy* è un fenomeno piuttosto recente, così come il successo della piattaforma Airbnb, il numero di studi a disposizione non è così esteso come nel caso del mondo dell'hotelleria.
Uno dei primi lavori è di Ikkala e Lampinen, che hanno elaborato una procedura qualitativa per analizzare il ruolo della reputazione dei padroni di casa nella determinazione del prezzo. I due autori hanno condotto delle interviste semi-strutturate ad host attivi sulla piattaforma e una delle tendenze che sono emerse è che i padroni di casa si sentono legittimati ad alzare il prezzo del soggiorno man mano che accumulano recensioni positive (questo fenomeno viene chiamato "*Reputational Capital*").

In coerenza con quanto analizzato in modo qualitativo da Ikkala, nel 2015 Gutt and Hermann hanno condotto uno studio su 14000 alloggi stanza locati a New York.
In particolare, gli autori hanno verificato che gli alloggi che nel corso del periodo di osservazione hanno sbloccato la possibilità di rendere visibile il proprio rating (la *policy* della piattaforma in quel periodo prevedeva di oscurare il rating degli host che non avessero superato una certa soglia di recensioni) sono riusciti ad imporre un premio di prezzo nei periodi successivi di quasi 3 $ per notte.

L'importanza della posizione degli alloggi è stata esplorata da Yang Li e al. nel 2016 e successivamente da Zangh e al. nella pubblicazione: "*Key Factors affecting the price of Airbnb Listings: A Geographically Weighted Approach".*
Nel suo studio Yang Li ha individuato una relazione positiva tra il prezzo e la vicinanza rispetto al punto strategico più vicino (normalmente il centro della città per le mete più turistiche).
Il secondo lavoro ha invece seguito un approccio radicalmente diverso: dopo aver individuato alcuni fattori che potrebbero influenzare il prezzo degli alloggi nella città metropolitana di Nashville in Tennessee, gli autori hanno mostrato come i vari fattori (tra cui Numero di recensioni, Distanza dalla principale autostrada, distanza dal centro, longevità dell'alloggio) abbiano un impatto nettamente diverso a seconda dell'area in cui si trova l'alloggio.

Sempre nel 2017 Wang e Nicolau hanno elaborato un modello di *Pricing* attraverso un modello di *regressione OLS* che include 31 variabili, raggruppate in 5 categorie:

- Site and Property Attributes
- Amenities and Services
- Rental rules
- Host Attributes
- Online reviews ratings

Le due categorie più interessanti sono sicuramente le ultime 2.

La categoria *Host Attributes* cerca di cogliere aspetti che inquadrino la figura del padrone di casa e esprimano informazioni sulla sua reputazione e affidabilità: in questa categoria compaiono la variabile che conta il numero di alloggi posseduti dall'host e 3 *dummy* che spiegano se l'host abbia verificato la propria identità, pubblicato un'immagine personale, e sia stato certificato *superhost.*
L'aspetto più interessante è l'inclusione nel modello di un indicatore di rating dell'alloggio e del suo host. Tale indicatore è il risultato dell'elaborazione delle opinioni dei clienti, che sono state raccolte attraverso un meccanismo di *feedback* costituito dall'attribuzione di un numero di "*stelle"* da 1 a 5.

Con il diffondersi della popolarità della piattaforma, negli anni successivi al 2015 si è potuto osservare un'evoluzione del modello di business di Airbnb: hanno iniziato ad aderire al servizio non solo host amatori che ricercano una rendita accessoria, ma anche delle società ben strutturate che possiedono una moltitudine di immobili.
Un gruppo di ricercatori del Dipartimento di Economia della città di Tessaly, in Grecia, ha analizzato più di 3500 appartamenti attivi sulla piattaforma ad Atene alla ricerca di differenze strutturali tra i prezzi di appartamenti gestiti da host professionali e host amatori. I risultati della modellizzazione mostrano che in media gli host professionali richiedono un prezzo superiore di 13 euro/notte rispetto agli amatori. Parte di tale gap è spiegato dal fatto che tendenzialmente gli host professionali gestiscono appartamenti più appetibili, ma una parte cospicua è da imputarsi ad un effetto rispettabilità legato alla più elevata organizzazione del business.

La rapida evoluzione delle esigenze dei consumatori ha portato ad un netto incremento dell'attenzione intorno alla figura dell'host, il cui comportamento in tutte le fasi della transazione è stato oggetto di studi sempre più dettagliati negli ultimi anni.
Lo scopo di questi studi è duplice, da un lato può essere di aiuto ai padroni di casi, che ottengono preziosi consigli su quali aspetti debbano curare maggiormente per migliorare la propria performance.
Dall'altro lato, consegnano strumenti molto potenti nelle mani degli utenti, che sono sempre più in grado di valutare a priori le diverse alternative e prendere decisioni migliori in relazione al tipo di esperienza di cui sono alla ricerca.
Ad esempio, la pubblicazione "*Accomodation prices on Airbnb, effects of host experience and market demand*" mostra che esiste una relazione molto forte tra il prezzo di un alloggio e l'esperienza del suo host, quantificata in base al numero di anni sulla piattaforma e al numero di persone ospitate.
La modellizzazione *edonistica* presentata in questo studio spiega come i clienti ripongano grandi aspettative in host con molti anni di esperienza alle spalle, dai quali si aspettano un servizio impeccabile, per il quale sono disposti a pagare in media un prezzo più alto.

2.2. Interazione umana nelle transazioni

Attribuire il successo di Airbnb, così come della *Sharing Economy* in generale, esclusivamente alla sua convenienza è una visione della realtà piuttosto limitata.
Se effettivamente i clienti scegliessero un alloggio sulla piattaforma rispetto ad una delle soluzioni alternative più tradizionali esclusivamente per la possibilità di risparmio, è ragionevole pensare che dopo un primo periodo di assestamento, i competitors avrebbero reagito riducendo le loro tariffe e, forti delle loro strutture tendenzialmente più appetibili (si

pensi ad esempio ad un maggior livello di privacy e alla maggior presenza di servizi) , avrebbero riconquistato la porzione di domanda che era stata catturata dal nuovo attore.

Diverse pubblicazioni hanno dimostrato che il punto di forza della piattaforma non risiede nei suoi prezzi, ma piuttosto nell'interazione umana con il padrone di casa.
Tra gli studi che si sono occupati di questo argomento Mingming Cheng e al. nel 2018 hanno indagato su quali fossero le fonti di soddisfazione per i clienti di Airbnb.
Analizzando le recensioni pubblicate dagli utenti e conducendo interviste semi-strutturate, l'autore spiega che i consumatori, al momento di valutare il loro soggiorno, raramente citano il prezzo e sono molto più interessati alla posizione dell'alloggio, ai suoi servizi, e soprattutto al loro rapporto con il padrone di casa.
Quest'ultimo aspetto è talmente importante che molti clienti, che si ritengono entusiasti nei confronti degli altri due aspetti chiave, non si possono ritenere soddisfatti della loro esperienza se non hanno la percezione di aver instaurato un rapporto umano con il padrone di casa.

Questa considerazione è strumentale per comprendere che la *Sharing Economy* ha contribuito a creare un nuovo paradigma di turismo, nel quale molti clienti, pur potendosi permettere di affittare un appartamento completo, scelgono volontariamente di condividere degli spazi con l'host per poter vivere un'esperienza che Sthapit e Jiménez-Barreto nel loro studio definiscono di "Comunione e di Divisione".

Gli studi che si occupano della dimensione sociale della Sharing Economy sono ormai svariati (di grande spessore quantitativo sono ad esempio i lavori di Lin e al. e quello di Sutherland e al. del 2019). Si tratta di un argomento molto popolare che in tempi recenti ha catturato l'attenzione non solo di economisti, ma anche di psicologi e esperti del comportamento, che stanno cercando di creare nuove procedure per cercare di analizzare e quantificare queste interazioni.

Uno degli studi più all'avanguardia dal punto di vista metodologico risale al 2017 ed è stato pubblicato da Alsudais e al.
Questo lavoro esplora il problema di quantificare le interazioni "offline" tra i due attori, che sono tendenzialmente uno scambio informativo molto difficile da trattare non lasciando alcuna traccia fisica o digitale.
La soluzione proposta consiste nell'indagare la recensione del cliente alla ricerca di considerazioni che rivelino qualche tipo di informazione sul rapporto avuto con l'host.
Per farlo, l'autore ha elaborato una procedura IT ad alta efficienza (superiore al 90%) che individua se in una recensione il guest include una menzione per l'host.
L'evidenza empirica ha mostrato che nell'80% dei casi il cliente rivela qualcosa sul padrone di casa. Questo risultato è molto importante perché ha aperto la strada per studi successivi (tra cui quello presentato in questa tesi) di inferire sul comportamento dell'host esclusivamente a partire dal giudizio pubblicato dal cliente.

Un ulteriore lavoro che vale la pena menzionare è "*Escaping loneliness through Airbnb host-guest interaction*" di Farmaki e al.
In questo studio gli autori mostrano una panoramica su come il senso di solitudine sia un sentimento sempre più radicato nella società, che non colpisce soltanto i cittadini più anziani ma individui di ogni età.

Questi, non riuscendo a instaurare dei rapporti sociali nei modi più convenzionali, sono sempre più alla ricerca di nuovi modi per interagire con altre persone, ed in generale di modi per "sentirsi parte di una rete coinvolgente di persone".
In uno studio precedente, Song e al. nel 2018 avevano spiegato come l'incremento a livello globale delle persone che soffrono di solitudine metta a dura prova i servizi sanitari e di assistenza sociale.
Lo studio mostra come il turismo attraverso le piattaforme come Airbnb possa contribuire ad alleviare questo problema sociale attraverso le interazioni tra ospiti e padroni di casa.
Grazie ad un sistema di interviste semi-strutturate con host e ospiti di Airbnb, gli autori hanno potuto verificare che una buona parte degli intervistati ha scelto questo tipo di turismo per "il desiderio di socializzazione e per la possibilità di interagire con persone provenienti da tutto il mondo". Molti addirittura affermano come la loro esperienza abbia portato ad un sollievo per il proprio disagio e parlano di Airbnb come un vero e proprio rimedio per la solitudine.
Nelle conclusioni gli autori suggeriscono che non sia remota la possibilità che nel futuro la piattaforma offra un servizio di abbinamento degli utenti che soffrono di solitudine agli host che sono più in grado di offrire loro l'esperienza di cui sono alla ricerca.

Se diversi studi affermano che la transazione *peer-to-peer* tipica della piattaforma sia in grado di generare grande valore per i clienti grazie alla potenzialità di creare un sensazione di "sentirsi a casa", come spiegato da Tussyadiah e Pesonen in una pubblicazione del 2015, è anche vero che il grande livello di incertezza legato a questo tipo di transazione può essere fonte di preoccupazione per entrambe le parti, che si trovano a dover condividere spazi personali con degli sconosciuti senza la sicurezza del supporto di un'entità commerciale come una struttura alberghiera tradizionale.
I riflessi di questo fenomeno sono stati indagati da H.Moon e al. nel paper "*Peer-to-peer interaction: Perspectives of Airbnb guests and hosts*".
Attraverso più di 500 interviste ad utenti della piattaforma, gli autori hanno cercato di indagare, sia dal punto di vista del guest ma anche del padrone di casa, quali fossero gli elementi che infondessero più preoccupazione negli attori e quali potessero essere delle misure per mitigare questo stato di disagio.
I risultati di questo studio sono particolarmente interessanti, perché mostrano che gli atteggiamenti di *self-disclousure* durante le fasi precedenti al soggiorno, e il meccanismo di *trust-building* dell'host durante il suo periodo di attività sulla piattaforma hanno un impatto eccezionale nel ridurre lo stato di apprensione delle due parti.
L'effetto virtuoso di tali meccanismi risulta addirittura più efficace dell'incontro "faccia a faccia" tra le parti.

2.3. Meccanismo di feedback tramite recensioni

2.3.1 Processo di valutazione Airbnb

Per capire l'importanza del sistema di recensioni, è indispensabile capire come funzioni il processo di valutazione.
Al momento della conclusione del soggiorno, il cliente ha a disposizione una finestra di 14 giorni per completare il processo di valutazione del proprio soggiorno.
La prima parte della valutazione consiste nell'attribuire da una 1 a 5 "stelle" ad ognuna delle 8 categorie predisposte da Airbnb.

Le diverse categorie cercano di cogliere al meglio tutti gli aspetti dell'esperienza, accettando i limiti intrinseci di un sistema numerico aggregato.

Le 8 categorie sono rispettivamente:

- Esperienza complessiva
- Pulizia
- Accuratezza
- Check-in
- Comunicazione
- Posizione
- Valore
- Servizi

Il Centro Assistenza di Airbnb fornisce delle linee guida per cercare di comprendere meglio cosa il cliente dovrebbe valutare in ciascuna delle categorie.

Categoria	Cosa i clienti dovrebbero valutare
Esperienza complessiva	Il soggiorno, nel suo complesso
Pulizia	Il livello di pulizia e ordine dell'alloggio al momento della consegna dello stesso
Accuratezza	Il livello di accuratezza con cui la pagina dell'annuncio (in particolare riferendosi a foto pubblicate e informazioni allegate) rappresentava lo stato attuale dell'alloggio
Check_in	Il livello di efficienza e professionalità mostrato durante il check-in
Comunicazione	L'efficacia della comunicazione con l'host prima e durante il soggiorno.
Posizione	Il quartiere in cui si trova l'alloggio, la facilità di accesso ai trasporti e la vicinanza rispetto al centro città, a centri commerciali e così via.
Valore	Il rapporto qualità prezzo dell'alloggio.
Servizi	I servizi disponibili e la corrispondenza tra gli stessi e l'elenco pubblicato nell'annuncio

La piattaforma consente inoltre di pubblicare una recensione in forma di testo libero.
In questa sezione i clienti sono liberi di esprimere un giudizio sulla loro esperienza, concentrandosi su qualunque aspetto preferiscano.
Gli unici due vincoli sono che il testo non superi le 1000 parole e che il contenuto non violi i termini sulle recensioni di Airbnb.

Dato che lo si ritiene un argomento molto rilevante, si presentano brevemente gli aspetti più salienti di tale regolamento, che è costituito da 3 sezioni:

1. Norme sul contenuto
2. Equità
3. Pertinenza

Per quanto riguarda le *Norme sul contenuto*, Airbnb individua una serie di contenuti tipo che non sono consentiti:

- contenuto creato esclusivamente a scopo pubblicitario o altro contenuto commerciale, inclusi loghi, link o nomi di società
- spam, contatti indesiderati o contenuti condivisi ripetutamente in modo fastidioso
- contenuto che approva o promuove attività illegali o dannose o che è sessualmente esplicito, violento, crudo, minaccioso o molesto
- contenuto discriminatorio
- tentativi di impersonare un altro individuo, account o entità, tra cui un rappresentante di Airbnb
- contenuto illegale o che viola i diritti di un'altra persona o entità, inclusi i diritti di proprietà intellettuale e i diritti alla privacy
- contenuto che include informazioni private o riservate di un'altra persona, sufficienti, ad esempio, a identificare la posizione di un alloggio

Le norme sull'*equità* sono atte a garantire che la recensione sia effettivamente prodotta da un cliente imparziale, in quanto questa è l'unica fattispecie che garantisce un contenuto informativo genuino, reale ed utile per clienti successivi. Per questo scopo, Airbnb non consente a individui o entità titolari o affiliati di un annuncio di pubblicare recensioni sulla loro stessa attività, né a individui che offrono annunci concorrenti di pubblicare recensioni sui loro competitor diretti.
Inoltre, è severamente vietato al padrone di casa incentivare recensioni positive e cercare di influenzarne il contenuto promettendo un compenso o minacciando di pubblicare una recensione negativa al cliente, che ne comprometterebbe il futuro uso della piattaforma.

Le *norme di pertinenza* sono atte a garantire che il contenuto della recensione sia pertinente ad Airbnb ed in particolare allo specifico soggiorno. La piattaforma in questo senso prescrive di evitare:

- commenti sulle opinioni sociali, politiche o religiose di una persona
- volgarità, insulti e supposizioni sul carattere o sulla personalità di un individuo
- contenuti che si riferiscono a circostanze completamente al di fuori del controllo della persona interessata dalla recensione
- contenuti su servizi non correlati a Airbnb (ad es. una compagnia aerea, un tragitto in *ride-sharing*, un ristorante, ecc.)
- commenti su prenotazioni, host o ospiti Airbnb precedenti o sul prodotto Airbnb non in riferimento all'annuncio

Una volta completato il processo di valutazione, il cliente ha ancora a disposizione 48 per modificare i propri giudizi.
Contestualmente alla finestra a disposizione del cliente, anche l'host ha a disposizione 14 giorni per valutare gli ospiti.

Airbnb ha predisposto un meccanismo di pubblicazione *double blinded* per tutelare entrambe le parti.
Il contenuto di ognuna delle due recensioni non è visibile fino a quando entrambe le parti non hanno pubblicato la loro valutazione (a meno che non si sia esaurita la finestra di tempo per l'inserimento).

In questo modo nessuno dei due attori può influenzare il comportamento dell'altra parte minacciando di aspettare di visionare il contenuto della recensione e pubblicando una recensione negativa come vendetta, nel caso la positività della recensione dell'altra parte non sia in linea con quanto auspicato.

2.3.2 L'importanza delle recensioni

I meccanismi di feedback tramite recensioni sono ormai una feature standard della maggior parte dei retailer online.
L'avvento di questa tecnologia risale ai primi anni 2000 in relazione alla vendita di prodotti fisici, ma oggi sono uno strumento importante anche per valutare servizi e addirittura beni esperienza.
Sono svariati gli studi che si sono occupati dell'importanza delle recensioni per la categoria di prodotti definiti *search goods.*
È ormai senza dubbio che queste siano uno strumento in grado di fornire grande valore sia per i consumatori, sia per la piattaforma di e-commerce.
Grazie alle recensioni degli altri utenti, i consumatori vengono agevolati nel loro processo decisionale perché queste rappresentano un giudizio sulle reali performance di tale prodotto, molto spesso difficilmente valutabili solo a partire dalle specifiche tecniche.

L'impatto delle recensioni non si limita solo al loro contenuto.
Kumar e Benbasat, infatti, nel 2006 hanno dimostrato che la presenza delle recensioni dei clienti su un sito (si pensi ad un player della grande distribuzione come Amazon.com) migliora la percezione dei clienti dell'utilità e della presenza sociale di tale sito web.
Le recensioni, infatti, hanno il potenziale per attirare le visite dei consumatori, aumentarne il tempo trascorso sul sito (ciò che nello studio viene chiamato "vischiosità") e creare un senso di comunità tra gli acquirenti.

Altri studi, tra i quali si vuole citare *"All Reviews Are Not Created Equal: The Disaggregate Impact of Reviews on Sales on Amazon.com"*di Chen e al. nel 2008 e quello di Clemons e al pubblicato nel 2006, hanno cercato di verificare se le recensioni dei clienti potessero avere un'influenza positiva sulle vendite. In particolare, Clemons e al. nello studio "*When Online Reviews Meet Hyperdifferentiation: A Study of the Craft Beer Industry*" hanno scoperto che valutazioni fortemente positive possono influenzare in modo molto convincente la crescita delle vendite di tale prodotto.

L'impatto potenziale delle recensioni è ancora superiore per i beni esperienza. Il concetto di *experience good* è stato coniato da Philip Nelson in contrasto con il concetto di *search good.* I beni esperienza sono quei prodotti/servizi le cui caratteristiche e qualità sono tali da essere difficilmente valutabili a priori. Le virtualmente infinite possibilità di espressione attraverso un testo libero aumentano a dismisura la capacità di "raccontare" un qualunque bene esperienza. Questo meccanismo è molto importante perché contribuisce così a ridurre le asimmetrie informative tra venditori e consumatori, i quali al crescere del numero di recensioni disponibili, si trovano sempre più agevolati nel processo di selezione.

La magnitudine di questo effetto è così grande da aver suscitato l'interesse dell'accademico giapponese Makoto Nakayama. In uno studio pubblicato nel 2019, l'economista ha potuto verificare che il diffondersi in modo sempre più capillare dei sistemi di feedback a recensioni ha innescato un fenomeno di trasformazione delle percezioni dei clienti.

Effettuando un confronto a distanza di quasi 10 anni tra svariate tipologie prodotti e servizi, è emerso che una grande quantità di prodotti che dieci anni fa erano tipicamente inclusi nella categoria degli *experience goods,* oggi sono "raccontabili" con una tale facilità da essere considerati alla stregua di un *search goods* qualsiasi.

3. Estrazione di una misura di gentilezza dell'host

Il punto di partenza del lavoro è il database "reviews" relativo alla città di Barcellona, consultabile tramite il portale InsideAirbnb.
AirBnb mette a disposizione dati aggiornati periodicamente, relativi a tutte le principali città.
Per continuità con studi precedenti, si è deciso di selezionare come città Barcellona e di utilizzare la versione del database più recente disponibile al momento dell'inizio del lavoro di tesi. I dati risultano aggiornati alla data 31 Aprile 2020.
Il database presenta, in forma tabellare, le recensioni relative ai soggiorni presso alloggi attivi sulla piattaforma Airbnb degli ultimi 10 anni.
Si tratta di una mole di dati impressionanti (circa 800 000 recensioni).
Siccome lo sforzo a livello computazionale del processamento dell'intero database sarebbe stato enorme, si è deciso di selezionare un subset dei dati che contiene le recensioni relative agli anni 2019 e la prima parte del 2020.
La scelta è stata motivata da una considerazione sul ciclo di obsolescenza dei dati: in un mondo in cui le esigenze dei consumatori sono in rapida e imprevedibile evoluzione, includere nell'analisi recensioni vecchie quasi un decennio avrebbe esposto al rischio di introdurre dei dati che non rispecchiano la realtà, che avrebbero potuto falsare i risultati di questa fase e, a cascata, di tutte quelle successive.
Il numero degli anni da includere è il risultato di un ragionamento di massa critica: le 287000 recensioni circa ottenute isolando gli ultimi 2 anni garantiscono ampiamente di ottenere risultati significativi.

Il database "reviews" è costituito da 5 colonne:

- Listing_id: è un codice che permette di identificare l'alloggio
- Id: è un codice numerico che permette di identificare in modo univoco la recensione
- Date: è la data in formato gg/mm/aaaa in cui il recensore ha inserito la recensione
- reviewer_id: codice che permette di identificare il cliente
- reviewer_name: è il nome (o i nomi nel caso di recensori corali) della persona che ha lasciato la recensione
- comments: stringa di testo contenente la recensione.

Per chiarezza espositiva, in Tabella1 si mostra una riga tipo del database.

Listing_Id	Id	date	reviewer_id	reviewer_name	comments
21974	152431809	16/05/2017	128118341	Anna	Great location!

Tabella 1: Riga tipo database

3.1 Pre-processamento delle recensioni

3.1.1 Formattazione del testo

La prima sfida è stata quella di standardizzare il più possibile il testo contenuto nelle recensioni.

Il *framework* predisposto da Airbnb per inserire una recensione è particolarmente aperto: l'utente può produrre un testo in forma libera, senza vincoli di lingua, formattazione, o lunghezza dell'elaborato.
Processare una stringa di caratteri di questo tipo risulta particolarmente difficile, specie per il fatto che la mole dei dati non permette di trattare in modo singolare tutti gli elementi problematici che si potrebbero verificare potenzialmente.
Per prima cosa, utilizzando dei pacchetti di funzioni pre-esistenti in *python*, il database è stato sottoposto a un ciclo di formattazione degli elementi terminatori di frase.
Questa operazione ha fatto si che tutte le frasi della recensione fossero delimitate da un singolo "punto", seguito da un singolo "spazio".
L'operazione è propedeutica alla manipolazione successiva, che utilizza uno strumento particolarmente instabile.

3.1.2 Gestione della lingua

L'algoritmo di classificazione degli host lavora utilizzando i vocaboli della lingua inglese.
In un primo momento si è valutata l'alternativa di isolare soltanto le recensioni scritte in tale lingua, ma questa semplificazione avrebbe eliminato una porzione consistente dei dati.
Questo non era preoccupante a livello di quantità di informazione da dare in pasto agli algoritmi, in quando la scarsità di dati poteva essere compensata includendo ulteriori anni di record.
Ciò che più preoccupava era l'effetto "cultura": soggetti appartenenti ad etnie diverse hanno modi diversi di interfacciarsi con le persone, e di conseguenza sensibilità diversa rispetto al comportamento dell'host.
Per questo motivo, limitarsi a considerarsi un'unica lingua avrebbe esposto al rischio di considerare come universali risultati che invece rispecchiano la visione di un numero ristretto di soggetti.
Per ovviare al problema della lingua si è deciso di tradurre le recensioni in lingua inglese.
Questo approccio è stato reso possibile dai servizi di traduzione di colossi come Microsoft e Google, che sfruttando meccanismi di Intelligenza Artificiale applicati alla NLP (*natural language processing*) hanno raggiunto livelli di accuratezza assolutamente inimmaginabili solo 5-10 anni fa.
Per motivazioni di reputazione si è deciso di utilizzare l'*API* di traduzione di Google chiamata *GoogleTrans API*.
Si tratta di un meccanismo di chiamata direttamente attraverso lo script *python* di un servizio di traduzione, che prende in input la lingua di origine, il testo contenuto in una stringa e lingua di destinazione e restituisce una stringa con il testo tradotto.
GoogleTrans è in grado di rilevare automaticamente la lingua di origine del testo, ma per ridurre l'incertezza nell'operazione (lanciare un'unica chiamata API per quasi 300000 cicli è una operazione piuttosto delicata, che ha causato più volte il *crash* dell'applicativo; ogni elemento di complessità che si riesce ad eliminare aumenta la probabilità che questa vada a buon fine), si è deciso di sfruttare *Detect*, una funzione inclusa nella liberia *Langdetect* di *python*, che sfrutta l'intelligenza artificiale per rilevare, con accuratezza in linea con lo stato dell'arte, la lingua in cui è stata generata una stringa.
L'output è stato salvato in una nuova colonna del database, in quanto si è valutato che potesse essere utile nel caso in cui, in un momento futuro, si decidesse di portare avanti un'analisi su come soggetti appartenenti a culture diverse esprimono giudizi.

In Tabella2 si mostra la distribuzione delle lingue delle recensioni, con la loro frequenza assoluta, percentuale e cumulata rispetto al totale delle recensioni.

Dai dati si può notare come eliminare tutte le recensioni non in lingua inglese avrebbe escluso circa 1/3 dei dati contenuti nel database, con conseguenze potenzialmente disastrose.

Lingua	Frequenza assoluta	Frequenza percentuale	Frequenza cumulata
Inglese	183276	67%	67%
Spagnolo	41576	15%	83%
Francese	22303	8%	91%
Italiano	6915	3%	93%
Tedesco	6172	2%	96%
Portoghese	4346	2%	97%
Altro	7372	3%	100%

Tabella 2: Distribuzione delle lingue nelle recensioni

Il 15% delle recensioni è scritto in lingua locale, ovvero lo spagnolo, e un ulteriore 15% è costituito da recensioni in francese, italiano, tedesco e portoghese.
L'occorrenza di lingue diverse dalle 6 sopra citate è pari al 3% del totale.

Monitorando l'esecuzione della chiamata *API*, durata più di 36 ore, ci si è reso conto che spesso la traduzione non funzionasse nonostante il servizio non presentasse messaggi di *warning*. L'errore si presentava con frequenza alta ma in modo causale, in quanto non si notava un trend sistematico per cui una determinata lingua non veniva mai tradotta.
Per ovviare a questo problema, è stato sviluppato uno *script* di controllo e correzione che lavora su una lingua per volta.
Per ogni lingua affetta da questo problema (per lo più spagnolo, francese e italiano) vengono innanzitutto selezionate tutte le recensioni in cui la lingua di origine corrisponde a tale lingua.
Fortunatamente, nei casi di mancata traduzione, il servizio restituisce una stringa identica a quella di origine, quindi al fine di individuare gli errori è bastato inserire un controllo di identicità tra stringa originale e tradotta.
A questo punto si lancia nuovamente il servizio di traduzione.
Il *loop* di controllo e traduzione è stato iterato, ottimizzandone a ogni ciclo i parametri, fino a quando il numero di "non tradotti" è sceso sotto all'1% dei dati totali.
L'analisi capillare dei casi residui ha mostrato che si trattava di recensioni molto brevi in lingua inglese, che quindi dovevano "*bypassare"* la chiamata al servizio, erroneamente classificate come lingue straniere.
Nonostante la fiducia nel servizio di Google, si è deciso di eliminare le recensioni la cui lingua di origine utilizza un alfabeto diverso da quello comune. In questo senso sono state ignorate recensioni in: giapponese, cinese, coreano, russo e greco.
Il numero di righe a disposizione a questo punto si è assestato intorno alle 272 000 unità.

3.1.3 Rilevamento della lunghezza della recensione

Utilizzando un *tool* di conteggio parole è stata aggiunta una colonna "num_parole" con il numero di parole contenute in ogni recensione.
Tabella3 riassume alcune statistiche descrittive per la variabile "num_parole".
Ci sono svariate recensioni che si assestano sotto le 13 parole (circa ¼ delle recensioni), mentre alcuni ospiti si impegnano in descrizioni dettagliate (elenchi puntati, pro/contro, etc).
Il massimo rilevato è di 1000 parole.

La lunghezza delle recensioni è stata utilizzata anche come strumento per eliminare le righe del record corrispondenti a recensioni vuote (eliminate in quanto non portatrici di informazione).

Media	41 parole
Deviazione standard	45 parole
Minimo	1 parola
Massimo	1000 parole
Primo quartile (25%)	13 parole
Secondo quartile (50%)	28 parole
Terzo quartile (75%)	53 parole

Tabella 3: Lunghezza delle recensioni

Uno spunto per un possibile studio futuro potrebbe essere cercare la presenza di relazioni sistematiche tra la positività della recensione e il numero di parole scritte: una recensione ottima è sistematicamente più lunga di una negativa oppure l'utente si impegna a creare un giudizio dettagliato anche nel caso il soggiorno non sia stato in linea con le sue aspettative?

3.1.4 Valutazione del sesso di chi scrive la recensione

La colonna "reviewer_name" presenta il nome della persona (o delle persone) che hanno scritto la recensione.
Sfruttando un *tool* di *Machine Learning* facilmente implementabile in *python* chiamato *GenderGuesser*, si è cercato di associare il sesso al nome di chi ha scritto la recensione.
Si tratta di un'operazione particolarmente delicata, che funziona particolarmente bene con nomi di matrice anglosassone, ma che perde efficacia con nomi di matrice latina o orientale.
Il metodo classifica ogni nome in: "maschio", "femmina", "sconosciuto", o "androgeno".
La categoria "androgeno" si riferisce a quei nomi che vengono utilizzati per entrambi i sessi (come Andrea) oppure al caso in cui in una recensione corale gli individui appartengano ad entrambi i sessi.
"Sconosciuto", invece, racchiude tutti quei nomi che non si è stato in grado di classificare.

Al fine di semplificare la classificazione si è deciso di attribuire i nomi "androgeni" alla categoria "femmina".
Il motivo principale è il seguente: l'indicatore potrebbe essere strumentale per verificare se una recensione scritta da una donna è strutturalmente diversa da quella di un uomo; di conseguenza, non è interessante tenere separata la classe "andy". Dovendo scegliere a quale delle due classi attribuire gli "androgeni", è stato considerato ragionevole che sia più probabile che in una coppia di sesso eterogeneo, sia la donna a scrivere la recensione.

Tabella4 presenta la frequenza assoluta delle classi "female", "male" e "unknown", la frequenza relativa rispetto al totale, e per le prime due classi, la composizione rispetto al totale dei nomi che sono stati riconosciuti, pari a 211028 recensioni.

Classe	Frequenza assoluta	Freq. Percentuale	Composizione %
Female	109486	43%	52%
Male	101542	40%	48%
Unknown	45538	17%	

Tabella 4: Sesso dei recensori

Dalla tabella si può notare come l'occorrenza di recensioni scritte da donne è tutto sommato simile alla frequenza di recensioni maschili. L'elevata presenza di sconosciuti riduce sensibilmente i dati a disposizione e costituisce un limite per analisi successive che volessero utilizzare questo indicatore.

È stato effettuato un tentativo di applicare la medesima procedura di riconoscimento del sesso anche sull'host. Questo dato sarebbe stato particolarmente interessante ai fini di verificare se ci sia un sesso che si comporta sistematicamente in modo più apprezzato dell'altro. Purtroppo, il tentativo ha restituito nel 99% dei casi la classe "unknown", confermando la perdita di efficacia del *tool* se ci si allontana dal vocabolario anglosassone (la maggior parte degli host hanno nomi di matrice latina).
Data la presenza di più di 20000 padroni di casa, una classificazione manuale sarebbe stata impossibile, e a malincuore si è deciso di trascurare questo tipo di analisi.

3.1.5 Verifica di segni di confidenza tra ospite e host

Per cercare di identificare segni di confidenza tra ospite e host si è creato un attributo binario "confidenza" che assume valore 1 se l'ospite nella recensione si riferisce all'host utilizzando il suo nome proprio.
Si tratta di un indicatore piuttosto sottile, che potrebbe anche non essere significativo, l'intuizione dietro a tale attributo è la seguente: nel contesto di soggiorni per lo più brevi e rivolti all'esplorazione della città, ricordare il nome dell'host potrebbe essere segnale di aver avuto un rapporto umano con quest'ultimo.
L'indicatore è stato creato con l'intenzione di analizzarne la potenziale correlazione con la positività della recensione: l'ospite ricorda il nome dell'host perché ha una buona considerazione dello stesso o perché l'applicazione tende a riportarne il nome in ogni fase?

