

Dipartimento di Informatica
Corso di laurea in Informatica

Ingegneria della Conoscenza

ANALISI DATI: AFFITTI IN BERLINO

DOCUMENTAZIONE



GitHub Repository: [BerlinRentals \(github.com\)](https://github.com/BerlinRentals)

Progetto a cura di:

Rosanna Fracchiolla (758347) r.fracchiolla12@studenti.uniba.it

Daniele Grandolfo (758354) d.grandolfo4@studenti.uniba.it

Anno Accademico **2023-2024**

SOMMARIO

1	Introduzione	2
2	Implementazioni	3
2.1	Strumenti utilizzati	3
2.2	Struttura.....	4
2.3	Approfondimenti teorici	5
3	Ragioni della Selezione delle Conoscenze Teoriche	6
3.1	Clustering (k-means)	6
3.2	Belief network	7
4	Meccanismi Operativi	8
4.1	Preprocessing.....	8
5	Requisiti.....	11
6	Guida all'esecuzione.....	11
6.1	Configurazione del progetto	11
6.2	Esecuzione del progetto	11
6.3	Esempi	12
7	Conclusione	15



Il settore dell'ospitalità è in costante evoluzione e richiede approcci innovativi nell'analisi dei dati per migliorare l'esperienza degli utenti. Airbnb, fondata nel 2008, ha trasformato il mercato degli affitti a breve termine, offrendo milioni di opzioni di alloggio in oltre 220 paesi. Per gestire e comprendere questa vasta quantità di dati, il progetto indipendente "Inside Airbnb" raccoglie e analizza informazioni pubblicamente disponibili dalla piattaforma, offrendo trasparenza su disponibilità, prezzi, recensioni e altro.

In questo contesto, abbiamo deciso di sviluppare una Base di Conoscenza per esplorare e ottimizzare i dati forniti da "Inside Airbnb". La nostra iniziativa si concentra su un'analisi dettagliata e trasparente delle attività di Airbnb svolte in particolare su Berlino, una delle città europee più vivaci e culturalmente ricche, che ha visto una crescita significativa nel settore degli affitti a breve termine.

Utilizzando metodologie avanzate di apprendimento automatico, miriamo a verificare dettagliatamente attributi essenziali come il numero di letti e bagni negli annunci di Airbnb a Berlino. Inoltre, intendiamo esplorare aspetti più complessi che influenzano l'attrattiva degli alloggi, come la disponibilità di strutture per la preparazione di cibi, connessioni internet ad alta velocità e parcheggi. La nostra analisi utilizzerà tecniche avanzate di machine learning e data mining per estrarre informazioni rilevanti dai dati grezzi, identificando tendenze, modelli e anomalie.

L'obiettivo è fornire dati utili per supportare la formulazione di politiche informate e efficaci sulla regolamentazione degli affitti a breve termine. La nostra iniziativa intende contribuire significativamente alla comprensione del panorama degli affitti a breve termine a Berlino, offrendo una risorsa preziosa per migliorare l'esperienza degli utenti e gestire in modo più efficace l'impatto di Airbnb sulla città. Con un'analisi dettagliata e trasparente, speriamo di fornire insight che possano aiutare a ottimizzare le politiche urbane e a soddisfare le esigenze dei viaggiatori e dei residenti locali.



2 IMPLEMENTAZIONI

2.1 STRUMENTI UTILIZZATI

Al fine di condurre il nostro Caso di Studio in maniera efficace e svolgere un'analisi approfondita del set di dati, abbiamo deliberato di utilizzare i seguenti strumenti: la scelta del linguaggio di programmazione per lo sviluppo del nostro progetto è ricaduta su Python. Questa selezione è stata motivata dalla vasta disponibilità di librerie e dalla facilità d'uso, con particolare attenzione a quelle strettamente rilevanti per i nostri obiettivi di ricerca. Tra le librerie chiave impiegate, figurano:

