DOCUMENTAZIONE PROGETTO

Corso di Ingegneria della Conoscenza – a.a.2020-2021

Costruzione di una Base di Conoscenza ed utilizzo di tecniche di apprendimento supervisionato (e non) per l'analisi del gradimento delle *Airbnb rentals*

Studenti:

Barbaro Nicola

mat. 665975 – email: n.barbaro@studenti.uniba.it

Gasbarro Roberto

mat. 652507 – email: r.gasbarro1@studenti.uniba.it

1. La *mission*

L'obiettivo di questo progetto è stato quello di analizzare un dataset di camere (*rentals*) proposte da Airbnb, un marketplace per il lodging, per poter offrire all'utente la possibilità di effettuare la ricerca di camere che più rispecchiano le sue esigenze: l'utente infatti è abilitato ad effettuare query personalizzate. Il secondo obiettivo è stato quello di trovare gruppi di camere simili tra loro attraverso tecniche di apprendimento non supervisionato. Infine, è stata addestrata una Belief Network per capire le relazioni che sussistono tra le varie features della camera e l'indice di gradimento di quest'ultima, per permettere all'utente di predire la probabilità di gradimento di una camera in relazione alle preferenze da lui indicate.

2. Linguaggi e strumenti utilizzati

Per la realizzazione del progetto è stato utilizzato principalmente il linguaggio Python, in quanto dispone di molte librerie che consentono di lavorare agevolmente con i dati, di utilizzare tecniche di apprendimento automatico e di costruire e utilizzare reti bayesiane. In particolare, sono state usate le seguenti librerie:

- Scikit-learn e Scikit-learn-extra, per l'utilizzo di modelli di apprendimento automatico e per la riduzione delle features;
- Pandas, per l'utilizzo dei Dataframe, che permettono agilmente di gestire grandi moli di dati;
- Pyswip, per l'interazione con il programma SWI-Prolog (richiede la presenza del software SWI-Prolog sulla propria macchina: https://www.swi-prolog.org/Download.html);
- Pgmpy, per la costruzione e manipolazione di Belief Network e per l'inferenza probabilistica.

È stato utilizzato, inoltre, il linguaggio Prolog, che adotta il paradigma della programmazione logica, per poter costruire una base di conoscenza a partire dal dataset e per poter sottoporre delle query all'agente.

Infine, l'utilizzo del software Weka ha permesso di addestrare una rete Bayesiana a partire dal dataset: di questa vengono apprese non solo la struttura ma anche le probabilità.

3. Fase di preprocessing

Il dataset è stato estratto dal sito *Insideairbnb*. Il dataset è formato da 74 features (contenenti informazioni dettagliate su ogni camera) e da 29581 osservazioni, aggiornate al 4 Agosto 2021.

La fase di preprocessing si articola in tre parti:

1. **Data Cleaning**
   questa fase si concentra inizialmente sull'eliminazione di colonne non importanti ai fini dello studio; la lista contiene, ma non si limita, posizione geografica delle camere, numero massimo di pernottamenti, informazioni superflue sul profilo dell'host. Segue una selezione dei tipi principali di proprietà e dei quartieri maggiormente presenti all'interno del dataset.
   Alcuni valori vuoti sono stati sostituiti con la mediana degli altri valori nel caso di variabili di tipo numerico ovvero con la moda nel caso di variabili di tipo categorico.
   In questa fase viene generato il file pre-processato per la costruzione della Base di Conoscenza in Prolog, che verrà completato successivamente con l'aggiunta della colonna *n_cluster*. Dopo aver estratto i dummies delle variabili categoriche, si passa alla fase successiva

2. **Creazione della matrice Termine-Documento per le amenities**
   L'obiettivo è creare due TD-Matrix: la prima contenente tutte le possibili amenities estraibili dal dataset (1028) e destinata al calcolo dei clusters; la seconda contenente un numero ridotto di amenities (solo coloro che appaiono un numero di volte almeno pari alla radice quadrata del numero di osservazioni) e destinata alla costruzione della rete Bayesiana.
   Il dataset presenta le amenities all'interno di un'unica feature, nella forma di una stringa di testo. In primo luogo, l'agente si assicura dell'assenza di amenities ripetitive o semanticamente simili. Segue poi la creazione della matrice sparsa riportante l'indice della camera sulle righe e le amenities sulle colonne.
   Viene eliminata la colonna *amenities* e sostituita con la matrice termine-documento.