Se a livello concettuale, la procedura di caratterizzazione dell'attributo sembra molto immediata, ha creato non pochi problemi a livello operativo.
Una semplice ricerca della stringa contenente il campo "nome_host" all'interno della stringa contenente la recensione ho restituito uno squilibrio importante verso il "non presente".
A questo punto non si era certi se la procedura fosse errata o se effettivamente la stragrande maggioranza dei clienti non utilizzasse il nome di persona dell'host.
Un rapido *troubleshooting* ha dimostrato che il metodo fosse assolutamente fallaceo.
Questo perché il meccanismo di confronto tra stringhe nei principali linguaggi di programmazione effettua un confronto carattere per carattere. È sufficiente una maiuscola al posto di una minuscola o uno spazio nella posizione sbagliata per rendere falso il confronto.
Il primo tentativo di correzione è stato quello di trasformare entrambe le stringhe in caratteri minuscoli per evitare il problema del cosiddetto linguaggio *case sensitive*.
Questo accorgimento ha aumentato il numero di "positivi" senza però raggiungere numeri soddisfacenti.
Esplorando un piccolo campione dei nomi inseriti nella colonna "nome_host" ci si è resi conto che in realtà, nella stragrande maggioranza dei casi, i padroni di casa non popolano tale campo con un singolo nome di persona, ma piuttosto utilizzando il nome commerciale dell'attività ("Mr. Apartement") oppure inseriscono una combinazione dei nomi di più persone, connessi nei modi più svariati ("y","+", "and", "&" per citarne alcuni).

Lo step successivo è stato quello di fare in modo che la *subroutine* riconoscesse la presenza di uno qualsiasi dei nomi di persona presenti.

Per farlo, la stringa “nome_host” è stata depurata di tutti i possibili connettori tra nomi (“y”, “+”, “()”,etc), e si è creata una nuova stringa dove i vari nomi sono separati da un singolo spazio.
Successivamente sono state eliminate tutte le buzzword come “Barcellona, flats, o apartement”. Così facendo, si perde la possibilità di rilevare la confidenza di tali host; si tratta comunque di una perdita calcolata che è compensata dalla protezione da un rischio molto peggiore che si sarebbe potuto verificare al momento dell’estrapolazione della parte di recensione relativa all’host (spiegata in una sezione successiva).

Generata la stringa formattata in modo conveniente, è stato creato un oggetto python “lista” temporaneo, costituito da tante stringhe quante parole presenti nel campo “nome_host” formattato. Ogni nome contiene un possibile modo con cui il cliente potrebbe riferirsi all’host all’interno della recensione. A questo punto si ricerca la presenza di uno qualsiasi degli elementi della lista all’interno della recensione. In caso di riscontro l’attributo “confidenza” salta a 1 e si passa all’elemento successivo.

Si è consapevoli che l’attributo potrebbe essere distorto verso il basso (sottidimensionato) in quanto l’algoritmo non è robusto rispetto ad eventuali errori di battitura e non è in grado di rilevare segni di confidenza nel caso i clienti abbiano interagito (e citino nella recensione) persone i cui nomi non sono stati registrati sul sito nel campo “nome_host” (si pensi ad esempio all’eventuale figlio dell’host).
Le fattispecie possibili sono così numerose da non poter prevedere una soluzione particolare per ognuna, e non è restato che accettare questo limite della procedura.

In Figura1 si può apprezzare in maniera visiva la “non popolarità” dell’uso del nome di persona, in quanto solo nel 25% dei casi l’attributo viene caratterizzato con un 1.

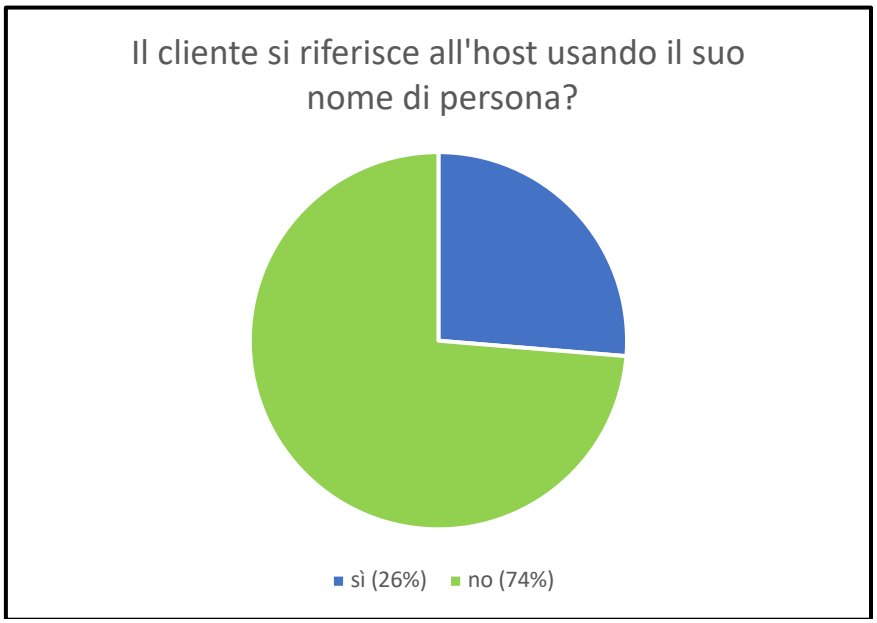


Figura 1: Effetto confidenza

3.1.6 Estrazione della parte di review relativa al comportamento dell'host

Il punto cruciale della fase di processamento delle recensioni è stato individuare le porzioni di recensione in cui si parla del comportamento dell'host e separarle dal resto.
Esplorando un campione di recensioni ci si è resi conto che tendenzialmente le persone condividono la loro esperienza parlando di:

- posizione ed estetica/funzionalità dell'alloggio
- città
- rapporto con host
- eventuali problemi che hanno avuto.

Per dare dati di qualità in pasto all'algoritmo di classificazione degli host era imperativo per questa fase conservare tutta l'informazione possibile relativa al comportamento delll'host e la sua interazione con il cliente ed escludere quanto più possibile legato ad altri aspetti.
Il meccanismo che si è applicato si chiama *tokenization* e, come nel caso precedente, è molto semplice a livello concettuale ma presenta delle insidie a livello operativo.
Per prima cosa si suddivide la review nelle frasi che la compongono, poi per ogni frase, se questa risponde ad un criterio di conformità sull'argomento, la si mantiene intatta, altrimenti la si sostituisce con una stringa vuota. Una volta analizzate tutte le frasi, si concatenano le stringhe residue per formare quella che nel database è denominata come "review_host".

I problemi di questa fase si distribuiscono lungo 3 direttrici:

- Lunghezza variabile delle review: il fatto che il numero di frasi non sia noto a priori non permette di creare una struttura fissa, ma impone di lavorare a livello locale su una recensione per volta, e di avere come output una unica stringa (in caso di struttura fissa si poteva evitare di ricorrere alla cancellazione di stringhe e concatenazione finale, non tra i metodi più eleganti per filtrare informazione)
- Contenuto e formattazione non prevedibile del testo. Il *tool* di "*tokenizzazione"* usa dei criteri di separazione frasi molto standard, che spesso non sono abbastanza flessibili per gestire del testo libero. Più le frasi sono corte e ben delimitate, più è facile estrarre il significato richiesto. Per questo motivo è stato necessario standardizzare in modo aggressivo la formattazione e il contenuto del testo. Sono stati eliminati tutti i caratteri speciali ed emoticon, e sostituiti tutti i segni di interpunzione con "virgole" e "punti".
- La scelta del criterio di conformità. Il problema è particolarmente complesso perché in generale è difficile inferire sull'argomento di una frase a partire dalle parole in essa contenute. In un primo momento era stata valutata l'idea di implementare *tool* che "scommettano" sull'argomento più probabile di una frase, tuttavia diverse volte all'interno di questo lavoro ci si è resi conto che strumenti di *natural lenguage processing* che funzionano molto bene in contesti circoscritti (molti strumenti di *Sentiment Analysis* considerati *state of the art* vengono validati attraverso un famoso database di recensioni di film chiamato "IMDB2"), perdono nettamente di efficacia in un *framework* così eccentrico come quello in esame. Data l'instabilità di questi strumenti si è deciso di ricorrere ad un approccio più tradizionale, usando come criterio di conformità di argomento la presenza di almeno una di una serie di parole "parlanti". Il problema si è trasformato nel selezionare il subset di parole che minimizzava il numero di errori di prima specie e di seconda specie. In generale l'idea era di inserire parole che possono essere associate esclusivamente ad una

persona, in modo di poter escludere tutte quelle frasi che si riferiscono all'appartamento oppure alla città. Il rischio di selezionare delle frasi che descrivono un soggetto terzo non host, è stato quantificato come trascurabile ricorrendo all'analisi manuale di un piccolo campione.
La selezione della lista ha richiesto diverse iterazioni con valutazione manuale del risultato, la versione finale si è assestata su:

- Una lista di pronomi personali da riferirsi ad uno o più host
- Il termine "host"
- Il termine "owner"
- Il termine "guy"
- Il termine "staff"
- Il termine "question"
- Il termine "helpful" e "help"
- Il termine "recommendation"
- Il termine "communication"
- Il termine "service"
- Il termine "friendly"
- Il termine "responsive"

In questa fase, al contrario della procedura spiegata in [3.1.5] si è potuto sfruttare a proprio vantaggio la ricerca carattere per carattere nel confronto tra stringhe in python. Ricercando "host ", ovvero la parola host seguita da uno spazio, si è riusciti ad escludere tutte quelle frasi in cui si usa il genitivo sassone "host's" per riferirsi ad aspetti non propriamente personali (ad esempio la stringa "host's location is awsome" si presenta con frequenza molto alta e avrebbe generato un numero importante di falsi positivi).

Si è consapevoli che il metodo, pur funzionando molto bene, non ha un'accuratezza pari al 100%. Nonostante questo, la capacità di riprodurre i risultati rieseguendo lo script e la possibilità di riparametrare la funzione che si occupa del criterio di congruenza, aprono alla possibilità di ottenere risultati migliori, integrando con facilità, nel momento in cui emergessero, nuovi *tool* di *guessing* stabili anche in contesti più difficili.

3.2 Classificazione delle recensioni

Questo paragrafo si occupa di descrivere il processo che ha portato all'elaborazione dell'algoritmo di classificazione delle singole review_host (per essere più precisi, di classificazione dell'host descritto nella review).

Ancora prima di sviluppare l'algoritmo di classificazione, parecchio tempo è stato speso cercando di capire quale fosse il tipo di quantificazione/classificazione più adatto. Tentativi precedenti di razionalizzare il comportamento dell'host avevano proposto una misura di gentilezza basata sul conteggio delle parole positive a lui riferite. In questo modo, ad ogni recensione veniva associato un punteggio numerico, rappresentato su una scala di rapporto.

L'approccio proposto in questa tesi è differente: a livello di singola recensione si è deciso non di ricorrere ad una misura numerica, ma piuttosto di tentare di attribuire ogni review ad una classe, mappata su una scala ordinale. La letteratura, in termini di *Sentiment Analysis,* ovvero quella disciplina che si occupa si estrarre il sentimento associato ad un testo, ricorre largamente ad una classificazione a 3 classi: negativa, positiva, neutra.

In un primo momento l'output dell'algoritmo era stato sviluppato in modo coerente con l'approccio a 3 classi, ma sono sorti dei dubbi relativi alla scarsa variabilità dei dati da dare in pasto ad analisi successive.
Analizzando un subset di dati estratti in modo casuale ci si è resi conto che l'occorrenza di recensioni negative è minima, e la stragrande maggioranza dei dati si divide tra recensioni neutre e buone.
Inoltre, la classe positiva racchiude al suo interno recensioni altamente eterogenee, in quanto si possono trovare recensioni che presentano una intensità minima ("l'host è gentile") e altre che invece presentano descrizioni dettagliate dell'esperienza fenomenale che i clienti hanno avuto grazie al padrone di casa.

L'elemento innovativo dell'approccio proposto in questa tesi consiste nel cercare di andare in profondità nella categoria delle recensioni positive e cercare di separare in modo sistematico quelle che sono generalmente positive, da quelle che invece sono state definite "super".
Le classi così ottenute sono 4:

- Recensione negativa
- Recensione neutra
- Recensione positiva
- Recensione super

La distinzione tra una recensione positiva e una super è una operazione molto semplice a livello inconscio: leggendo un giudizio su una persona risulta immediato avere un'idea se il sentimento ad essa associato sia mite, oppure se l'intensità sia tale da distinguere tale recensione dalla normalità.
Purtroppo, esternalizzare la logica dietro a un ragionamento del genere risulta particolarmente difficile, e ancora più complicato è generare un algoritmo che leggendo singole parole, sia in grado di cogliere la semantica della frase e l'intensità del sentimento ad essa associato. Questo è il motivo per cui gli approcci tradizionali tendono a fallire quando si tratta di Natural Language Processing.
Per cercare di riprodurre la logica intrinseca con cui il cervello prende decisioni, si è deciso di ricorrere all'Intelligenza Artificiale e in particolare al *Machine Learning*.

3.2.1 Piano di lavoro originale

Il piano di lavoro originale consisteva nello sviluppare un modello di *Machine Learning* che fosse in grado di analizzare la porzione di recensione relativa all'host (il campo "review_host" generato precedentemente) e attribuirla autonomamente a una delle 4 classi.
L'idea su cui si basa l'approccio è la seguente: fornendo in input un campione sufficientemente ampio ed eterogeneo, e associando a tale campione l'output desiderato, ovvero la classe più appropriata, le reti neurali del modello di *Machine Learning* sono in grado di internalizzare la logica di ragionamento umana riconoscendo pattern all'interno dei dati, in questo caso particolari sequenze di parole.
L'onere associato a tale approccio è la mole di dati da processare manualmente per generare il cosiddetto *training set*. La letteratura afferma che con dati numerici, la massa critica per avere un risultato soddisfacente, si assesta su almeno 500 occorrenze per ogni classe.
Sfortunatamente la pratica di apprendimento è molto più complicata quando si tratta di dati testuali, sia per la moltitudine di parole presenti nel vocabolario comune, che rendono virtualmente impossibile che durante la fase di *training* l'algoritmo entri in contatto con

ogni parola potenzialmente significativa, sia per il fatto che il modo di esprimersi e il senso legato ad una espressione trascende la pura combinazione di parole.
Per tali motivi, durante i primi test di validazione, ci si è resi conto che l'algoritmo "*performava"* in modo sorprendente nella distinzione tra "positivo" e "super", ma sistematicamente non era in grado di cogliere recensioni negative a meno di non rilevare aggettivi estremamente intensi ("terribile, peggior, pessimo", etc).

In Tabella5 presenta il risultato di un test effettuato durante lo studio di fattibilità dell'algoritmo.

Classe	Percentuale di recensioni classificate correttamente
Super	85%
Positivo	90%
Neutro	100%
Negativo	0%

Tabella 5: Performance dell'algoritmo di classificazione iniziale

Alla luce di tale risultato si è deciso si ridurre lo *scope* di applicazione del modello di *Machine Learning* alla distinzione tra "positivo" e "super" e di utilizzare strumenti alternativi per individuare le recensioni negative.

3.2.2 Procedura finale di classificazione

3.2.2.1 Gestione di recensioni neutre

Il riconoscimento di recensione "neutra" è stato immediato, in quanto la fase di processamento descritta in [3.1.6] che ha generato il campo "review_host", ha restituito una stringa vuota ogni volta che la recensione originale non presentava frasi riferite all'host. In questo modo, è bastato associare alla classe neutra tutte quelle recensioni che presentavo lunghezza pari a zero.

Per correttezza formale si è voluto controllare l'occorrenza di falsi negativi: in generale, si è potuto verificare che ogni qual volta il cliente si riferisce al cliente (utilizzando una delle formule incluse nei criteri di congruenza), poi procede esprimendo un giudizio polarizzato, in quanto non si sono individuati messaggi neutri come ad esempio "l'host è neutro" oppure "niente da dire sull'host".

Per fornire qualche statistica preliminare, la frequenza assoluta di recensioni neutre è stata pari a circa 125416 recensioni, pari al 48% del totale dei dati.

3.2.2.2 Gestione delle recensioni negative

Alla luce dell'insuccesso del modello di Machine Learning custom, si è deciso di ricorrere a degli strumenti di *Sentiment Analysis* che sono stati parametrati in modo specifico per individuare sentimenti negativi associati ad un contenuto testuale.
Anche questi strumenti si basano sull'Intelligenza Artificiale e in particolare sulle reti neurali.
Il successo di questi pacchetti è garantito dalla mole enorme di dati che sono stati utilizzati per parametrarli in fase di *training*. La scala dei dati in input, infatti, è probabilmente il fattore a più alto ritorno rispetto all'accuratezza nei modelli di machine learning.
Non sorprende, infatti, che questi vengano considerati *state of the art* nel 2020.

Il primo strumento utilizzato è un pacchetto chiamato SID, ovvero *Sentity Intensity Analyzer*. Si tratta di una funzionalità implementabile in python che riceve in input un testo, in questo caso il campo "review_host", ne analizza il contenuto ricercando parole ad alta intensità, che sono state modellizzate in un dizionario che associa a ogni parola un punteggio, e restituisce un valore compreso tra -1 e +1. Ciò che questo valore vuole rappresentare è il grado di positività del sentimento legato al testo. Valori minori di zero individuano recensioni negative, valori positivi crescenti individuano recensioni sempre più positive.
Il punteggio in output è stato salvato in una colonna del database, ed è stata generata una *dummy* di riepilogo che assume valore unitario se la recensione è stata classificata in modo negativo da SID.

Il secondo strumento è la funzione Polarity estratta dal pacchetto Textblob.
Il principio di funzionamento è esattamente lo stesso, la funzione riceve in input il testo e restituisce un valore compreso tra -1 e 1. Come nel caso precedente è stata generata una variabile binaria per individuare in modo immediato le recensioni classificate in modo negativ da Polarity.

È ragionevole, a questo punto chiedersi, quale sia l'utilità di sfruttare due strumenti invece che uno, dato che il loro impianto logico è molto simile.
La scelta è dipesa dalla gestione dei falsi negativi: analizzando i risultati di entrambi gli strumenti ci si è resi conto che questi funzionano bene nell'individuare le recensioni effettivamente negative, ma spesso, in presenza di negazioni o di espressioni con significato non letterale, classificavano come negativa una recensione positiva o addirittura super.
Considerare come effettivamente negative, solo le recensioni classificate come negative da entrambi i metodi, ha permesso di ridurre in modo sistematico gli errori di seconda specie.

L'occorrenza di recensioni "negative" si è assestata in questo modo su circa 2218 unità, pari a circa il 9 per mille del totale dei dati.

3.2.2.3 Gestione delle recensioni positive e super

In questa sezione si discute il processo di costruzione e di parametrazione del modello di *machine learning* utilizzato per distinguere le recensioni positive da quelle super.

Le fasi di sviluppo necessarie ad arrivare ad un modello utilizzabile sono le seguenti

- Preparazione dei dati in input
- Selezione dell'output desiderato
- Architettura della rete e parametrazione del modello
- Fase di training
- Fase di validazione

Non si tratta di fasi strettamente sequenziali, al contrario l'attività più critica durante lo sviluppo del modello è capire come ottimizzare architettura, parametri o funzione di attivazione (si capirà a breve di cosa si tratta) alla luce dei risultati di test effettuati su un piccolo sottoinsieme dei dati.

3.2.2.3.1. Preparazione dei dati in input

Come un normale applicativo, una rete neurale prende in input dei dati numerici, li elabora e restituisce un output coerente con la funzione di attivazione scelta.
La prima sfida è stata trasformare dati testuali (salvati come stringhe nel campo "reviews_host" del database) in modo da essere elaborabili dal modello.
Si è scelto di utilizzare l'approccio noto in letteratura come *bag of words*.
Questo consiste nel generare uno *script* che legga le recensioni e salvi le N parole a frequenza più alta. A questo punto si crea una struttura dati chiamata "dizionario": ad ognuna delle parole più frequenti si associa un indice tra 0 e N-1, di solito preservando come ordine la frequenza di apparizione decrescente nel testo.
A questo punto, ogni recensione può essere convertita in un vettore binario di N elementi, in cui in ogni posizione è presente un 1 se la parola associata a quella posizione nel dizionario è presente nella recensione.
L'implementazione di una procedura piuttosto complessa è stata in realtà resa molto agevole grazie al pacchetto di funzioni Preprocessing, incluso nell'ambiente Keras di Tensorflow.
(Tensorflow è il *framework* di sviluppo di modelli di *Machine learning* elaborato da Google).

L'elemento cruciale di questa operazione è stata la scelta del parametro N: come è facile intuire, aumentare il numero di parole da dare in input aumenta la capacità di imparare del modello, in quanto aumenta il numero di informazioni su cui inferire.
Nonostante quanto detto sopra, dimensionare N il più grande possibile non è sempre la scelta ottima per due motivi:

1. Aumentandone la dimensione di partenza del modello, ne aumenta il costo computazione, in quando aumentano in modo esponenziale le operazioni da effettuare per processare i dati; selezionare un N troppo grande può addirittura pregiudicare la possibilità di compilare il modello in tempi sensibili su un computer personale
2. Aumentare eccessivamente il parametro N espone al rischio di *overfitting*: il modello diventa estremamente efficiente ad inferire nel contesto delle parole che ha imparato, ma riconosce un significato talmente forte a determinate combinazioni particolari di parole, che perde la capacità di generalizzare ciò che ha imparato e quindi di inferire su dati esterni alla propria base dati. Un modello che soffra di o*verfitting* è completamente inutilizzabile.

Dopo svariati tentativi, partendo da N=1000, fino a tentativi estremi effettuati con 100 000 parole, si è deciso di selezionare N pari a 20000 (nel codice è inglobato nella variabile VOCAB_SIZE).

3.2.2.3.2. Selezione dell'output

Il fine ultimo del modello è quello di esibire uno 0 se la recensione è classificata come positiva, ed un 1 se questa invece risulta super.
L'unico accorgimento in questa fase è stato generare una relazione biunivoca tra il valore numerico e il nome della classe in modo da poter passare agevolmente da una all'altra.
Come nell'approccio *bag of words* questo problema si è risolto sfruttando una struttura "dizionario" di *python*, che salva in memoria la corrispondenza.

3.2.2.3.3 Architettura della rete

Il concetto di rete neurale si può riassumere in modo molto semplicistico come una struttura computazionale parallelizzata composta da neuroni che trasformano degli input in output.
Viene definita parallela perché ogni strato (*layer*) è composto da un certo numero di neuroni che elaborano in modo indipendente.
La rete vera e propria si ottiene combinando, in diversi modi possibili, un certo numero di *layer*. Per questa applicazione si è scelto di utilizzare un modello sequenziale e denso: ciò significa che ogni neurone di uno strato successivo è collegato a tutti i neuroni dello strato precedente. Sfruttare questo grande numero di connessioni permette di esplorare uno spazio delle combinazioni virtualmente infinito.

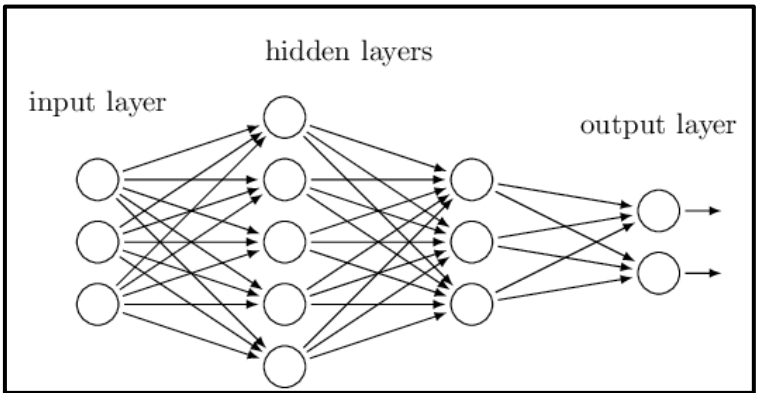


Figura 2: Schema di rete neurale sequenziale densa

Scelta la struttura della rete, il passo successivo consiste nel selezionare il numero di strati, ed il numero di neuroni per ogni strato.
La scelta è stata il risultato del bilanciamento di due aspetti:

- Capacità di spiegazione del fenomeno, che cresce al crescere dei parametri
- Economicità del modello: man mano che la dimensione cresce, i tempi di *training* e di *deployment* della rete esplodono

È difficile spiegare cosa rappresenti a livello fisico il numero di strati di una rete, il parametro è stato posto pari a 6 tramite una ottimizzazione manuale.
È più facile invece rappresentare il significato del numero dei neuroni per strato.
È pratica comune che il primo strato abbia la dimensione dei dati in input, in questo caso 20000 (scegliere un qualunque numero inferiore vorrebbe dire eseguire calcoli per estrarre dei dati e poi non includerli nell'analisi) e che l'ultimo strato abbia in output le dimensioni delle possibili alternative della classificazione, in questo "positivo" e "super", quindi 2.
Il numero di neuroni degli strati intermedi rappresenta come l'informazione contenuta del vocabolario delle parole viene elaborata e fatta convergere a due sole possiblità.

La scelta di questi valori è stata effettuata in modo piuttosto intuitivo: per il primo strato intermedio ci si è immaginato che molte parole frequenti nel dizionario, fossero in realtà di scarsa informazione, come ad esempio congiunzioni ed articoli. Per questo motivo si è scelto si conservare solo una parola ogni 4 e di generare uno strato da 5000 neuroni.
La sequenza degli altri strati procede nel seguente modo: 1000 parole per il terzo strato, 500 per il quarto, 100 per il quinto.
Il sesto strato trasforma le ultime 20 parole nell'output 0/1 richiesto.

L'ultimo elemento da sviluppare è la cosiddetta *funzione di attivazione*: questa esprime come l'input dell'ultimo strato viene convertito nell'informazione in output.
Per questa applicazione si è scelto una funzione di attivazione *sigmoide*, uno strumento statistico, che sfruttando la regressione logistica effettuata a partire dai dati dell'ultimo strato, restituisce la probabilità che la recensione appartenga alla classe positiva, e per complemento a 1, alla classe super.
Per tradurre l'output della funzione *sigmoide* nell'output effettivo del modello, si è deciso di associare la recensione alla classe per cui mostrava una probabilità di appartenenza massima.

3.2.2.3.4 Fase di training

Per far si che il modello fosse in grado di internalizzare le logiche di assegnazione umane, è stata necessaria una fase di *training* consistente.
Un subset di recensioni estratto dal database originale, costituito da recensioni positive e super è stato classificato manualmente.
Le recensioni, convertite in vettori numerici con lo stesso approccio *bag of words*, e la colonna contenente le classi associate manualmente, sono poi state date in pasto alla funzionalità *model.fit* dell'ambiente *tensorflow.*
Questo comando effettua una ottimizzazione ricorsiva dei parametri interni del modello, guidata da due funzioni obbiettivo chiamate *accuracy* e *val_accuracy*.

Il meccanismo è il seguente: si continua a "ciclare" sul dataset, fino a quando entrambe le funzioni obbiettivo restituiscono valori maggiori rispetto all'iterazione precedente.
La funzione *accuracy* rappresenta la capacità di associare l'output corretto all'interno del subset utilizzato per il training.
La funzione *val_accuracy* cerca di catturare la capacità di generalizzazione del modello su dati diversi da quelli utilizzati durante la fase di *training* (su dati non "familiari").
Tendenzialmente la funzione *accuracy* cresce sempre con il numero nei cicli, in quanto la rete neurale è in grado di esplorare sempre di più lo spazio delle possibilità e attribuire significato a nuove combinazioni di neuroni.
Il fatto che ad una certa iterazione *val_accuracy* tenda a decrescere, è sintomo di *overfitting*, la rete ha raggiunto il suo potenziale di apprendimento e l'unico modo per migliorare ulteriormente la performance, è l'ottimizzazione di altri parametri (ritornando a lavorare sulla fase di architettura).

Il modello, una volta ottimizzato, si è assestato su valori di *accuracy* interna intorno al 90%, e di *val_accuracy* dell'85%.

3.2.2.3.5 Fase di validazione

Si tratta di una fase di test effettuato su un campione diverso rispetto al subset utilizzato durante la fase di *training*.
Si effettua tramite il comando *model.evalute* ed ha restituito valori di accuratezza non di molto inferiori rispetto ai risultati della fase di *training*, come auspicabile.

Una volta validato il modello, ne è stato lanciato il *deployment* sul database completo.
Alla fine dell'esecuzione, durata più di 6 ore, ad ogni recensione, corrispondeva una colonna che specificava la classe di appartenenza.
Ai fini di successive elaborazioni, sono state create 4 variabili binarie, la cui combinazione permette di individuare in modo numerico la classe di riferimento.

In Tabella6 si può osservare la frequenza assoluta, percentuale e cumulata di ognuna delle 4 classi.

Classe	Freq. assoluta	Freq. percentuale	Freq. comulata
Negativa	2218	1%	1%
Neutra	125416	49%	50%
Positiva	97591	37%	87%
Super	32255	13%	100%

Tabella 6: Composizione dei diversi tipi di recensione

Per facilità di interpretazione del risultato, le frequenze assolute sono state inserite in un grafico a torta, mostrato in Figura3.

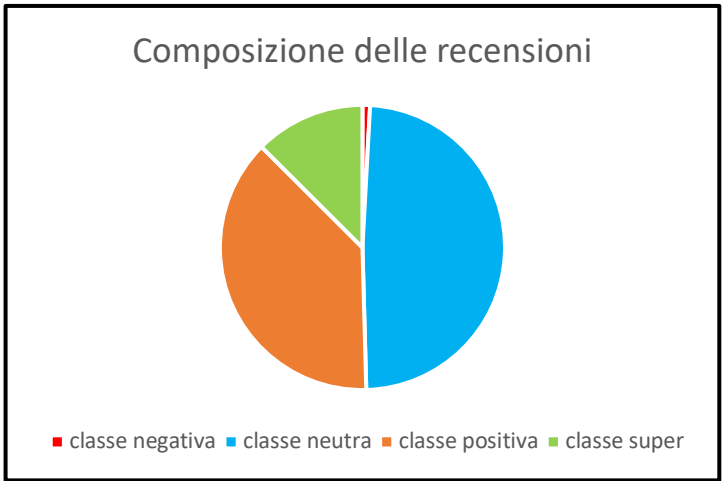


Figura 3: Grafico a torta sulla classificazione delle recensioni

Al fine di avere ulteriori indicatori a disposizione per analisi successive, si è deciso di salvare in una colonna del database i punteggi restituiti dalle funzioni "SID" e "polarity", applicate a tutte le recensioni del database.

3.3 Elaborazione delle misure di gentilezza

3.3.1 Aggregazione delle recensioni

Una volta classificate le singole recensioni, il passo successivo è stato raggruppare tutte le recensioni relative ad uno stesso alloggio (ciò che Airbnb chiama *listing*) per poter inferire sul comportamento dell'host.
Si è valutato ex-ante la possibilità di aggregare non su base alloggio ma su base host, ovvero di mettere insieme tutte le recensioni relative allo stesso padrone di casa (che ovviamente può possedere più alloggi).

Questo approccio è stato scartato, in quanto si è valutato che lo stesso host può comportarsi in modo diverso con clienti di alloggi diversi (si pensi, nel più semplice dei casi, ad un host che possiede un alloggio vicino alla propria abitazione, e di conseguenza può accogliere agevolmente i clienti, ed un alloggio in un altro quartiere della città, per il quale gli è conveniente fare il *check in* in modo digitale).
Per cercare di tenere conto, nel contesto di analisi successive, che la performance di un particolare alloggio possa essere influenzata dal fatto che il padrone di casa possieda altri listing, è stata inserita una variabile, che tiene memoria del numero di alloggi di cui un host è titolare.

A livello operativo l'attività di aggregazione è piuttosto elementare, e permette non solo di separare recensioni di listing diversi, ma anche di applicare ad ogni gruppo delle funzioni di aggregazione.
In particolare, si è deciso di calcolare la media di gruppo dei seguenti attributi:

- Confidenza
- Punteggio_SID
- Polarity
- Sulle 4 variabili binarie che individuano la classe delle recensioni (classe_negativo, classe_neutro, classe_positivo, classe_super)

Per rendere più chiaro il risultato del processo di aggregazione, in Tabella7 viene mostrato una riga del database aggregato.

Listing_id	Confidenza	SID	Polarity	Freq. Negativo	Freq. Neutro	Freq. Positivo	Freq. Super
393717	0.51	0.36	0.25	0.0	0.43	0.50	0.07

Tabella 7: Esempio di funzione di aggregazione di gruppo

3.3.2 Algoritmo di classificazione del listing

La funzione di aggregazione applicata alle 4 *dummies* che rappresentano la classe, restituisce la frequenza relativa di appartenenza ad ognuna delle classi.
Questo è un indicatore molto potente, perché permette di capire come sono distribuite le opinioni relative allo stesso alloggio.
Per decidere come attribuire una classe di gentilezza unica ad ogni *listing*, si è implementata una procedura in 3 parti.

Prima parte
Se la frequenza di recensioni negative risultava maggiore o uguale al 10% del totale, l'host è stato assegnato alla classe negativa. Una soglia così severa ha l'obbiettivo di valorizzare l'informazione molto significativa, ma scarsa in quanto a bassa occorrenza, delle recensioni in cui il cliente si lamenta del comportamento dell'host.

Seconda parte
Per ogni listing, si è individuata la classe a frequenza massima, e si è salvato in memoria la frequenza e la classe ad essa collegata.
Questa procedura si è dimostrata scorretta, in quanto un listing che mostrasse il 40% di neutre, il 30% di positive e il 30% di super veniva assegnato alla classe neutra nonostante presentasse una porzione di recensioni "buone" (il 60%) sostanzialmente superiore rispetto a quelle neutre. Per correggere questa distorsione, si è deciso di accorpare temporaneamente

la classe positiva e neutra in una *macroclasse* "buona", ed effettuare il confronto delle frequenze, al fine di individuarne la moda, tra classe negativa, classe neutra e macroclasse buona.
Per i casi in cui la frequenza massima cadeva in corrispondenza della macroclasse buona, si è deciso di assegnare il listing alla classe super se la frequenza di recensioni super era superiore ad un terzo della frequenza della macroclasse.
Una soglia inferiore a quella più naturale, pari al 50%, permette, come nel caso delle recensioni negative, di enfatizzare il contenuto informativo di una classe poco popolata.