1. **Pandas:** Utilizzata per la gestione del dataset e le attività di preprocessing. La sua capacità di gestire grandi quantità di dati in modo rapido e flessibile ha reso possibile una pulizia e una trasformazione del dataset senza intoppi.
2. **PySwip:** Questa libreria è stata fondamentale per la costruzione e l'interrogazione della nostra Knowledge Base, integrando efficacemente la programmazione in Python con le funzionalità di Prolog.
3. **Scikit-learn:** Abbiamo fatto uso di Scikit-learn per la parte relativa al clustering dei dati. Questa libreria ci ha fornito gli strumenti necessari per implementare metodi avanzati di analisi dei dati, in particolare quelli legati al clustering.
4. **Pgmpy:** Utilizzata per l'inferenza probabilistica, Pgmpy è stata fondamentale per la costruzione e l'analisi della Belief Network. Ci ha permesso di modellare relazioni probabilistiche complesse tra le diverse caratteristiche delle stanze. Grazie a Pgmpy, siamo stati in grado di eseguire inferenze e aggiornamenti delle credenze basate sui dati osservati.

Per quanto riguarda la creazione della struttura della Belief Network, abbiamo scelto di addestrare un modello di rete Bayesiana direttamente da Python, evitando la necessità di strumenti esterni. Questo processo ci ha permesso di costruire e verificare le distribuzioni di probabilità condizionate (CPDs) e di creare una Belief Network robusta.

Attraverso l'uso combinato di questi strumenti, siamo riusciti a condurre un'analisi approfondita e dettagliata del set di dati, fornendo insight significativi e supportando i nostri obiettivi di ricerca con metodologie avanzate e tecnologie all'avanguardia.



Package **BeliefNetwork**:

Il modulo "*beliefNetwork*" crea ed addestra un modello di rete bayesiana, calcolandone le *CPD* (*Conditional Probability Distributions*), esegue inferenze sulla rete utilizzando il metodo di eliminazione delle variabili e restituisce i risultati basati sulle preferenze specificate come input.

Il modulo "*ui*" implementa un'interfaccia utente per interrogare la Knowledge Base tramite query Prolog e condurre inferenze su una rete bayesiana. Gli utenti possono inserire query Prolog o specificare preferenze tramite attributi discreti e booleani per ottenere risultati statistici basati su una rete bayesiana predefinita.

Package **clustering**:

Il modulo "*clustering*" esegue il clustering utilizzando l'algoritmo K-Means a partire da un file CSV. Il numero di cluster viene determinato tramite il metodo del gomito, che identifica il numero ottimale di cluster per la suddivisione dei dati. Successivamente, aggiunge le etichette dei cluster al dataframe e salva le modifiche.

Package **datasets**:

Il package in questione contiene principalmente il dataset originale, nonché il file "*listings.csv*". Con le fasi di preprocessing, questo package viene arricchito di ulteriori file CSV contenenti dati sempre più leggibili. Inoltre, ospita anche il file Prolog contenente le regole che consentono la predisposizione delle query.

Package **KnowledgeBase**:

Il modulo "*Kb*" consente la creazione del file Prolog. Include informazioni sulle caratteristiche delle stanze e definisce regole basate su tali dati, come la disponibilità di prenotazioni, le caratteristiche delle stanze e i criteri di prezzo.

Package **preprocessing**:

Il modulo "*amenities*" definisce una funzione per creare una matrice di booleani che rappresenta la presenza o l'assenza di determinati comfort in un dataframe di annunci immobiliari, utilizzando una soglia per filtrare i comfort meno comuni.

Il modulo "*bayesianDataframe*" applica una serie di trasformazioni al dataset iniziale per prepararlo all'uso in una belief network. Le colonne non necessarie vengono eliminate, i valori delle colonne vengono discretizzati per creare categorie, e infine vengono effettuate operazioni di pulizia e concatenazione dei dati.

Il modulo "*cleaning*" esegue il preprocessing del dataset, preparandolo per l'uso in una belief network e per l'analisi di clustering, salvando i risultati in file CSV separati.



Il modulo "*clusteringDataframe*" esegue il preprocessing per il clustering di un dataset, includendo la creazione di una matrice di comfort, la rimozione di colonne non numeriche e non necessarie, l'applicazione della PCA (Principal Component Analysis) per ridurre le dimensioni e la normalizzazione dei dati. Il risultato è un dataframe pronto per l'analisi di clustering.

