

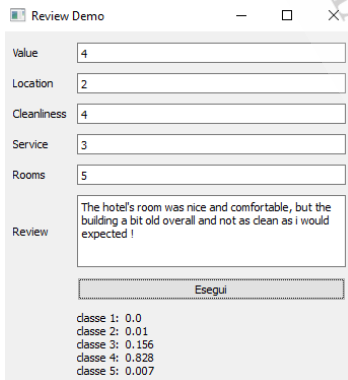
# **Applicazione di una Rete Bayesiana alle recensioni di TripAdvisor**

Bertini Simone, Moretti Luca, Nigrelli Riccardo

7 luglio 2019

# Obiettivo

Sviluppare una demo che permetta di predire la valutazione di una recensione di un hotel fatta da un utente.



Review Demo

Value: 4

Location: 2

Cleanliness: 4

Service: 3

Rooms: 5

Review: The hotel's room was nice and comfortable, but the building a bit old overall and not as clean as i would expected !

Esegui

classe 1: 0.0  
classe 2: 0.01  
classe 3: 0.156  
classe 4: 0.828  
classe 5: 0.007

Figura 1: Interfaccia della demo

# Passi Eseguiti

- rappresentare in modo vettoriale le recensioni del dataset
- selezionare le features
- modellare il problema utilizzando una rete bayesiana
- effettuare o indurre una stima dei parametri e delle CPT
- stimare le performance del modello
- verificare il corretto funzionamento del modello attraverso una demo

# Dataset

Il dataset fornito è composto da:

- 1459 blocchi di recensioni per il Training
- 299 blocchi di recensioni per il Testing

Ogni blocco è composto da :

- un header con Rating medio, Prezzo medio e URL dell'Hotel
- un numero variabile di recensioni comprendenti:
  - una recensione scritta
  - dei metadati con punteggio da 1 a 5

# Selezione Features

Sono state estratte ed analizzate tutte le recensioni scritte ed eseguite le seguenti operazioni:

- conversione del testo in minuscolo
- rimozioni di tutte le forme contratte
- conversione delle parole in token
- rimozione delle stopwords
- rimozione di tutti i simboli di punteggiatura

# Selezione Features

I termini rilevati con maggiore frequenza e con maggiore influenza sono i seguenti:

Parola	Frequenza
great	143.473
good	110.660
comfortable	27.861
clean	48.595
bad	18.381
old	16.955

Tabella 1: Parole selezionate con relativa frequenza

# Rappresentazione Vettori

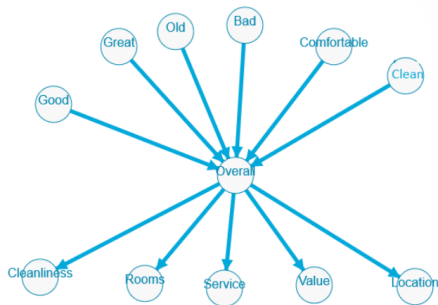
I vettori sono stati costruiti per ogni recensione del dataset di Training e contengono 12 parametri:

- 6 metadati
- 6 keywords

I parametri dei metadati contengono il loro valore assegnato dall'utente (da 1 a 5); mentre quelli delle keywords la presenza (1) o meno (0) nella recensione scritta. Infine sono state filtrate ed eliminate tutte le recensioni, e relativi vettori, con presenza di metadati di valore -1.

# Modello Knowledge-Based

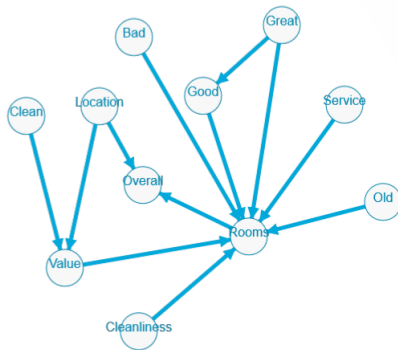
Il primo modello è stato scritto a mano e considera i metadati dipendenti dall'Overall che a sua volta è influenzato dalle keywords significative che sono state selezionate. Le CPT del modello sono state stimate tramite la libreria libpgm che utilizza come parametri la struttura della rete e i dati riguardanti le reviews.





## Modello Indotto

Il secondo modello è stato indotto tramite la libreria libpgm dal Dataset analizzato. Il modello è stato poi lievemente modificato tramite approccio Knowledge-Based per modificare alcuni archi. Le CPT sono state generate come per il modello Knowledge-Based.



# Esperimenti

Prima di testare il modello ottenuto sui vettori delle recensioni della cartella di Testing sono state eliminate quelle con metadati con valore -1. Il test è stato eseguito tramite un task di inferenza esatta con parametri:

- Query: il metadato Overall
- Evidenze: i metadati presenti nella recensione e la presenza o meno delle keywords nella recensione scritta

Il modello è stato valutato contando quante volte classifica nel modo corretto la recensione rispetto a quante volte invece sbaglia.

# Performance del Modello

Le performance del modello sono state calcolate per accuratezza ed hanno dato i seguenti risultati:

Classe	Modello Knowledge-Based	Modello Indotto
1	83%	68%
2	62%	36%
3	61%	20%
4	66%	73%
5	85%	73%

Tabella 2: Confronto dell'accuratezza

Con un risultato complessivo pari a 83% per il modello Knowledge-Based e 71% per il modello indotto.

# Analisi Risultati

- le analisi tengono conto del fatto che i dataset di test sono sbilanciati verso le classi 4 e 5;
- performance migliori per il modello Knowledge-Based in quanto ci si focalizza sull'overall e non sulla globalità del modello;

# Conclusioni

- le parole influenzino molto meno il modello rispetto ai metadati;
- aumento performance se si passasse da un problema multiclasse a un problema binario;
- approcci più sofisticati di scelta delle keywords potrebbero migliorare le performance;
- nonostante non avessimo conoscenze specifiche o approfondite del dominio, il modello risultante ha discrete performance.