Terza parte
Si è operata una ulteriore correzione della baseline ottenuta: per tutti quei listing che sono stati classificati come "neutri", se la frequenza della classe neutro era inferiore al 65%, e se il restante 25% era diviso tra positivo e neutro, allora si andava a controllare l'attributo "confidenza". Se in almeno una recensione ogni due il cliente si riferiva all'host utilizzando il proprio nome di persona, allora la classe veniva corretta in positiva.
I risultati di questa procedura sono stati salvati in una variabile testuale "tipo", e come nel caso dell'attributo "classe" sono state generate delle dummies, indispensabili per includere in test statistici questo tipo di informazione ordinale.
Si è valutato se proporre l'upgrade di classe anche per la classe "negativa" e "positiva".
Per quanto riguarda la classe negativa, si è valutato che l'occorrenza di recensioni, e a maggior ragione di listing negativi, è estremamente bassa, quasi nulla, e ridurre ulteriormente la loro presenza in modo artificiale non avrebbe fatto altro che distruggere variabilità all'interno dai dati, già fortemente limitata dal fatto che una parte consistente dei listing sono stati classificati neutri.
Per l'upgrade da positivo a super la motivazione è stata differente: si è deliberatamente deciso di selezionare una soglia di intensità del sentimento alta per considerare una recensione super. Alla luce di questo, il semplice segnale di confidenza espresso dall'attributo "confidenza" non è sufficiente per giustificare l'upgrade della classe.
In Figura4 si raffigura un grafico a torta che mostra la composizione dei listing in seguito alla procedura di classificazione.

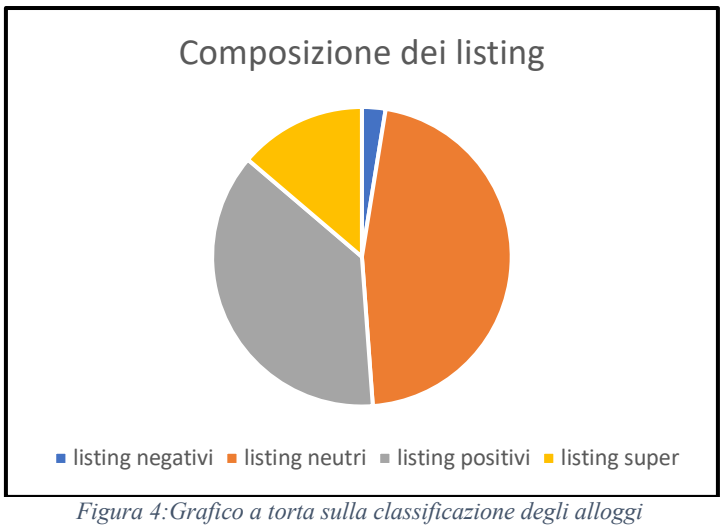


Figura 4:Grafico a torta sulla classificazione degli alloggi

I listing negativi sono il 3% del totale, quelli neutri il 46%, la frazione di listing positivi è pari al 37%, i listing super, infine occupano il 14% del totale.

3.3.3 Misura di gentilezza a partire dai tool di Sentiment Analysis

La funzione di aggregazione media di gruppo, applicata agli attributi "punteggio_SID" e "polarity", ha restituito un indicatore di positività medio delle recensioni di un certo listing espresso sul continuo, che permette di effettuare confronti omogenei tra *listings* diversi.
Si è deciso, per facilità di lettura e di interpretazione, di "*rimappare"* i valori assunti dai due punteggi in una scala da 1 a 10, dove 1 rappresenta un'opinione media super negativa, 5 un'opinione media neutra e 10 una entusiasta. I nuovi indicatori sono stati nominati, rispettivamente: "Score_1" e "Score_2" e in Tabella8 si presentano alcune statistiche descrittive di tali variabili.

Indicatore	Media	Dev.std	Min	Max	Primo quartile	Mediana	Terzo quartile
Score_1	6.54	0.83	3.77	9.48	5.91	6.49	7.12
Score_2	6.1	0.6	4.06	9.18	5.67	6.1	6.54

Tabella 8: Statistiche sui primi indicatori di gentilezza

Il passo successivo nell'esplorazione dei due indicatori è stato plottarne la distribuzione cumulata empirica, mostrata rispettivamente in Figura5 e Figura6.

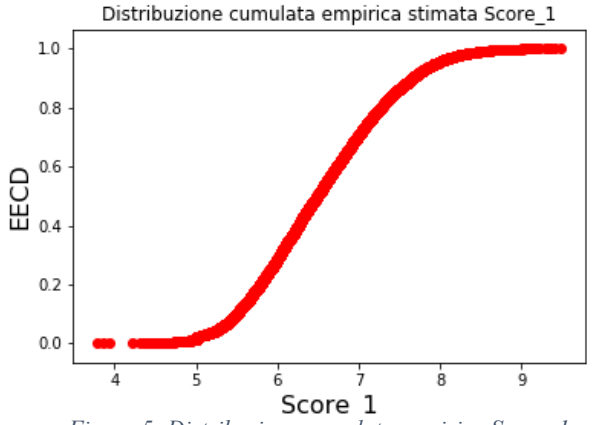


Figura 5: Distribuzione cumulata empirica Score_1

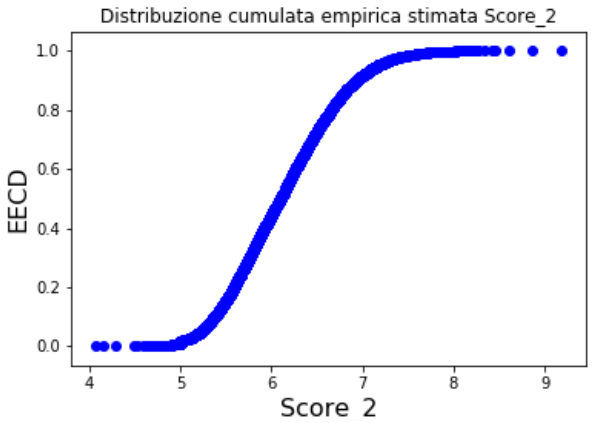


Figura 6: Distribuzione cumulata empirica Score_2

Essendo i due indicatori generati da una combinazione di variabili che si possono ritenere indipendenti e identicamente distribuite, è stato interessante verificare se la distribuzione risultante fosse approssimabile in modo accettabile attraverso una distribuzione normale.

Per prima cosa si è deciso di raggruppare i valori in intervalli e plottare per ogni intervallo, la sua frequenza di realizzazione. Figura7 e Figura8 mostrano un grafico a istogrammi per ogni indicatore.

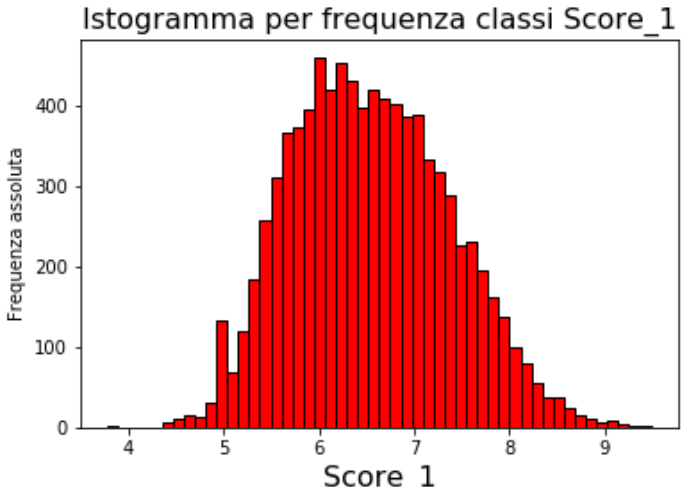


Figura 7:Istogramma per Score_1

Gli istogrammi mostrano la sicura unimodalità delle distribuzioni dei due indicatori.
Come si poteva già apprezzare dall'analisi delle due deviazioni standard, rispettivamente 0,83 per "Score_1" e 0,6 per "Score_2", nel caso del secondo indicatore i valori si stringono maggiormente intorno il valor medio, restituendo una distribuzione empirica più alta e più stretta.
Il fatto che entrambe le mediane sono praticamente identiche alle rispettive medie, poneva buoni presupposti per poter ipotizzare una distribuzione simmetrica.
A livello visivo, nonostante la massa dei punti sia ugualmente distribuita a sinistra e a destra del valor medio, si può apprezzare come la coda destra della distribuzione sia maggiormente popolata di valori estremi rispetto alla coda sinistra.
Questo comportamento è compatibile con quello che ci si aspettava a priori, in quanto, sin dalle prime analisi delle recensioni, appariva ovvio come performance degli host eccellenti fossero molto più probabili di performance terribili.

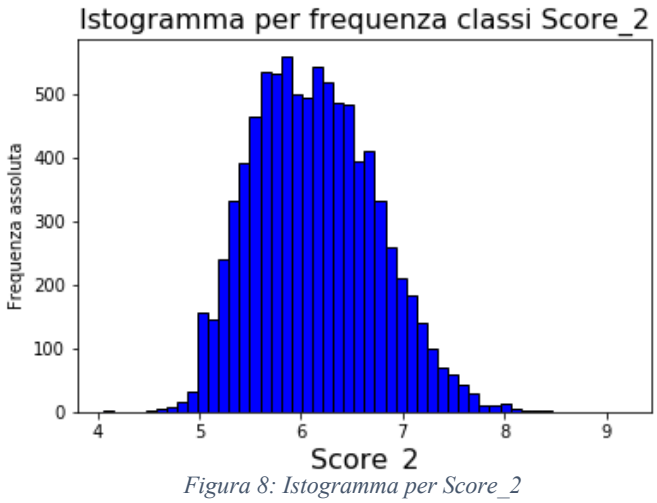


Figura 8: Istogramma per Score_2

Il passo successivo è stato cercare di approssimare le serie di dati attraverso una distribuzione normale. Questa operazione è stata effettuata attraverso la funzione *kdensity* che effettua una ottimizzazione iterativa dei parametri della *gaussiana*, per restituire la distribuzione stimata che meglio approssima quella empirica. In Figura9 e Figura10 si mostra l'output dell'approssimazione.

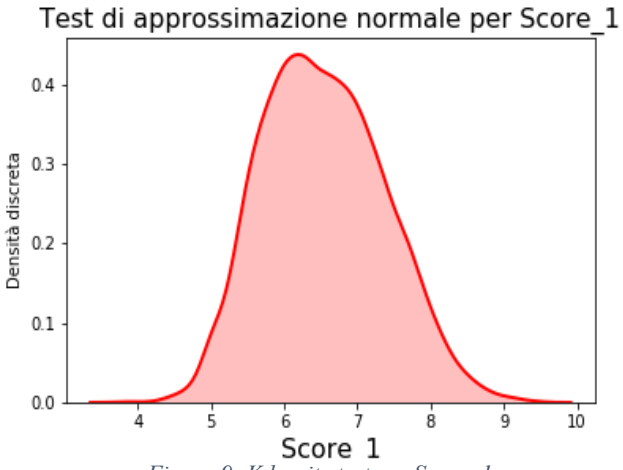


Figura 9: Kdensity test per Score_1

I due grafici mostrano che i dati sono ben approssimati attraverso due distribuzioni normali. I due tratti "irregolari" evidenziati dal grafico sono giustificati dal ragionamento fatto in precedenza sull'eterogeneità tra coda destra e sinistra.

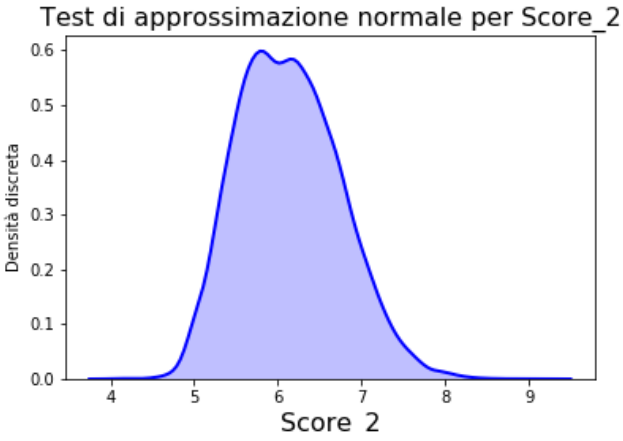


Figura 10: Kdensity test per Score_2

3.3.4 Misura di gentilezza derivata dal modello di Machine Learning

Ai fini di analisi regressive, la procedura di classificazione attraverso una struttura ordinale non è la scelta ottimale, in quanto distrugge un sacco di eterogeneità dell'informazione contenuta nei dati.
Di fatto la classe del listing viene associata solo sulla base della frequenza massima, ignorando la composizione dei dati residui.

In questo senso, due listing entrambi con il 90% di recensioni neutre, ma uno con il restante 10% di recensioni negative, e l'altro con il 10% ripartito tra recensioni positive e super, sono considerati equivalenti.
Tabella9 presenta un esempio del primo limite dell'algoritmo di classificazione.

Listing_id	Freq_neg	Freq_neutro	Freq_pos	Freq_super	Classe assegnata
23456	0,1	0,9	0	0	NEUTRO
65473	0	0,9	0,05	0,05	NEUTRO

Tabella 9: Caso anomalo tipo1

Una considerazione ulteriore è la seguente: viene classificato come neutro sia un listing con il 90% di neutre, che un listing con il 51% di neutre, 25% di positive e 24% di super.
Tabella10 rappresenta in modo tabellare la situazione sopra descritta.

Listing_id	**Freq_neg**	**Freq_neutro**	Freq_pos	Freq_super	Classe assegnata
23456	0	0,9	0,1	0	NEUTRO
65473	0	0,51	0,25	0,24	NEUTRO

Tabella 10: Caso anomalo tipo2

Per superare il limite che emerge dal considerare solo la moda di una distribuzione e cercare di introdurre informazione sul corpo della distribuzione, si è proposta una misura aggregata estratta a partire dalle frequenze.
La misura si è ottenuta moltiplicando le frequenze percentuali per dei pesi.
La scelta dei pesi è fortemente strategica e ha richiesto diversi tentativi, l'ispirazione che si è seguito è la seguente: si voleva che i listing negativi avessero un punteggio compreso tra il 3 e il 5; i *listing* neutri occupassero la fascia centrale fino al 7.5 (si ricordi che un soggiorno definito neutro non deve essere interpretato come qualcosa di negativo, per essere tale l'host deve aver adempiuto a tutti i propri doveri, semplicemente non ha mostrato una empatia tale da spingere il cliente a scrivere qualcosa sul suo comportamento).
Proseguendo un *listing* positivo avrebbe dovuto ottenere punteggi fino al 9, e un *listing* super occupare l'ultimo gradino della scala tra 9 e 10.

Per far sì che si ottenesse qualcosa di simile a quanto sperato i pesi selezionati sono stati:

- 3 per la frequenza negativa
- 7 per la frequenza delle neutre
- 8 per le positive
- 10 per le super

I risultati ottenuti seguivano per la maggior parte la scala che si aveva in mente, ma ci sono stati dei problemi di inversione d'ordine in cui comparivano *listing* positivi con un punteggio inferiore a quello di un *listing* neutro. Per risolvere questi problemi sono state introdotte delle soglie di saturazione, che hanno ripristinato l'ordine auspicato, riportando valori fuori scala al limite superiore della classe.
La misura è stata salvata nel database con il nome "rank".

Si procede ora con un'analisi preliminare dell'attributo.
Per prima cosa, l'ambiente *Pandas* di *python*, utilizzato per tutte le manipolazioni del database, con il semplice comando "*describe( )",* propone alcune statistiche descrittive sull'attributo, che sono state riprodotte in Tabella11.

Indicatore	Media	Dev.std	Min	Max	Primo quartile	Mediana	Terzo quartile
Rank	7,665	0.421	3	9.5	7,47	7,5	7,91

Tabella 11: Statistiche indicatore di gentilezza ad hoc 1,"rank"

Successivamente, i valori sono stati raggruppati in intervalli e si è prodotto un grafico di frequenze assolute a istogrammi, Figura11.

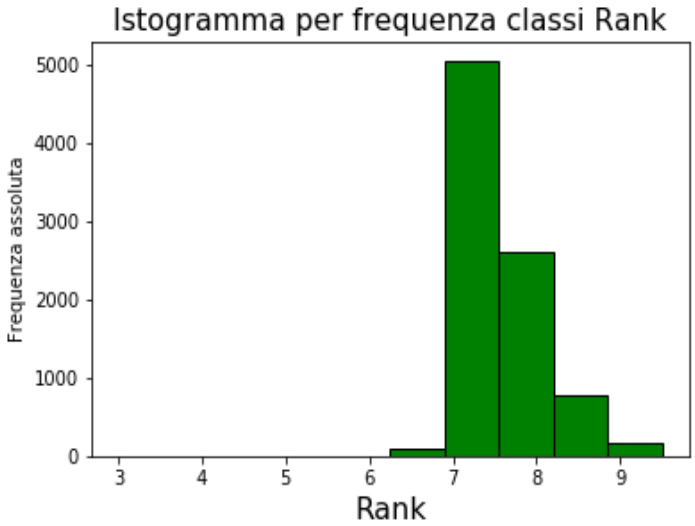


Figura 11: Istotgramma di frequenza variaible rank

Il grafico mostra un andamento interessante: la modellizzazione della gentilezza dell'host, generata tramite i *tool* non specializzati *SID* e *POLARITY*, e mappata tramite gli attributi "Score_1" e "Score_2" restituiva una distribuzione simmetrica intorno al valor medio (che coincideva con la mediana). Questo vuol dire, in altri termini, che tra i listing attivi sulla piattaforma, anche quelli con scarse performance a livello di gentilezza, sono in grado di catturare domanda e di conseguenza rimanere all'interno del mercato.
Al contrario, la modellizzazione tramite "rank" restituisce una distribuzione praticamente troncata al di sotto del valor medio, che cattura il comportamento che è stato definito come neutro.
Si può notare infatti come la maggior parte della massa, si concentri tra valori compresi tra 7 e 8, ovvero i valori che erano stati individuati per identificare la classe neutra, mentre la restante parte della massa si divide tra punteggi propri di host positivi e host super.

Il fatto che, al di sopra della classe neutra, la frequenza sia decrescente con il rank è di facile interpretazione a livello pratico: si può ragionevolmente pensare, infatti, che un comportamento tale per essere riconosciuti come host "positivi" richieda uno sforzo non indifferente che solo una piccola parte degli host sia disposta (o possibilitata) a produrre.
Il ragionamento è analogo nel valutare la frequenza degli host super, per i quali lo sforzo incrementale per garantirsi il riconoscimento di tale titolo è sicuramente importante, a fronte di un vantaggio competitivo che non è così ovvio a priori.

Sarebbe stato preoccupante, se invece, la maggior parte della massa fosse stata concentrata su host "super". Questa configurazione comporterebbe sicuramente un guadagno in utilità economica per i clienti, che in media ottengono un servizio superiore rispetto al caso esplorato in precedenza, ma allo stesso tempo inibisce ogni possibilità di vantaggio competitivo per gli host, che sono forzati a fornire una performance "super" pena il rischio di rimanere tagliati fuori dal mercato (in questo senso il servizio "super" rappresenterebbe una barriera all'entrata importante), ma non sono in grado di capitalizzare a livello di guadagno il loro comportamento.
Una situazione di questo tipo non è incompatibile con la realtà, anzi è probabile che si manifesti sotto forma di lento aggiustamento di mercato, nel caso lo stesso prenda coscienza dell'esistenza di un vantaggio legato alla "gentilezza".
Il tipo di servizio in cui si collocano piattaforme come Airbnb e Booking non è ancora giunto a maturazione, in particolare si verificano ancora fenomeni di Asimmetria Informativa, sia per quanto riguarda i clienti, che non sono in grado di valutare ciò che otterranno a priori, sia riferendosi alla scarsa capacità di analizzare e comprendere dall'esterno i meccanismi in atto all'interno del mercato.
È ragionevole pensare, per lo meno dal punto di visto dell'autore, che man mano che il business si avvicinerà a maturazione, ci sarà più *information disclosure* su quale siano effettivamente i fattori critici di successo, e qualora diventi validato che il comportamento del padrone di casa garantisca una leva importante sul prezzo o sui ricavi, allora si potrebbe verificare un meccanismo di adeguamento a uno standard "super", indipendentemente dallo sforzo umano richiesto.

Per quanto riguarda la non presenza di host con punteggi basso, questo potrebbe far ipotizzare ad un meccanismo di autoregolamentazione del mercato.
L'idea è la seguente: host che ricevono un numero significativo di recensioni negative sul loro comportamento, i quali secondo la modellizzazione proposta verrebbero classificati come host negativi, perderebbero progressivamente domanda.
A questo punto ci si immaginano due possibili scenari, o il padrone di casa prende coscienza della propria perdita di performance e devia il proprio comportamento, in modo da rientrare tra i canoni degli host neutri ed essere attivo sulla piattaforma, oppure è destinato ad essere escluso dal mercato, per mano degli utenti che preferiscono sistematicamente alloggi con recensioni migliori o per mano dei regolatori della piattaforma, nel caso esista qualche meccanismo di controllo qualità che limiti la visibilità degli host con performance pessime.
Sarebbe stato particolarmente interessante provare a verificare questa ipotesi andando ad analizzare gli andamenti dei punteggi degli host nel tempo e i loro relativi tassi di occupazione, ma purtroppo i dati pubblicati da Airbnb sono insufficienti e inadatti ad effettuare una indagine di tale tipo.

Siccome la scelta dei pesi rappresenta un'operazione particolarmente delicata, che si basa molto su esperienza ed intuizione, in quanto non esiste una procedura scientifica da seguire o un test per verificare che la scelta sia adeguata, si è deciso di produrre una seconda misura, chiamata "rank_2", generata tramite una combinazione alternativa di pesi.
In particolare, i pesi selezionati sono stati:

- 3 per la frequenza negativa
- 7 per la frequenza delle neutre
- 9 per le positive
- 10 per le super

In fase di modellizzazione statistica, attraverso il *software* STATA, saranno degli A/B testing tra i due indicatori a validare quale sia la misura che meglio si adatta a descrivere la realtà. Forti della volontà di polarizzare comportamenti estremi, i pesi della classe negativa e super non sono stati modificati, invece il peso relativo della classe positiva è stato aumentato rispetto al peso della classe neutra, per promuovere un comportamento virtuoso, anche vista l'eccessiva presenza di host neutri.
Come nel caso di "rank" si presentano statistiche descrittive in Tabella12.

Indicatore	Media	Dev.std	Min	Max	Primo quartile	Mediana	Terzo quartile
Rank_2	8,07	0.5	3	9.75	7.76	8	8,43

Tabella 12: Statistiche indicatore di gentilezza ad hoc 2

Dalla tabella si può notare uno *shift* dei punteggi verso l'alto, originato dall'aumento dei pesi assoluti. Ciò che in realtà è più interessante osservare è la distribuzione dei punteggi, presentata nel grafico di frequenza a istogrammi in Figura12.

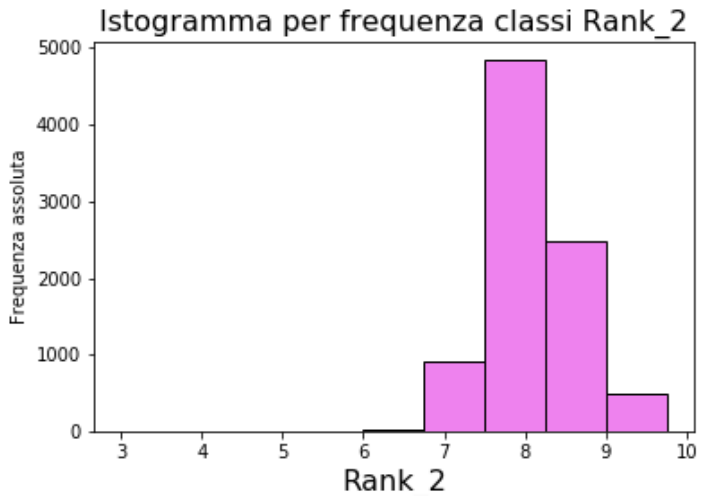


Figura 12: Istogramma di frequenza variabile rank_2

Dall'analisi della composizione delle classi (si veda ad esempio Figura3) emerge che circa il 50% dei dati appartiene alla classe neutra e che il 49% dei dati appartiene alle classi positivo e neutro.
Per come è costruito l'indicatore "rank_2", il punteggio massimo che può raggiungere un host neutro che abbia solo ricevuto recensioni neutre o negative è 7.
Siccome la mediana è 8, questo significa che necessariamente, la maggior parte degli host classificati come neutri, in realtà ottiene una cospicua parte di recensioni positive o neutre.
Questo risultato, anche se può sembrare di poco conto, mostra l'enorme passo avanti passando dalla classificazione in 4 classi, all'utilizzo di una misura nel continuo.
Se il 50% degli host erano neutri riduceva di molto l'eterogeneità dei dati per successive analisi, le misure "rank_1" e "rank_2" sono state strumentali nel capire che anche gli host che sembrano non fornire un servizio degno di nota, in realtà, con alcuni clienti sono in grado di distinguersi dalla media.
Sulla stessa lunghezza d'onda del ragionamento fatto per l'esclusione dal mercato degli host negativi, sarebbe interessante provare a capire a livello temporale e geografico, quali siano le ragioni dietro a questa differenza di comportamento, e se ci sia effettivamente una tendenza ad evolvere, tendendo ad un comportamento "super".

4. Costruzione del database finale

I modelli statistici, che verranno presentati a partire dal capitolo 6, cercano di legare il prezzo di un alloggio a delle variabili target e a delle cosiddette variabili di controllo, che, come si vedrà in seguito, hanno il compito di mitigare il problema della distorsione dei risultati indotta da variabili omesse.
Le variabili target per questo studio sono state elaborate a partire dai dati presenti nel database "reviews", mentre le variabili di controllo sono state estratte dal database "listings".
Così come il database "reviews", anche il database "listings" è reso disponibile direttamente da Airbnb attraverso la piattaforma proprietaria InsideAirbnb.
"Listings" contiene informazioni riguardo aspetti molto diversi tra di loro, come la struttura dell'alloggio, aspetti burocratici legati alla transazione (*policy* di cancellazione e deposito cauzionale ad esempio) e alcuni indicatori generati direttamente da Airbnb.
In Tabella13 si presenta la lista e una descrizione delle colonne che sono state selezionate dal database "listings", al fine di includerle nell'analisi.

Nome_attributo	Tipo	Descrizione	Richiede elaborazione
Price	€	Prezzo per notte	Sì
Guests_included	Num	Numero di ospiti per i quali il prezzo si riferisce	No
Accommodates	Num	Numero massimo di ospiti che l'alloggio può ospitare	No
Listing_type	Stringa	Informazioni sul tipo di alloggio	Sì
Bathrooms	Num	Numero di bagni	No
Bedrooms	Num	Numero di camere da letto	No
Beds	Num	Numero di letti	No
Number_of_ reviews	Num	Numero di recensioni per tale alloggio	No
Number_of_ reviews_ltm	Num	Numero di recensioni negli ultimi 12 mesi	No
Reviews_per_month	Num	Numero di recensioni medie in un mese	No
Calculated_host_listings_count	Num	Numero di alloggi che l'host di tale alloggio possiede	No
Host_is_superhost	Binario	Se 1, l'host è certificato Super dalla piattaforma	No

Host_response_rate	%	Percentuale di richieste di prenotazione alle quali l'host risponde	No
Host_acceptance_rate	%	Percentuale di richieste che si traducono in prenotazione	No
Security deposit	€	Importo del deposito cauzionale	No
Cleaning_fee	€	Costo per servizio di pulizia (opzionale)	No
Host_identity_verified	Binario	1 se l'host ha effettuato con successo la procedura di verifica della propria identità	No
require_guest_profile_picture	Binario	1 se host richiede che il cliente abbia una immagine del profilo per procedere alla prenotazione	No
require_guest_phone_verification	Binario	1 se host richiede che il cliente effettui la procedura di verifica del numero di cellulare	No
Cancellation policy	Stringa	Informazione su qualora sia possibile cancellare gratuitamente la prenotazione ed entro quanti giorni	Sì
review_scores_rating	Num	Punteggio generato da Airbnb, rating dell'alloggio/host	No
review_scores_accuracy	Num	Punteggio generato da Airbnb, corrispondenza tra quanto descritto nell'annuncio e l'esperienza reale	No
review_scores_cleanliness	Num	Punteggio generato da Airbnb, pulizia e ordine dell'alloggio	No
review_scores_checkin	Num	Punteggio generato da Airbnb, livello di servizio offerto durante il check-in	No
review_scores_communication	Num	Punteggio generato da Airbnb, facilità di comunicazione con l'host	No
review_scores_location	Num	Punteggio generato da Airbnb, percezione della posizione	No

review_scores_value	Num	Punteggio generato da Airbnb, rapporto qualità/prezzo del soggiorno	No
Longidute/latitude	Stringa	Individua posizione alloggio	Sì

Tabella 13: Colonne estratte dal database "listings"

Variabili interessanti come il prezzo settimanale, il prezzo mensile e il numero di metri quadrati dell'appartamento non sono state incluse, in quanto il numero eccessivo di valori mancanti avrebbe ridotto in modo rilevante la dimensione del campione utilizzabile.

4.1. Processamento delle variabili

Alcune delle variabili presenti in Tabella13 hanno richiesto delle operazioni di pre-processamento, o per trasformarle in modo da poter essere utilizzate nei modelli statistici, o più in generale per far emergere il contenuto informativo che dai dati grezzi non era così evidente.

4.1.1 Listing_type

Si tratta di una stringa di caratteri che specifica il tipo di alloggio (ad esempio, stanza di albergo, o stanza in appartamento). Ai fini di questo studio, un'informazione così granulare non è particolarmente utile.
Ciò che invece interessa maggiormente è distinguere gli alloggi dove il padrone di casa affitta l'intero appartamento (o l'intera casa) dagli alloggi dove l'host affitta soltanto una camera. Per fare questo, a partire dai dati testuali, si è creato una variabile *dummy* "App_intero" che assume valore 1 se l'host affitta un alloggio intero, 0 altrimenti.

4.1.2 Cancellation_policy

Si tratta di un campo testuale che esprime la facilità con cui un cliente può cancellare gratuitamente la sua prenotazione.
Tra i valori assunti dalla variabile, i più frequenti sono:

- *Flexible*
- *Moderate*
- *Strict_15*
- *Strict_30*
- *Strict_60*
- *Strict_90*

Per modellare in modo analitico questo fenomeno, si è deciso di creare una *dummy* "cancellation_strict" che assume valore 1 se la variabile "cancellation_policy" assume valori: Strict_15, Strict_30, Strict_60, Strict_90 e 0 altrimenti.
Si è valutata la possibilità di considerare come *Non Strict* (e quindi associare valore 0) tutti i casi in cui "cancellation policy" assume valore Strict_15.
A livello operativo tale valore significa che il cliente può cancellare la prenotazione solo almeno 15 giorni prima della sua prenotazione.

Se per alcuni contesti, come ad esempio nel mondo dell'hotelleria, un preavviso di 15 giorni può essere considerato ragionevole, per un tipo di turismo più informale e giovane, tipico delle piattaforme come Airbnb, non è stato valutato un livello di flessibilità che tuteli in modo adeguato le esigenze del consumatore.

4.1.3 Bathrooms

Un numero copioso di bagni in un alloggio è tendenzialmente un indicatore di spaziosità dell'ambiente e di lusso.
Per questi motivi si è deciso di includere la variabile originale nell'analisi.
Un altro dettaglio interessante, che "bathrooms" non è in grado di cogliere per come è strutturata, è la presenza di almeno un bagno nell'alloggio.
Se per gli appartamenti è un elemento abbastanza scontato, molto spesso, affittando una singola stanza all'interno di un appartamento può succedere di dover ricorrere ad un bagno condiviso tra gli inquilini.
Per cogliere questo aspetto, si è generate una variabile *dummy* "bagno" che assume valore 1 se è presente almeno un bagno privato nell'alloggio, 0 altrimenti.

4.1.4 Calculated_host_listings_count

Rappresenta il numero di alloggi diversi che possiede l'host.
Il fatto di possedere più alloggi potrebbe essere un modo per discriminare host amatori, che affittano l'appartamento o stanze dello stesso per avere una rendita accessoria, da agenzie che invece hanno il servizio di locazione come business primario.
Come per l'attributo precedente, spesso non importa solamente il numero di alloggi ma è interessante capire se l'host ne possieda più di uno.
A questo fine è stato generata la *dummy* "multiprop" che assume valore 1 se l'host possiede più di un alloggio.

4.1.5 Prezzo

Dopo una rapida formattazione dei dati, al fine di creare un campo numerico a partire da uno testuale, un'esplorazione preliminare dell'attributo ha mostrato la presenza di un cospicuo numero di valori a prima vista anomali.
Indipendentemente dal livello di lusso di un alloggio, osservare stanze singole in appartamento, per 2 persone, con un prezzo per notte di 2000 € è un fatto quantomeno curioso. Allo stesso modo, erano presenti degli alloggi per 2 persone o più, per un modico prezzo di 10 € per notte.

È difficile elaborare una procedura che a priori stabilisca se il prezzo di un alloggio è completamente fuori mercato, data la sua composizione.
Alla luce di questa incapacità di valutare se il prezzo presente nel database fosse affetto da errore, la possibilità di eliminare tutti gli alloggi con prezzi sospetti è stata rifiutata per il rischio di eliminare della genuina eterogeneità tra i valori.
Al fine di mitigare l'eventualità di errori di inserimento di dati nel record, a partire dai valori più estremi, si è effettuato un confronto tra il prezzo presente sul database ed il prezzo che tale alloggio presenta attualmente sulla piattaforma.
Consapevoli che l'effetto della recessione del turismo causata dal Coronavirus possa creare delle distorsioni dei prezzi attuali osservabili su Airbnb, si è valutato che fosse improbabile che un alloggio che ad Ottobre 2020 presenta un prezzo per la stagione 2021 di 80 €/notte, possa aver avuto a Luglio 2020 un prezzo di 2000 €/notte.

Per tutti gli alloggi per cui si è identificata una incompatibilità di questo tipo, il prezzo è stato sostituito con una media dei prezzi che l'alloggio mostra per l'anno 2021.
La stessa procedura empirica è stata effettuata, con grande fatica, per gli alloggi che a prima vista presentavano un prezzo troppo basso date le loro caratteristiche.