2.3 APPROFONDIMENTI TEORICI

Knowledge Base (KB):

Una Knowledge Base è un archivio centralizzato di informazioni strutturate e organizzate, utilizzato per raccogliere, gestire e condividere conoscenze su un particolare argomento o ambito. Nel nostro caso, questa funzionalità è fondamentale per chiunque stia cercando una sistemazione specifica in base a requisiti particolari, come il numero di letti, la presenza di una cucina o la disponibilità di servizi aggiuntivi. Gli utenti possono interrogare la KB attraverso query mirate, ottenendo risposte accurate e pertinenti in tempo reale.

Clustering:

Il clustering è una tecnica di apprendimento che raggruppa oggetti simili tra loro. Nel nostro progetto, gli utenti possono trovare rapidamente gruppi di stanze con caratteristiche simili, come valutazioni o comodità specifiche, migliorando l'efficienza della ricerca. Abbiamo utilizzato l'algoritmo K-means per suddividere i dati in cluster ottimali. Per determinare il numero ideale di cluster, abbiamo applicato il metodo dell'Elbow, che identifica il punto in cui l'aggiunta di ulteriori cluster non migliora significativamente la varianza intra-cluster. Questo approccio garantisce gruppi di stanze ben distinti e rappresentativi.

Belief Network:

Una Belief Network, o rete bayesiana, è un modello probabilistico utilizzato per rappresentare le relazioni tra diverse variabili casuali. Per noi, è stato cruciale per identificare correlazioni nascoste tra le caratteristiche delle stanze. Questo ci permette di calcolare probabilità a posteriori sulla qualità della stanza, basate su preferenze espresse dagli utenti. Utilizzando le reti bayesiane, possiamo fornire raccomandazioni personalizzate e informazioni dettagliate, come la probabilità che una stanza soddisfi determinate preferenze in base ai dati storici. Questo strumento migliora l'accuratezza delle raccomandazioni e la soddisfazione dell'utente.

In sintesi, la combinazione di Knowledge Base, Clustering e Belief Network promette di fornire agli utenti uno strumento potente per esplorare, valutare e selezionare le opzioni di alloggio più adatte alle loro esigenze. Questo approccio integrato migliora significativamente l'esperienza nel settore degli affitti a breve termine a Berlino, consentendo agli utenti di prendere decisioni informate e personalizzate, ottimizzando il processo di ricerca e selezione degli alloggi. La nostra iniziativa mira a rendere l'analisi dei dati più trasparente ed efficace, offrendo un valore aggiunto sia per i viaggiatori che per gli operatori del settore.

