RELAZIONE DEL PROGETTO DI MODELLI PROBABILISTICI

Università degli Studi di Milano-Bicocca

Dipartimento di Informatica Sistemistica e Comunicazione

Autori:

Andrea Vegetti, 794298

Ricardo Matamoros, 807450

Sommario

[1 Introduzione 2](#_Toc12217405)

[2 Data Management 4](#_Toc12217406)

[2.1 Descrizione e considerazioni sui dati 4](#_Toc12217407)

[2.2 Sentiment analysis 5](#_Toc12217408)

[3 Generazione Dei Modelli 7](#_Toc12217409)

[3.1 Naive Bayes 7](#_Toc12217410)

[3.2 Rete bayesiana modellata secondo conoscenze a priori 8](#_Toc12217411)

[3.3 Rete bayesiana generata con R 9](#_Toc12217412)

[4 Confronto Delle Performance 11](#_Toc12217413)

[5 Rating Complessivo 15](#_Toc12217414)

[6 Conclusioni 16](#_Toc12217415)

# Introduzione

Il progetto che è stato scelto per questo lavoro è quello riguardante TripAdvisor, il cui l’obiettivo è di calcolare il rating di un determinato hotel partendo dalla classificazione delle sue recensioni. Questo passaggio viene eseguito tramite l’utilizzo di diversi tipi di modelli e l’approccio di sentiment analysis; i risultati sono stati poi confrontati per selezionare il più performante.

Il problema nasce dal bisogno di fornire all’utente informazioni quanto più veritiere possibili tali che possano semplificare la scelta di un hotel sulla piattaforma di booking TripAdvisor. I dati di riferimento sono le opinioni scritte da persone che hanno usufruito del servizio. Il valore dei commenti spesso però non rappresenta l’esperienza del cliente in quanto la recensione risulta priva di significato o porta poca informazione. Questo incide sulla reputazione complessiva del servizio, e negli ultimi anni ha acquisito sempre più importanza dato che influisce sull’aumento o diminuzione della clientela.

Il progetto è stato svolto secondo un determinato flusso di lavoro suddiviso in diverse parti:

* Pulizia e creazione dei dati da utilizzare (**data management**);
* Scelta e definizione dei modelli utilizzati (**generazione dei modelli**);
* Analisi dei risultati (**confronto delle performance**);
* Generazione dell’output del problema (**rating complessivo**)

Viene ora descritto l’ambiente di lavoro sul quale è stato sviluppato il progetto:

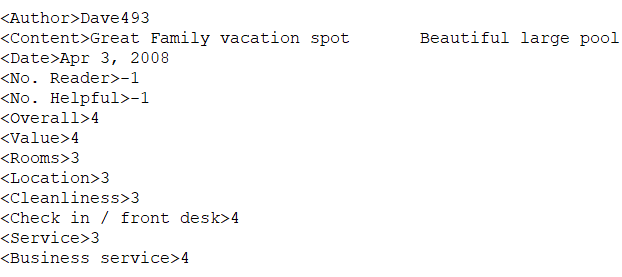
* Python 2.7.16
* Rstudio versione 1.2.1335
* R versione 3.5.3 (2019-03-11)
* [RStudio 1.2.1335](https://download1.rstudio.org/desktop/windows/RStudio-1.2.1335.exe)
* Weka versione 3.8

Verranno confrontate le performance di tre differenti modelli, i quali apprendono da due training set differenti. Il primo è basato su un’assunzione a priori fatta sul campo target, ovvero il valore della recensione, in cui si considerano tutte le recensioni con il campo overall < 4 come “bad”, mentre “good” le restanti. Il secondo è il risultato ottenuto dall’analisi del testo scritto da un cliente tramite sentiment analysis.