4.1.6 Location

A partire dai dati di longitudine e latitudine si è cercato di elaborare un indicatore che mostri quanto sia strategica la posizione di un alloggio.
Per prima cosa, facendo una ricerca incrociata su diversi blog di turisti, si sono individuati i 14 luoghi più gettonati dai turisti di Barcellona.
Dopo averne ricavato la posizione nella stessa struttura "*long-lat*", per ogni alloggio si è calcolata la distanza chilometrica da ognuno dei 14 punti turistici.
La riflessione su come combinare le distanze per far creare un indicatore ha fatto emergere 2 possibili linee di azione.

Indicatore 1: minimo tra le distanze

Per ogni alloggio, date le 14 distanze, se ne è calcolato il minimo e lo si è salvato nella variabile "min_dist". L'indicatore è tanto migliore quanto il valore è più piccolo.
La logica dietro tale costruzione è la seguente: un turista, piuttosto che essere in una posizione equidistante da tutto, potrebbe preferire l'estrema vicinanza rispetto alla zona/monumento/attrazione che preferisce, e nel quale passerebbe più tempo, e accettare di spostarsi per raggiungere le altre.
Analisi successive hanno mostrato che questa non è la logica che i clienti seguono nel valutare la posizione di un alloggio. Per questo motivo, ai fini dell'analisi statistica tale indicatore è stato scartato in favore di "avg_dist".

Indicatore 2: media tra le distanze

Per ogni alloggio si è calcolata la media tra le 14 distanze. L'indicatore è tanto migliore quanto ovviamente la distanza media è minore, che esemplifica una posizione centrale comoda per raggiungere tutte le attrazioni di interesse. Come anticipato in precedenza, la nuova variabile è stata nominata "avg_dist" e inserita in grande misura, e con ottimo successo, nei modelli econometrici proposti.

Un possibile spunto per migliorare potenzialmente *Indicatore 2* sarebbe non utilizzare una media semplice ma una media pesata.
I vari pesi dovrebbero riflettere l'importanza relativa della vicinanza rispetto ad una località. Si propone un esempio banale per rendere più semplice la comprensione: se una particolare località è ben servita da mezzi pubblici di superficie, allora è sicuramente meno critico esserne particolarmente vicini.
Per modellare questo meccanismo di compensazione, si dovrebbe inserire un peso relativo minore (un km di distanza arbitrario da tale attrazione dovrebbe avere un impatto marginale minore sulla distanza media rispetto ad un km ulteriore di distanza rispetto ad una località non facilmente raggiungibile).
Il punto cruciale di tale pratica sarebbe la determinazione dei 14 pesi, che risulta particolarmente difficile in quanto richiederebbe una conoscenza capillare della città e delle abitudini dei turisti.

In Tabella14 si presenta la lista delle località che sono state utilizzate come punti di attrazione focali.
In Figura13, le stesse località sono state posizionate su una mappa di Barcellona.

Località	Codice per mappa
Piazza di Spagna	1
Piazza Cataluna	2
Sagrada Familia	3
La Rambla	4
Casa Battlo	5
Casa Mila	6
Spiaggia di Barcelonetta	7
Boqueria (mercato tipico)	8
Parco Guell	9
Museo Historia	10
Castello Montjuic	11
Fondazione Mirò	12
Quartiere Gracia	13
Palazzo della Musica	14

Tabella 14: Legenda località per indicatore posizione

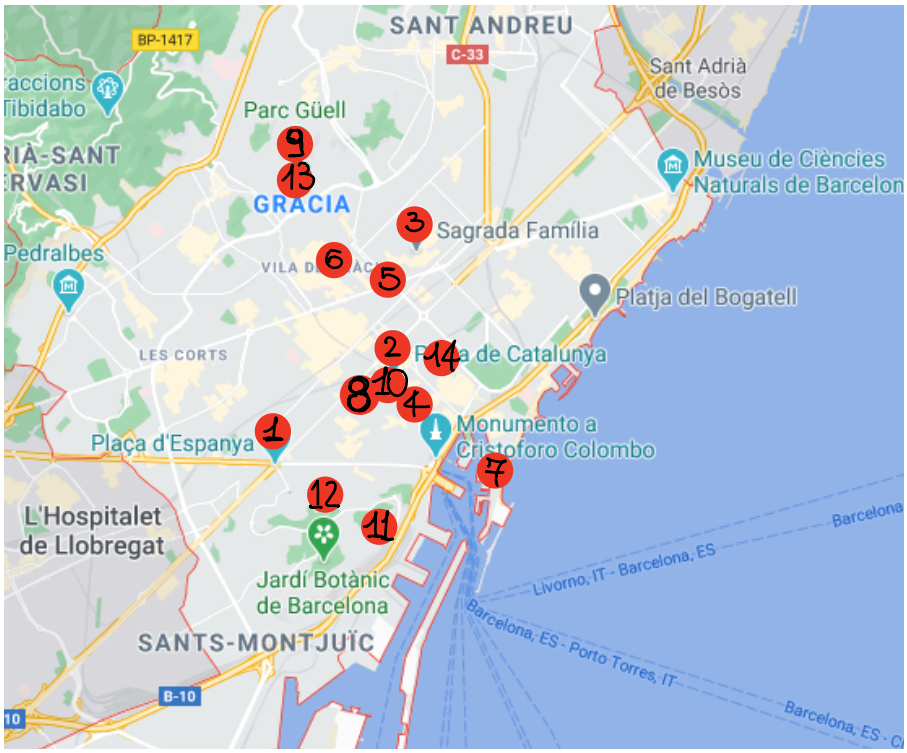


Figura 13: Mappa di Barcellona con località strategiche

5. Esplorazione del database

5.1 Presentazione del database

In questa sezione si presenta la struttura del database finale.
Il database in oggetto è composto da 8758 righe e 43 colonne.
Una parte delle colonne deriva dall'elaborazione del database "reviews" e coglie aspetti relativi alla gentilezza del padrone di casa, le restanti variabili sono state estratte a partire dal database "listings".
Si propone ora una tabella di riepilogo, in cui vengono mostrate tutte le variabili con una breve descrizione. Un'analisi più in dettaglio verrà effettuata nelle sezioni successive.

Nome variabile	Tipo	Descrizione
Listing_id	Stringa	Codice identificativo alloggio
Price	€	Prezzo per notte
Guests_included	Numero	Numero di clienti inclusi nel prezzo
Accommodates	Numero	Numero max di clienti che l'alloggio può accogliere
App_intero	Binario	1 se host affitta un appartamento intero, 0 altrimenti
Bedrooms	Numero	Numero di camere da letto
Beds	Numero	Numero di letti
Bathrooms	Numero	Numero di bagni
Bagno	Binario	1 se l'alloggio possiede almeno 1 bagno privato
Avg_dist	Km	Distanza media rispetto alle attrazioni principali
Host_is_superhost	Binario	1 se Host viene riconosciuto come Superhost dalla piattaforma Airbnb
Host_acceptance rate	%	% di richieste che si traduce in prenotazione
Host_response_rate	%	% di richieste alle quale l'host fornisce una risposta
Host_identity_verified	Binario	1 se l'host ha effettuato con successo la procedura di verifica della propria identità
Cancellation_strict	Binario	1 se il cliente non può cancellare gratuitamente la prenotazione
Security_deposit	€	Importo del deposito cauzionale
Cleaning_fee	€	Importo per servizio di pulizia (opzionale)

Require_guest_profile_picture	Binario	1 se host richiede che il cliente abbia inserito una immagine del profilo per procedere alla prenotazione
Require_guest_phone_verification	Binario	1 se host richiede che il cliente effettui la procedura di verifica del numero di cellulare
Calculated_host_listings_count	Numero	Numero di alloggi posseduti dall'host
Multiprop	Binario	1 se l'host possiede più di un alloggio
Number_of_reviews	Numero	Numero di recensioni relative a tale alloggio
Number_of_reviews_ltm	Numero	Numero di recensioni negli ultimi 12 mesi
Reviews_per_month	Numero	Numero medio di recensioni per mese
Reviews_scores_rating	1-100	Punteggio generato da Airbnb, rating dell'alloggio/host
Reviews_scores_cleanliness	1-10	Punteggio generato da Airbnb, livello di pulizia dell'alloggio
Reviews_scores_checkin	1-10	Punteggio generato da Airbnb, livello di servizio offerto durante il check-in
Reviews_scores_communication	1-10	Punteggio generato da Airbnb, facilità di comunicazione con l'host
Reviews_scores_location	1-10	Punteggio generato da Airbnb, percezione della posizione
Reviews_scores_value	1-10	Punteggio generato da Airbnb, rapporto qualità/prezzo del soggiorno
Confidenza	%	% di recensioni nelle quali il cliente si riferisce all'host utilizzando il suo nome di persona
Freq_neg	%	% di recensioni relative all'host in cui il suo comportamento (gentilezza) viene classificato come negativo
Freq_neutro	%	% di recensioni relative all'host in cui il suo comportamento (gentilezza) viene classificato come neutro
Freq_pos	%	% di recensioni relative all'host in cui il suo comportamento (gentilezza) viene classificato come positivo

Freq_super	%	% di recensioni relative all'host in cui il suo comportamento (gentilezza) viene classificato come super
Rank	3-10	Indice di gentilezza dell'host (primo metodo)
Rank_2	3-10	Indice di gentilezza dell'host (riparametrazione)
Score_1	1-10	Indice di gentilezza dell'host (generato dal tool *SID*)
Score_2	1-10	Indice di gentilezza dell'host (generato dal tool *Polarity*)
Tipo_negativo	Binario	Se 1 individua un host classificato come negativo
Tipo_neutro	Binario	Se 1 individua un host classificato come neutro
Tipo_positivo	Binario	Se 1 individua un host classificato come positivo
Tipo_super	Binario	Se 1 individua un host classificato come super

Tabella 15: Variabili del database finale

5.2 Statistiche descrittive delle singole variabili

In questa sezione si esplora più nel dettaglio l'informazione catturata dalle variabili più critiche, delle quali si mostrano delle statistiche descrittive e si cerca di stimarne la distribuzione.

5.2.1 Prezzo

Valore minimo	Primo quartile	Mediana	Terzo quartile	Valore massimo	Media	Dev_std	*Skewness*	*Kurtosis*
10	40	65	114	1001	94.55	97.7	3.80	24.41

Tabella 16: Statistiche descrittive variabile prezzo

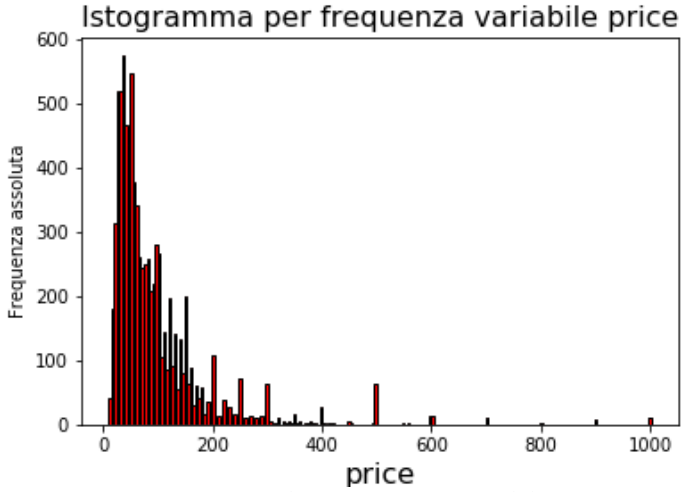


Figura 14:Distribuzione empirica variabile prezzo

Si tratta di una distribuzione molto concentrata intorno al valor medio (lo si può capire dal valore molto alto di *kurtosis*, che prova a cogliere proprio questo aspetto) ma che allo stesso tempo presenta anche valori molto estremi (diversi alloggi presentano un prezzo per notte superiore ai 400 €).
Se a livello statistico una distribuzione di questo tipo potrebbe sembrare atipica, nella realtà è di facile comprensione, in quanto un prezzo particolarmente elevato è compatibile, ad esempio, con un alloggio lussuoso o in una posizione super conveniente.

5.2.2 Guests_included e accommodates

Variabile	Primo quartile	Mediana	Terzo quartile	Valore massimo	Media	*Kurtosis*
Accomodates	2	3	5	20	3.59	6.34
Guests_included	1	1	2	16	2,14	8,56

Tabella 17: Statistiche variabili Accommodates e Guests Included

Il valor medio e la mediana della variabile "Accommodates" danno informazione sulla dimensione media degli appartamenti.
Siccome sono entrambe superiori o uguali alle 3 persone, è ragionevole pensare che una quota cospicua degli alloggi sia rappresentata da appartamenti con almeno 2 stanze, piuttosto che da monolocali.
In modo complementare, il fatto che almeno nel 50% dei casi il prezzo sia espresso per una sola persona, piuttosto che per due, fa pensare che sia molto frequente che venga affittata solo una stanza di un alloggio.
Sarebbe interessante capire se ciò avviene perché il proprietario di casa vive con la famiglia nelle restanti stanze, oppure se le agenzie preferiscano suddividere appartamenti in multiple stanze e affittarle singolarmente (magari perché affittare l'intero appartamento potrebbe risultare più difficile, specie per alloggi particolarmente grandi).

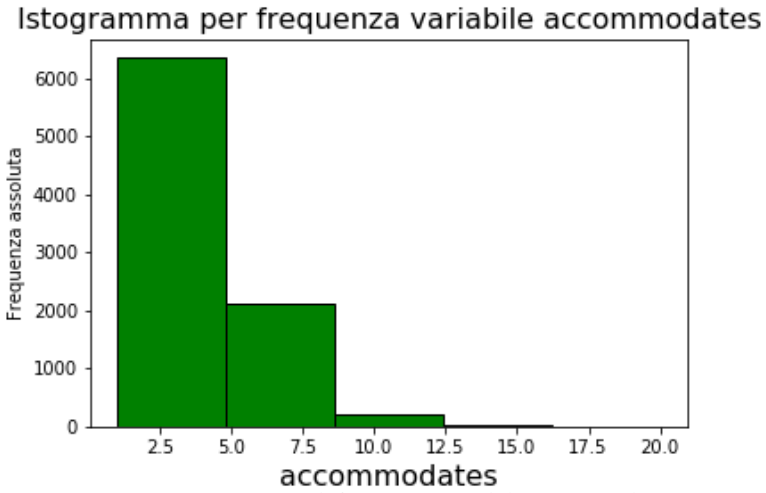


Figura 15:Istogramma di frequenza variabile Accommodates

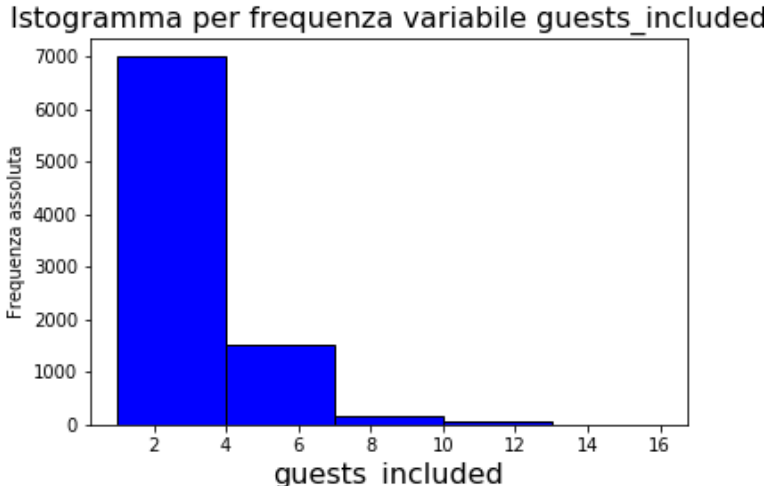


Figura 16: Istogramma di frequenza variabile guests included

I valori massimi delle 2 variabili, pari rispettivamente a 16 e 20 potrebbero far pensare a strutture particolarmente grosse e lussuose, come ad esempio ville, ad alloggi modesti di grande dimensione oppure ad ostelli con grandi camerate (per i quali è possibile affittare non solo un singolo posto letto ma anche una porzione di camerata o addirittura la camerata intera).
Questo dubbio si può facilmente risolvere esplorando la distribuzione dei prezzi per gli alloggi con valore della variabile "Accommodates" superiore alle 10 unità (valore abbondantemente superiore al 75esimo percentile).
Per questo sottoinsieme di 90 alloggi, si è generata una variabile che rappresenta il costo per persona per notte e se ne è plottata su un grafico la distribuzione empirica, raffigurata in Figura17.

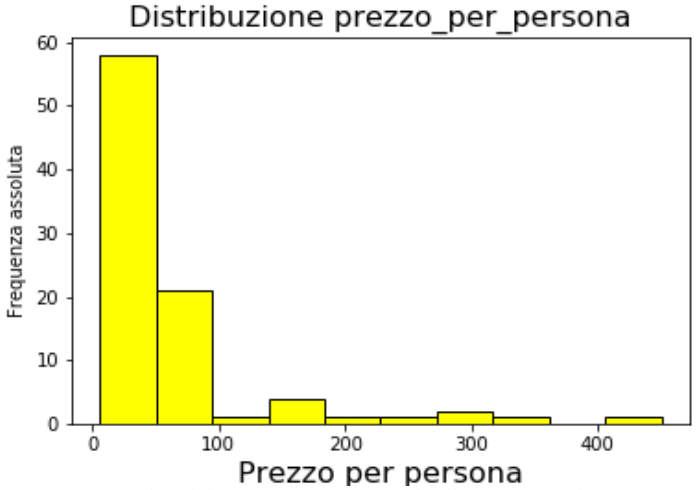


Figura 17: Analisi del prezzo per persona di alloggi particolarmente capienti

Il 75% percentile della distribuzione empirica risulta inferiore ai 65€.
Per questi motivi, è ragionevole pensare che almeno tre quarti degli alloggi grandi non siano lussuosi. La restante parte è costituita molto probabilmente da alloggi di lusso, tali da giustificare un prezzo per persona per notte superiore ai 300 €.

5.2.3 App_intero

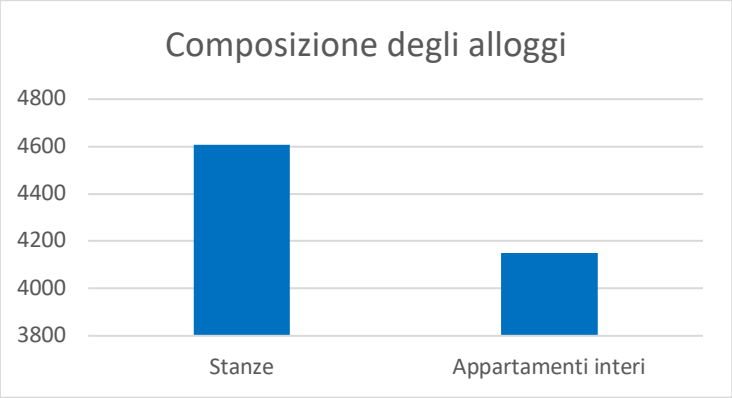


Figura 18:Composizione degli alloggi tra stanze singole e appartamenti

Su un totale di 8758 alloggi, 4150 sono appartamenti completi (pari al 47% del totale) e 4608 sono stanze private.
La variabile "app_intero" risulta particolarmente importante perché permette di discriminare tra 2 tipi di transazioni strutturalmente diverse, che potrebbero essere soggette a meccanismi di equilibrio di prezzo completamente diversi.
Inserire questa variabile in un modello regressivo permette di tenere conto di questo aspetto all'interno dell'analisi.

5.2.4 Bedrooms e beds

Si tratta di variabili molto semplici da interpretare e da includere in modelli statistici.
Posso essere utilizzate come *proxy* della dimensione dell'alloggio, mentre il confronto tra il numero di camere da letto e il numero dei letti può dare delle informazioni sull'abitabilità delle zone giorno (ad esempio la eventuale presenza di divani letti).
In Tabella18 si presentano delle semplici statistiche descrittive delle due variabili.

Variabile	Primo quartile	Mediana	Terzo quartile	Valore massimo	Media
Bedrooms	1	1	2	9	1.63
Beds	1	2	3	20	2,372

Tabella 18:Statistiche descrittive variabili bedrooms e beds

La distribuzione empirica della variabile "bedrooms" rafforza le considerazioni fatte analizzando la variabile "accommodates": il fatto che la media sia compresa tra 1 e 2 alla luce del fatto che almeno il 50% degli alloggi sono singole stanze (e per questo motivo ci si aspetta una unica camera da letto), fa pensare che gli appartamenti completi siano generalmente composti da almeno 2 stanze da letto.
I valori massimi seguono le stesse considerazioni fatte sulla doppia possibilità villa/ostello.

5.2.5 Bagno

Questa variabile binaria è stata creata per indagare la presenza di un bagno privato negli alloggi stanza. La media della distribuzione e il fatto che primo quartile e mediana siano pari a 1 farebbero pensare che la presenza del bagno sia quasi scontata negli alloggi, ma il risultato potrebbe essere inflazionato dalla massiccia presenza di appartamenti interi.
Per questo motivo, si è deciso di isolare un subset costituito solo dagli alloggi stanza e si è indagata la variabile "bagno" solo per questi.

Anche per gli alloggi stanza la media è molto prossima all'unità così come il primo quartile. Gli alloggi senza un bagno privato risultano solo 35: questo valore stupisce un po' data la grande popolarità di ostelli a basso prezzo nelle grandi città turistiche come Barcellona.

5.2.6 Avg_dist

La variabile, espressa in km, fornisce informazioni sulla strategicità della posizione di un alloggio.
Si hanno grandi speranze per questo indicatore, che si immagina sia in grado di cogliere una cospicua parte di varianza della distribuzione dei prezzi di mercato.
In Tabella19 se ne presentano delle statistiche descrittive.

Minimo	25%	50%	75%	Massimo	Media	Dev_std	Skewness
1,5	1,82	2,15	2,83	8,5	2,475	0,94	1,74

Tabella 19: Statistiche descrittive variabile Avg Dist

Dai dati in tabella non si è in grado di dire molto sulla variabile; interessante è il valore positivo della *skewness*, che denota un'asimmetria verso sinistra che genera una coda destra della distribuzione molto lunga.
Ci si immagina che gli alloggi appartenenti alla coda destra della distribuzione siano fortemente penalizzati a livello di prezzo e di domanda che sono in grado di catturare.
Sarà particolarmente interessante valutare se gli alloggi appartenenti al primo quartile, che si trovano in posizioni particolarmente centrali, siano in grado di imporre un premio di prezzo importante.
Per andare ad analizzare ancora più in profondità questo indicatore, Figura19 propone un grafico a istogrammi che prova a stimarne la distribuzione a partire dai dati empirici.

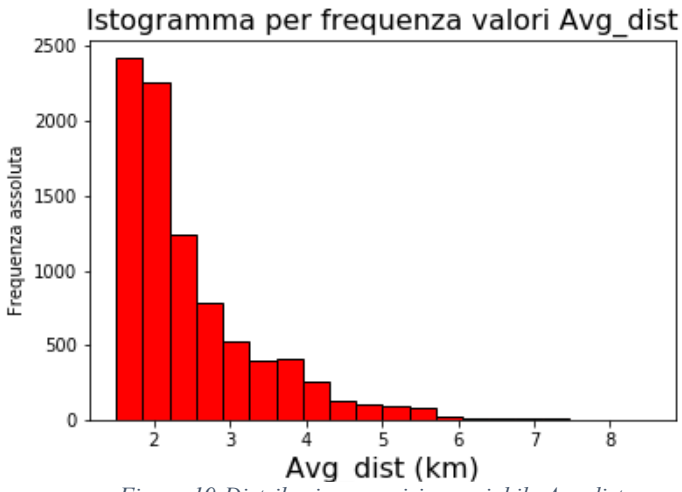


Figura 19:Distribuzione empirica variabile Avg dist

Dal grafico, è quasi immediato ipotizzare una distribuzione iperbolica degli alloggi rispetto alla distanza dal centro.
Una distribuzione di questo tipo sarebbe stata inspiegabile fino a non più di 5 anni fa.
All'albore delle piattaforme come Airbnb e Booking era infatti comune che il centro città fosse occupato dai residenti e che gli alloggi affittabili dai turisti si trovassero nelle zone più periferiche.

Negli ultimi anni, si è verificata una forte inversione di questo trend: le grandi possibilità di guadagno affittando gli alloggi hanno portato ad uno "svuotamento" dei centri città, e al conseguente trasferimento dei residenti nelle periferie.
Questo fenomeno, sicuramente positivo per il turismo, ma meno per la vivibilità della città e per la conservazione della tradizione storica è descritto nel dettaglio nell' articolo: "Come *stanno cambiando le città per colpa di Airbnb*", pubblicato da Gianfrancesco Turano nel dicembre del 2017.

A causa della grande eterogeneità di valori, il grafico così costruito non permette di avere una chiara visione delle code della distribuzione.
Per questo motivo si è deciso di analizzare singolarmente le due code, seguendo un approccio simile a quello applicato nella *Teoria dei valori estremi*.
Partendo dalla coda destra, si è selezionato il 5% dei valori più grandi e se ne è plottata la frequenza assoluta. A livello operativo, ciò è stato effettuato isolando gli alloggi con una distanza media superiore a 4.4 km. In Figura20 si mostra la distribuzione della coda destra così individuata.

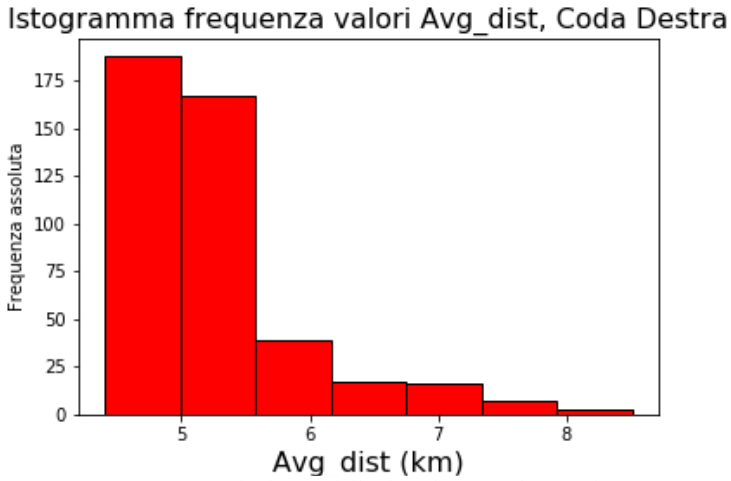


Figura 20:Coda destra della distribuzione di Avg_dist

Più del 50% dei valori si mantiene su valori inferiori ai 5 km. Pur essendo in una posizione sicuramente non invidiabile, tali alloggi sono probabilmente in grado di attirare comunque domanda, a patto di essere in una zona ben servita dai mezzi pubblici.
Circa una 50 di alloggi si trovano ad almeno 7-8 km dal centro; tale distanza sembra particolarmente eccessiva perché questi possano avere successo, a meno di non imporre prezzi stracciati.
È molto probabile che tali alloggi non siano dedicati alla clientela turistica, ma piuttosto a quella business nel caso si trovassero vicino ad aeroporti o zone industriali o sedi di grandi società.

L'analisi della coda sinistra ha restituito valori peculiari.
Sostanzialmente tutto il primo quartile di distribuisce in modo uniforme tra 1.5 km e 1.8 km. Alla luce di questi risultati si è immaginato che tutti questi alloggi abbiamo grossomodo lo stesso vantaggio competitivo a livello di posizione, essendo poco probabile che un turista sia molto sensibile ad una differenza di distanza media di 300 m, dato il fatto che egli già si trova in una posizione molto invidiabile.

5.2.7 Host_is_superhost

Il numero di alloggi certificati *Superhost* è pari a 2739, corrispondente al 31% del totale.
Dalle informazioni disponibili sulla piattaforma non si è riuscito a capire se il titolo di *Superhost* sia assegnato ad un host sulla base di tutti gli alloggi che possiede, oppure se sia relativo ad un host per un singolo alloggio.
La qualifica di *Superhost* potrebbe avere un impatto rilevante su prezzo e tasso di domanda, in quanto è una certificazione affidabile (in quanto assegnata direttamente da Airbnb) sulla professionalità del padrone di casa.
Come nel caso dell'attributo "App_intero", includere una variabile di questo tipo in un modello di regressione lineare è un'operazione delicata, in quanto il modello riesce al più ad individuare un effetto marginale medio della qualifica Superhost sulla variabile dipendente.
Il rischio, in questo senso, è che nella realtà non esista un premio medio di prezzo per i *Superhost*, come eventualmente individuato dal modello lineare, ma piuttosto che il meccanismo di formazione del prezzo di equilibrio segua delle dinamiche che sono completamente diverse per i Superhost rispetto al resto degli alloggi.

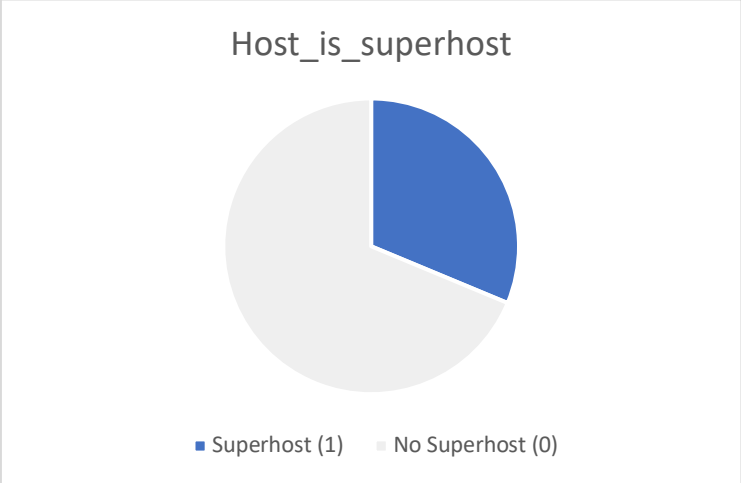


Figura 21: Percentuale di Superhost

5.2.8 Cancellation_Strict

L'attributo binario assume valore 1 se il cliente non può cancellare in modo gratuito la sua prenotazione.
Si pensa che ci possa essere un *malus* di prezzo legato alla presenza di vincoli di cancellazione, se un host vuole limitare la flessibilità del cliente, allora si deve accontentare di un prezzo più basso.
Effettuando il ragionamento speculare, garantire flessibilità nella cancellazione si può considerare un servizio *optional* al quale potrebbe essere associato un *benefit* di prezzo (esattamente con lo stesso meccanismo con cui le compagnie aeree fanno pagare un *surplus* per un posto vicino al finestrino).
A questo livello di analisi, l'unico dato interessante è la frequenza relativa dei valori della variabile; una distribuzione piuttosto equa tra le due possibilità è indispensabile per garantire risultati robusti nei successivi test statistici.
La media, pari a 0,51 è perfetta a questo fine.

5.2.9 Calculated_host_listings_count e Multiprop

La variabile "calculated_host_listings_count è utile per discriminare host amatori, che affittano una parte della loro casa, o possiedono qualche alloggio, da organizzazioni strutturate come ad esempio agenzie.
Un'analisi esplorativa della variabile restituisce una media di 13 alloggi per host a fronte di una mediana pari a 2.
Questo significa che il tipo di host è fortemente polarizzato, gli host amatori sono almeno il 50% e possiedono 1 o 2 alloggi, mentre le agenzie per controbilanciare investono in un numero molto grande di alloggi (almeno il 25% degli host possiede 8 alloggi o più).
Questi dati vengono riassunti in Tabella 20.

25%	50%	75%	Massimo	Media	Dev_std	Skewness
1	2	8	155	12,81	28,44	4.

Tabella 20:Statistiche descrittive sul numero di alloggi per host

Ragionando a priori su quale potesse essere l'impatto sulla qualità del soggiorno di avere a che fare con un host che possiede molti alloggi, sono emerse due linee di pensiero:

- Effetto fiducia: un'organizzazione come un'agenzia tendenzialmente è riconosciuta più affidabile e più efficiente rispetto ad un privato. È dunque possibile che questo effetto reputazione possa giustificare un *delta* di prezzo positivo tra agenzie e host privati. Allo stesso tempo è verosimile anche l'atteggiamento contrario: le agenzie, sostenendo un rischio di mercato diversificato, possono anche permettersi di fare *undercutting* su un numero selezionato di listings, al fine di catturare domanda, sussidiando tale perdita di margine attraverso altri alloggi a ritorno più alto.
- Effetto "freddezza": è ragionevole pensare che un host che possieda un numero elevato di alloggi non sia in grado di garantire un servizio particolarmente empatico nei confronti dei clienti. Nell'ottica di uno studio atto a valorizzare la gentilezza dell'host come elemento di successo, un servizio più standard e meno empatico potrebbe essere la causa di un prezzo di equilibrio più basso.

La variabile "Multiprop" coglie lo stesso aspetto sotto forma di *dummy*.

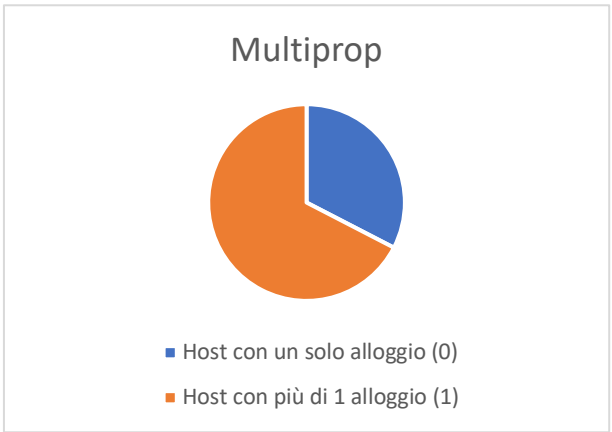


Figura 22: Percentuale di host che possiedono più di un alloggio

In Figura22 si può vedere la composizione della variabile binaria "Multiprop".
Il 67% degli host possiede più di un alloggio, mentre il numero di host che possiede solo un alloggio è pari a solo 2854.