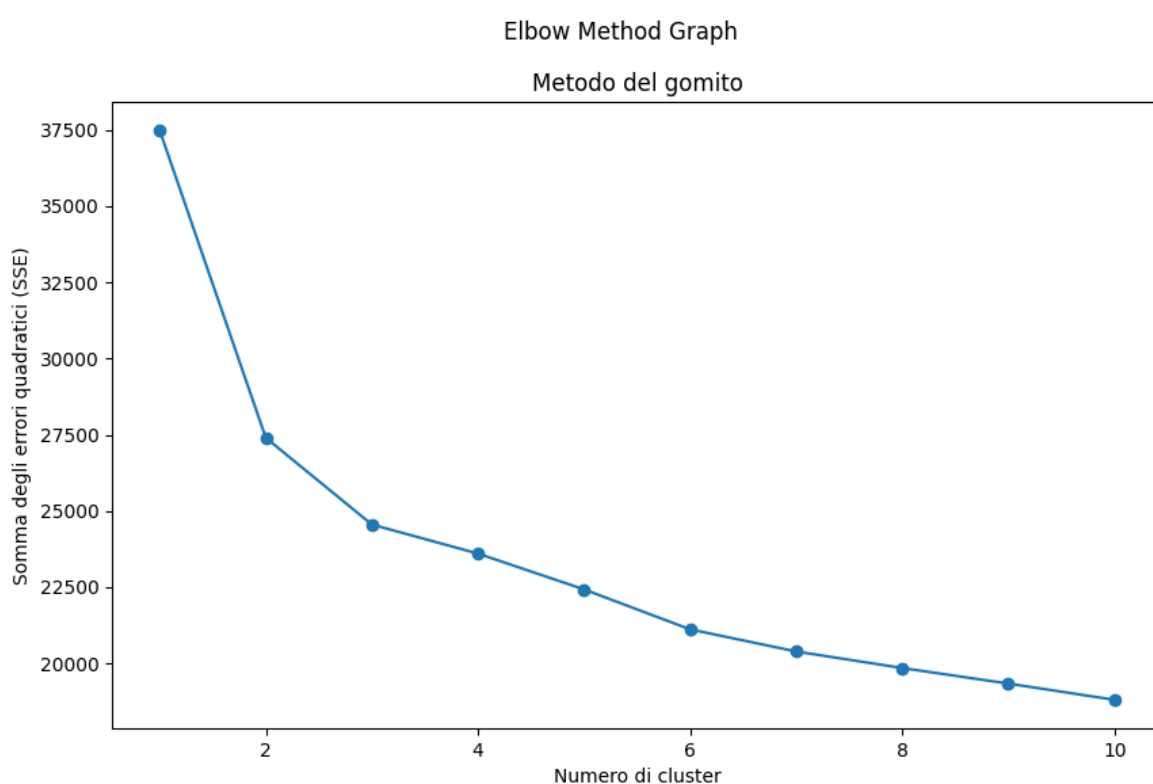


3 RAGIONI DELLA SELEZIONE DELLE CONOSCENZE TEORICHE

3.1 CLUSTERING (K-MEANS)

Utile per identificare e raggruppare stanze che condividono caratteristiche simili. Più nello specifico, abbiamo impiegato l'algoritmo K-Means.

L'algoritmo K-Means prende in input un insieme di dati ed il numero desiderato di clusters (k) in cui suddividere i dati. Dopodichè divide i dati in k cluster, assegnando ogni punto al cluster con il centroide più vicino. Iterativamente, ricalcola i centroidi come la media dei punti assegnati e riassegna i punti fino a convergenza, minimizzando la somma delle distanze quadrate tra punti e centroidi.



La scelta del numero ottimale di classi k è cruciale e può essere determinata utilizzando il "metodo del gomito". Questo metodo consiste nel tracciare un grafico dell'errore quadratico in funzione del numero di cluster e individuare il punto in cui l'aggiunta di ulteriori cluster non fornisce un miglioramento significativo, identificato come il "gomito" della curva. Abbiamo calcolato il numero ottimale di cluster analizzando le differenze successive nel cambiamento dell'errore, individuando il punto in cui l'errore diminuisce in modo significativo.



3.2 BELIEF NETWORK

Nel nostro progetto, abbiamo deciso di utilizzare la Belief Network per analizzare i dati e comprendere le relazioni tra le diverse features. Questo approccio ci consente di visualizzare in modo chiaro e strutturato come le varie caratteristiche sono correlate tra loro. Inoltre, la realizzazione della Belief Network ci offre la possibilità di eseguire interrogazioni basate sull'inferenza probabilistica. Questo significa che possiamo ottenere informazioni più approfondite sui dati e fare previsioni basate sulle relazioni probabilistiche identificate nella rete. In sostanza, l'utilizzo della Belief Network ci consente di esplorare e comprendere meglio la complessità dei nostri dati, fornendo una base solida per le nostre analisi e decisioni future.

Una struttura concettuale di notevole rilievo è rappresentata dalla rete bayesiana. Questo modello, delinea una distribuzione di probabilità congiunta su un insieme di variabili, alcune delle quali interdipendenti. La rete visualizza chiaramente le connessioni tra le caratteristiche, offrendo una serie di probabilità condizionate. Questo ordinamento delle caratteristiche si basa su una distinzione tra variabili genitore e figlie, dove i genitori costituiscono il minimo insieme di predecessori di una variabile figlia, garantendo che gli altri predecessori siano condizionalmente indipendenti dalla variabile figlia stessa, tenuto conto dei genitori.

Dopo aver completato il processo di apprendimento della struttura della Belief Network, gli utenti hanno l'opportunità di formulare query basate sulle loro preferenze personali riguardanti varie proprietà delle strutture, come il quartiere di ubicazione, lo status di "superhost" dell'host e la disponibilità di diversi servizi. Queste preferenze vengono utilizzate come evidenze per l'algoritmo di Variable Elimination, dove la variabile da predire è il "review scores rating", ovvero la media delle recensioni rilasciate dagli utenti per la struttura in questione.