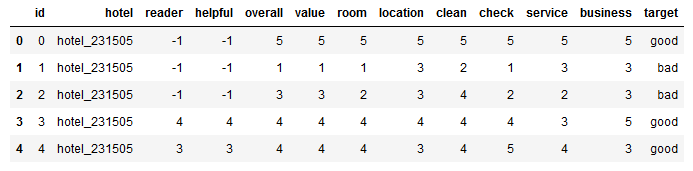
Per eseguire il progetto è necessario inserire i file *.dat* all’interno delle cartelle ./*Training* e ./Testing. Inoltre, è necessario modificare i path presenti nei file dentro la cartella *./code* e nel file *bayesian.R* con l’indirizzo dove è stato salvato il progetto.

# Data Management

## Descrizione e considerazioni sui dati

I dati di input sono dei file *.dat*, ed ognuno di essi corrisponde all’insieme di recensioni di un determinato hotel. Ognuna di esse è caratterizzata da una descrizione scritta in linguaggio naturale, da alcuni metadati e dal rating complessivo calcolato da TripAdvisor. Viene ora presentato un esempio.

Gli hotel sono suddivisi in due insiemi (training e testing) per essere utilizzati nella successiva parte apprendimento.

La prima operazione di manipolazione dei dati è stata quella di convertire il formato di recensione sopra descritto in un vettore in cui sono presenti i valori dei metadati, del rating, e dell’hotel di appartenenza.

Come risultato si è ottenuto un file csv contenente tutte le recensioni di ogni hotel. Questa operazione è stata eseguita sull’insieme di trainig e su quello di testing. I file che eseguono queste operazioni sono [*gen\_csv\_sentimental.py*](http://localhost:8888/edit/Documents/ModelliProbabilistici/Progetto/prediction_tripadvisor/code/gen_csv_sentimental.py) e *[gen\_csv\_assumption.py](http://localhost:8888/edit/Documents/ModelliProbabilistici/Progetto/prediction_tripadvisor/code/gen_csv_sentimental.py" \t "_blank)* e si sono ottenuti i seguenti file risultanti:

* *training\_initial\_assumption.csv*;
* *testing\_initial\_assumption.csv;*
* *training\_initial\_sentimental.csv;*
* *testing\_initial\_sentimental.csv*.

Lo step successivo è stato quello di eliminare dai csv ottenuti tutte le recensioni che forniscono poca informazione. A tale proposito sono stati rimossi tutti i vettori in cui era presente un valore inferiore a 1. Questo perché ai fini delle analisi sono stati considerati solo i valori compresi tra 1 e 5, mentre nelle recensioni si possono trovare anche valutazioni pari a 0 oppure a -1.

Il filtraggio viene eseguito all’interno dei notebook che definiscono le diverse tipologie di rete, i quali verranno presentati in dettaglio nella sezione dedicata alla generazione dei modelli. Per questo procedimento e per tutta la parte di gestione dei csv è stata utilizzata la libreria per Python *pandas*, che permette di definire delle query per selezionare solo le recensioni utili.

I csv risultanti saranno gli input per la fase di apprendimento dei rispettivi modelli a cui si riferiscono.

## Sentiment analysis

Il commento scritto nel linguaggio naturale all’interno del tag *<Content>* non è stato incluso nella fase di vettorizzazione delle recensioni ma è stato elaborato secondo l’approccio di sentiment analysis.

Per task è stato utilizzato il dizionario AFINN-111.txt che contiene parole singole, e ad ognuna di esse è associato un punteggio tra -5 (molto negativo) e 5 (molto positivo). Ad ogni recensione corrisponde un valore calcolato come il rapporto tra la somma dei valori associati ad ogni sentimento la radice quadra del numero di sentimenti.

L’obiettivo è di determinare il possibile valore “good” o “bad” di una recensione basandosi sull’analisi del commento presente nel tag *<Content>*. In particolare, in questo progetto si sono considerati come positivi (etichettati come “good”) i commenti aventi un voto maggiore di 0 e come negativi quelli con voto minore di 0 (etichettati come “bad”). Le recensioni con una valutazione pari a 0 sono state classificate come neutre. Tale assunzione è derivata dalla volontà di determinare i commenti complessivamente più positivi.