5.2.10 Number_of_reviews, Number_of_reviews_ltm e Reviews_per_month

Si tratta di 3 variabili che colgono aspetti diversi dello stesso fenomeno e possono essere utilizzate come *proxy* del tasso di domanda di un alloggio.
La variabile "number_of_reviews", essendo una variabile aggregata a partire dall'iscrizione dell'alloggio sulla piattaforma, dà un'indicazione della longevità dell'alloggio.
Le considerazioni fatte su *listing* presenti sulla piattaforma da diversi anni tendono ad essere più robuste rispetto all'inferenza su un alloggio attivo da cui pochi mesi, la cui performance potrebbe ancor essere in fase di assestamento.
La variabile "number_of_reviews_ltm" è una forma standardizzata della variabile precedente, in quanto conta le recensioni accumulate negli ultimi 12 mesi.
Questa può essere utilizzata come *proxy* ottima del tasso di domanda, in quando è invariante rispetto al momento dell'iscrizione, e coprendo esattamente 12 mesi, è invariante anche rispetto ad eventuali fenomeni di stagionalità, a meno di periodi di oscillazione estremamente ampi, pari ad esempio ad anni, come nel caso di crisi strutturali.
La variabile "reviews_per_month" è semplicemente una versione media su base mensile del tasso di domanda. Effettuando analisi di correlazione, ci si è resi conto che non risulta perfettamente correlata con nessuna delle due variabili sopra citate, e di conseguenza, anche per la mancanza di informazioni pubbliche al riguardo, risulta particolarmente difficile individuare che tipo di media sia stata applicata e su quale campione di dati.

Variabile	Number_of_reviews	Number_of_reviews_ltm	Reviews_per_month
Min	6	4	0.1
25%	20	10	1,03
50%	48	20	1,83
75%	103	32	2,9
Max	731	319	28
Media	76,37	23,52	2,13
Dev_std	80,88	17,38	1,46
Kurtosis	9,07	26,42	24,82
Skewness	2,12	2,66	2,29

Tabella 21: Statistiche descrittive sul numero di recensioni per ogni alloggio

Come per le variabili già trattate, in Tabella21 si mostrano alcune statistiche descrittive.

Tutte e tre le variabili presentano *skewness* positiva, che esemplifica una asimmetria verso sinistra con code destre molto più allungate.
Il valore di *kurtosis* di "reviews_per_month" molto alto e molto simile a quello di "number_of_reviews_ltm" fa pensare che probabilmente è stata utilizzata quest'ultima variabile come punto di partenza per l'elaborazione dell'indicatore, piuttosto che "number_of_reviews".
In un primo momento ci si era immaginati che il tasso di domanda potesse essere distribuito in modo normale, ma come è chiaro da Figura23, che mostra un istogramma di frequenze

per “reviews_per_month”, si tratta in realtà di una distribuzione fortemente asimmetrica, che ricorda addirittura una curva iperbolica.
La forma della distribuzione empirica di “number_of_reviews”, e di “number_of_reviews_ltm” è stata omessa in quanto molto simile a quella di “reviews_per_month”.

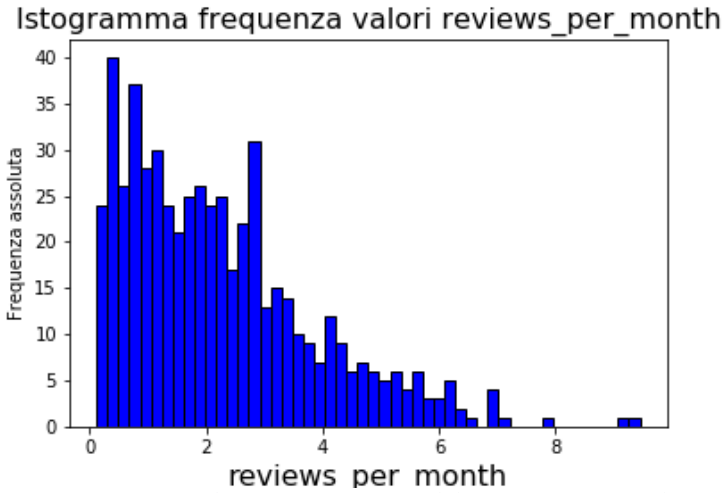


Figura 23: Distribuzione empirica variabile Reviews_per_month

5.2.11 Review_scores_rating, Review_scores_location, Review_scores_value, Review_scores_checking, Review_scores_communication e Review_scores_cleanliness

Si tratta di indicatori pubblicati direttamente da Airbnb e dei quali non si sa esattamente come siano stati costruiti.
Per iniziare ad esplorare questi indicatori, in Tabella 22 se ne mostrano alcune statistiche descrittive.

Variabile	Rating	Value	Location	Communication	Check-in	Cleanliness
Min	46	4	6	6	5	5
25%	89	9	9	9	9	9
50%	93	9	10	10	10	9
75%	96	10	10	10	10	10
Max	100	10	10	10	10	10
Media	91,88	9,135	9,72	9,65	9,64	9,33
Dev_std	6,02	0,68	0,495	0,562	0,6	0,734

Tabella 22: Indicatori generati da Airbnb

Già a partire da questi primi dati si può capire che i 6 indicatori presentano distribuzioni a bassissima varianza, e con valori medi che sono tremendamente vicini al valore massimo consentito (100 per il primo e 10 per i restanti 5).
Per avere una visione più chiara, si procede a produrre un istogramma di frequenze al fine di verificare se effettivamente questi siano poveri di informazione, in quanti infetti da un *bias* inflativo, oppure se nonostante la bassa varianza esistano dei valori estremi o dei trend strutturali.

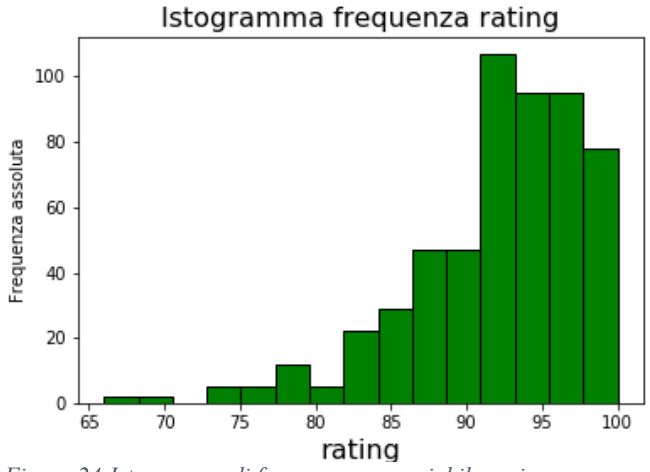


Figura 24:Istogramma di frequenza per variabile review_scores_rating

Da Figura24 si può notare come per l'indicatore "Review_scores_rating", la metà inferiore della scala di valori consentita non sia praticamente popolata.
Inoltre, più del 50% dei valori è concentrata tra il 90 e il 100.
La restante parte dei valori popola, in modo piuttosto omogeneo, il *range* tra 50 e 90.

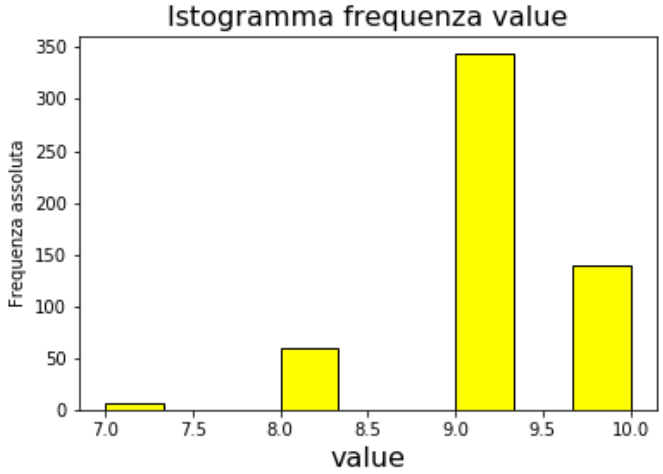
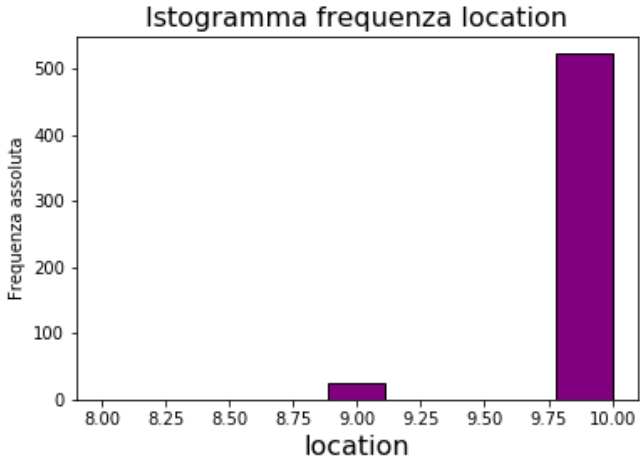


Figura 25: Istogramma di frequenza variabile Review_Scores_Value

L'indicatore "Rewiew_scores_value" risulta poco efficace, in quanto nonostante sia mappato nel continuo tra 1 e 10, nella pratica i valori distinti che assume sono pochissimi. Così come per "Review_scores_rating", anche in questo caso si ipotizza, data la grande massa concentrata intorno al valor massimo, che un bias inflativo distorca i giudizi espressi dai clienti.



Nell'analizzare l'indicatore "Review_scores_location" sono valide le stesse considerazioni fatte per l'indicatore "Review_scores_value", ma addirittura in questo caso la povertà di informazione è ancora più estrema.

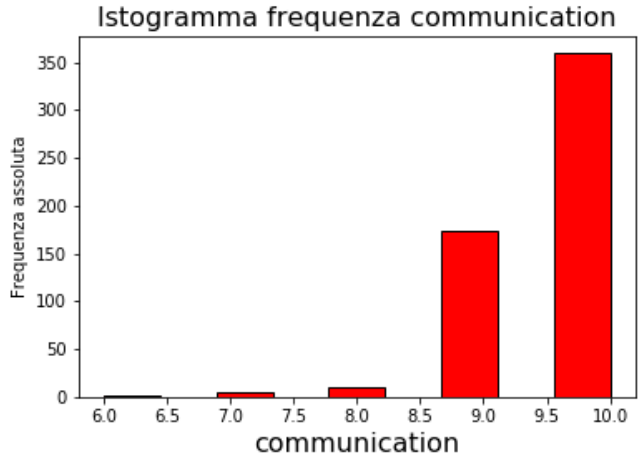


Figura 26:: Istogramma di frequenza variabile Review_Scores_Communication

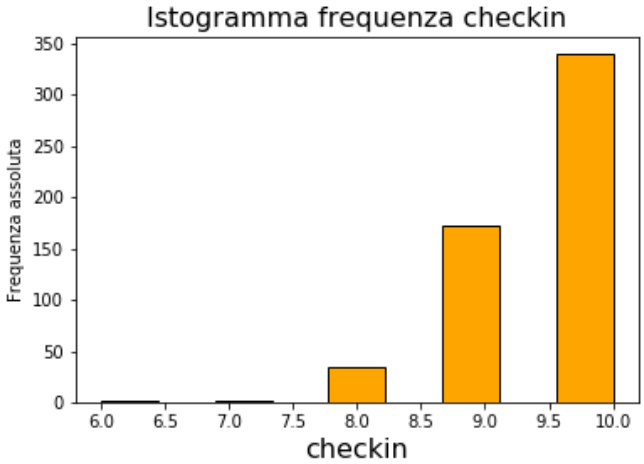


Figura 27:: Istogramma di frequenza variabile Review_Scores_Checkin

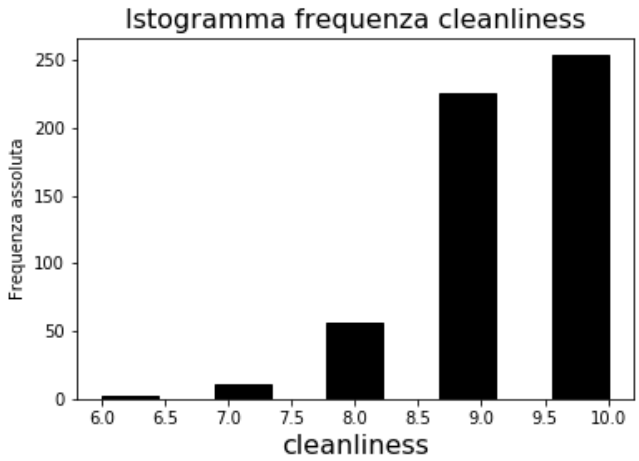


Figura 28: Istogramma di frequenza variabile Review_Scores_Cleanliness

I tre indicatori sopra mostrati, “Review_scores_communication”, “Review_scores_checkin” e “Review_scores_cleanliness” soffrono degli stessi problemi individuati per gli indicatori precedenti.
Il numero di valori distinti è molto ristretto, ma almeno il livello di varianza un po’ più elevato permette di portare qualche tipo di informazione.

6. Relazione prezzo-gentilezza

Una volta elaborata una misura di gentilezza dell'host, si vuole cercare di capire quale sia il suo effetto sulla performance di un determinato alloggio.
La prima relazione che si vuole andare a indagare è quella tra prezzo e gentilezza.
Per farlo si vuole cercare di modellizzare il meccanismo di formazione del prezzo di equilibrio, al fine di capire se la gentilezza sia un determinante del prezzo, e in tale caso in che modo lo influenzi.
Nelle scienze statistiche, ogni qual volta vi sia a disposizione un campione esteso di dati, si ricorre tendenzialmente a modelli di regressione multipla.
Un modello, nella forma più generale, non è altro che un'equazione composta da una variabile dipendente, una serie di variabili esplicative e da coefficienti (o costanti).
Data una forma funzionale, che si ipotizza la migliore per cercare di spiegare la variabile dipendente attraverso le variabili indipendenti, l'approccio regressivo sfrutta un gran numero di osservazioni per stimare i parametri che meglio si adattano a tale modello.
Normalmente la tecnica di ottimizzazione dei parametri è la minimizzazione della somma degli scarti quadratici: si individuano i parametri che minimizzano la somma dei quadrati delle differenze tra i valori osservati della variabile dipendente e i valori predetti attraverso il modello.

6.1 Regressione semplice

Il modello più atomico di regressione è la regressione lineare semplice, in cui si sfrutta una unica variabile indipendente per cercare di spiegare di spiegare la variabile dipendente.
La regressione lineare semplice viene utilizzata quando si ipotizza che esista un effetto lineare tra variabile indipendente e variabile dipendente, e nella pratica significa stimare la migliore retta che leghi le due variabili.

Nel contesto delle scienze economico/sociali, è molto raro che un fattore sia influenzato da una unica variabile, ma trattandosi di una analisi molto semplice, vale la pena di effettuare un tentativo.
Trattandosi di uno studio sull'effetto della gentilezza, è naturale, per prima cosa, provare a legare il prezzo dell'alloggio ad una tra le misure di gentilezza.
Come misura di partenza, si è deciso di utilizzare l'indicatore "rank", generato attraverso il modello ad hoc di *Machine Learning*.

Come software di modellizzazione statistica, si è deciso di utilizzare STATA per la sua popolarità nel mondo accademico; per questo motivo gli output presentati provengono direttamente da esportazioni di videate di tale software.

```
. regress price rank, robust

Linear regression                               Number of obs     =      8,758
                                                F(1, 8756)        =     386.76
                                                Prob > F          =     0.0000
                                                R-squared         =     0.0413
                                                Root MSE          =     95.667

------------------------------------------------------------------------------
             |               Robust
       price |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        rank |  -47.03862   2.391858   -19.67   0.000    -51.72723   -42.35002
       _cons |   455.1386   18.71355    24.32   0.000     418.4556    491.8215
------------------------------------------------------------------------------
```

Trattandosi del primo output mostrato, si spiega brevemente quali siano le informazioni rappresentate.
La sintassi "regress price rank" ipotizza un modello del tipo:

$$price = a + b \text{ x } rank$$

La stima dei parametri restituisce i valori mostrati nella colonna Coef: l'intercetta della retta stimata è 455.1386 e il coefficiente angolare è -47.0386.
Oltre a segno e modulo dei coefficienti è importante verificarne la significatività; ogni qual volta si lanci una regressione, il software include un test statistico per ogni coefficiente stimato, che permette di capire se questo sia diverso da zero in modo robusto.
Il modo più semplice a livello operativo per valutare la significatività di un coefficiente è analizzare il *p-value* mostrato in quarta colonna: quanto più piccolo è questo valore, tanto maggiore è la probabilità che il coefficiente sia diverso da zero.
A scopo puramente esplicativo, si può osservare la prima riga, in cui si effettua un test sul coefficiente della variabile "rank"; essendo il p-value pari a 0, significa che il coefficiente (e in modo transitivo l'effetto della variabile "rank"), è sicuramente significativo.
Un ulteriore indicatore molto utile per guidare le analisi è il cosiddetto *R-squared*, un indicatore mappato nel continuo tra 0 e 1, che è tanto più alto quando il modello è in grado di spiegare i valori osservati nella realtà.
Il valore restituito da questa particolare regressione mostra che il modello è completamente inadatto.

Un modello di regressione lineare semplice potrebbe essere inadatto per almeno 3 motivi (o ovviamente per una loro combinazione):

- La misura scelta non è efficace
- La forma funzionale non è adatta
- La variabile dipendente non è influenzata da un solo fattore (distorsione da fattori omessi)

Per controllare che la misura non sia fallacea si è sostituito "rank" con le altre misure di gentilezza a disposizione: "Score_1", "Score_2" e "Rank_2".
Si ricorda che "Score_1" è una misura di gentilezza che è stata estratta utilizzando un *tool* preconfezionato di Sentiment Analysis chiamato *SID*, mentre Score_2 è stata estratto

tramite il *tool* Polarity. “Rank_2”, invece, deriva da una riparametrizzazione della procedura che ha generato “Rank”.

```
. regress price score_1, robust

Linear regression                               Number of obs     =      8,758
                                                F(1, 8756)        =     687.96
                                                Prob > F          =     0.0000
                                                R-squared         =     0.0699
                                                Root MSE          =     94.227

------------------------------------------------------------------------------
             |               Robust
       price |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     score_1 |  -31.07308   1.184687   -26.23   0.000    -33.39535   -28.75082
       _cons |   297.7604   8.265767    36.02   0.000     281.5576    313.9633
------------------------------------------------------------------------------
```

```
. regress price score_2, robust

Linear regression                               Number of obs     =      8,758
                                                F(1, 8756)        =     611.91
                                                Prob > F          =     0.0000
                                                R-squared         =     0.0627
                                                Root MSE          =     94.59

------------------------------------------------------------------------------
             |               Robust
       price |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     score_2 |  -40.31204   1.629634   -24.74   0.000    -43.50651   -37.11758
       _cons |   341.7757   10.45678    32.68   0.000      321.278    362.2734
------------------------------------------------------------------------------
```

```
. regress price rank_2, robust

Linear regression                               Number of obs     =      8,758
                                                F(1, 8756)        =     458.54
                                                Prob > F          =     0.0000
                                                R-squared         =     0.0556
                                                Root MSE          =     94.951

------------------------------------------------------------------------------
             |               Robust
       price |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      rank_2 |  -45.83633   2.140529   -21.41   0.000    -50.03227   -41.64039
       _cons |   464.6343   17.69059    26.26   0.000     429.9566     499.312
------------------------------------------------------------------------------
```

Essendo i risultati robusti rispetto alla sostituzione della misura “rank” con altre misure generate in modo indipendente l’una dalle altre, si può affermare con confidenza che non si tratti di un problema di inadeguatezza della misura.

Per verificare se esista una distorsione da errata forma funzionale, si può provare a cambiare il modo in cui "*price"* è legato a "rank".

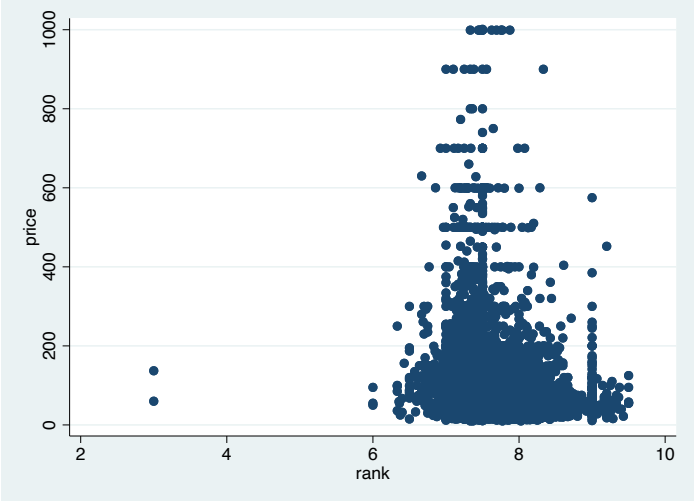


Figura 29:Plot price vs Rank

Avendo osservato empiricamente un andamento che potrebbe sembrare prima crescente e poi decrescente, si è deciso di effettuare un tentativo utilizzando una forma quadratica. La sintassi "qrank" sottointende *quadratic rank*, ovvero la potenza seconda della variabile "rank".

Il modello così ottenuto è del tipo:

$$price = a + b \text{ x } rank + c \text{ x } rank^2$$

```
. regress price rank qrank, robust

Linear regression                               Number of obs     =      8,758
                                                F(2, 8755)        =     220.29
                                                Prob > F          =     0.0000
                                                R-squared         =     0.0417
                                                Root MSE          =     95.651

------------------------------------------------------------------------------
             |               Robust
       price |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        rank |  -116.7833   93.72251    -1.25   0.213    -300.5015    66.93481
       qrank |   4.471197   5.920274     0.76   0.450    -7.133931    16.07632
       _cons |   726.2442   369.7616     1.96   0.050     1.424509    1451.064
------------------------------------------------------------------------------
```

Il basso valore di *R-squared* e la non significatività di entrambi i coefficienti (*p-value* superiori al 20%) mostrano l'inefficacia del modello.

Il tentativo successivo è stato effettuato con una trasformazione logaritmica della variabile indipendente.

Un modello del tipo:

$$price = a + b \text{ x } \log(rank)$$

risulta particolarmente interessante perché permette di valutare in modo immediato quale sia l'impatto in valore assoluto di un aumento percentuale della variabile "rank" (impatto espresso tramite il coefficiente di tale variabile).

```
. regress price lograrank, robust

Linear regression                               Number of obs     =      8,758
                                                F(1, 8756)        =     230.41
                                                Prob > F          =     0.0000
                                                R-squared         =     0.0407
                                                Root MSE          =     95.693

------------------------------------------------------------------------------
             |               Robust
       price |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     lograrank |  -358.7872   23.63667   -15.18   0.000    -405.1207   -312.4538
       _cons |   824.7769   48.41891    17.03   0.000     729.8644    919.6893
------------------------------------------------------------------------------
```

Anche questo output conferma la performance terribile in questa applicazione dei modelli di regressione semplice.
L'analisi è stata ripetuta con altre trasformazioni logaritmiche (log-lineare e log-log) senza un miglioramento dell'esito.

Questi risultati sono in realtà coerenti con quanto ci si aspettava a priori, ovvero che siano svariati i fattori che influenzano in modo importante la formazione del prezzo di equilibrio. È necessario quindi ricorrere a modelli di regressione multipla.

6.2. Modelli di regressione multipla

Come nel caso della regressione semplice, anche nelle implementazioni di regressioni multiple si è soliti partire utilizzando un modello di tipo lineare, per poi procedere introducendo forme funzionali diverse nel caso si ipotizzi che relazioni di tipo non lineare siano più adatte a rappresentare il fenomeno.
In quanto naturale estensione di una regressione semplice lineare, la regressione lineare multipla consiste nello spiegare la variabile dipendente attraverso una combinazione lineare delle variabili esplicative.
Modelli di questo tipo sono di facile interpretazione, in quanto ogni coefficiente della combinazione mostra l'effetto marginale di un aumento unitario della variabile a cui è collegato, a parità di tutte le altre variabili.
Il processo di selezione delle variabili da includere in un modello è un'attività molto strategica e dispendiosa, specie dato un numero di variabili a disposizione molto elevato (più di 40) che genera un insieme delle possibili combinazioni sterminato.

Per questo motivo, si è deciso di omettere, per il momento, la logica di selezione e di mostrare subito uno degli output ottenuti.

```
Linear regression                               Number of obs     =      8,731
                                                F(12, 8718)       =     259.07
                                                Prob > F          =     0.0000
                                                R-squared         =     0.3566
                                                Root MSE          =     78.475

------------------------------------------------------------------------------
             price |      Coef.   Robust Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------------+----------------------------------------------------------------
              rank |  -1.492791   2.041462    -0.73   0.465    -5.494538    2.508956
      accommodates |   11.83267   1.389999     8.51   0.000     9.107947     14.5574
  guests_included |   3.539662   1.240004     2.85   0.004     1.108961    5.970363
          bedrooms |   5.900845   2.409988     2.45   0.014       1.1767    10.62499
              beds |   2.113873   1.741656     1.21   0.225    -1.300184    5.527931
         bathrooms |   19.10468   2.810567     6.80   0.000      13.5953    24.61405
reviews_per_month |  -4.547298   .7103178    -6.40   0.000    -5.939689   -3.154908
          avg_dist |  -10.17991   .7563364   -13.46   0.000    -11.66251   -8.697315
 cancellation_strict |  -4.497593   1.713125    -2.63   0.009    -7.855723   -1.139464
calculated_host_li~t |  -.2601182   .0386702    -6.73   0.000    -.3359209   -.1843156
host_acceptance_rate |   55.08753   5.990082     9.20   0.000     43.34555    66.82951
          app_intero |   26.54462   3.407701     7.79   0.000     19.86472    33.22452
               _cons |  -8.483044   18.28838    -0.46   0.643    -44.33258    27.36649
------------------------------------------------------------------------------
```

In questo modello si è deciso di utilizzare come variabili esplicative:

- Rank
- Accommodates
- Guests_included
- Bedrooms
- Beds
- Bathrooms
- Reviews_per_month
- Avg_dist
- Cancellation_strics
- Calculated_host_listings_count
- Host_acceptance_rate
- App_intero

Indipendentemente dalla scelta delle singole variabili, che verrà spiegata in seguito, il risultato di questo output è importante per svariati motivi:

- Il valore di *R-squared* è sostanzialmente superiore a quello ottenuto attraverso le regressioni semplici e raggiunge valori molto soddisfacenti se confrontati con i valori target del mondo accademico. Questo dato è importante perché fa capire che non si tratta di un fenomeno casuale, ma che invece vi è la possibilità di elaborare una modellizzazione che si avvicini alla realtà

- Gran parte delle variabili hanno valori di significatività molto alta e segni dei coefficienti (e di conseguenza impatti marginali) coerenti con quanto ci si aspetta avvenga nella realtà
- Nonostante una moltitudine di tentativi effettuati, cambiando la combinazione delle variabili inserite e/o la misura di gentilezza utilizzata, non si è riuscito ad ottenere un modello in cui la variabile "rank" risultasse significativa. Questo significa che l'impatto marginale della gentilezza in tale modello potrebbe essere anche nullo, e che di conseguenza non si riesce a capire quale sia l'effetto di tale fattore sul prezzo. Alla luce di questi risultati si è iniziato a pensare che l'impatto della gentilezza probabilmente non è di tipo lineare, e che ci fosse la necessità di esplorare nuove forme funzionali

Come detto in precedenza, non esiste una metodologia standard per il processo di esplorazione delle forme funzionali, tuttavia è sempre una buona idea farsi guidare dall'idea che ci si era fatta a priori sul fenomeno. In fase di definizione della domanda di ricerca per questa tesi, ci si era immaginati che la gentilezza potesse avere un effetto virtuoso sul prezzo e che questo effetto potesse essere più netto in corrispondenza di host a punteggi medio/alti, e che si mitigasse all'ulteriore salire dei punteggi.
Il modo più efficace per rappresentare questo effetto di *smoothing* è l'utilizzo delle trasformazioni logaritmiche.
I tre casi più utilizzati sono:

- Lineare-log in cui si trasforma la variabile indipendente
- Log-lineare in cui si trasforma la variabile dipendente
- Log-log in cui si trasformano entrambe le variabili

```
Linear regression                               Number of obs     =      8,731
                                                F(12, 8718)       =     258.96
                                                Prob > F          =     0.0000
                                                R-squared         =     0.3566
                                                Root MSE          =     78.475

------------------------------------------------------------------------------
             |               Robust
       price |      Coef.  Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     logrank |  -11.62674   15.68468    -0.74   0.459    -42.37241    19.11894
 accommodates |   11.83139   1.390383     8.51   0.000     9.105911    14.55687
guests_included |   3.540488   1.240122     2.85   0.004     1.109556     5.97142
    bedrooms |   5.901685   2.409904     2.45   0.014     1.177703    10.62567
        beds |   2.114879    1.74158     1.21   0.225     -1.29903    5.528788
   bathrooms |   19.10562   2.810386     6.80   0.000      13.5966    24.61463
reviews_per_month |  -4.543403   .7096538    -6.40   0.000    -5.934492   -3.152314
    avg_dist |  -10.17856   .7563642   -13.46   0.000    -11.66122    -8.69591
cancellation_strict |  -4.498051   1.713044    -2.63   0.009    -7.856021   -1.140081
calculated_host_listings_count |  -.2602215   .0387039    -6.72   0.000    -.3360903   -.1843528
host_acceptance_rate |   55.09515   5.992484     9.19   0.000     43.34847    66.84184
    app_intero |   26.54258   3.405387     7.79   0.000     19.86722    33.21795
         _cons |    3.71852   34.04186     0.11   0.913    -63.01156     70.4486
------------------------------------------------------------------------------
```

L'esplorazione della prima tipologia non ha portato a nessun risultato interessante; in qualunque combinazione delle variabili provata l'effetto di gentilezza continuava a non essere significativo.

Diverso è stato l'esito dei tentativi effettuati sfruttando la trasformazione lineare della variabile prezzo.

```
Linear regression                               Number of obs     =      8,731
                                                F(12, 8718)       =     892.05
                                                Prob > F          =     0.0000
                                                R-squared         =     0.5505
                                                Root MSE          =     .50845

------------------------------------------------------------------------------
                             |               Robust
                    logprice |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-----------------------------+------------------------------------------------
                        rank |   .0461113   .0142321     3.24   0.001     .0182129    .0740097
                 accommodates |   .1257043   .0086636    14.51   0.000     .1087216    .1426869
             guests_included |   .0192306   .0061962     3.10   0.002     .0070846    .0313766
                    bedrooms |   .0140198   .0147708     0.95   0.343    -.0149345    .0429741
                        beds |   .0006862   .0109003     0.06   0.950     -.020681    .0220534
                   bathrooms |   .0445143   .0164164     2.71   0.007     .0123342    .0766944
           reviews_per_month |  -.0255023   .0045192    -5.64   0.000     -.034361   -.0166436
                    avg_dist |  -.1176005   .0057888   -20.32   0.000    -.1289479   -.1062531
          cancellation_strict |  -.0408701   .0109932    -3.72   0.000    -.0624193   -.0193208
calculated_host_listings_count |  -.0013624   .0002176    -6.26   0.000    -.0017889   -.0009358
          host_acceptance_rate |    .553516   .0499737    11.08   0.000     .4555557    .6514762
                    app_intero |   .5166815   .0197607    26.15   0.000     .4779459    .5554172
                         _cons |   2.915952    .128656    22.66   0.000     2.663756    3.168148
------------------------------------------------------------------------------
```

In un modello di questo tipo, il coefficiente della variabile "rank" rappresenta l'incremento percentuale della variabile prezzo in conseguenza di un aumento unitario della variabile "rank".
Finalmente, sfruttando questa trasformazione logaritmica emerge in modo altamente significativo (*p-value* praticamente nullo) l'impatto virtuoso della gentilezza dell'host.
A sostegno della validità del modello è il fatto che anche le variabili di controllo mantengono alti valori di significatività, per la maggior parte, e soprattutto segni degli impatti marginali coerenti con la realtà empirica.

Nonostante l'indubbia efficacia del modello sopra mostra mostrato, i risultati migliori, e soprattutto più robusti al cambiamento di variabili di controllo e misura di gentilezza, sono stati ottenuti effettuando una trasformazione logaritmica non solo della variabile prezzo ma anche della variabile "rank".

La struttura del modello risulta del tipo:

$$\log(price) = a + b \times \log(rank) + c \times "variabile\ di\ controllo"$$

Anche modelli di questo tipo sono di interpretazione molto intuitiva:

- Il coefficiente b rappresenta l'incremento percentuale della variabile prezzo, in relazione ad un incremento percentuale della variabile "rank"
- Il coefficiente c, invece, rappresenta l'incremento percentuale del prezzo, in seguito ad un incremento unitario di una generica variabile di controllo.

6.3. Processo di modellizzazione finale

Forti della scelta della struttura ottima per il modello, lo step successivo è stata la selezione delle varabili di controllo da includere.

Per la presentazione della logica di inserimento si è deciso di seguire un approccio di tipo sequenziale: le variabili sono state raggruppate in base all'aspetto che cercano di cogliere e si è incluso un aspetto per volta, anche per avere un'idea di massima di quanto sia il contributo di ogni singolo fattore nella spiegazione della varianza totale dei prezzi.