In pratica, ciò significa che le preferenze espresse dall'utente vengono utilizzate per calcolare la probabilità che le camere presenti nel dataset abbiano un determinato livello di qualità. Questo permette agli utenti di valutare con una certa affidabilità la probabilità che una determinata struttura soddisfi le loro aspettative di qualità e possa piacere loro.

La probabilità calcolata sarà quindi $P(\text{review_scores_rating} | \text{preferenze utente})$, fornendo agli utenti un mezzo potente per prendere decisioni informate sulla scelta dell'alloggio.



4 MECCANISMI OPERATIVI

4.1 PREPROCESSING

La fase iniziale del processo è stata il *preprocessing* dei dati, che consiste in una serie di operazioni mirate a pulire e organizzare le informazioni, per renderle utilizzabili nelle analisi successive. Questo processo ha coinvolto diverse attività cruciali:

- **Riduzione del dataset:** Selezione e mantenimento delle informazioni rilevanti, eliminando dati ridondanti o non necessari.
- **Gestione dei comforts:** Creazione di una matrice di booleani che rappresenta la presenza o assenza di specifici comfort nelle strutture, utilizzando una soglia per filtrare i comfort meno comuni.
- **Preparazione per il clustering:** Applicazione di tecniche di preprocessing per il clustering, come la normalizzazione dei dati, la rimozione delle colonne non numeriche e non necessarie, e l'applicazione della *PCA (Principal Component Analysis)* per ridurre le dimensioni.
- **Costruzione di un dataset utile alla Belief Network:** Preparazione dei dati per l'uso in una Belief Network, che include la discretizzazione delle variabili, la pulizia delle colonne e la creazione di nuove variabili.

Questi passaggi chiave hanno contribuito a rendere i dati più comprensibili e adatti all'analisi, producendo un dataset strutturato e ottimizzato, pronto per essere esplorato attraverso varie tecniche di analisi e modellazione.

4.1.1 RIDUZIONE DEL DATASET

La prima attività consiste nella pulizia del dataset, eliminando le colonne considerate poco rilevanti ai fini del nostro studio. Questa fase è stata fondamentale per ridurre il rumore nei dati e concentrarci solo sulle informazioni più pertinenti. A tal proposito, il dataset è stato filtrato eliminando i valori non inerenti all'affitto a breve termine di interi appartamenti, ovvero le righe che fornivano dati su singole stanze.

Per affrontare i dati mancanti all'interno del dataset, è stata adottata una strategia di riempimento utilizzando *Simple Imputer*: una classe utile a mantenere l'integrità dei dati, evitando di scartare intere righe o colonne a causa della mancanza di alcuni valori, garantendo un dataset pronto per successive analisi. *Simple Imputer* permette, infatti, di sostituire i valori mancanti con dei valori calcolati in base a strategie differenti: per le features numeriche i valori mancanti sono stati rimpiazzati dalla media degli altri valori presenti nella stessa colonna; per le features categoriche, invece, con il valore più frequente presente nella colonna.



Infine, è stata introdotta una colonna per identificare le stanze situate nelle zone centrali della città, basandoci su una lista creata attraverso una mappa di Berlino. Quest'ultima colonna è stata utile per valutare, ad esempio, se la localizzazione in città influisce sul prezzo delle stanze.

4.1.2 GESTIONE DEI COMFORTS

Anche la colonna contenente informazioni sui servizi offerti dalle strutture ha subito una pulizia. Per semplificare e organizzare questi dati, è stata eseguita un'operazione testuale per pulire e standardizzare le descrizioni dei servizi.

Dopodichè, abbiamo eliminato le colonne corrispondenti ai servizi presenti in meno del 30% delle strutture totali. Questo threshold ha permesso di concentrarsi sui servizi più comuni e rilevanti, riducendo la complessità del dataset.

Si è poi costruita una matrice binaria dei comforts, dove ogni colonna rappresenta un servizio e ogni riga rappresenta una struttura. In questa matrice, la presenza del servizio è stata indicata dal valore '1'. Quest'ultima è stata poi integrata nel dataset pulito, rendendo più efficiente la gestione e l'analisi di queste informazioni.