Questo approccio però non tiene conto di eventuali quantificatori presenti prima delle parole del dizionario ed inoltre non rileva sentimenti come il sarcasmo e l’ironia.

Durante la fase conclusiva di creazione del training set con questo approccio, si è notato che i risultati ottenuti nel campo target erano fortemente sbilanciati. Per questo motivo si è deciso di utilizzare Weka per sottoporre l’insieme di training generato ad una fase di undersampling.

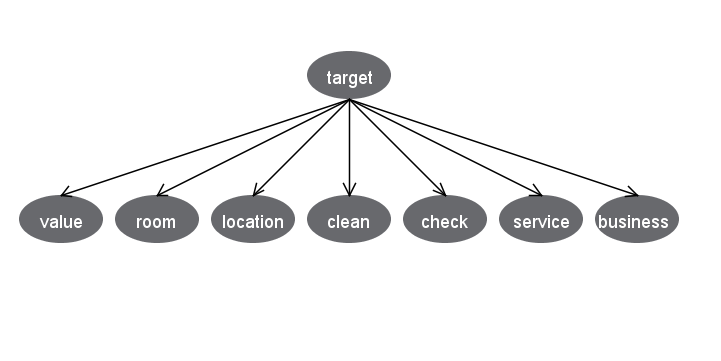
L’insieme di trainig con sentimental analysis ottenuto con undersampling risulta essere circa il 13% di quello ottenuto dopo le operazioni di filtraggio.

Il codice riguardante la sentiment analysis è consultabile nel file [*gen\_csv\_assumption.py*](http://localhost:8888/edit/Documents/ModelliProbabilistici/Progetto/prediction_tripadvisor/code/gen_csv_sentimental.py).

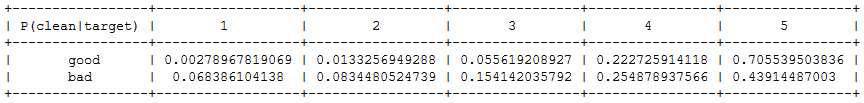
# Generazione Dei Modelli

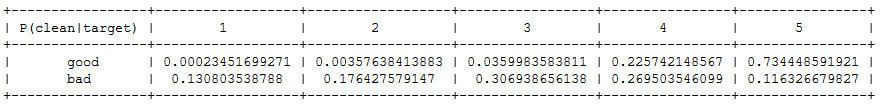
Ai fini del lavoro è stato scelto di definire tre diversi modelli da poter applicare all’analisi dei dati. Per ognuno di essi è stato poi eseguito il task di inferenza per poter apprendere la variabile target. Inoltre, ogni modello è stato utilizzato con il training set ottenuto dall’operazione di sentiment analysis.

## Naive Bayes

Il primo modello utilizzato è quello basato su naive Bayes. Data la variabile target le altre risultano essere indipendenti le une dalle altre. La rete complessiva è la seguente:

Il codice che la genera è reperibile nel file [*naive\_bayes\_assumption.ipynb*](http://localhost:8888/notebooks/Documents/ModelliProbabilistici/Progetto/prediction_tripadvisor/naive_bayes_assumption.ipynb)e [*naivebayes\_sentimental.ipynb*](http://localhost:8888/notebooks/Documents/ModelliProbabilistici/Progetto/prediction_tripadvisor/naivebayes_sentimental.ipynb). Per definirla è stato necessario introdurre le conditional probabilities tables (CPT) per ogni nodo e sono state calcolate verificando tramite delle query la frequenza dei valori assumibili da ogni variabile rispetto al target. stata utilizzata la libreria *libpgm* per Python per definire la struttura complessiva della rete nel file *skeleton.txt*.