6.3.1 Capienza dell'alloggio

Il primo aspetto che si è cercato di cogliere è stata la capienza dell'alloggio.
Le variabili a disposizione sono 2:

- Guests_included: numero di persone incluse nel prezzo mostrato sulla piattaforma
- Accommodates: numero massimo di persone che l'alloggio può accogliere

Il senso comune, e l'alto livello di correlazione tra prezzo e ognuna delle due variabili ha fatto ipotizzare che l'effetto capienza avesse un impatto positivo importante sul prezzo.

	price	guests~d	accomm~s
price	1.0000		
guests_inc~d	0.4402	1.0000	
accommodates	0.5633	0.7115	1.0000

Figura 30: Correlazione tra prezzo e indicatori di capienza alloggio

Non restava che capire se le variabili fossero intercambiabili, se ce ne fosse una che meglio spiega il fenomeno oppure se fosse necessario includerle entrambe.
L'alto livello di correlazione tra le due variabili (coefficiente di correlazione pari a 0.7115) mostra che queste cercano di cogliere essenzialmente lo stesso aspetto.
A livello operativo includere in un modello variabili fortemente correlate, espone a problemi di multicollinearità che potenzialmente creano distorsioni nei valori dei coefficienti e sul loro livello di significatività.
Dovendo scegliere quale variabile inserire, si è scelto "Accommodates" in quanto, in tutti i modelli proposti, garantisce un livello di spiegazione del fenomeno marginalmente superiore rispetto a "guests_included".
La scelta non è particolarmente strategica, in quanto, anche nel modello finale, non si registrano scostamenti rilevanti inter cambiando le due variabili o inserendole entrambe.
Per completezza si mostra anche il modello con la variabile "guests_included" utilizzata singolarmente, e quello con entrambe le variabili esplicative.

```
. regress logprice logrank accommodates, robust

Linear regression                               Number of obs     =      8,758
                                                F(2, 8755)        =    2055.49
                                                Prob > F          =     0.0000
                                                R-squared         =     0.4614
                                                Root MSE          =     .55615

------------------------------------------------------------------------------
             |               Robust
    logprice |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     logrank |  -.5478679   .1156368    -4.74   0.000    -.7745432   -.3211925
accommodates |   .2207486    .004054    54.45   0.000     .2128019    .2286954
       _cons |   4.553669   .2431399    18.73   0.000     4.077058    5.03028
------------------------------------------------------------------------------
```

```
. regress logprice lograng guests_included, robust

Linear regression                               Number of obs     =      8,758
                                                F(2, 8755)        =    1316.26
                                                Prob > F          =     0.0000
                                                R-squared         =     0.2870
                                                Root MSE          =      .6399

------------------------------------------------------------------------------
             |               Robust
    logprice |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     lograng |  -1.649894   .1415015   -11.66   0.000    -1.92727   -1.372518
guests_incl~d|   .2130194   .0050175    42.46   0.000     .2031838    .2228549
       _cons |   7.134102   .2933377    24.32   0.000     6.559091    7.709113
------------------------------------------------------------------------------
```

```
. regress logprice lograng accommodates guests_included, robust

Linear regression                               Number of obs     =      8,758
                                                F(3, 8754)        =    1643.31
                                                Prob > F          =     0.0000
                                                R-squared         =     0.4645
                                                Root MSE          =     .55459

------------------------------------------------------------------------------
             |               Robust
    logprice |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     lograng |  -.4941334   .1130087    -4.37   0.000    -.715657   -.2726099
accommodates |   .2024968   .0061023    33.18   0.000     .1905349    .2144587
guests_incl~d|   .0343411   .0066169     5.19   0.000     .0213703    .0473118
       _cons |   4.436474     .23715    18.71   0.000     3.971604    4.901344
------------------------------------------------------------------------------
```

Come ci si aspettava, un aumento del numero di persone ospitabili nell'appoggio porta ad un aumento super significativo del prezzo.
Il ruolo della gentilezza, a questo livello, è ancora difficile da spiegare, ma ci si aspetta che l'impatto vero emerga man mano che si includano nuovi aspetti, che permettano di ridurre la distorsione da fattori omessi, che può portare ad una non significatività dei risultati, o addirittura ad inversioni di segno nei coefficienti.

6.3.2 Elementi strutturali dell'alloggio

Il passo successivo è stato includere variabili che dessero informazioni sulla struttura dell'appartamento. A questo fine si avevano a disposizione 3 variabili:

- Bedrooms: numero di camere da letto
- Beds: numero di letti
- Bathrooms: numero di bagni

Come nel caso precedente, ci si aspetta che ognuna di queste variabili abbia un impatto positivo sul prezzo.
Ancora prima di analizzare le correlazioni tra queste 3 variabili, si pensava di voler includere sicuramente "bathrooms", accompagnata da una tra le rimanenti variabili. Questo perché si pensava che il numero dei bagni fosse portatore di informazione sostanzialmente diversa rispetto a quella rappresentata dalle variabili sulla zona notte.

D’altro canto, inserire la variabile “beds”, data la presenza nel modello di “bedrooms”, non avrebbe aggiunto alta informazione, se non quella sulla dimensione media delle camere da letto.

```
. pwcorr price bathrooms bedrooms beds

             |    price bathro~s bedrooms     beds
-------------+------------------------------------
       price |   1.0000
   bathrooms |   0.3704   1.0000
    bedrooms |   0.5136   0.5298   1.0000
        beds |   0.5055   0.5035   0.8180   1.0000
```

Figura 31: Correlazione tra prezzo e variabili strutturali

L’alto livello di correlazione tra la variabile bedrooms e beds (0,818) e la comparazione delle performance dei vari modelli ha confermato la correttezza del piano di lavoro iniziale.

```
. regress logprice logrank accommodates bathrooms bedrooms,robust

Linear regression                               Number of obs     =      8,754
                                                F(4, 8749)        =    1188.32
                                                Prob > F          =     0.0000
                                                R-squared         =     0.4627
                                                Root MSE          =     .55559

------------------------------------------------------------------------------
             |               Robust
    logprice |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     logrank |  -.5463963   .1155892    -4.73   0.000    -.7729784   -.3198143
accommodates |   .2114026   .0074818    28.26   0.000     .1967365    .2260687
   bathrooms |  -.0375376   .0192238    -1.95   0.051    -.0752207    .0001454
    bedrooms |   .0382449   .0151035     2.53   0.011     .0086384    .0678513
       _cons |   4.571864   .2431999    18.80   0.000     4.095135    5.048593
------------------------------------------------------------------------------
```

Coerentemente con quanto ci si aspettava, il numero di stanze da letto ha un impatto positivo sul prezzo.
Di difficile interpretazione è invece il segno negativo della variabile “bathrooms”: come per la misura di gentilezza ci si immagina che questo sia influenzato dalla residua omissione di fattori rilevanti.

Come ultimo elemento legato alla struttura dell’alloggio, si immagina che ci sia un premio di prezzo nell’affittare un intero appartamento rispetto ad una sola stanza: per cogliere questo aspetto si è incluso nel modello una *dummy* “app_intero” che assume valore 1 se l’host affitta un appartamento intero.
Per come è stata costruita, ci si aspetta un coefficiente positivo per tale variabile.

```
. regress logprice lograk accommodates bathrooms bedrooms app_intero ,robust

Linear regression                               Number of obs     =      8,754
                                                F(5, 8748)        =    1844.91
                                                Prob > F          =     0.0000
                                                R-squared         =     0.5220
                                                Root MSE          =     .52408

------------------------------------------------------------------------------
             |               Robust
    logprice |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     lograk |   .1823674   .1094445     1.67   0.096    -.0321695    .3969043
accommodates |   .1356526    .007353    18.45   0.000     .1212389    .1500663
   bathrooms |   .0596915   .0176618     3.38   0.001     .0250702    .0943128
    bedrooms |   .0130531   .0131199     0.99   0.320     -.012665    .0387713
  app_intero |   .5282021   .0201376    26.23   0.000     .4887276    .5676766
       _cons |   3.021569   .2286474    13.21   0.000     2.573367    3.469772
------------------------------------------------------------------------------
```

Come ci si aspettava, il coefficiente di "app_intero" è positivo, molto elevato (è l'impatto marginale massimo tra quello delle variabili espresse in forma assoluta) e altamente significativo.
Ma non solo, l'inclusione di questa variabile, ha causato l'inversione del segno del coefficiente della variabile "bathrooms", riportando il significato dell'impatto marginale in linea con il senso comune.
Ancora più importante ai fini dello studio, è l'effetto sulla trasformata logaritmica del "rank": l'impatto marginale della gentilezza è finalmente emerso come positivo.
Risulta ancora difficile fare delle considerazioni sul modulo di tale coefficiente.

6.3.3 Effetto posizione

Si è ritenuto strategico includere un indicatore sulla posizione dell'alloggio.
Per questo motivo, si è inclusa la variabile "avg_dist": trattandosi di una misura di distanza media espressa in km.
Se la distanza è effettivamente un elemento importante per i turisti di Barcellona, ci si aspetterebbe un segno negativo del relativo coefficiente.

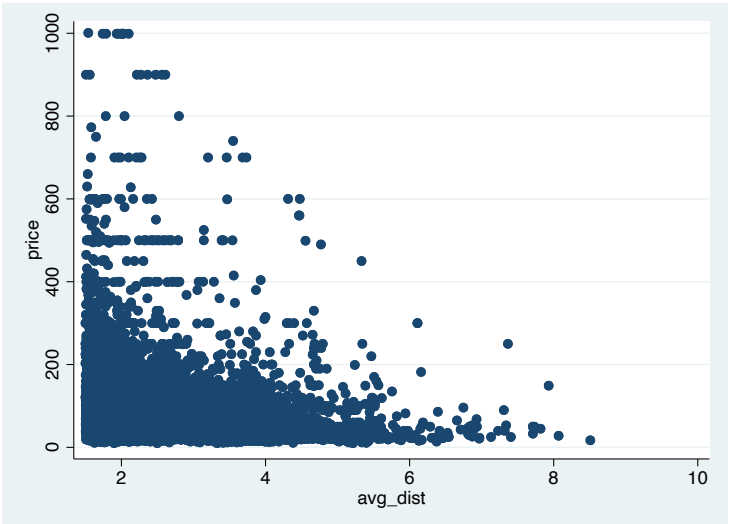


Figura 32: Plot Price vs Avg_dist

La distribuzione empirica dei prezzi in relazione alla distanza, e la correlazione negativa tra le due variabili (coefficiente di correlazione pari a -0.1526), sembrano avvalorare le ipotesi fatte.

```
Linear regression                               Number of obs     =      8,754
                                                F(6, 8747)        =    1680.42
                                                Prob > F          =     0.0000
                                                R-squared         =     0.5408
                                                Root MSE          =     .51369

------------------------------------------------------------------------------
             |               Robust
    logprice |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     logrank |   .2466265   .1074994     2.29   0.022     .0359024    .4573505
accommodates |    .131168    .007178    18.27   0.000     .1170975    .1452386
   bathrooms |   .0460754   .0172218     2.68   0.007     .0123167    .0798341
    bedrooms |   .0222088   .0128706     1.73   0.084    -.0030205    .0474382
  app_intero |   .5176776   .0197698    26.19   0.000     .4789242     .556431
    avg_dist |  -.1119891   .0057341   -19.53   0.000    -.1232293    -.100749
       _cons |   3.192349   .2256093    14.15   0.000     2.750102    3.634597
------------------------------------------------------------------------------
```

L'output del modello mostra che l'indicatore lavora in modo eccezionale, ottenendo un valore di significatività molto elevato (come mostrato dal quantile t pari a -19.53).
Il segno è coerente con quanto ci si aspetta dall'atteggiamento di un turista, e non solo, avg_dist ha un effetto virtuoso anche sulle altre variabili incluse, rafforzando la significatività del coefficiente di "bedrooms" e portando il *p-value* della misura di gentilezza quasi al 98%.

6.3.4 Relazione inversa prezzo/tasso di domanda

L'effetto successivo che si è voluto includere è l'impatto del tasso di domanda sul prezzo. Come spiegato nel capitolo precedente, si avevano a disposizione 3 variabili usabili come *proxy* del tasso di domanda:

- Number_of_reviews
- Number_of_reviews_ltm (recensioni negli ultimi 2 mesi)
- Reviews_per_month (media per mese)

Nonostante le possibili distorsioni create dalla non conoscenza del numero di notti per ciascuna prenotazione, che sarebbe stato un indicatore importante per pesare le recensioni (distorsione che avrebbe comunque impattato allo stesso modo tutte e 3 le variabili), si è deciso di includere nel modello "reviews_per_month".
Alla luce delle principali teorie di equilibrio economico ci si aspetta che la relazione tra prezzo e domanda sia negativa.

```
Linear regression                               Number of obs     =      8,754
                                                F(7, 8746)        =    1443.41
                                                Prob > F          =     0.0000
                                                R-squared         =     0.5415
                                                Root MSE          =     .51334

------------------------------------------------------------------------------
             |               Robust
   logprice |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     logrank |   .2430511   .1076041     2.26   0.024     .0321218    .4539803
accommodates |    .131424   .0071776    18.31   0.000     .1173543    .1454937
   bathrooms |   .0452016   .0172226     2.62   0.009     .0114413     .078962
    bedrooms |   .0208033   .0129115     1.61   0.107    -.0045063     .046113
  app_intero |   .5137652   .0198846    25.84   0.000     .4747868    .5527436
    avg_dist |  -.1141305   .0057863   -19.72   0.000    -.1254731    -.102788
reviews_per_~h |  -.0136277   .0042121    -3.24   0.001    -.0218843   -.0053711
       _cons |    3.23831   .2263289    14.31   0.000     2.794652    3.681967
------------------------------------------------------------------------------
```

6.3.5 Cancellazione gratuita

Si voleva inoltre verificare se esistesse una relazione tra il prezzo e la flessibilità di cancellazione. Per fare ciò si è cercato di includere nel modello la *dummy* "cancellation_strict" che assume i seguenti valori:

- 0: se il cliente ha flessibilità nel cancellare la sua prenotazione gratuitamente
- 1: se il padrone di casa è molto rigido, ovvero non c'è possibilità di cancellazione

```
Linear regression                               Number of obs     =      8,754
                                                F(8, 8745)        =    1266.74
                                                Prob > F          =     0.0000
                                                R-squared         =     0.5421
                                                Root MSE          =     .51303

------------------------------------------------------------------------------
             |               Robust
   logprice |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     logrank |   .2407786   .1073203     2.24   0.025     .0304056    .4511516
accommodates |   .1316561   .0071637    18.38   0.000     .1176135    .1456987
   bathrooms |   .0465658   .0172767     2.70   0.007     .0126994    .0804323
    bedrooms |   .0202348   .0128902     1.57   0.117    -.0050329    .0455026
  app_intero |   .5138826   .0198763    25.85   0.000     .4749204    .5528449
    avg_dist |  -.1158136    .005806   -19.95   0.000    -.1271948   -.1044324
reviews_per_~h |  -.0135272   .0042029    -3.22   0.001    -.0217659   -.0052886
cancellation~t |   -.037257   .0110163    -3.38   0.001    -.0588515   -.0156624
       _cons |   3.264139   .2260608    14.44   0.000     2.821007    3.707272
------------------------------------------------------------------------------
```

Il segno negativo di tale coefficiente è interpretabile in due modi:

- La "flessibilità" è un servizio extra rispetto alla baseline di impossibilità di cancellazione, se il cliente vuole assicurarsi questo tipo di servizio, deve essere disposto a pagare un premio di prezzo.
- Il fatto di ridurre la flessibilità di cancellazione è una forma di assicurazione che il padrone di casa impone per salvaguardare il proprio investimento. Per potersi garantire un tasso di domanda adeguato nonostante questo costo ombra per il cliente, è costretto ad abbassare il prezzo dell'alloggio (la scelta di rigidità ha un effetto malus sul prezzo)

6.3.6 Tasso di accettazione delle richieste di prenotazione

Un ulteriore aspetto legato alla procedura di prenotazione, è la probabilità data una richiesta di prenotazione, che il padrone di casa confermi la prenotazione.
Valutarne a priori l'effetto sul prezzo è una operazione sottile, in quanto l'impatto non è così esplicito. Si può però pensare, specie nel caso di prenotazioni last minute, che la sicurezza che la propria richiesta venga accettata possa essere un elemento ben visto dai clienti, che potrebbero essere disposti a pagare un premio di prezzo.

```
Linear regression                               Number of obs     =      8,743
                                                F(9, 8733)        =    1174.14
                                                Prob > F          =     0.0000
                                                R-squared         =     0.5480
                                                Root MSE          =     .50971

------------------------------------------------------------------------------
             |               Robust
    logprice |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     logrank |    .435765   .1073828     4.06   0.000     .2252695    .6462605
accommodates |   .1316693   .0069804    18.86   0.000     .1179862    .1453524
   bathrooms |   .0465232   .0167655     2.77   0.006     .0136589    .0793874
    bedrooms |   .0208675   .0125865     1.66   0.097    -.003805      .04554
  app_intero |   .5047166   .0195511    25.82   0.000     .4663918    .5430414
    avg_dist |  -.1156847   .0057599   -20.08   0.000    -.1269754   -.104394
reviews_per_~h |  -.0214238   .0043633    -4.91   0.000    -.0299769   -.0128708
cancellation~t |  -.0375285   .0109461    -3.43   0.001    -.0589853   -.0160716
host_acceptan~e |   .5446325   .0509823    10.68   0.000     .4446953    .6445697
       _cons |    2.37152   .2333466    10.16   0.000     1.914106    2.828934
------------------------------------------------------------------------------
```

Effettivamente l'output conferma la presenza di un bonus legato all'effetto "sicurezza".

6.3.7 Confronto tra host amatore e agenzia

Ci si è chiesti se esistesse una differenza nel meccanismo di formazione del prezzo tra host amatori e organizzazioni più strutturate. Nel capitolo precedente si erano ipotizzati due possibili tipologie di effetti sul prezzo:

- Effetto "reputazione"
- Effetto "freddezza"

Per cercare di comprendere quale dei due effetti prevalesse si è cercato di aggiungere al modello una variabile che rispecchiasse la differenza tra queste tipologie di host.
La variabile "calculated_host_listings_count", che rappresenta il numero di alloggi posseduti da un certo host, sembra perfetta a questo fine, in quanto tendenzialmente host amatori possiedono pochi annunci, mentre invece organizzazioni come agenzie hanno grandi investimenti in real estate.
In alternativa a "calculated_host_listings_count", la *dummy* "multiprop" cerca di cogliere esattamente lo stesso aspetto in forma compatta.
Per scegliere quale modello preferire ci si è fatti guidare dalla performance mostrate nell'output.

```
Linear regression                               Number of obs     =      8,743
                                                F(10, 8732)       =    1059.77
                                                Prob > F          =     0.0000
                                                R-squared         =     0.5480
                                                Root MSE          =     .50972

------------------------------------------------------------------------------
             |               Robust
    logprice |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     logrank |   .4146991   .1088265     3.81   0.000     .2013736    .6280246
accommodates |   .1318577   .0069745    18.91   0.000     .1181862    .1455293
   bathrooms |   .0472413   .0168205     2.81   0.005     .0142692    .0802134
    bedrooms |   .0205238   .0125836     1.63   0.103    -.0041431    .0451907
  app_intero |   .5049637   .0195713    25.80   0.000     .4665993     .543328
    avg_dist |  -.1161585   .0057795   -20.10   0.000    -.1274877   -.1048293
reviews_per_~h |  -.0216868   .0043913    -4.94   0.000    -.0302947   -.0130788
cancellation~t |  -.0373908   .0109573    -3.41   0.001    -.0588697   -.0159119
host_acceptan~e |    .549259   .0511205    10.74   0.000     .4490507    .6494672
   multiprop |  -.011087   .0112141    -0.99   0.323    -.0330693    .0108952
       _cons |   2.417962   .2363288    10.23   0.000     1.954702    2.881222
------------------------------------------------------------------------------
```

```
Linear regression                               Number of obs     =      8,743
                                                F(10, 8732)       =    1062.22
                                                Prob > F          =     0.0000
                                                R-squared         =     0.5494
                                                Root MSE          =     .50893

------------------------------------------------------------------------------
             |               Robust
    logprice |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     logrank |   .3391617   .1086764     3.12   0.002     .1261305     .552193
accommodates |   .1331541   .0070319    18.94   0.000       .11937    .1469382
   bathrooms |   .0470565   .0167028     2.82   0.005     .0143151    .0797979
    bedrooms |   .0185592   .0126064     1.47   0.141    -.0061524    .0432707
  app_intero |   .5193937   .0198457    26.17   0.000     .4804914     .558296
    avg_dist |  -.1179446   .0057892   -20.37   0.000    -.1292928   -.1065964
reviews_per_~h |  -.0255727   .0045232    -5.65   0.000    -.0344392   -.0167063
cancellation~t |  -.0410078   .0109987    -3.73   0.000    -.0625678   -.0194478
host_acceptan~e |   .5656876   .0508136    11.13   0.000     .4660811    .6652942
calculated_ho~t |  -.001126   .0001942    -5.80   0.000    -.0015066   -.0007454
       _cons |   2.569622   .2357699    10.90   0.000     2.107457    3.031786
------------------------------------------------------------------------------
```

L'output mostra come il prezzo sia decrescente (anche se il moltiplicatore è molto piccolo) rispetto al numero di alloggi posseduti. Alla luce di questo risultato sembra prevalere l'effetto freddezza: le agenzie non sono in grado di garantire un servizio personalizzato e di conseguenza sono costrette ad adeguare il prezzo a tale mancanza.
Allo stesso tempo, è anche possibile che il prezzo più basso derivi da una politica di *undercutting* e *cross subsidization*, ma purtroppo non c'è modo di testare tale teoria.

La variante con "multiprop" non sembra "performare" allo stesso modo del modello con la variabile estesa, permane il segno negativo coerente con l'impatto marginale, ma la variabile perde fortemente di significatività.

Come forma finale del modello, per questo motivo, si è scelta la variante con le seguenti variabili.

- Logrank
- Accommodates
- Bathrooms
- Bedrooms
- App_intero
- Avg_dist
- Reviews_per_month
- Cancellation_strict
- Host_acceptance rate
- Calculated_host_listings_count

6.4. Analisi di robustezza

Si vuole verificare che il modello sia robusto rispetto a cambiamenti della misura di gentilezza. Se la modellizzazione è effettivamente accurata, e le varie misure di gentilezza cercano di cogliere lo stesso aspetto, allora ci si aspetterebbe di ottenere risultati simili sostituendo con "rank" con ognuna delle altre 3 alternative.
La prima verifica è stata effettuata utilizzando "rank_2".

Trattandosi di una riparametrazione abbastanza sottile, ottenere un output sostanzialmente diverso sarebbe un segnale piuttosto preoccupante di inadeguatezza del modello proposto.

```
Linear regression                               Number of obs     =      8,743
                                                F(10, 8732)       =    1061.17
                                                Prob > F          =     0.0000
                                                R-squared         =     0.5491
                                                Root MSE          =      .5091

------------------------------------------------------------------------------
             |               Robust
    logprice |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    logrank_2 |   .1884025     .10615     1.77   0.076    -.0196766    .3964815
 accommodates |   .1325997    .007036    18.85   0.000     .1188075    .1463919
    bathrooms |   .0469525   .0167159     2.81   0.005     .0141854    .0797197
     bedrooms |   .0190835   .0126185     1.51   0.130    -.0056517    .0438188
   app_intero |   .5175449   .0199804    25.90   0.000     .4783787    .5567111
     avg_dist |  -.1179888   .0057941   -20.36   0.000    -.1293466    -.106631
reviews_per_month |  -.0256569   .0045231    -5.67   0.000    -.0345233   -.0167905
cancellation_strict |  -.0411142   .0110111    -3.73   0.000    -.0626986   -.0195299
host_acceptance_rate |   .5571612   .0507031    10.99   0.000     .4577713    .6565512
calculated_host_listings_count |  -.0011469    .000198    -5.79   0.000     -.001535   -.0007588
         _cons |   2.877598   .2368927    12.15   0.000     2.413232    3.341963
------------------------------------------------------------------------------
```

Fortunatamente, i risultati non sono così diversi a parte una po' di perdita di significatività del coefficiente di "logrank_2" che è comprensibile, dato che l'ottimizzazione del modello non è stata effettuata su questa misura.

```
Linear regression                               Number of obs     =      8,743
                                                F(10, 8732)       =    1061.12
                                                Prob > F          =     0.0000
                                                R-squared         =     0.5491
                                                Root MSE          =      .5091

------------------------------------------------------------------------------
                             |               Robust
                    logprice |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-----------------------------+------------------------------------------------
                  logscore_1 |   .1009911   .0573423     1.76   0.078    -.0114134    .2133955
                accommodates |   .1326145   .0070402    18.84   0.000     .1188141    .1464149
                   bathrooms |   .0471346   .0166783     2.83   0.005     .0144411     .079828
                    bedrooms |   .0191739    .012601     1.52   0.128    -.0055271    .0438749
                  app_intero |   .5192779   .0203175    25.56   0.000     .4794508     .559105
                    avg_dist |  -.1180005    .005789   -20.38   0.000    -.1293483   -.1066527
           reviews_per_month |   -.025703   .0045233    -5.68   0.000    -.0345698   -.0168362
         cancellation_strict |  -.0410604   .0110148    -3.73   0.000     -.062652   -.0194688
       host_acceptance_rate |   .5581953   .0507991    10.99   0.000     .4586171    .6577735
calculated_host_listings_count |    -.00112   .0002016    -5.56   0.000    -.0015152   -.0007248
                       _cons |   3.079409   .1286312    23.94   0.000     2.827261    3.331556
------------------------------------------------------------------------------
```

Anche con un po' di stupore, l'output mostra come la struttura del modello si adatti perfettamente anche alla misura "Score_1", che ottiene risultati molto simili a quelli utilizzando "Rank_2".

```
Linear regression                               Number of obs     =      8,743
                                                F(10, 8732)       =    1060.12
                                                Prob > F          =     0.0000
                                                R-squared         =     0.5489
                                                Root MSE          =     .50919

------------------------------------------------------------------------------
                             |               Robust
                    logprice |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-----------------------------+------------------------------------------------
                  logscore_2 |   .0550274   .0726688     0.76   0.449    -.0874206    .1974753
                accommodates |   .1320383   .0070246    18.80   0.000     .1182684    .1458082
                   bathrooms |   .0469578   .0166956     2.81   0.005     .0142304    .0796851
                    bedrooms |   .0195237    .012616     1.55   0.122    -.0052067     .044254
                  app_intero |   .5149139   .0202695    25.40   0.000      .475181    .5546468
                    avg_dist |  -.1177732   .0057934   -20.33   0.000    -.1291297   -.1064167
           reviews_per_month |  -.0257108   .0045203    -5.69   0.000    -.0345717    -.01685
         cancellation_strict |  -.0413067   .0110223    -3.75   0.000    -.0629131   -.0197004
       host_acceptance_rate |   .5482233    .050717    10.81   0.000      .448806    .6476405
calculated_host_listings_count |  -.0011882   .0002011    -5.91   0.000    -.0015824   -.0007939
                       _cons |   3.182403   .1506199    21.13   0.000     2.887153    3.477654
------------------------------------------------------------------------------
```

Un po' meno vicino al caso migliore è l'output con la trasformata dello "Score_2" al posto della trasformata di "rank".
Il segno è comunque coerente con l'effetto desiderato, ma il coefficiente perde di significatività. Come nel caso precedente, è probabile che una diversa ottimizzazione della selezione delle variabili avrebbe portato a risultati simili al caso base.

6.5. Confronto tra stanze e appartamenti

L'introduzione nel modello finale della variabile "app_intero" permette di individuare un premio di prezzo "medio" derivante dal passaggio da una stanza singola ad un appartamento completo, a parità di altre condizioni.
Nei capitoli precedenti si era riflettuto su come la ricerca di un appartamento e quella di un alloggio stanza siano due transazioni completamente diverse, sia per il genere di clientela che cerca ognuna delle due tipologie, sia per i "desiderata" di un alloggio per le differenti categorie. Per questi motivi, dire che il meccanismo di equilibrio di prezzo è il medesimo, a meno di una costante, è un'affermazione piuttosto ambiziosa.

Si è dunque ritenuto indispensabile effettuare un'analisi separando il database in due sottocampioni, quello degli alloggi stanza e quello degli appartamenti completi, al fine di verificare se il modello generale si adatti bene ai due sottocampioni, o se sia necessario procedere ad una modellizzazione personalizzata per i due tipi di transazione.

6.5.1 Meccanismo di equilibrio per le stanze

Il primo controllo da fare è quello di adeguatezza del modello generale se applicato al solo sottocampione delle stanze singole.
In caso di risultati non ottimali, diventa importante capire se è sufficiente un raffinamento del modello generale, per potersi adattare ad un meccanismo specifico che non è così diverso da quello della piattaforma in aggregato, oppure se l'equilibrio di prezzo per le stanze singole è governato da altri fattori (o dagli stessi ma espressi in modo alternativo) ed è necessaria la costruzione di un modello ad hoc.

```
Linear regression                               Number of obs     =      4,599
                                                F(9, 4589)        =     124.61
                                                Prob > F          =     0.0000
                                                R-squared         =     0.2928
                                                Root MSE          =     .48387

------------------------------------------------------------------------------
                             |               Robust
                    logprice |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-----------------------------+------------------------------------------------
                     logrank |   .4968712   .1408082     3.53   0.000     .2208195    .7729229
                accommodates |   .2523732   .0133693    18.88   0.000      .226163    .2785834
                    bedrooms |  -.0002425   .0373544    -0.01   0.995    -.0734752    .0729902
                   bathrooms |  -.1306113   .0268286    -4.87   0.000    -.1832083   -.0780143
          reviews_per_month |  -.0323194   .0063167    -5.12   0.000    -.0447032   -.0199355
                    avg_dist |  -.1328931   .0068434   -19.42   0.000    -.1463094   -.1194768
          cancellation_strict |  -.0572081   .0143349    -3.99   0.000    -.0853115   -.0291048
calculated_host_listings_count |  -.0004084   .0020976    -0.19   0.846    -.0045207     .003704
        host_acceptance_rate |    .610503   .0596924    10.23   0.000     .4934771    .7275289
                       _cons |   2.247796   .3154608     7.13   0.000     1.629341    2.866251
------------------------------------------------------------------------------
```

Analizzando segno e significatività del coefficiente di "logrank", l'impatto della gentilezza sul prezzo degli alloggi stanza sembrerebbe essere lo stesso che nel caso generale.
In questa iterazione il coefficiente di "logrank" è addirittura più significativo che nel caso generale (il valore *t* è pari e 3.53, rispetto al 3.12 del caso aggregato).
Per analizzare il comportamento delle variabili di controllo:

- La variabile "accommodates" ha comportamento molto simile rispetto al caso generale
- La variabile "host_acceptance_rate" ha comportamento molto simile rispetto al caso generale
- La variabile "cancelleation_strict" ha comportamento molto simile rispetto al caso generale
- La variabile "avg_dist" ha comportamento molto simile rispetto al caso generale
- La variabile "reviews_per_month" ha comportamento molto simile rispetto al caso generale
  - La variabile "calculated_host_listings_count" mantiene il suo segno negativo, ma perde completamente di significatività.
  - La variabile "bedrooms" perde di significatività
- La variabile "bathrooms" cambia di segno e diventa significativamente negativa.

In base a quanto mostrato dall'output sopra presentato, si può affermare con confidenza che il modello generale si adatti molto bene al mondo degli alloggi stanza.

Allo stesso tempo, è possibile ottimizzarlo alla luce di alcune considerazioni ovvie su questo tipo di transazione.
La variabile "bedrooms" risulta completamente superflua, in quanto è ragionevole pensare che nell'affittare una stanza singola, il numero di camere da letto sia unitario.
L'inversione di segno della variabile "bathrooms" permette di ottenere delle informazioni che si erano perse nel meccanismo di aggregazione: il fatto che ad un aumento del numero di bagni sia associato una diminuzione del prezzo fa capire che i clienti che scelgono alloggi stanza non sono disposti a pagare di più per avere un bagno privato.

Per cercare di raffinare il modello alla luce di questo meccanismo, è più corretto sostituire la variabile "bathrooms" con la variabile bagno.
Questa è una *dummy* che assume valore 1 nel caso la stanza abbia un bagno privato.
Per coerenza con quanto detto in precedenza, si esclude anche la variabile "bedrooms".

```
Linear regression                               Number of obs     =      4,599
                                                F(8, 4590)        =     110.37
                                                Prob > F          =     0.0000
                                                R-squared         =     0.2816
                                                Root MSE          =     .48761

------------------------------------------------------------------------------
                             |               Robust
                    logprice |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-----------------------------+------------------------------------------------
                     logrank |   .5509798   .1406761     3.92   0.000     .2751869    .8267726
                accommodates |   .2351131   .0183334    12.82   0.000     .1991708    .2710553
                       bagno |   -.219559   .1362948    -1.61   0.107    -.4867624    .0476445
           reviews_per_month |  -.0315552   .0059398    -5.31   0.000    -.0432001   -.0199104
                    avg_dist |  -.1312544    .006841   -19.19   0.000     -.144666   -.1178428
        cancellation_strict  |  -.0560014   .0144178    -3.88   0.000    -.0842673   -.0277355
calculated_host_listings_count |  -.0024184   .0020854    -1.16   0.246    -.0065068    .0016701
        host_acceptance_rate |   .6208846   .0629599     9.86   0.000      .497453    .7443163
                       _cons |   2.220866   .3377106     6.58   0.000     1.558791    2.882941
------------------------------------------------------------------------------
```

Il coefficiente negativo della variabile "bagno" rafforza la convinzione secondo cui i clienti di alloggi stanza (probabilmente coppie o piccoli gruppi di giovani) non sono disposti a pagare per un bagno privato e si accontentano di quello condiviso.
Rimangono valide le considerazioni fatte sul modello precedente.

Nonostante si sia convinti che la relazione che meglio esplica l'influenza della gentilezza sul prezzo sia una forma funzionale di tipo "log-log", si ottengono buoni risultati sul sottocampione alloggi stanza anche eseguendo la trasformazione logaritmica solo sul prezzo e mantenendo in forma assoluta la variabile "rank". Nonostante ai fini della modellizzazione non restituisca i risultati più adatti, il fatto che anche in questo caso permanga l'effetto marginale positivo rafforza la convinzione secondo cui la qualità dell'interazione con il cliente sia un fattore virtuoso rispetto alla sua performance.