4.1.3 PREPARAZIONE PER IL CLUSTERING

L'ultimo preprocessing è stato dedicato al clustering. Si sono selezionate le principali feature per valutare la similarità tra le camere, (prezzo, numero di recensioni e rating della struttura). Si è applicata la *Principal Component Analysis (PCA)*, una tecnica utilizzata per ridurre la dimensionalità dei dati, e il *MinMaxScaler*, per scalare le feature in modo che siano comprese in un intervallo ottimale.

Questi passaggi hanno ottimizzato la gestione delle features per il clustering, introducendo una colonna con il cluster relativo per ciascuna stanza all'interno del dataset, ottenuto dall'operazione di cleaning.



4.1.4 COSTRUZIONE DI UN DATASET UTILE ALLA BELIEF NETWORK

La costruzione di un dataset utilizzabile per la Belief Network ha richiesto ulteriori operazioni di preprocessing. Oltre all'eliminazione di colonne non funzionali all'inferenza probabilistica, si è proseguiti verso la discretizzazione di alcune delle colonne restanti, rendendo i valori numerici e testuali più comprensibili. (Esempio: Colonna *'review_scores_rating'* scomposta negli intervalli *'low_rating'*, *'nice_rating'* e *'top_rating'*)

Infine, per implementare il metodo di inferenza "Variable Elimination", si sono definiti intervalli che suddividono il dataframe in modo omogeneo. Questi intervalli permettono di gestire meglio la complessità dei dati e di facilitare le analisi probabilistiche.



5 REQUISITI

Per eseguire correttamente l'applicativo sviluppato per il progetto di Ingegneria della Conoscenza, è necessario soddisfare una serie di requisiti di sistema e configurazioni specifiche.

REQUISITI DI SISTEMA

- ✓ **Versione di Python:** Python 3.12+
- ✓ **Strumenti aggiuntivi:** Git Bash (per utenti Windows)

6 GUIDA ALL'ESECUZIONE

6.1 CONFIGURAZIONE DEL PROGETTO

1. **Installazione di SWI-Prolog:** Prima di procedere con l'installazione e l'esecuzione del progetto, è necessario installare SWI-Prolog sulla propria macchina. È possibile scaricare l'installer dal seguente link: [SWI-Prolog Download](#)
2. **Clonazione del Repository:** Utilizzare il terminale o Git Bash (per utenti windows) per clonare il repository del progetto.
3. **Creazione e attivazione del Virtual Environment:** Creare un ambiente virtuale per isolare le dipendenze del progetto

```
python -m venv venv  
source venv/Scripts/activate
```

4. **Installazione delle Dipendenze:** Installare tutte le dipendenze necessarie utilizzando il file 'requirements.txt' fornito nel repository

```
pip install -r requirements.txt
```

6.2 ESECUZIONE DEL PROGETTO

1. **Esecuzione delle fasi di preprocessing:**
 - **ESECUZIONE AUTOMATICA:** Per eseguire automaticamente tutte le fasi di preprocessing è possibile lanciare il seguente comando:

```
python preprocessing/preprocessing.py
```

- **ESECUZIONE MANUALE:** Per eseguire manualmente ogni singola fase di preprocessing è possibile eseguire la seguente lista di comandi individualmente:

```
python preprocessing/cleaning.py  
python clustering/clustering.py <numero di iterazioni (opzionale)>  
python KnowledgeBase/Kb.py
```

2. **Interfaccia Utente per le queries:** Per avviare l'interfaccia utente che permette di eseguire query sulla Knowledge Base e condurre inferenze sulla Belief Network

```
python BeliefNetwork/ui.py
```



6.3.1 QUERIES

- Quali sono gli appartamenti che rispettano un determinato intervallo di prezzo?

OUTPUT :

Index	Room
0	18
1	24
2	25
3	35
4	39
5	59
6	60
7	61
8	77
9	102
10	106
11	155

`price_range(Room, "expensive")`

- Quali sono gli appartamenti dotati di un determinato comfort?