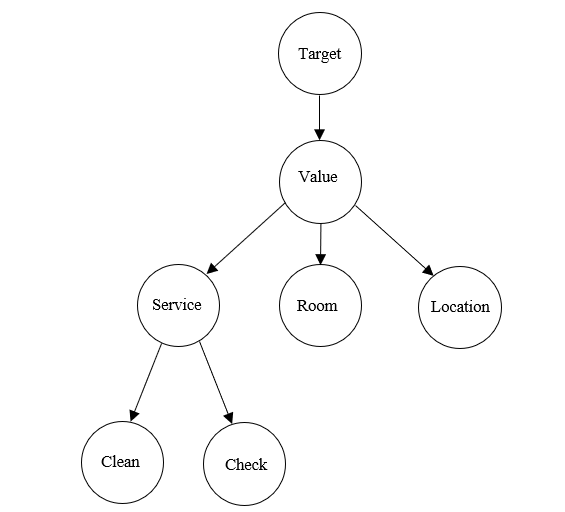
Vengono ora riportati due esempi di cpt, la prima ottenuta dall’utilizzo del training con l’assunzione sul rating a priori, e l’altra ottenuta utilizzando il training risultante dall’operazione di sentiment analysis.



Per la CPT del nodo target è stata fatta un’assunzione a priori, ovvero di considerare tutte le recensioni con il campo overall < 4 come “bad”, mentre “good” le restanti.

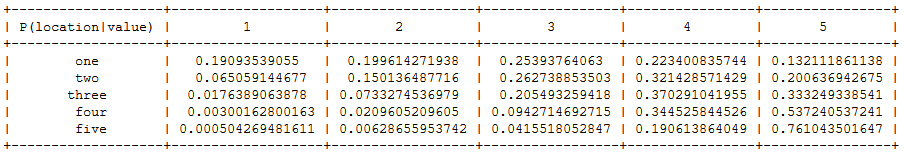
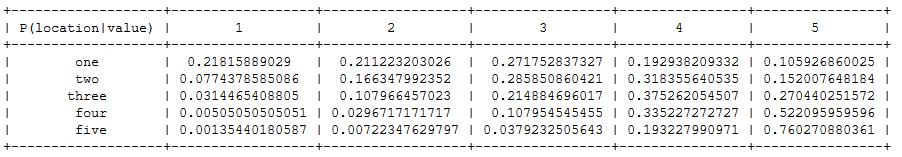
Come verifica della correttezza delle tabelle è stato utilizzato il software Weka, con il quale è stata ottenuta la stessa rete e le stesse CPT della corrispondente rete in Python.

## Rete bayesiana modellata secondo conoscenze a priori

Il secondo modello è stato definito partendo dall’analisi della pagina di TripAdvisor che l’utente deve compilare per lasciare una recensione. Analizzando i campi di valutazione si è assunto che il valore overall dell’hotel dipendesse in primo luogo dal valore del rapporto qualità-prezzo.

I valori di service, room e location sono i primi indicatori ad essere valutati dopo il campo value, che è stato impostato come nodo padre. Infine, i valori di clean e check sono stati considerati come degli indici di qualità del servizio e per questo motivo dipendono da esso.

Rispetto alla rete bayesiana ingenua non è presente business come nodo. Durante la scrittura di una recensione infatti non è sempre presente una domanda per questo indice, e perciò è stato deciso di non includerla nel modello.

Vengono ora presentati due esempi di cpt per i due tipi di training set, il primo con assunzione a priori e il secondo con sentimental analysis.