3. **Raffinamento dei dati**
   Vengono discretizzate in classi le features *price, minimum_nights* e *number_of_reviews* nella copia del dataset destinata alla creazione della rete Bayesiana. Vengono poi studiati, nel dataset principale, gli outliers, eliminando tutte le features presentanti un numero di outliers maggiore del 20%.
   Segue la riduzione del numero di features del dataset del calcolo dei cluster a 26, utilizzando tecniche di Principal Component Analysis. Il numero di features da mantenere è stato scelto in base alla riduzione della varianza, come riportato nel grafico.

MinMaxScaler

L'algoritmo MinMaxScaler prende in input ogni valore di una feature, ne sottrae il valore minimo e lo divide per il range, ovvero la distanza tra il valore massimo e minimo originari. MinMaxScaler preserva la curva della distribuzione originaria, senza perdere le informazioni semantiche dei dati originari.

$$MinMaxScaler(x) = \frac{x - x_{min}}{x_{max} - x_{min}}$$

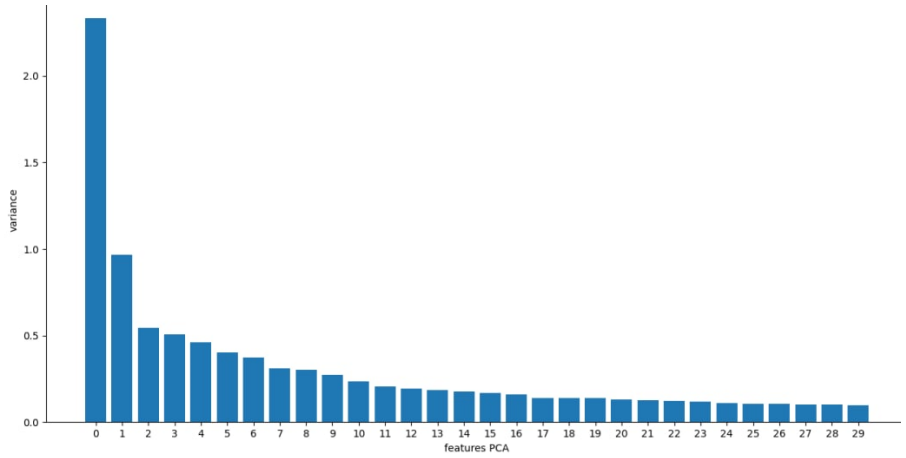


Figura 2-Scelta del numero di Features per PCA

In seguito è stata effettuata la normalizzazione del dataset destinato al calcolo dei cluster attraverso l'uso del MinMaxScaler.

4. Clustering

Si è voluto sfruttare una tecnica di apprendimento non supervisionato per individuare indicativamente le classi di camere più simili tra loro, per agevolare lo studio dei pattern di features che influenzassero di più il rating della camera.

La tecnica scelta è stata quella del Clustering.

Per facilitare la scelta del k ideale con cui effettuare il clustering, si utilizza il metodo dell'elbow: si esegue un plot dell'errore ottenuto in funzione di k e il k ideale è quello per cui successivamente l'errore diminuisce in maniera non rilevante. In corrispondenza del k ideale, quindi, sul grafico si viene a creare un vero e proprio "gomito".

Inizialmente è stata adottata la tecnica del K-Means: essa mira a partizionare le osservazioni in un numero predefinito di clusters, creando dei centroidi, ovvero dei prototipi del cluster stesso, e assegnando il valore di cluster ad ogni esempio in base al centroide più vicino in termini del quadrato della distanza Euclidea.

Si riporta il grafico ottenuto con K-Means, facendo variare il numero di cluster k da 2 a 10.