OUTPUT :

Index	Room
0	1
1	2
2	3
3	5
4	6
5	8
6	9
7	11
8	12
9	13
10	14
11	16

`amenities(Room, "wifi")`



- Quali sono gli appartamenti definiti come piccoli?
(max 2 camere da letto, max 3 letti, max 1 bagno)

OUTPUT:

Index	Result
0	1
1	3
2	5
3	6
4	7
5	11
6	12
7	14
8	18
9	23
10	26
11	29

`small_rooms(Result)`

- Quali sono gli appartamenti definiti come grandi?
(min 4 camere da letto, min 4 letti, min 2 bagni)

OUTPUT:

Index	Result
0	2
1	16
2	39
3	204
4	419
5	450
6	600
7	660
8	914
9	978
10	991
11	1208

`big_rooms(Result)`

- Le stanze indicate, sono simili (in base al cluster associato)?

```
similar_rooms(10,99)
false
```

`similar_rooms(X,Y) :- cluster(X,C),cluster(Y,D), C = D`



6.3.2 INFERENCES

Dopo aver acquisito la struttura della *Belief Network*, gli utenti possono effettuare query basate sulle loro preferenze personali riguardo diverse caratteristiche delle strutture, come il quartiere di ubicazione, lo status di "superhost" dell'host e la disponibilità di vari servizi.

Queste preferenze degli utenti vengono impiegate come evidenze per l'algoritmo di Variable Elimination, dove la variabile da prevedere è "*review scores rating*", ossia la media delle recensioni date dagli utenti per quella specifica struttura. In pratica, ciò significa che le preferenze degli utenti vengono utilizzate per calcolare la probabilità che le camere presenti nel dataset abbiano un certo livello di qualità. Questo permette agli utenti di valutare con una certa affidabilità la probabilità che una determinata struttura soddisfi le loro aspettative di qualità e risulti di loro gradimento.

La probabilità calcolata sarà quindi $P(\text{review_scores_rating}|\text{preferenze utente})$.

- Ricerca di alloggi gestiti da *superhost*, che si trovino in *Alexanderplatz*, rientrino nella fascia di prezzo *media* e consentano una connessione *Wi-Fi*.

neighbourhood_cleansed = alexanderplatz, class_of_price = medium, host_is_superhost = True, wifi = True

```
neighbourhood_cleansed = alexanderplatz, class_of_price = medium, host_is_superhost = True, wifi = True
RATING      PROBABILITY
top_rating   0.2518
nice_rating  0.1681
good_rating  0.5596
low_rating   0.0205
```

- Ricerca di alloggi che si trovino in *Alexanderplatz*, rientrino nella fascia di prezzo *costoso*, abbiano *tante* recensioni e consentano soggiorni a lungo termine.

neighbourhood_cleansed = alexanderplatz, class_of_price = expensive, number_of_reviews = lot_of_reviews, long_term_stays_allowed = True

```
neighbourhood_cleansed = alexanderplatz, class_of_price = expensive, number_of_reviews = lot_of_reviews, long_term_stays_allowed = True
RATING      PROBABILITY
top_rating   0.3024
nice_rating  0.6976
good_rating  0.0
low_rating   0.0
```



Lo scopo della ricerca svolta è l'ottimizzazione dei servizi offerti da Airbnb nella zona di Berlino, avvalendoci di tecniche avanzate di apprendimento automatico: partendo dalla definizione di una base di conoscenza, strutturando e organizzando le informazioni relative alle diverse proprietà registrate.

Sono state adottate tecniche di apprendimento supervisionato in forma probabilistica, come le Belief Network, per comprendere le relazioni complesse tra le varie caratteristiche delle camere e dei servizi offerti. Ciò ci ha consentito di effettuare interrogazioni sofisticate sulla nostra base di conoscenza e di ottenere informazioni dettagliate sui servizi, migliorando così la qualità complessiva dell'esperienza dell'utente.

Inoltre, sono state impiegate tecniche di apprendimento non supervisionato, come il clustering con l'algoritmo K-Means, per identificare camere simili e raggrupparle in base alle loro caratteristiche comuni, permettendo di fornire raccomandazioni personalizzate agli utenti, migliorando ulteriormente l'esperienza di ricerca e prenotazione delle camere su Airbnb e consentendo una ricerca più accurata da parte degli utenti.

È possibile sfruttare questo prodotto per evidenziare le possibili migliorie da apportare ai servizi già offerti da portali per affitti a breve termine. Enfatizzando anche l'aspetto della *user experience*, è stato possibile andare oltre le semplici caratteristiche delle camere e di considerare anche le preferenze e le abitudini degli utenti, contribuendo così a migliorare l'esperienza complessiva di ricerca dell'appartamento perfetto.