Per completezza espositiva si riporta anche l'output di questa variante.

```
Linear regression                               Number of obs     =      4,599
                                                F(8, 4590)        =     110.90
                                                Prob > F          =     0.0000
                                                R-squared         =     0.2819
                                                Root MSE          =     .48754

------------------------------------------------------------------------------
                             |               Robust
                    logprice |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-----------------------------+------------------------------------------------
                        rank |   .0726052   .0174351     4.16   0.000     .0384241    .1067863
                accommodates |   .2351889   .0183246    12.83   0.000      .199264    .2711139
                       bagno |  -.2199171   .1363243    -1.61   0.107    -.4871784    .0473441
           reviews_per_month |  -.0313818     .00594    -5.28   0.000     -.043027   -.0197366
                    avg_dist |  -.1312049   .0068399   -19.18   0.000    -.1446144   -.1177953
            cancellation_strict |  -.0560022   .0144147    -3.89   0.000     -.084262   -.0277424
calculated_host_listings_count |  -.0023957   .0020837    -1.15   0.250    -.0064808    .0016894
          host_acceptance_rate |    .622862   .0629998     9.89   0.000     .4993521    .7463719
                         _cons |   2.783261   .2100757    13.25   0.000     2.371412     3.19511
------------------------------------------------------------------------------
```

6.5.2 Meccanismo di equilibrio per gli appartamenti completi

In modo esattamente analogo al caso precedente, la prima operazione è l'applicazione del modello generale al subset degli appartamenti completi.

```
Linear regression                               Number of obs     =      4,144
                                                F(9, 4134)        =     180.58
                                                Prob > F          =     0.0000
                                                R-squared         =     0.2865
                                                Root MSE          =      .5018

------------------------------------------------------------------------------
                             |               Robust
                    logprice |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-----------------------------+------------------------------------------------
                     logrank |   .0812673    .171141     0.47   0.635    -.2542612    .4167958
                accommodates |   .0557312   .0068081     8.19   0.000     .0423837    .0690787
                   bathrooms |   .2575825   .0175327    14.69   0.000     .2232089     .291956
                    bedrooms |   .0458352   .0120936     3.79   0.000     .0221252    .0695452
                    avg_dist |  -.0745119   .0100654    -7.40   0.000    -.0942454   -.0547783
           reviews_per_month |  -.0148167   .0067204    -2.20   0.028    -.0279924   -.0016411
            cancellation_strict |  -.0283256   .0158497    -1.79   0.074    -.0593996    .0027483
          host_acceptance_rate |   .5131077   .0896495     5.72   0.000     .3373465    .6888689
calculated_host_listings_count |  -.0008993   .0002016    -4.46   0.000    -.0012945   -.0005041
                         _cons |   3.558026   .3612891     9.85   0.000     2.849705    4.266347
------------------------------------------------------------------------------
```

Le variabili di controllo svolgono la stessa funzione che hanno nel modello aggregato. Anche l'impatto marginale della gentilezza mantiene il segno positivo del modello generale, ma il livello di significatività è estremamente basso.
È un chiaro esempio che il modello generale non è ottimizzato per estrarre in modo corretto il contributo della gentilezza su questo subset.
Si tratta molto probabilmente di un problema di selezione delle variabili di controllo.
Il primo intervento che si può attuare è la sostituzione di alcune variabili con le loro analoghe che cercano di cogliere lo stesso fenomeno.
In questo caso si è deciso di iniziare sostituendo "calculated_host_listings_count" con la versione *dummy* "multiprop", dato che a livello di modello generale non si era osservato un vero motivo per preferire l'una all'altra.

```
Linear regression                               Number of obs     =      4,144
                                                F(9, 4134)        =     181.88
                                                Prob > F          =     0.0000
                                                R-squared         =     0.2842
                                                Root MSE          =      .5026

------------------------------------------------------------------------------
             |               Robust
    logprice |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     logrank |   .2693438   .1702969     1.58   0.114    -.0645298    .6032175
accommodates |   .0529686   .0068217     7.76   0.000     .0395943    .0663429
    bedrooms |   .0496469   .0120451     4.12   0.000     .0260321    .0732618
   bathrooms |   .2583608    .017536    14.73   0.000     .2239808    .2927408
    avg_dist |  -.0690102   .0100821    -6.84   0.000    -.0887766   -.0492438
reviews_per_~h |  -.0062522   .0063857    -0.98   0.328    -.0187715    .0062672
cancellation~t |  -.0202801   .0156816    -1.29   0.196    -.0510245    .0104644
host_acceptan~e |   .4667812   .0911852     5.12   0.000     .2880092    .6455532
   multiprop |   .0361817   .0183042     1.98   0.048     .0002957    .0720677
       _cons |   3.145818   .3610832     8.71   0.000       2.4379    3.853735
------------------------------------------------------------------------------
```

Già con questa semplice operazione i risultati appaiono molto più solidi.
L'impatto della gentilezza sale in modo consistente (coefficiente pari a 0,269 vs 0,08 del caso di partenza) e il livello di significatività raggiunge quasi il 90%.
Operando una ulteriore sostituzione dell'attributo "bathrooms" con la sua versione binaria, si riesce ad estrarre in modo soddisfacente l'impatto della gentilezza.

```
Linear regression                               Number of obs     =      4,145
                                                F(9, 4135)        =     146.55
                                                Prob > F          =     0.0000
                                                R-squared         =     0.2411
                                                Root MSE          =     .51747

------------------------------------------------------------------------------
             |               Robust
    logprice |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     logrank |   .4362176   .1734097     2.52   0.012     .0962414    .7761939
accommodates |    .082469    .006951    11.86   0.000     .0688413    .0960967
    bedrooms |   .0993774   .0125213     7.94   0.000     .0748288    .1239259
       bagno |   .1374881   .1805903     0.76   0.447     -.216566    .4915423
    avg_dist |  -.0838879   .0105442    -7.96   0.000    -.1045601   -.0632156
reviews_per_~h |  -.0099879   .0065509    -1.52   0.127    -.0228313    .0028555
cancellation~t |  -.0072008    .016099    -0.45   0.655    -.0387635    .0243619
host_acceptan~e |   .4674331   .0956328     4.89   0.000     .2799414    .6549248
   multiprop |    .030113   .0189171     1.59   0.111    -.0069747    .0672008
       _cons |   2.822745   .4162019     6.78   0.000     2.006766    3.638725
------------------------------------------------------------------------------
```

Il nuovo livello di significatività raggiunge quasi il 99% e il modulo del coefficiente si assesta su valori più importanti. Inoltre:

- La variabile "logrank" si comporta in modo simile al modello generale
- La variabile "accommodates si comporta in modo simile al modello generale
- La variabile "bedrooms si comporta in modo simile al modello generale
- La variabile "avg_dist si comporta in modo simile al modello generale
- La variabile "host_acceptance_rate si comporta in modo simile al modello generale
- La variabile "reviews_per_month" si comporta in modo simile al modello generale

- La variabile “bagno” mantiene il segno corretto ma perde di significatività
- La variabile “cancellation_strict” mantiene il segno corretto ma perde di significatività
- ◊ La variabile “multiprop” perde di significatività e subisce una curiosa inversione di segno

In generale, malgrado l’incapacità di estrarre il vero impatto del tipo di host (host amatore vs host professionista), il modello generale si adatta molto bene anche al secondo subset.
Di conseguenza, si può affermare che ragionevolmente il meccanismo di formazione del prezzo per le due transazioni non sia poi così diverso, e non ci siano motivi per sostenere l’inadeguatezza dell’impianto del modello generale.

Come nel caso generale (e quello degli alloggi stanza), anche per gli appartamenti completi si è in grado di elaborare un modello subottimale ma robusto effettuando la trasformata logaritmica solo della variabile dipendente, come mostrato dal seguente ouptut.

```
Linear regression                               Number of obs     =      4,145
                                                F(8, 4136)        =     164.61
                                                Prob > F          =     0.0000
                                                R-squared         =     0.2410
                                                Root MSE          =     .51742

------------------------------------------------------------------------------
             |               Robust
    logprice |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        rank |    .060348   .0242649     2.49   0.013     .0127759    .1079202
accommodates |   .0825344   .0069541    11.87   0.000     .0689006    .0961682
    bedrooms |   .0995589   .0125174     7.95   0.000      .075018    .1240998
    avg_dist |  -.0843659   .0104852    -8.05   0.000    -.1049226   -.0638092
reviews_per_~h |  -.0098666   .0065435    -1.51   0.132    -.0226953    .0029621
cancellation~t |   -.007437   .0160859    -0.46   0.644    -.0389741    .0241001
host_acceptan~e |   .4678956   .0956772     4.89   0.000     .2803169    .6554743
    multiprop |   .0304299   .0189395     1.61   0.108    -.0067017    .0675616
        _cons |   3.385436   .2117048    15.99   0.000     2.970381    3.800492
------------------------------------------------------------------------------
```

6.5.3 Confronto tra i due subset

Un altro motivo per cui valeva la pena di applicare il modello generale sui due subset separati era quello di verificare se esistessero delle differenze sostanziali nel modo in cui la gentilezza influenza il prezzo. A questo fine, non è tanto importante analizzare il livello di confidenza, fortemente impattato dalla scelta delle variabili di controllo, ma piuttosto è importante analizzare il modulo del coefficiente di tale variabile.

Effettuare un confronto risulta particolarmente facile utilizzando i modelli di tipo log-log, in quanto un confronto tra i coefficienti permette di confrontare direttamente le elasticità del prezzo rispetto alla gentilezza.
Si può osservare come l’elasticità del prezzo rispetto alla gentilezza negli alloggi stanza (0,496) sia sostanzialmente superiore a quella negli appartamenti (0,436).
Si ritiene che questo fatto sia perfettamente coerente con la realtà, in quanto l’impatto “umano” del padrone di casa è molto più rilevante in un’esperienza a stretto contatto come nel caso di alloggio in una stanza e si ritiene giusto che il virtuosismo nel servizio si rifletta in modo più netto sul prezzo.

6.6. Superhost e Review_scores_rating

In questo paragrafo si vuole esplorare il ruolo nel meccanismo di formazione del prezzo di equilibrio delle variabili “host_is_superhost” e “review_scores_rating” e si vuole fornire una spiegazione per la loro esclusione nelle analisi effettuate nel capitolo 6 e quelle che seguiranno nel capitolo 7.

6.6.1 Host_is_superhost

Il badge di Superhost è una certificazione della qualità di un alloggio e della professionalità del proprio padrone di casa. Il fatto che sia assegnato direttamente dalla piattaforma Airbnb, mediante una procedura standardizzata, rafforza la credibilità agli occhi dei clienti di tale attributo.
Ragionando sul potenziale impatto di tale certificazione durante il processo decisionale dei clienti, è facile immaginare che essi possano preferire un listing certificato Superhost, e che possano essere disposti a pagare un prezzo più elevato, aspettandosi un’esperienza di alto livello.

Source	SS	df	MS		
Model	29891986.1	11	2717453.28	Number of obs	= 8,743
				F(11, 8731)	= 442.80
Residual	53582032.3	8,731	6136.98686	Prob > F	= 0.0000
				R-squared	= 0.3581
				Adj R-squared	= 0.3573
Total	83474018.4	8,742	9548.61798	Root MSE	= 78.339

price	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
host_is_superhost	9.329374	1.87002	4.99	0.000	5.663694	12.99505
accommodates	13.23063	.8429817	15.70	0.000	11.57818	14.88307
guests_included	3.658693	.7184923	5.09	0.000	2.250279	5.067108
bedrooms	6.724914	1.611687	4.17	0.000	3.565627	9.8842
bathrooms	19.18576	1.731123	11.08	0.000	15.79235	22.57917
reviews_per_month	-4.897422	.6088877	-8.04	0.000	-6.090985	-3.703858
avg_dist	-10.278	.9130585	-11.26	0.000	-12.06781	-8.488185
cancellation_strict	-4.311786	1.688481	-2.55	0.011	-7.621607	-1.001964
host_acceptance_rate	55.68458	7.977215	6.98	0.000	40.04736	71.3218
app_intero	25.49813	2.426205	10.51	0.000	20.7422	30.25406
calculated_host_listings_count	-.2247471	.0340478	-6.60	0.000	-.2914887	-.1580054
_cons	-24.18876	8.071416	-3.00	0.003	-40.01064	-8.366887

Immaginandosi che l’effetto reputazione dell’attributo potesse essere lineare, si è effettuata una modellizzazione sfruttando un impianto di regressione lineare multipla, utilizzando un set di variabili di controllo simile a quello usato nella modellizzazione attraverso la misura di gentilezza.
I risultati mostrati sono coerenti con quanto ci si aspettava: l’effetto sul prezzo della certificazione Superhost è positivo e super significativo.
In particolare, il coefficiente di tale variabile mostra che la certificazione porta ad un aumento del prezzo di equilibrio di 9,33 €/notte.
Se confrontato con la mediana della distribuzione dei prezzi, pari a 65 €, ciò comporta un aumento percentuale pari al 14,3%.

6.6.2 Review_scores_rating

Si tratta di un indicatore compreso tra 1 e 100 che viene estratto a partire dai giudizi che i clienti rilasciano durante il processo di valutazione, assegnando da 1 a 5 "stelle" ad una serie di categorie.
Dato che è un'elaborazione dei giudizi ex-post rispetto al momento del soggiorno, ci si immagina che abbia un buon potere esplicativo rispetto alla variabilità dei prezzi di mercato. In particolare, se tale indicatore è efficiente, ci si aspetta che esista una relazione positiva tra il prezzo e il rating di un alloggio.
Anche in questo caso ci si è immaginati che l'impatto sul prezzo potesse essere di tipo lineare.

Source	SS	df	MS		
Model	30446639.5	11	2767876.32	Number of obs	= 8,743
Residual	53027378.9	8,731	6073.45996	F(11, 8731)	= 455.73
				Prob > F	= 0.0000
Total	83474018.4	8,742	9548.61798	R-squared	= 0.3647
				Adj R-squared	= 0.3639
				Root MSE	= 77.932

price	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
review_scores_rating	1.580376	.1464355	10.79	0.000	1.293328	1.867424
accommodates	13.81692	.840644	16.44	0.000	12.16906	15.46478
guests_included	3.464016	.7149757	4.84	0.000	2.062495	4.865537
bedrooms	6.403094	1.603379	3.99	0.000	3.260093	9.546095
bathrooms	18.67663	1.722964	10.84	0.000	15.29922	22.05405
reviews_per_month	-5.444896	.607492	-8.96	0.000	-6.635724	-4.254068
avg_dist	-10.12639	.9078656	-11.15	0.000	-11.90602	-8.346755
cancellation_strict	-3.897561	1.680191	-2.32	0.020	-7.191132	-.6039903
host_acceptance_rate	62.70284	7.962673	7.87	0.000	47.09413	78.31156
app_intero	24.54995	2.413551	10.17	0.000	19.81882	29.28108
calculated_host_listings_count	-.1714904	.0342134	-5.01	0.000	-.2385567	-.1044241
_cons	-173.258	16.15656	-10.72	0.000	-204.9286	-141.5873

L'evidenza empirica conferma le ipotesi fatte a priori, ad ogni punto di miglioramento del rating è associato un incremento del prezzo pari a 1,58 €/notte.

Confrontando i valori di prezzo previsti dai due modelli, ci si è resi conto che entrambi generano previsioni molto simili. Questo risultato non stupisce alla luce del fatto che le due variabili sono fortemente correlate (coefficiente di correlazione pari a 0,54) dato che per essere considerati Superhost, gli alloggi non devono aver avuto un rating inferiore a 96/100 negli ultimi 12 mesi.

Questo significa, che al fine di spiegare la variabilità dei prezzi di mercato, il contenuto informativo di "host_is_superhost" se aggiunto al modello già contente "reviews_score_rating" è minimo, come esemplificato nel modello successivo.
In questo output si può osservare come, a causa della distorsione generata dalla multicollinearità, la variabile "host_is_superhost" perda di significatività e come il modello abbia la stessa capacità di esplicazione di quello con la sola variabile "review_scores_rating" (*R-squared* pressochè identici).

Source	SS	df	MS	Number of obs	=	8,743
				F(12, 8730)	=	417.74
Model	30448028.4	12	2537335.7	Prob > F	=	0.0000
Residual	53025990	8,730	6073.99656	R-squared	=	0.3648
				Adj R-squared	=	0.3639
Total	83474018.4	8,742	9548.61798	Root MSE	=	77.936

price	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
host_is_superhost	-1.029319	2.15249	-0.48	0.633	-5.248707	3.19007
review_scores_rating	1.621129	.1694341	9.57	0.000	1.288998	1.95326
accommodates	13.80951	.8408239	16.42	0.000	12.1613	15.45772
guests_included	3.458457	.7151018	4.84	0.000	2.056689	4.860225
bedrooms	6.417099	1.603717	4.00	0.000	3.273435	9.560763
bathrooms	18.67545	1.723042	10.84	0.000	15.29788	22.05301
reviews_per_month	-5.430087	.6083077	-8.93	0.000	-6.622513	-4.23766
avg_dist	-10.11029	.9085297	-11.13	0.000	-11.89122	-8.329354
cancellation_strict	-3.904734	1.680332	-2.32	0.020	-7.198582	-.6108865
host_acceptance_rate	62.88097	7.971732	7.89	0.000	47.2545	78.50744
app_intero	24.59582	2.415563	10.18	0.000	19.86075	29.33089
calculated_host_listings_count	-.1726903	.0343068	-5.03	0.000	-.2399396	-.1054409
_cons	-176.9057	17.86742	-9.90	0.000	-211.93	-141.8813

6.6.3 Giustificazione dell'omissione del rating

Dato che l'obbiettivo di questa tesi è valutare l'impatto della gentilezza sulla performance dell'host, e dato che il "rating" per come è costruito coglie in modo aggregato una moltitudine di aspetti, sarebbe stato interessante produrre una modellizzazione che includesse contemporaneamente la variabile "rating" (che implicitamente spiega anche l'effetto Superhost) e una misura di gentilezza.
Questa operazione non è stata possibile in quanto l'alto livello di correlazione tra le due variabili (coefficiente di correlazione tra "rating" e "rank" pari a 0,4), causava delle distorsioni da multicollinearità che non permettevano di valutare in alcun modo l'effetto della gentilezza.
Non potendo inserire contestualmente "rank" e "rating" nell'analisi, ed essendo il focus della tesi legato alla gentilezza, si è deciso di tralasciare l'informazione racchiusa dalla misura di rating e di concentrare l'attenzione sui modelli basati sulla misura di gentilezza.

Effettuando delle analisi comparative tra le due famiglie di modelli, e in particolare confrontando la capacità di spiegare la variabilità nei prezzi di mercato, si è comunque potuto verificare che l'inefficienza generata dall'omissione della variabile "review_score_rating" è piuttosto contenuta.
Questo approccio è stato replicato non solo per i modelli di *pricing* ma anche per l'analisi sui tassi di domanda presentata nel capitolo 7.

6.6.4 Rank come determinante del rating

La distorsione da multicollinearità tra "rank" e "rating" è generata dal fatto che senza dubbio il rating di un alloggio è portatore di informazione sulla qualità dell'interazione tra il padrone di casa e i suoi clienti.
A questo punto è risultato naturale provare a scorporare l'effetto di gentilezza a partire dalla misura di "rating", cercando a livello operativo di dimostrare che la gentilezza è un fattore determinante di quest'ultimo.
Dato che l'indicatore "review_scores_rating" nella pratica è una combinazione lineare di una serie di indicatori, di cui almeno 2 ("review_scores_communication" e

"review_scores_checkin") sono portatori di informazione legata al comportamento dell'host, si è ipotizzato che anche la relazione tra "rating" e "gentilezza" potesse essere di tipo lineare.
Per questo motivo, si è effettuato un tentativo di modellizzazione lineare semplice.

```
. regress review_scores_rating rank , robust

Linear regression                               Number of obs     =      8,758
                                                F(1, 8756)        =     931.68
                                                Prob > F          =     0.0000
                                                R-squared         =     0.1540
                                                Root MSE          =     5.5394

------------------------------------------------------------------------------
             |               Robust
review_sco~g |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        rank |   5.600733   .1834893    30.52   0.000     5.241051    5.960416
       _cons |   48.95291    1.41918    34.49   0.000     46.17098    51.73483
------------------------------------------------------------------------------
```

Dall'output si può vedere come la misura gentilezza sia molto efficiente al fine di spiegare la variabilità tra i rating: il suo coefficiente ha segno positivo, come si attendeva, e il livello di significatività è eccellente (t-value pari a 30.5).

Dato che l'indicatore "rating" è molto incentrato sul concetto di comunicazione si è voluto inserire nel modello la variabile "confidenza", che rappresenta la percentuale di volte in cui i clienti si riferiscono al padrone di casa nella recensione utilizzando il suo nome di persona. La motivazione dietro tale inserimento è la seguente: se la comunicazione tra il padrone di casa e il cliente è di alto livello ci si aspetta che il cliente tenda a ricordare il suo nome, e nella recensione tenderà ad utilizzare proprio questo, piuttosto che riferirsi a lui con il generico termine "host".

```
Linear regression                               Number of obs     =      8,758
                                                F(2, 8755)        =     869.96
                                                Prob > F          =     0.0000
                                                R-squared         =     0.1853
                                                Root MSE          =     5.4361

------------------------------------------------------------------------------
             |               Robust
review_sco~g |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        rank |   3.825641   .2181014    17.54   0.000     3.398112    4.253171
  confidenza |   5.139513   .3150344    16.31   0.000     4.521971    5.757054
       _cons |   60.79826   1.607519    37.82   0.000     57.64714    63.94937
------------------------------------------------------------------------------
```

Come si può vedere dall'output, la variabile "confidenza" "performa" in modo eccellente e in particolare il suo inserimento permette di migliorare in modo sostanziale la capacità esplicativa del modello.

Alla luce dell'ultima modellizzazione si può affermare che la gentilezza sia un determinante del rating di un alloggio.
Dato che nel paragrafo [6.6.2] si era mostrato come il rating influenzi positivamente il prezzo, la composizione dei due effetti permette di mostrare, con un approccio completamente diverso rispetto a quello impiegato nell'analisi centrale dello studio, che la gentilezza abbia un impatto virtuoso rispetto al prezzo.

7. Relazione tasso di domanda-gentilezza

La modellizzazione presentata nel capitolo precedente ha permesso di affermare con sicurezza che ad un atteggiamento particolarmente positivo del padrone di casa è associato un premio di prezzo.
L'analisi dei coefficienti stimati attraverso la regressione però, ha mostrato che tale impatto è piuttosto contenuto e che la determinazione del prezzo di equilibrio è dominata da altri fattori (come ad esempio la capienza dell'appartamento e la sua centralità).
Questo risultato non implica che non esista nessun vantaggio rilevante associato alla gentilezza, in quanto il premio di prezzo è solo uno dei potenziali *boost* di performance.
È infatti possibile che i padroni di casa particolarmente gentili non siano in grado di imporre un prezzo troppo diverso da quello di host "neutri" (a parità delle altre condizioni), ma che attraverso il loro servizio superiore alla media, siano in grado di catturare molta più domanda rispetto ai concorrenti.
In questo modo, otterrebbero un incremento rilevante dei loro ricavi, da imputarsi ad una maggiore rotazione della clientela.
Lo scopo di questo capitolo è verificare se questo meccanismo si rifletta effettivamente nella realtà.

Il modello di equilibrio di prezzo è stato estratto a partire dal database aggregato (contenente stanze e appartamenti) perché si credeva che il meccanismo di equilibrio potesse essere simile. In questo capitolo si è invece seguito un approccio differente: si è deciso di iniziare costruendo un modello per uno dei due subset parziali (quello degli appartamenti completi) per poi cercare, nel caso si fossero ottenuti risultati soddisfacenti, di generalizzarlo per creare un modello universale.

7.1. Appartamenti completi

Il primo tentativo è stato effettuato attraverso un modello di regressione multipla.
Pensando che l'effetto della gentilezza potesse essere di tipo lineare, si è deciso di iniziare con un modello di regressione lineare multipla.

7.1.1 Regressione lineare

Per prima cosa si è individuata la variabile dipendente, che si vuole cercare di spiegare tramite il set di variabili esplicative: come *proxy* del tasso di domanda si è selezionata la variabile "reviews_per_month", ma i risultati non cambiano molto sostituendola con "number_of_reviews_ltm" (o addirittura con "number_of_reviews").
Come variabile target è stata selezionata "rank", per creare un parallelo diretto con la modellizzazione del prezzo di equilibrio.

A differenza del modello di prezzo, dove non si osservava una correlazione positiva tra variabile dipendente e target, le analisi preliminari di correlazione mostrano dei risultati più speranzosi.

	review~h
reviews_pe~h	1.0000
rank	0.1200
rank_2	0.1398
score_1	0.1661
score_2	0.1552

Figura 33:Correlazione tasso di domanda e indicatori di gentilezza

L'operazione più critica è stata sicuramente la selezione delle variabili di controllo, che verranno presentate raggruppate per tipologia di aspetto che cercano di cogliere.

7.1.1.1 Aspetti economici

In modo esattamente analogo ai modelli presentati nel capitolo precedente, si pensa che il tasso di domanda sia negativamente influenzato dal prezzo.

```
. regress reviews_per_month rank price, robust

Linear regression                               Number of obs     =      4,150
                                                F(2, 4147)        =      35.81
                                                Prob > F          =     0.0000
                                                R-squared         =     0.0180
                                                Root MSE          =     1.3613

------------------------------------------------------------------------------
             |               Robust
reviews_pe~h |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        rank |    .454948   .0603042     7.54   0.000     .3367195    .5731765
       price |  -.0007539   .0001811    -4.16   0.000     -.001109   -.0003989
       _cons |  -1.355064   .4486024    -3.02   0.003    -2.234565   -.4755627
------------------------------------------------------------------------------
```

Un'altra variabile legata ad aspetti economici è il deposito cauzionale, quantificato dalla variabile "security_deposit": ci si immagina che un deposito cospicuo possa essere un deterrente rispetto alla prenotazione, e di conseguenza ci si aspetta un coefficiente negativo che lo leghi alla variabile dipendente.

```
. regress reviews_per_month rank price security_deposit, robust

Linear regression                               Number of obs     =      3,988
                                                F(3, 3984)        =      46.25
                                                Prob > F          =     0.0000
                                                R-squared         =     0.0363
                                                Root MSE          =     1.3463

------------------------------------------------------------------------------
             |               Robust
reviews_per_mo~h |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-----------------+----------------------------------------------------------------
            rank |   .4828965    .060854     7.94   0.000     .3635885    .6022044
           price |  -.0004935   .0001874    -2.63   0.008    -.0008609   -.0001262
security_deposit |  -.0008749   .0001016    -8.61   0.000    -.0010741   -.0006758
           _cons |  -1.407794   .4517317    -3.12   0.002    -2.293441   -.5221471
------------------------------------------------------------------------------
```

L'output mostra risultati coerenti con quanto ci aspettava.

7.1.1.2 Effetto location

```
. regress reviews_per_month rank price security_deposit avg_dist, robust

Linear regression                               Number of obs     =      3,988
                                                F(4, 3983)        =      38.28
                                                Prob > F          =     0.0000
                                                R-squared         =     0.0401
                                                Root MSE          =     1.3438

------------------------------------------------------------------------------
             |               Robust
reviews_per~h|      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        rank |   .4927766   .0614027     8.03   0.000     .3723929    .6131603
       price |    -.000589    .000188    -3.13   0.002    -.0009575   -.0002205
security_de~t|   -.0008806    .000102    -8.63   0.000    -.0010807   -.0006806
    avg_dist |    -.109646   .0244558    -4.48   0.000    -.1575932   -.0616989
       _cons |    -1.20601   .4523956    -2.67   0.008    -2.092958   -.3190613
------------------------------------------------------------------------------
```

Coerentemente con quanto ci si aspettava, il tasso di domanda risulta negativamente influenzato anche dall'indicatore di distanza medio "avg_dist".
Non solo, l'introduzione di questa ulteriore variabile porta ad un rafforzamento della significatività delle altre variabili presenti.

7.1.1.3. Tasso di risposta e accettazione del padrone di casa

Si tratta di indicatori che riflettono il comportamento del padrone di casa durante le fasi precedenti alla conferma della prenotazione.
Ovviamente un alto tasso di risposta dovrebbe favorire il tasso di domanda, così come un tasso alto di accettazione.
È necessario verificare se le due variabili portino la stessa informazione, oppure se sia ottimale inserirle congiuntamente.

```
Linear regression                               Number of obs     =      3,915
                                                F(5, 3909)        =      43.13
                                                Prob > F          =     0.0000
                                                R-squared         =     0.0574
                                                Root MSE          =     1.3351

------------------------------------------------------------------------------
             |               Robust
 reviews_per~h|      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        rank |   .4573623   .0629292     7.27   0.000     .3339852    .5807395
       price |    -.000586   .0001886    -3.11   0.002    -.0009558   -.0002161
  security_de~t|   -.0009619   .0001036    -9.29   0.000    -.0011649   -.0007588
    avg_dist |   -.1399244   .0250109    -5.59   0.000      -.18896   -.0908888
host_respon~e|    1.348931   .1810361     7.45   0.000     .9939966    1.703865
       _cons |    -2.09742   .4626739    -4.53   0.000    -3.004525   -1.190315
------------------------------------------------------------------------------
```

```
Linear regression                               Number of obs     =      3,986
                                                F(5, 3980)        =      38.91
                                                Prob > F          =     0.0000
                                                R-squared         =     0.0481
                                                Root MSE          =     1.3384

------------------------------------------------------------------------------
                             |               Robust
   reviews_per_month         |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-----------------------------+------------------------------------------------
                        rank |    .529523   .0615217     8.61   0.000      .408906    .6501401
                       price |  -.0006285   .0001873    -3.35   0.001    -.0009958   -.0002612
            security_deposit |  -.0008377   .0001009    -8.30   0.000    -.0010356   -.0006398
                    avg_dist |  -.1078883   .0243513    -4.43   0.000    -.1556306   -.0601461
host_acceptance_rate         |   1.434436   .2825066     5.08   0.000     .8805649    1.988307
                       _cons |  -2.867173   .5567677    -5.15   0.000     -3.95875   -1.775597
------------------------------------------------------------------------------
```

Entrambe le variabili, usate singolarmente, sono altamente significative: dato un valore di *R-squared* molto più elevato, sembra che il tasso di risposta sia più efficace di quello di accettazione.

```
Linear regression                               Number of obs     =      3,915
                                                F(6, 3908)        =      39.92
                                                Prob > F          =     0.0000
                                                R-squared         =     0.0632
                                                Root MSE          =     1.3312

------------------------------------------------------------------------------
                             |               Robust
   reviews_per_month         |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-----------------------------+------------------------------------------------
                        rank |   .4925221   .0629811     7.82   0.000     .3690433     .616001
                       price |  -.0006313   .0001876    -3.36   0.001    -.0009992   -.0002635
            security_deposit |  -.0009152   .0001044    -8.77   0.000    -.0011198   -.0007105
                    avg_dist |  -.1365805    .024976    -5.47   0.000    -.1855477   -.0876133
  host_response_rate         |   1.274687   .1808708     7.05   0.000     .9200765    1.629297
host_acceptance_rate         |   1.263704   .3016458     4.19   0.000     .6723064    1.855103
                       _cons |  -3.518761   .5733051    -6.14   0.000    -4.642766   -2.394755
------------------------------------------------------------------------------
```

L'introduzione di entrambe le variabili permette di aumentare la potenza esplicativa del modello senza che queste inficino reciprocamente la loro significatività: per questo motivo si è deciso di mantenerle entrambe nel modello finale.

7.1.1.4 Host_identity_verified

"Host_identity_verified" è un attributo binario che assume valore 1 se il padrone di casa ha verificato la propria identità. Si tratta di un indicatore di affidabilità dell'host, certificato direttamente dalla piattaforma Airbnb, che ci si aspetta che abbia un impatto positivo sul tasso di domanda.
Essendo un attributo di tipo binario, il coefficiente stimato risulta particolarmente intuitivo in quanto esprime il potenziale incremento di clientela sottoponendosi al processo di verifica dell'identità.

```
Linear regression                               Number of obs     =      3,915
                                                F(7, 3907)        =      39.25
                                                Prob > F          =     0.0000
                                                R-squared         =     0.0695
                                                Root MSE          =     1.3269

------------------------------------------------------------------------------
                             |               Robust
     reviews_per_month       |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-----------------------------+------------------------------------------------
                        rank |   .4444982   .0629654     7.06   0.000     .3210499    .5679464
                       price |  -.0005901   .0001883    -3.13   0.002    -.0009593   -.0002209
            security_deposit |  -.0009281   .0001021    -9.09   0.000    -.0011283   -.0007279
                    avg_dist |  -.1410915   .0248724    -5.67   0.000    -.1898555   -.0923274
          host_response_rate |   1.223018   .1817851     6.73   0.000     .8666157    1.579421
        host_acceptance_rate |   1.316019    .301861     4.36   0.000     .7241986    1.907838
host_identity_verified       |   .2308879   .0443069     5.21   0.000      .144021    .3177548
                       _cons |  -3.23603    .5701491    -5.68   0.000    -4.353848   -2.118212
------------------------------------------------------------------------------
```

L'output mostra che l'effetto della verifica è molto significativo (*p-value* pari a 0).
Il segno del relativo coefficiente (così come quello di tutte le variabili inserite fino ad ora) è in linea con il senso comune e le ipotesi fatte a priori.

7.1.1.5 Azioni richieste al cliente

Gli host possono chiedere al cliente di effettuare alcune azioni che assicurino la loro buonafede durante le operazioni di prenotazione.
Sul database pubblicato da Airbnb si hanno informazioni su due azioni di verifica:

- "Require_guest_phone_verification" è una *dummy* che assume valore 1 se il padrone di casa richiede tassativamente al cliente di sottoporsi alla procedura di verifica del numero di telefono cellulare
- Require_guest_profile_picture è una *dummy* che assume valore 1 se il padrone di casa richiede tassativamente che il cliente carichi una propria foto identificativa

Nonostante si tratti di operazioni che nel periodo storico in cui viviamo non richiedono più di qualche minuto, si è curiosi di vedere se i risultati numerici avvalorino l'ipotesi secondo la quale queste azioni rappresentino un deterrente alla prenotazione.

	Phone_verification	Profile_picture
Phone_verification	1	0,4672
Profile_picture	0,4672	1

Figura 34:Correlazione tra Phone_verification e Profile_pic

Trattandosi di variabili fortemente correlate, è poco probabile che riescano a sopravvivere il simultaneo inserimento nel modello senza far insorgere problemi di multicollinearità.