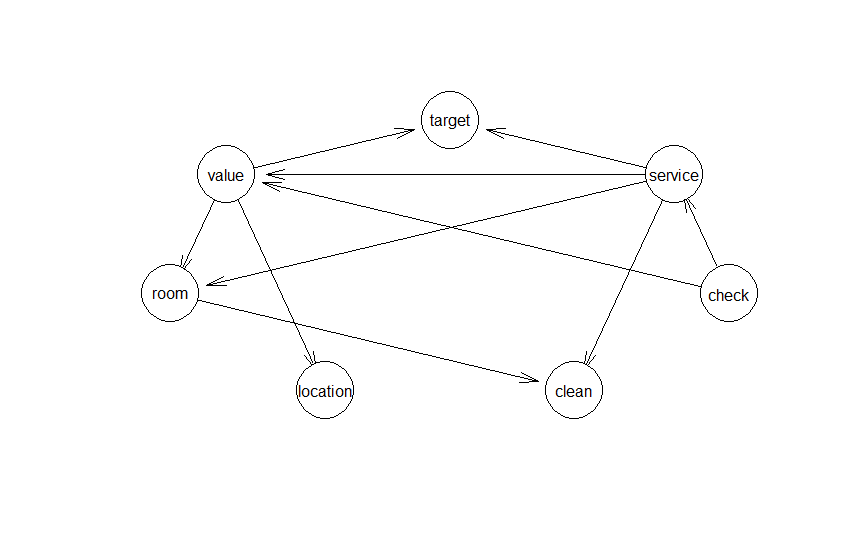
Il codice di riferimento è consultabile nei notebook *bayesian\_net\_assumption.ipynb* e *bayesian\_net\_sentimental.ipynb*.

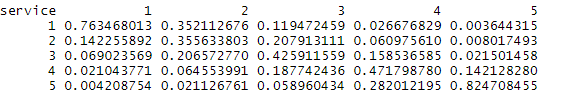
## Rete bayesiana generata con R

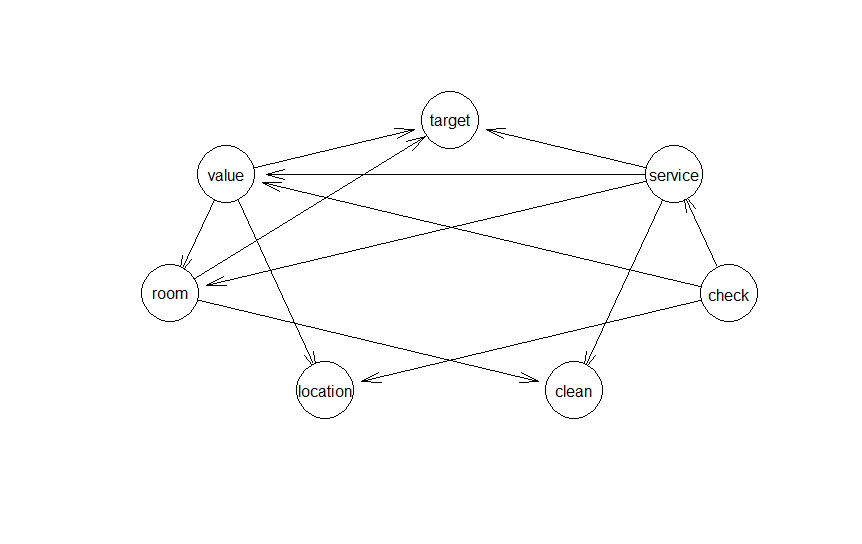
Il terzo modello è stato implementato utilizzando la libreria *bnlearn* disponibile per R. Per la generazione della rete con questa libreria è stato utilizzato l’algoritmo HILL CLIMBING, il quale implementa una tecnica di ottimizzazione matematica.

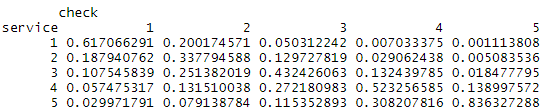
Inizialmente viene generata una soluzione random, e in modo iterativo, genera delle possibili modifiche incrementali alla rete. Le modifiche generate vengono applicate se e solo se si verifica un miglioramento delle perfomance, altrimenti vengono scartate. Negli spazi vettoriali discreti, ogni possibile valore può essere visualizzato come un vertice in un grafico. Hill climbing seguirà il grafo dal vertice al vertice, aumentando sempre localmente (o diminuendo) il valore di , fino a raggiungere un massimo locale (o minimo locale).