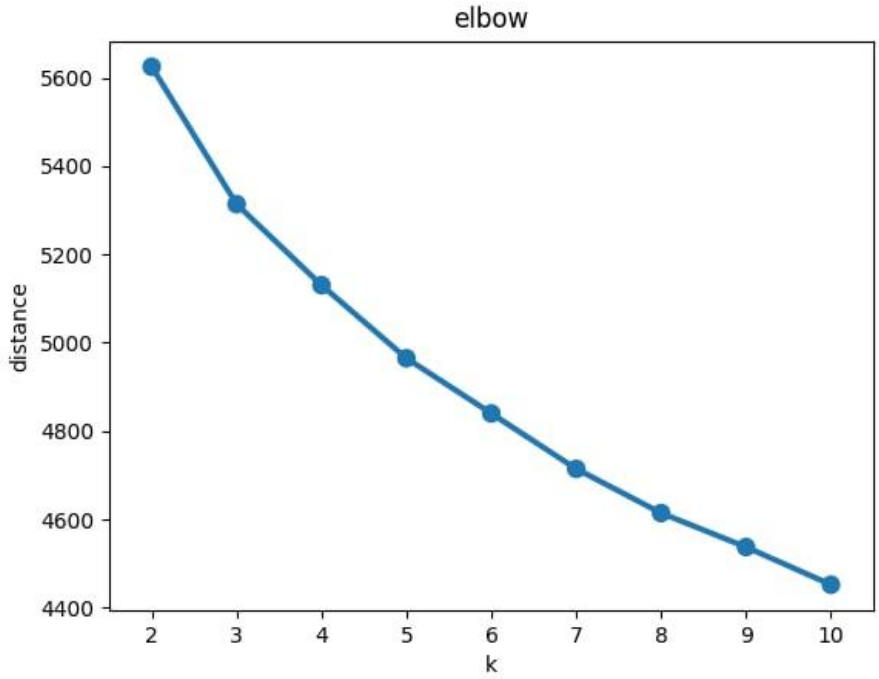
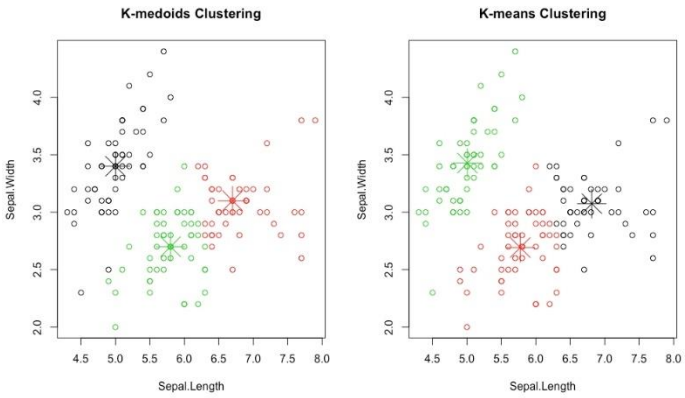


Figura 1 - K-Means

Come si evince dal grafico, non si apprezza una sufficiente tendenza del dataset a clusterizzare. Inoltre, il K-Means presenta una difficoltà a livello semantico: i centroidi trovati rappresentano la media dei valori delle features degli esempi appartenenti a quel determinato cluster, per cui il centroide potrebbe non corrispondere a nessuno degli esempi del dataset. Per questo motivo, si è scelto di virare verso una tecnica alternativa di clustering, chiamata K-Medoids.

Un *Medoid* di un cluster è definito come l'esempio all'interno del cluster stesso la cui dissimilarità media rispetto a tutti gli altri esempi del cluster è minimale: esso è il punto più centrale del cluster. La differenza sostanziale con il K-Means, oltre a presentare un algoritmo di calcolo leggermente modificato e una complessità leggermente maggiore, è che un *Medoid* è un esempio effettivo del dataset.



Si riporta un'immagine esplicativa della differenza tra le due tecniche di clustering.

Si riporta il grafico ottenuto con K-Medoids, facendo variare il numero di cluster k da 2 a 10.