Linear regression

Number of obs	3,915
F(8, 3906)	35.41
Prob > F	0.0000
R-squared	0.0711
Root MSE	1.3259

reviews_per_month	Coef.	Robust Std. Err.	t	P>|t|	[95% Conf.	Interval]
rank	.4471376	.0629815	7.10	0.000	.3236579	.5706173
price	-.0005852	.0001877	-3.12	0.002	-.0009532	-.0002173
security_deposit	-.000934	.000102	-9.16	0.000	-.001134	-.000734
avg_dist	-.145039	.024863	-5.83	0.000	-.1937846	-.0962933
host_response_rate	1.239218	.1823149	6.80	0.000	.881777	1.59666
host_acceptance_rate	1.289671	.3010754	4.28	0.000	.699391	1.879951
host_identity_verified	.2394999	.0443373	5.40	0.000	.1525733	.3264264
require_guest_phone_verification	-.2397693	.0764291	-3.14	0.002	-.389614	-.0899247
_cons	-3.224849	.5692694	-5.66	0.000	-4.340943	-2.108756

Linear regression

Number of obs	3,915
F(8, 3906)	34.61
Prob > F	0.0000
R-squared	0.0699
Root MSE	1.3267

reviews_per_month	Coef.	Robust Std. Err.	t	P>|t|	[95% Conf.	Interval]
rank	.4397857	.0629888	6.98	0.000	.3162916	.5632798
price	-.0005861	.0001883	-3.11	0.002	-.0009554	-.0002169
security_deposit	-.0009293	.0001024	-9.08	0.000	-.00113	-.0007286
avg_dist	-.1405528	.0248884	-5.65	0.000	-.1893483	-.0917574
host_response_rate	1.216237	.1818404	6.69	0.000	.8597254	1.572748
host_acceptance_rate	1.327158	.3026933	4.38	0.000	.7337058	1.92061
host_identity_verified	.2268563	.0443524	5.11	0.000	.1399002	.3138124
require_guest_profile_picture	.2194526	.1594318	1.38	0.169	-.0931248	.5320301
_cons	-3.20886	.5708919	-5.62	0.000	-4.328134	-2.089586

Linear regression

Number of obs	3,915
F(9, 3905)	33.02
Prob > F	0.0000
R-squared	0.0731
Root MSE	1.3246

reviews_per_month	Coef.	Robust Std. Err.	t	P>|t|	[95% Conf.	Interval]
rank	.4371033	.0628271	6.96	0.000	.3139263	.5602803
price	-.0005728	.0001875	-3.06	0.002	-.0009403	-.0002052
security_deposit	-.0009403	.0001026	-9.17	0.000	-.0011414	-.0007392
avg_dist	-.1459911	.0248767	-5.87	0.000	-.1947637	-.0972186
host_response_rate	1.231918	.1819804	6.77	0.000	.8751324	1.588704
host_acceptance_rate	1.301812	.3019796	4.31	0.000	.709759	1.893864
host_identity_verified	.2345737	.044307	5.29	0.000	.1477067	.3214408
require_guest_phone_verification	-.3778186	.0754029	-5.01	0.000	-.5256514	-.2299858
require_guest_profile_picture	.5380475	.1684839	3.19	0.001	.2077227	.8683723
_cons	-3.151796	.569113	-5.54	0.000	-4.267583	-2.03601

Il primo output mostra il chiaro effetto deterrenza della richiesta di verifica del numero di telefono sul tasso di domanda (coefficiente negativo e *p-value* pari a 0).
La non significatività nel secondo output del coefficiente della variabile "require_guest_profile_picture" non permette di inferire su quale sia l'effetto di tale variabile.
L'inserimento contestuale di entrambe le variabili fa emergere, come temuto, problemi di multicollinearità, senza aumentare la potenza esplicativa del modello.

Alla luce di queste considerazioni, si è deciso di selezionare come modello esplicativo finale la variante mostrata nel primo output, i cui risultati permettono di affermare con sicurezza che per gli appartamenti completi esista sicuramente un impatto marginale positivo della gentilezza sul tasso di domanda: gli host che riescono a fornire un servizio eccellente catturano sistematicamente più domanda.

7.2. Database completo

Dati gli ottimi risultati ottenuti sul subset degli appartamenti completo, è stato naturale effettuare un tentativo di scalare il modello definitivo sul database completo.

```
Linear regression                               Number of obs     =      6,464
                                                F(8, 6455)        =      63.98
                                                Prob > F          =     0.0000
                                                R-squared         =     0.0798
                                                Root MSE          =     1.3981
```

reviews_per_month	Coef.	Robust Std. Err.	t	P>|t|	[95% Conf.	Interval]
rank	.2527595	.0428793	5.89	0.000	.1687019	.3368172
price	-.0012473	.0001869	-6.67	0.000	-.0016137	-.0008808
security_deposit	-.0007455	.0001546	-4.82	0.000	-.0010486	-.0004424
avg_dist	-.1925568	.0174201	-11.05	0.000	-.226706	-.1584076
host_response_rate	1.049533	.1307582	8.03	0.000	.7932031	1.305862
host_acceptance_rate	2.427945	.1770034	13.72	0.000	2.08096	2.774931
host_identity_verified	.1184307	.036017	3.29	0.001	.0478254	.1890361
require_guest_phone_verification	-.3053018	.0680052	-4.49	0.000	-.4386146	-.171989
_cons	-2.416914	.4054082	-5.96	0.000	-3.211648	-1.62218

Il modello elaborato per gli appartamenti si comporta molto bene per il database generale.
Risulta molto immediato interpretare l'impatto della gentilezza: come nel caso degli appartamenti si può osservare un forte impatto virtuoso sulla domanda, anche se il modulo del coefficiente (che si ricorda riassume l'aumento assoluto del tasso di domanda in risposta ad un aumento unitario della gentilezza) è inferiore rispetto al caso degli appartamenti (0,2527 vs 0,44 nel modello precedente).

Il ruolo delle variabili di controllo è il medesimo che nel caso sopra trattato:

Variabile	Impatto su domanda	Interpretazione
Prezzo	Negativo	Coerente con la relazione tipica prezzo/quantità delle teorie microeconomiche
Deposito cauzionale	Negativo	Un deposito cospicuo è deterrente alla prenotazione
Distanza media	Negativo	I clienti preferiscono gli alloggi con posizione migliore, anche se hanno un prezzo più alto
Tasso di risposta	Positivo	Effetto fiducia/sicurezza
Tasso di accettazione	Positivo	Effetto fiducia/sicurezza
Identità verificata	Positivo	Effetto fiducia/reputazione
Richiesta di verifica numero di cellulare	Negativo	Si tratta di un'azione che comporta uno sforzo da parte del cliente. È un deterrente alla prenotazione

7.2.1. Analisi di robustezza del modello generale

In questo paragrafo si vuole testare la robustezza della relazione tasso di domanda/gentilezza.
Si inizia sostituendo la misura di gentilezza con uno tra gli altri indicatori a disposizione.
Il primo test non è così drastico in quanto la misura "rank_2" non è indipendente rispetto a "rank", si tratta più di una validazione del modello che un vero e proprio *stress test*.

```
Linear regression                               Number of obs     =      6,464
                                                F(8, 6455)        =      66.52
                                                Prob > F          =     0.0000
                                                R-squared         =     0.0831
                                                Root MSE          =     1.3956

------------------------------------------------------------------------------
                             |               Robust
reviews_per_month            |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-----------------------------+------------------------------------------------
                     rank_2  |   .2790649   .0345565     8.08   0.000     .2113227    .3468071
                      price  |  -.0011806   .0001869    -6.32   0.000    -.0015469   -.0008143
           security_deposit  |  -.0007307   .0001532    -4.77   0.000    -.0010311   -.0004304
                   avg_dist  |  -.1956404   .0174152   -11.23   0.000    -.2297799   -.1615009
          host_response_rate |    .990905   .1298614     7.63   0.000     .7363335    1.245476
        host_acceptance_rate |   2.510923   .1795255    13.99   0.000     2.158993    2.862853
      host_identity_verified |   .1069654   .0361187     2.96   0.003     .0361608    .1777699
require_guest_phone_verification |  -.3047852   .0678897    -4.49   0.000    -.4378715    -.171699
                       _cons |  -2.747359   .3682889    -7.46   0.000    -3.469328   -2.025391
------------------------------------------------------------------------------
```

Il modello si adatta perfettamente alla variabile “Rank_2”: addirittura in questa versione l’impatto della gentilezza è ancora più significativo (*t-value* pari a 8 rispetto al valore 5.89 ottenuto nel modello base).

Il primo vero test di robustezza lo si ottiene sostituendo a “rank” la variabile “Score_1”, che è stata generata con una procedura completamente indipendente (si ricorda che “Score_1” è il punteggio di positività assegnato alla recensione dal *tool SID*).

```
Linear regression                               Number of obs     =      6,464
                                                F(8, 6455)        =      69.54
                                                Prob > F          =     0.0000
                                                R-squared         =     0.0866
                                                Root MSE          =      1.393

------------------------------------------------------------------------------
                             |               Robust
          reviews_per_month |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-----------------------------+------------------------------------------------
                     score_1 |   .2084918     .02201     9.47   0.000      .165345    .2516386
                       price |  -.0011027   .0001871    -5.89   0.000    -.0014695   -.0007358
            security_deposit |  -.0007117   .0001514    -4.70   0.000    -.0010086   -.0004149
                    avg_dist |  -.1974693   .0173698   -11.37   0.000    -.2315198   -.1634187
          host_response_rate |   .9394361   .1289323     7.29   0.000      .686686    1.192186
       host_acceptance_rate |    2.59152   .1824196    14.21   0.000     2.233917    2.949123
     host_identity_verified |   .1018085   .0360129     2.83   0.005     .0312113    .1724056
require_guest_phone_verifi~n |  -.2990211   .0675644    -4.43   0.000    -.4314698   -.1665724
                      _cons |  -1.882943   .2690116    -7.00   0.000    -2.410294   -1.355591
------------------------------------------------------------------------------
```

L’output mostra che il modello è sicuramente invariante rispetto alla misura di gentilezza: alla luce di questa analisi si può affermare con sicurezza che esiste un impatto virtuoso della gentilezza sul tasso di domanda.

Per completezza, si riporta anche l’output ottenuto sostituendo “Rank” con “Score_2”, ma i risultati sono pressoché identici agli altri casi.

```
Linear regression                               Number of obs     =      6,464
                                                F(8, 6455)        =      68.72
                                                Prob > F          =     0.0000
                                                R-squared         =     0.0873
                                                Root MSE          =     1.3924

------------------------------------------------------------------------------
                             |               Robust
          reviews_per_month |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-----------------------------+------------------------------------------------
                     score_2 |   .2925637   .0300209     9.75   0.000     .2337129    .3514145
                       price |  -.0011123   .0001864    -5.97   0.000    -.0014777   -.0007468
            security_deposit |  -.0007196   .0001512    -4.76   0.000     -.001016   -.0004233
                    avg_dist |  -.1963166   .0173628   -11.31   0.000    -.2303533   -.1622798
          host_response_rate |   .9233301   .1291399     7.15   0.000     .6701731    1.176487
       host_acceptance_rate |   2.602352   .1845798    14.10   0.000     2.240514     2.96419
     host_identity_verified |    .099397    .035961     2.76   0.006     .0289016    .1698924
require_guest_phone_verifi~n |  -.2963982   .0675072    -4.39   0.000    -.4287346   -.1640618
                      _cons |  -2.309736   .3003904    -7.69   0.000      -2.8986   -1.720871
------------------------------------------------------------------------------
```

Come ultima alterazione del modello base, si vuole mostrare che l'impatto positivo della gentilezza è robusto anche ad una perturbazione della forma funzionale.

La prima perturbazione consiste in una trasformazione logaritmica della misura di gentilezza. Il coefficiente ottenuto grazie alla stima regressiva si interpreta come la variazione assoluta del tasso di domanda in risposta ad una variazione percentuale della misura di gentilezza.
Si auspica che tale coefficiente continui a rimanere positivo e significativo.

Nel secondo output, invece, si può osservare la medesima trasformazione applicata alla variabile dipendente (il tasso di domanda).
In questo caso, il coefficiente pari 0,11592 rappresenta l'aumento percentuale del tasso di domanda in risposta ad una variazione unitaria dell'indicatore di gentilezza.

```
Linear regression                               Number of obs     =      6,464
                                                F(8, 6455)        =      65.58
                                                Prob > F          =     0.0000
                                                R-squared         =     0.0810
                                                Root MSE          =     1.3972

------------------------------------------------------------------------------
                             |               Robust
          reviews_per_month |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-----------------------------+------------------------------------------------
                     logrank |   2.152931   .3110125     6.92   0.000     1.543243    2.762619
                       price |  -.0012341   .0001865    -6.62   0.000    -.0015998   -.0008685
              security_deposit |  -.0007409   .0001541    -4.81   0.000     -.001043   -.0004388
                    avg_dist |  -.1930662   .0174106   -11.09   0.000    -.2271967   -.1589357
          host_response_rate |   1.037748   .1304586     7.95   0.000     .7820064    1.293491
        host_acceptance_rate |   2.444847   .1764732    13.85   0.000     2.098901    2.790793
      host_identity_verified |   .1153793   .0360081     3.20   0.001     .0447915    .1859671
require_guest_phone_verification |  -.3063682   .0680088    -4.50   0.000     -.439688   -.1730484
                       _cons |  -4.864676   .6860797    -7.09   0.000    -6.209619   -3.519732
------------------------------------------------------------------------------
```

```
Linear regression                               Number of obs     =      6,464
                                                F(8, 6455)        =      75.69
                                                Prob > F          =     0.0000
                                                R-squared         =     0.1021
                                                Root MSE          =     .69782

------------------------------------------------------------------------------
                             |               Robust
                      logrev |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-----------------------------+------------------------------------------------
                        rank |     .11592   .0263631     4.40   0.000     .0642397    .1676003
                       price |    -.00075    .000098    -7.65   0.000    -.0009421   -.0005578
              security_deposit |  -.0004758   .0000845    -5.63   0.000    -.0006414   -.0003103
                    avg_dist |  -.0825161   .0096168    -8.58   0.000    -.1013682    -.063664
          host_response_rate |   .5767009   .0696176     8.28   0.000     .4402274    .7131744
        host_acceptance_rate |    1.45132   .1021595    14.21   0.000     1.251053    1.651586
      host_identity_verified |    .076659   .0185927     4.12   0.000     .0402112    .1131068
require_guest_phone_verification |  -.1740822   .0419318    -4.15   0.000    -.2562823    -.091882
                       _cons |  -1.958411   .2418833    -8.10   0.000    -2.432582   -1.484239
------------------------------------------------------------------------------
```

In entrambe le perturbazioni si può vedere come l'impatto marginale della gentilezza continui ad essere positivo e fortemente significativo (*t-value* pari a 6,92 nel primo caso e 4,40 nel secondo).

Come si è già spiegato nei test di robustezza precedenti, si vuole ribadire che ai fini della modellizzazione tali modelli siano subottimali per rappresentare il fenomeno.
In questo senso, lo scopo di queste analisi non era quello di fornire rappresentazioni alternative dello stesso fenomeno, ma piuttosto di riuscire a continuare ad osservare l'effetto virtuoso della gentilezza nonostante la forma funzionale non fosse quella ideale.

7.3. Alloggi stanza

Durante le analisi preliminari, si è iniziato a temere che si potessero presentare serie difficoltà nella modellizzazione degli alloggi stanza.
Tali timori sono stati poi confermati dagli svariati tentativi effettuati, in nessuno dei quali si riusciva a far emergere significatività nella relazione tra tasso di domanda e gentilezza.

Si pensa che tale inefficacia sia da imputarsi all'inadeguatezza della variabile "reviews_per_month" se utilizzata come *proxy* del tasso di domanda.
Per come è costruito, l'indicatore pesa allo stesso modo recensioni relative a soggiorni di durate diverse, e di conseguenza non in tutte le situazioni è in grado di approssimare con successo il vero tasso di occupazione dell'alloggio.

Per gli appartamenti completi ci si aspetta che i valori di durata oscillino relativamente poco intorno alla durata media di soggiorno (pari a 7 giorni per la città di Barcellona, dato pubblicato da Airbnb), e di conseguenza non si verificano grandi distorsioni rispetto alla realtà sfruttando la *proxy* proposta.

Al contrario, è molto probabile che il comportamento dei clienti di alloggi stanza sia molto più eterogeneo, creando potenzialmente uno scostamento rilevante tra tasso di domanda vero e approssimato.

Utilizzare una *proxy* subottimale come variabile dipendente di un modello regressivo è un'operazione molto delicata. Questo perché, se lo scostamento è rilevante, la regressione stima il modello che meglio si adatta a delle osservazioni di un fenomeno "fittizio" che a sua volta non coincide con il fenomeno reale che si vuole esplorare.
Calando questa situazione ipotetica nel caso reale, significa che il modello non sta effettivamente cercando di comprendere le oscillazioni nel tasso di domanda, ma quelle di un fenomeno a sè stante, potenzialmente molto diverso da quello reale.
Il rischio, nel caso peggiore, è che nonostante il set di variabili e la forma funzionale siano idonee a cogliere il fenomeno vero, la distorsione generata dalla *proxy* possa portare alla non significatività di alcuni coefficienti, o addirittura ad inversioni di segno degli stessi.

Analizzando i risultati dell'applicazione del modello generale al subset delle stanze, si pensa che sia proprio questo lo scenario verificato.

```
Linear regression                               Number of obs     =      2,549
                                                F(8, 2540)        =      59.33
                                                Prob > F          =     0.0000
                                                R-squared         =     0.1085
                                                Root MSE          =     1.4617

------------------------------------------------------------------------------
                             |               Robust
           reviews_per_month |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-----------------------------+------------------------------------------------
                        rank |  -.0958554   .0645862    -1.48   0.138    -.2225024    .0307917
                       price |   -.002244   .0005124    -4.38   0.000    -.0032488   -.0012391
            security_deposit |  -.0001459   .0002532    -0.58   0.565    -.0006423    .0003506
                    avg_dist |  -.2458666   .0245093   -10.03   0.000    -.2939268   -.1978063
          host_response_rate |   .3787874    .199234     1.90   0.057    -.0118902     .769465
        host_acceptance_rate |    3.49062   .2048262    17.04   0.000     3.088977    3.892264
      host_identity_verified |  -.0621577   .0591878    -1.05   0.294     -.178219    .0539037
require_guest_phone_verification |  -.3916836   .1365505    -2.87   0.004    -.6594454   -.1239219
                       _cons |   .238168    .600374     0.40   0.692    -.9391045     1.41544
------------------------------------------------------------------------------
```

Dall'output si può osservare come svariate variabile che si pensa abbiano un impatto sul tasso di domanda, abbiano perso significatività ("rank", "security_deposit" e "host_identity_verified") e alcune abbiano subito addirittura una inversione di segno rispetto a ciò che ci si aspettava (i segni dei coefficienti di "rank" e "host_identity_verified" sono entrambi negativi).

Non potendo lavorare per ottimizzare l'approssimazione (è proprio Airbnb che per politica interna ha deciso di non rendere pubblici i dati sulla domanda degli alloggi), l'unica strada percorribile è quella di raffinare il modello accettando un certo livello di inefficienza di partenza.
Tendenzialmente, modificare le variabili di controllo permette di riportare il segno di "rank" coerente con quanto ci si aspetta che sia il suo impatto marginale nella realtà e con quanto ottenuto nel modello generale, senza però riuscire ad ottenere livelli di significatività soddisfacenti.

Si è deciso così di provare ad utilizzare una misura diversa di gentilezza, ottenendo risultati leggermente migliori selezionando come variabile target "Score_2".

```
           reviews_per_month |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-----------------------------+------------------------------------------------
                     score_2 |   .0331174   .0503605     0.66   0.511    -.0656343    .1318692
                       price |  -.0021485   .0005179    -4.15   0.000    -.0031641   -.0011328
            security_deposit |  -.0001509   .0002554    -0.59   0.555    -.0006517    .0003498
                    avg_dist |   -.249389    .024428   -10.21   0.000    -.2972898   -.2014883
          host_response_rate |   .3510127   .1983001     1.77   0.077    -.0378336     .739859
        host_acceptance_rate |   3.601095   .2050849    17.56   0.000     3.198944    4.003246
      host_identity_verified |  -.0751086   .0589098    -1.27   0.202    -.1906247    .0404076
require_guest_phone_verification |  -.4001811   .1376689    -2.91   0.004    -.6701359   -.1302263
                       _cons |  -.7894432   .4554276    -1.73   0.083    -1.682491    .1036041
------------------------------------------------------------------------------
```

Effettuando una trasformazione logaritmica della misura di gentilezza si inizia ad intravvedere il vero impatto della gentilezza.
Il coefficiente che lega il tasso di domanda alla gentilezza è fortemente positivo, e il livello di significatività raggiunge quasi l'80%.

reviews_per_month	Coef.	Robust Std. Err.	t	P>|t|	[95% Conf.	Interval]
logscore_2	.3998051	.3194015	1.25	0.211	-.2265088	1.026119
price	-.0021089	.0005189	-4.06	0.000	-.0031263	-.0010914
security_deposit	-.0001534	.0002561	-0.60	0.549	-.0006556	.0003487
avg_dist	-.250582	.0244153	-10.26	0.000	-.2984579	-.2027062
host_response_rate	.3439896	.1983151	1.73	0.083	-.0448861	.7328654
host_acceptance_rate	3.633037	.2049908	17.72	0.000	3.231071	4.035003
host_identity_verified	-.0788482	.0589131	-1.34	0.181	-.1943708	.0366744
require_guest_phone_verification	-.4011071	.1378972	-2.91	0.004	-.6715095	-.1307046
_cons	-1.338261	.699864	-1.91	0.056	-2.710623	.0341016

L'ulteriore trasformazione in forma logaritmica della variabile dipendente (denominata come "logrev" per semplicità nell'output) ha rappresentato l'ultimo step di raffinamento in cui si è riusciti a migliorare la performance del modello.
Il livello di significatività della gentilezza viene amplificato fino a quasi l'85%.

logrev	Coef.	Robust Std. Err.	t	P>|t|	[95% Conf.	Interval]
logscore_2	.2136914	.1563733	1.37	0.172	-.0929408	.5203237
price	-.0014534	.000218	-6.67	0.000	-.0018808	-.0010259
security_deposit	-.0000637	.0001005	-0.63	0.526	-.0002607	.0001334
avg_dist	-.1067908	.0121716	-8.77	0.000	-.1306581	-.0829235
host_response_rate	.1716312	.1002619	1.71	0.087	-.0249723	.3682347
host_acceptance_rate	1.99413	.1245434	16.01	0.000	1.749913	2.238347
host_identity_verified	-.0522107	.0281141	-1.86	0.063	-.1073396	.0029183
require_guest_phone_verification	-.2813144	.0837189	-3.36	0.001	-.4454786	-.1171502
_cons	-1.409529	.3452451	-4.08	0.000	-2.086519	-.7325381

Nonostante i risultati siano lontani dal caso ideale, alla luce del fatto che la gentilezza ha impatto positivo sulla domanda negli appartamenti, a maggior ragione si pensa che sia un elemento virtuoso negli alloggi stanza, dove ci sono infinite più possibilità di fornire un servizio più "umano" e personalizzato, e sicuramente il rapporto con il padrone di casa è un aspetto importante del soggiorno nel suo complesso.

Il fatto che il modello generale sia molto efficace nonostante la potenziale distorsione da *proxy* è un fatto particolare: cercando di darne una spiegazione ci si è immaginati che probabilmente, dato che il campione a disposizione è molto più esteso (circa il doppio delle osservazioni rispetto al subset delle stanze), scostamenti di segno diverso si siano compensati in modo più consistente.

8. Conclusioni e implicazioni

In questa tesi si è cercato di comprendere quale fosse l'impatto della gentilezza dell'host sulla performance del suo alloggio.
Per farlo, si è elaborato un modello di Pricing che contenesse tra le variabili esplicative un indicatore della gentilezza del padrone di casa.
Tale indicatore è stato estratto a partire dei giudizi contenuti nelle recensioni pubblicate dai clienti successivamente al soggiorno.
I risultati della modellizzazione mostrano che la gentilezza ha sicuramente un impatto positivo sul prezzo: questo significa che un host che sia in grado di migliorare in modo rilevante il servizio fornito al cliente (e soprattutto migliorare la percezione che il cliente ha di tale servizio) può permettersi di imporre un prezzo per notte più elevato.
Una volta verificato che tale effetto esiste, bisogna anche quantificare il premio di prezzo ad esso associato.
I modelli di regressione multipla utilizzati estensivamente nel capitolo 6 hanno permesso di stimare un coefficiente che rappresenta l'elasticità del prezzo rispetto alla gentilezza.
Nel modello ricavato a partire dal database completo, che include sia appartamenti completi che alloggi stanza, l'elasticità è stata stimata pari a 0,34.
Ciò significa che per ogni punto percentuale di miglioramento dell'indicatore di gentilezza, il prezzo di equilibrio di tale alloggio subisce un incremento di 0,34 punti percentuali.

Per calare questi numeri in un esempio verosimile si può immaginare un alloggio con un prezzo per notte di 100 euro, il cui host viene inizialmente classificato con un punteggio di gentilezza pari a 8 (si ricorda che per come è costruito, l'indicatore "rank" mappa i valori nel continuo tra 3 e 10).
Se il padrone di casa fosse in grado di migliorare il proprio livello di gentilezza passando da 8 a 9, a cui corrisponde un aumento del 12,5%, potrebbe imporre un premio di prezzo pari al 4,25%, e il nuovo prezzo di equilibrio sarebbe pari a 104,25 euro.

I risultati presentati nell'esempio permettono di capire che il premio di prezzo è sicuramente non trascurabile, e garantisce un flusso di ricavi accessorio sicuramente ben accetto al padrone di casa.
Detto questo, è anche evidente che si tratti di un effetto piuttosto contenuto, che sicuramente non è in grado di modificare in modo troppo importante il posizionamento di tale alloggio.
Sarebbe stato un po' troppo pretenzioso, infatti, pensare che l'effetto della gentilezza avrebbe permesso ad un host eccezionale ma titolare di una struttura piuttosto mediocre, di imporre lo stesso prezzo di un host più anonimo ma avente a disposizione un alloggio di alto livello.
Questo perché la modellizzazione ha permesso di capire che il meccanismo di formazione del prezzo di equilibrio è fortemente dominato da altri fattori, tra i quali più importanti sono sicuramente la posizione dell'alloggio e le sue caratteristiche strutturali.

Dato che la performance degli alloggi è influenzata non solo dal prezzo ma anche dal loro tasso di occupazione, si è voluto indagare la possibilità che la gentilezza del padrone di casa avesse un impatto anche sulla quantità di domanda che l'alloggio è in grado di catturare.
In modo analogo a quanto effettuato per la variabile prezzo, si è elaborato un modello che permettesse di legare il livello di gentilezza dell'host al tasso di domanda.

I risultati delle regressioni effettuate nel capitolo 7 hanno confermato che il miglioramento dell'interazione con il cliente ha un impatto eccezionale sul tasso di domanda.
In tale analisi è stimato che ogni punto di miglioramento dell'indicatore di gentilezza, comporta un aumento del tasso di domanda pari a 0,25.
Alla luce del fatto che media e mediana della distribuzione empirica del tasso di domanda cadono vicini al valore 2, significa che ogni salto intero dell'indicatore gentilezza comporta un miglioramento del tasso di domanda superiore al 12%.
Questo effetto, combinato al premio di prezzo presentato in precedenza, permette di affermare con sicurezza che un miglioramento della gentilezza abbia un impatto molto rilevante sui guadagni di tale alloggio.

A questo punto però bisogna porsi due interrogativi:

1. Lo sforzo legato al fornire un servizio migliore giustifica il ritorno marginale sulla gentilezza?
2. L'investimento in reputazione e gentilezza è la strada migliore che si possa decidere di intraprendere per migliorare la performance?

Per quanto riguarda il primo interrogativo, la risposta è tendenzialmente affermativa. Normalmente fornire un servizio migliore non richiede investimenti di tipo materiale, ma piuttosto un incremento di impegno da parte del personale. Nonostante non si voglia sottostimare in nessun modo il livello di attenzione e sforzo richiesti al padrone di casa per garantire che il soggiorno dei clienti sia indimenticabile, si pensa che l'entità del ritorno sui ricavi sia tale da rendere conveniente questa trasformazione.

Rispondere alla seconda domanda, invece è molto più complicato.
Si è visto nel capitolo 6 che l'equilibrio di prezzo dell'alloggio è sicuramente dominato da alcuni elementi come la posizione e delle caratteristiche strutturali che manifestano la qualità dell'alloggio.
Per cercare di far leva su questi fattori, le uniche strade percorribili sono investire per ristrutturare il locale (al fine di migliorarne la percezione) oppure acquistare alloggi più appetibili.
Il problema legato a questi interventi è che richiedono tempo e soprattutto una grande quantità di capitale da investire, asset che non è così scontato che il padrone di casa (specie riferendosi ad un host amatore) abbia a disposizione.

Alla luce di queste considerazioni, l'investimento in "gentilezza" sembra essere una strategia assolutamente da attuare, in quanto richiede virtualmente zero capitale, e garantisce un elemento di diversificazione rispetto al gran parte dei padroni di casa (che attualmente forniscono un servizio molto "neutro").

9. Indice delle tabelle

10. Indice delle figure

Referenze

Bibliografia

- O., Bull Adrien (1994). "Pricing a Motel's Location." *International Journal of Contemporary Hospitality Management*, *6*(6), 10–15. https://doi.org/10.1108/09596119410070422
- Hung, W.-T., Shang, J.-K., & Wang, F.-C. (2010). "Pricing determinants in the hotel industry: Quantile regression analysis." *International Journal of Hospitality Management*, *29*(3), 378–384. https://doi.org/https://doi.org/10.1016/j.ijhm.2009.09.001
- Ikkala, T., & Lampinen, A. (2014). "Defining the price of hospitality: Networked hospitality exchange via Airbnb". *Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW*, *February 2014*, 173–176. https://doi.org/10.1145/2556420.2556506
- Gutt, D., & Herrmann, P. (2015). "Sharing Means Caring? Host' Price Reaction to Rating Visibility*." Association for Information Systems* http://aisel.aisnet.org/ecis2015_riphttp://aisel.aisnet.org/ecis2015_rip/54
- Li, Y.; Pan, Q.; Yang, T.; Guo, L. "Reasonable price recommendation on Airbnb using Multi-Scale clustering". *In Proceedings of the 2016 35th Control Conference (CCC), Chengdu, China*, 27–29 July 2016; pp. 7038–7041
- Zhang, Z., Chen, R. J. C., Han, L. D., & Yang, L. (2017). "Key factors affecting the price of Airbnb listings: A geographically weighted approach". *Sustainability (Switzerland)*, *9*(9), 1–13. https://doi.org/10.3390/su9091635
- Magno, F., Cassia, F., & Ugolini, M. M. (2018). "Accommodation prices on Airbnb: effects of host experience and market demand." *TQM Journal*,*30*(5), 608–620. https://doi.org/10.1108/TQM-12-2017-0164

- Tussyadiah, I.P., Pesonen, J., 2015. "Impacts of peer-to-peer accommodation use on travel patterns." *J. Travel. Res*. 55 (8), 1022–1040.
- Cheng, M., & Jin, X. (2019). "What do Airbnb users care about? An analysis of online review comments". *International Journal of Hospitality Management*, 76*, 58-70.*
- Sthapit & Jiménez-Barreto (2018): Sthapit E., Jiménez-Barreto J., "Sharing in the host–guest relationship: perspectives on the Airbnb hospitality experience", *Anatolia, 2018, Volume* 29, [282-284]
- Lin, P. M. C., Fan, D. X. F., Zhang, H. Q., & Lau, C. (2019). "Spend less and experience more: Understanding tourists' social contact in the Airbnb context". *International Journal of Hospitality Management*, *83*(January), 65–73. https://doi.org/10.1016/j.ijhm.2019.04.007
- Sutherland, I., & Kiatkawsin, K. (2020). "Determinants of guest experience in Airbnb: A topic modeling approach using LDA". *Sustainability (Switzerland)*, *12*(8). https://doi.org/10.3390/SU12083402

- Alsudais, A. (2017). “Quantifying the offline interactions between hosts and guests of Airbnb”. *AMCIS 2017 - America’s Conference on Information Systems: A Tradition of Innovation*, *2017-Augus*(August).
- Farmaki, A., & Stergiou, D. P. (2019). “Escaping loneliness Through Airbnb host-guest”. *Tourism Management*, *74*(April), 331333. https://doi.org/10.1016/j.tourman.2019.04.006
- Song, H., Altinay, L., Sun, N., & Wang, X. L. (2018). “The influence of social interactions on senior customers' experiences and loneliness.” *International Journal of Contemporary Hospitality Management*, 30(8), 2773–2790.
- Moon, H., Miao, L., Hanks, L., & Line, N. D. (2019). “Peer-to-peer interactions: Perspectives of Airbnb guests and hosts”. *International Journal of Hospitality Management*, *77*(July 2018), 405–414. https://doi.org/10.1016/j.ijhm.2018.08.004

- Kumar, N., and Benbasat, I. 2006. “The Influence of Recommen- dations on Consumer Reviews on Evaluations of Websites,” *Information Systems Research* (17:4), pp. 425-439.
- Chen, P., Dhanasobhon, S., and Smith, M. 2008. “All Reviews Are Not Created Equal: The Disaggregate Impact of Reviews on Sales on Amazon.com,” *working paper, Carnegie Mellon Uni- versity (available at SSRN: http://ssrn.com/abstract=918083).*
- Clemons, E., Gao, G., and Hitt, L. 2006. “When Online Reviews Meet Hyperdifferentiation: A Study of the Craft Beer Industry,” *Journal of Management Information Systems* (23:2), pp. 149-171.
- Nakayama, M., Sutcliffe, N., & Wan, Y. (2010). “Has the web transformed experience goods into search goods*?”* Pp. 251–262. https://doi.org/10.1007/s12525-010-0041-z

Sitografia

- “Inside Airbnb”, http://insideairbnb.com
- “Come stanno cambiando le città per colpa di Airbnb”, Gianfrancesco Turano. *https://espresso.repubblica.it/inchieste/2017/12/11/news/come-stanno-cambiando-le-citta-per-colpa-di-airbnb-1.315812*