Inoltre, è stato scelto dato che spesso può produrre un risultato migliore rispetto ad altri algoritmi nel caso in cui ci siano vincoli di tempo computazionale

 Di seguito vengono presentate le reti definite con il training con assunzione e con sentiment analysis, riportando per ciascuna un esempio di cpt. Il codice è reperibile nel file *bayesian.R*.



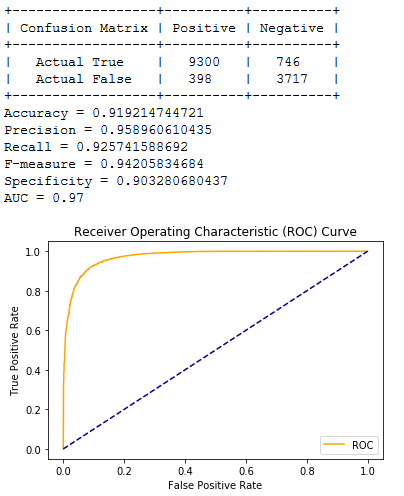


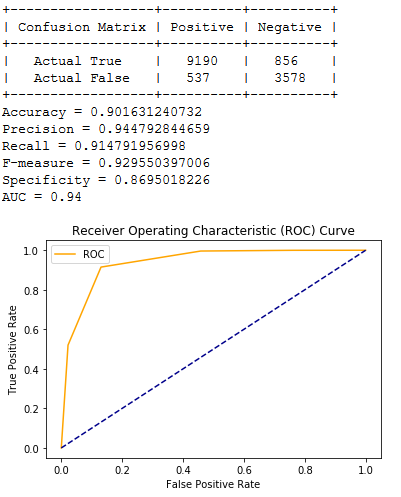


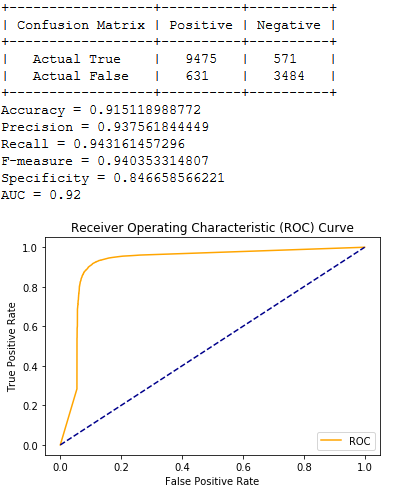
# Confronto Delle Performance

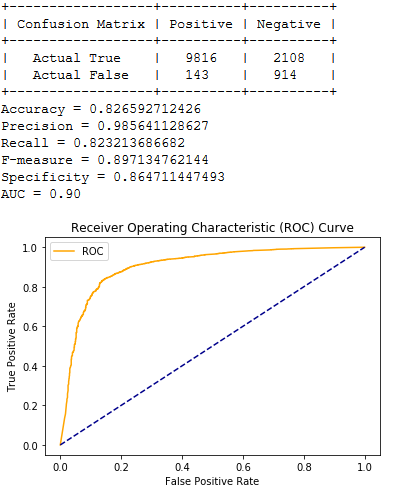
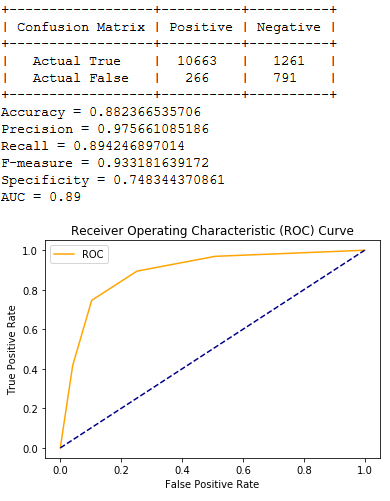
Per ogni modello si sono ottenuti dei risultati in termini di valore del target, ed in totale sono stati eseguiti sei diversi confronti e calcolate le misure di performance. Il codice inerente a questa sezione è consultabile nel file *confusion\_matrix.ipynb*.