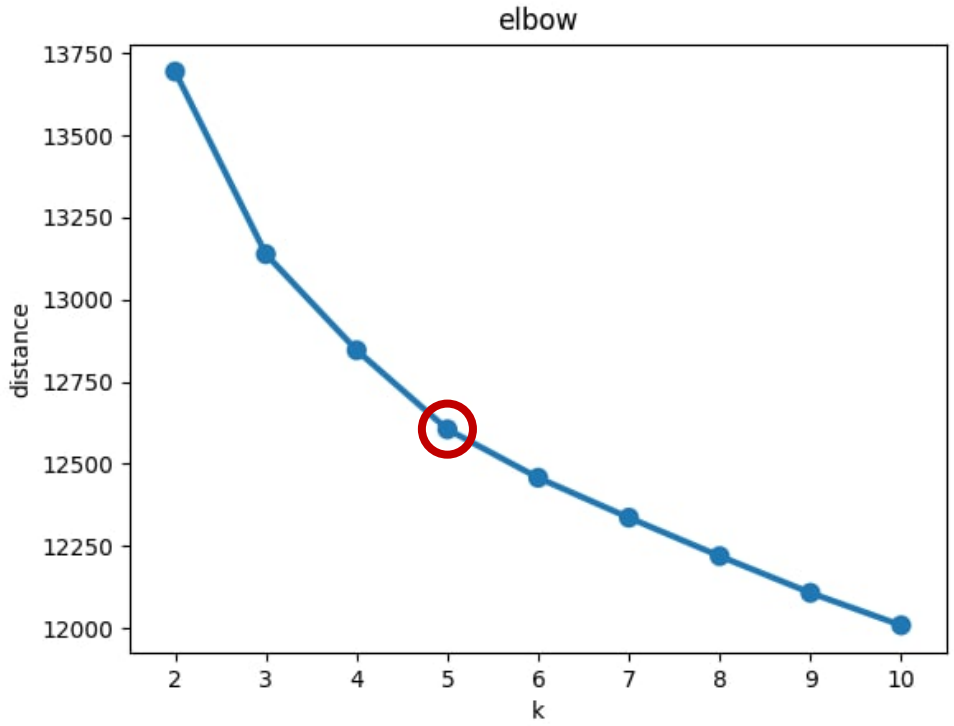


Figura 2 - K-Medoids

Secondo quanto detto precedentemente, il k ideale è risultato essere 5: la curva ha una forte discesa per $k \leq 5$ e tende a non mostrare una significativa ripidità per valori successivi.

Di seguito, vengono riportati anche i *medoid* dei cluster individuati attraverso, corredati delle caratteristiche riscontrate per ogni cluster e dal numero di esempi per ogni cluster:

Id	*Neighbourhood*	*Neighbourhood group*	*Property type*	*Room type*	*Price*	*Number of reviews*	*Characteristics found*	*Number of examples in the cluster*
8900	Harlem	Manhattan	Private room in apartment	Private room	18	2	basso costo, massima centralità, poche amenities, poca metratura	7706
10247	Kips Bay	Manhattan	Private room in apartment	Private room	150	0	meno centrale, più amenities, metratura bassa	7214
12051	Upper East Side	Manhattan	Entire apartment	Entire home/apt	123	36	grande metratura, poche amenities, centrale	5619
11483	Sheepshead Bay	Brooklyn	Entire house	Entire home/apt	79	24	grande metratura, tante amenities, lontana dal centro	5179
15741	East New York	Brooklyn	Private room in apartment	Private room	54	46	lontana dal centro, bassa metratura, poche amenities, economica	3863

5. Bayesian Network

Nel dataset sono presenti informazioni dettagliate circa il gradimento delle camere offerte da Airbnb, in forma di punteggio. Uno degli obiettivi del progetto è stato utilizzare tali informazioni per addestrare una rete Bayesiana che riuscisse a predire la percentuale di gradimento di una stanza a seconda delle preferenze espresse dall'utente in fase di query. Una Belief Network è un grafo orientato aciclico, i cui nodi rappresentano le features, ed un insieme di distribuzioni di probabilità condizionate, dato $P(X_i|parents(X_i))$ per ogni variabile X_i . Questa ha come compito quello di rappresentare le dipendenze probabilistiche tra features. Tuttavia, la struttura del grafo non era conosciuta a priori, per cui è stato necessario apprenderla attraverso i dati. Per far ciò, è stato scelto il software Weka, e i parametri risultati migliori, a seguito di un estensivo testing dell'apprendimento, sono i seguenti:

- Estimator: SimpleEstimator con $\alpha = 0.5$. Il parametro α viene utilizzato per l'apprendimento delle CPT e viene interpretato come il conteggio iniziale per ogni valore
- Search Algorithm: HillClimber (maxNrOfParents = 3, scoreType = BDeu)

Per valutare l'accuratezza del modello appreso, abbiamo usato la k-fold cross validation, ponendo k = 10. Gli score ottenuti sono i seguenti:

LogScore Bayes:	-1002519.8029036967
LogScore BDeu:	-1025427.5521024137
LogScore MDL:	-1024294.3040627859
LogScore ENTROPY:	-1002628.7132299761
LogScore AIC:	-1006837.7132299761

	Accuracy	*Precision*	*Recall*	*F-Measure*
Like	96.376%	0.951	0.919	0.935
Dislike		0.968	0.975	0.975

Classificato come	*Like*	*Dislike*
Like	20828	393
Dislike	679	7681

Appresa la belief network, adesso si è in grado di fare inferenza probabilistica esatta facendo uso dell'algoritmo di Eliminazione di variabili. Per poter utilizzare la rete bayesiana, questa è stata esportata in formato *BIF* ed è stata acquisita dal programma in Python attraverso l'uso della libreria *pgmpy*. La belief network è stata quindi utilizzata per predire la probabilità di gradimento di una camera a partire dalle evidenze recuperate dal dataset, secondo le preferenze dell'utente, che specifica quali features debbano essere utilizzate per il calcolo della probabilità.

6. Costruzione della Base di Conoscenza e inferenza

Si è scelto di rappresentare la conoscenza fornita dal dataset e manipolata (in parte) nella fase di preprocessing per permettere all'utente di accedere alla conoscenza effettuando opportune query. A tal scopo, viene costruita una base di conoscenza, scritta in Prolog, che rappresenti il dataset. A tale base di conoscenza sono state aggiunte alcune regole per poter fare ragionamento automatico e quindi permettere all'utente di poter sottomettere query più complesse. È possibile usare la KB in Prolog (facendo assunzione di conoscenza completa) oppure si può fare uso del metainterprete Ailog (facendo assunzione di conoscenza non completa).

Si riportano alcuni esempi di query che l'utente può sottomettere:

Esempio di query falsa per la KB:

```
room_type(3,"entire_home/apt").
false.
```

Esempio di query vera per la KB:

```
room_type(3,"private_room").
true.
```

Esempio di query non-ground. In questo caso, vengono restituiti gli individui per i quali la query è avvalorata:

```
room_type(3,Type).
              Type
0    'private_room'
```

Esempio di query non-ground più complessa. È possibile scrivere query composte da congiunzioni di atomi attraverso l'uso della "," ed è possibile scrivere la negazione di un atomo usando il simbolo "\+":

```
room_type(Room,Type), price(Room,Price), \+ Type="private_room",
Price < 100.
       Room                Type  Price
0        17  b'entire_home/apt'   99.0
1        22  b'entire_home/apt'   99.0
2        42  b'entire_home/apt'   80.0
3        43  b'entire_home/apt'   85.0
4        49  b'entire_home/apt'   83.0
```

Sono state inserite nella Base di Conoscenza alcune regole (clausole dotate di un corpo, espresso come congiunzione di atomi), che sono disponibili all'utente per poter effettuare una query.

Ad esempio, con la query `same_amenities(Room1,Room2,Amenity)`, vengono restituite tutte le Amenities che le camere Room1 e Room2 hanno in comune.