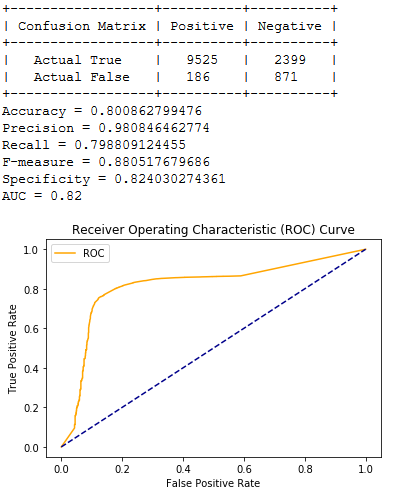
1. **Confronto tra modello naive Bayes con assunzione e l’insieme di testing con assunzione**

****

1. ** Confronto tra la rete definita con conoscenze a priori con assunzione e l’insieme di testing con assunzione**
2. **Confronto tra la rete generata con R con assunzione e l’insieme di testing con assunzione**

****

1. **Confronto tra modello naive Bayes con sentimental analysis e l’insieme di testing con sentimental analysis**
2. **Confronto tra la rete definita con conoscenze a priori con sentimental analysis e l’insieme di testing con sentimental analysis**
3. **Confronto tra la rete generata con R con sentimental analysis e l’insieme di testing con sentimental analysis**

****

Il risultato migliore è stato ottenuto dal modello Naive Bayes, ma non è stato considerato come risultato finale perché l’obiettivo del progetto è di sviluppare una rete bayesiana che permetta di fare inferenza.

Un classificatore di Naive Bayes è un modello semplice che descrive una particolare classe di rete bayesiana, in cui tutte le variabili sono indipendenti dalla classe. Per questo motivo ci sono alcuni problemi che Naive Bayes non può risolvere. Tuttavia, la sua semplicità ne facilita l'applicazione e richiede meno dati per ottenere un buon risultato in molti casi.

Una rete bayesiana invece modella le relazioni tra le variabili in un modo molto generale ed è appropriato usarla quando si conoscono quali sono le relazioni o quando si hanno abbastanza dati per ricavarle.

Per questi motivi il modello più adatto al problema è quello della rete bayesiana definita con conoscenze a priori in cui è stata fatta l’assunzione sul target.

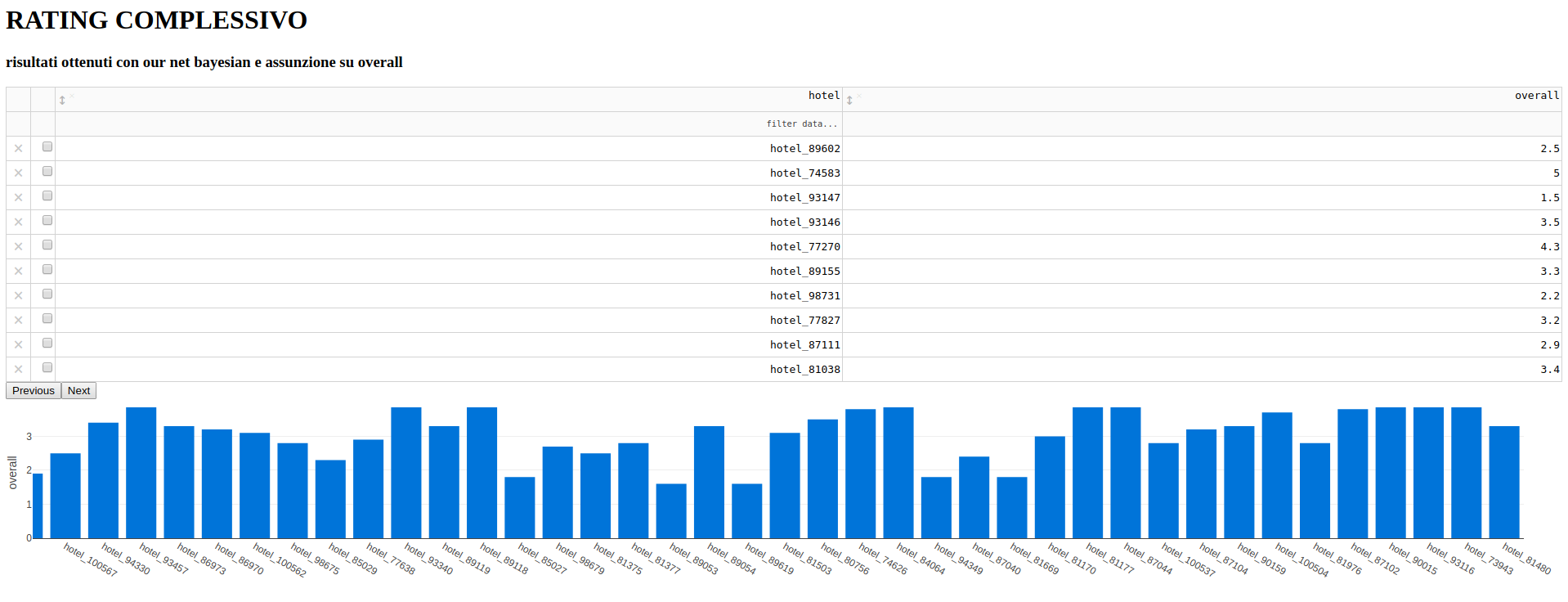
# Rating Complessivo

Utilizzando i risultati ottenuti con il modello di rete bayesiana, è stato generato un CSV contenente due colonne. La prima corrisponde all’ID dell’hotel, mentre la seconda colonna contiene il ranking medio associato.

Il processo utilizzato per la realizzazione del csv è stato implementato nel notebook *calculate\_rating.ipynb*.

Per ogni hotel si considerano il numero di recensioni “bad” e “good”, si sommano il numero di recensioni targate good e si divide per il totale delle recensioni per tale hotel, moltiplicando infine per 5.

Alla fine di questa procedura si ottiene il csv descritto inizialmente. È presentato ora il risultato ottenuto nella demo.



# Conclusioni

Tramite l’utilizzo di diversi modelli e approcci per la risoluzione del problema posto in partenza è stato calcolato il rating per ciascun hotel, basando la valutazione su possibili relazioni tra le variabili. In questo modo i risultati generati possono considerarsi più conformi rispetto all’esperienza vissuta da un cliente presso un hotel.

Durante la fase di testing è stato notato come nonostante la rete definita con naive Bayes fosse migliore a livello di performance, non sia adeguata per le motivazioni esposte in precedenza.

La conoscenza di dominio del problema è risultata essere una buona base per la costruzione di un modello affidabile, dato che le relazioni tra variabili spesso sono deducibili senza ricorrere ad uno strumento di apprendimento.

La rete definita tramite R invece è più complessa rispetto alle precedenti; i risultati ottenuti possono essere spiegati con la maggiore connessione tra le variabili, che rende il calcolo per la classificazione di nuovi hotel più complesso.

L’approccio di sentiment analysis ha comportato un calo delle performance dei modelli in cui è stato applicato. L’analisi però si basa su un commento scritto direttamente dall’utente che ha usufruito di un servizio, il quale certamente fornisce più informazione rispetto che a mettere una valutazione su una scala da 1 a 5. Si può quindi assumere che sotto questo punto di vista la classificazione sia stata più veritiera rispetto alla realtà. Bisogna comunque specificare che il valore di AUC ottenuto è comunque sufficientemente alto. Infine, sono disponibili ulteriori metodi di calcolo del sentimento di una recensione, che per lo scopo del lavoro potrebbero essere più performanti.