Attraverso il comando `ailog` è possibile avviare il meta-interprete Ailog, di cui è possibile trovare la documentazione al link https://www.cs.ubc.ca/~poole/aibook/code/ailog/ailog_man.html. All'avvio di Ailog, viene caricata automaticamente la Base di Conoscenza. Ailog consente di approfondire le risposte ad una query facendo uso dei comandi `how, why` e `whynot`. Inoltre, nella Base di Conoscenza sono stati dichiarati alcuni atomi *askable*, cioè atomi il cui valore sarà richiesto all'utente a run-time.

Attraverso il comando `satisfaction`, è possibile attivare il calcolo della probabilità con cui una camera possa piacere all'utente in base ad una sua query predefinita. La probabilità viene calcolata facendo inferenza probabilistica a partire dalla Belief Network addestrata. All'utente viene chiesto di scrivere la lista delle features di suo interesse. Queste saranno utilizzate come evidenze nel calcolo della probabilità di gradimento. Successivamente l'utente potrà scrivere una query in Prolog per poter ricercare determinate camere. Su tali camere viene quindi calcolata la probabilità di gradimento. Per motivi legati alle complessità computazionali, il programma limita il calcolo del gradimento soltanto alle prime 10 camere che soddisfano la query dell'utente.

Ad esempio, l'utente specifica di voler conoscere la probabilità di gradimento per le camere il cui prezzo è inferiore a 100 e specifica come preferenze wifi, tv, pool. La query scritta dall'utente sarà `price(Room, Price), Price<100`. Supponendo che la prima camera trovata nella KB che soddisfa la query abbia il wifi, ma non la Tv né la pool, allora la probabilità calcolata sarà la seguente: `P(Like=true | Wifi=true, Tv=false, Pool=false)`.

7. Conclusione e idee

Il progetto qui esposto è una dimostrazione, seppur sperimentale, della potenza dell'apprendimento supervisionato quando associato alla controparte non supervisionata: le informazioni estratte da entrambe le tecniche di learning sono il frutto della scoperta di pattern interessanti circa l'influenza di caratteristiche salienti delle location: l'analisi delle camere relazionata all'apprezzamento dei customers ha dimostrato quanto un costo per notte cospicuo o l'altolocatezza di una zona in una città come NYC non influenzino la scelta dell'utente così come fattori più vicini alla pulizia, alla cortesia dell'host o alle *amenities* fornite.
Questi risultati e questi indici di gradimento sono convoluti in un unico servizio, vicino all'utente sia per chiarezza che per semplicità di utilizzo rispetto ai classici metodi di analisi dei dati, che punta ad agevolare (e potenzialmente influenzare) l'utilizzatore a scegliere una lodge adatta alle sue esigenze, o quantomeno ridurre il campo di osservazione a pochi esemplari di camere in linea con i gusti del futuro cliente.
Il sistema può sicuramente ampliare il suo raggio di azione: infatti sono stati tralasciati fattori di studio basati sulle tecniche di NLP: si sarebbe potuto osservare la distribuzione delle parole chiave più *catchy* presenti nelle descrizioni delle lobbies, oppure studiare le recensioni testuali dedicate ad ogni camera per analizzare il sentiment degli score numerici usati in questa sede per i nostri scopi predittivi.
Inoltre, data la forte influenza che ha la zona in cui la location è sita sull'indice di gradimento,

sarebbe stato altrettanto interessante potenziare l'apprendimento e la base di conoscenza per valutare non solo il numero di ospiti e la scelta delle amenities, ma anche il mezzo con cui essi si spostano o la motivazione del pernottamento.
Si conclude facendo notare quanto questi approcci siano estremamente validi per migliorare i servizi come quelli forniti da *Airbnb*: per quanto l'esperienza utente nei nostri giorni sia a livelli altissimi, e per quanto il sistema offerto dal servizio sia ampiamente preparato e fornito di ogni possibile informazione strutturata per aiutare e influenzare l'utente nella scelta, metodi di apprendimento come queli riportati permettono di potenziare la conoscenza e agevolare il futuro cliente fornendo dati facilmente convertibili in un concetto "umano" che vanno oltre la somma delle singole caratteristiche di ogni camera.