Università degli studi di Milano–Bicocca

Scuola di Economia e Statistica

Corso di Laurea in

Scienze statistiche ed economiche

Monitoraggio di presenze per una riduzione degli sprechi

Autori:

Cecilia Trevisi[2.2], mat.862637

Giacomo Rabuzzi[2.1,3.1], mat.864452

Riccardo Pajno[1,2.2.6,3.2], mat.864557

Anno Accademico 2023/2024

Indice

Abstract

Questo studio investiga il monitoraggio dell'occupazione negli ambienti interni mediante l'impiego di sensori non invasivi quali CO_2, temperatura e illuminazione, supportato da tecniche di Machine Learning. L'analisi si concentra sull'ottimizzazione dell'efficienza energetica degli spazi interni. Dopo un'approfondita analisi preliminare dei dati ambientali, vengono utilizzati algoritmi di clustering, per identificare modelli, e algoritmi di classificazione, per stimare il numero di occupanti. I risultati evidenziano che l'algoritmo K-medie mostra un'eccellente capacità di clustering, mentre le Reti Neurali mostrano prestazioni superiori nelle operazioni di classificazione. La valutazione è stata condotta considerando sia l'efficacia che l'efficienza dei metodi impiegati.

Introduzione

Le informazioni in tempo reale sulla numerosità di persone che occupano un ambiente interno consentono, non solo di risparmiare energia, ma anche di fornire un miglior comfort agli occupanti.
Per evitare l'utilizzo di sistemi basati su video, negli ultimi anni si è cercato di approcciare il problema del rilevamento dell'occupazione tramite l'uso di sensori ambientali non instrusivi, come ad esempio la rilevazione di CO_2, temperatura e luce.
Pertanto per questo lavoro abbiamo scelto di analizzare, tramite l'utilizzo del Machine Learning, dati riguardanti parametri ambientali, provenienti dai seguenti cinque sensori: CO_2, temperatura, luce, movimento e suono, per la stima del numero di occupanti in una stanza.

L'obiettivo di queste analisi è individuare la tecnica migliore al fine di massimizzare il risparmio energetico all'interno della stanza. Nello specifico, dopo una preliminare analisi esplorativa dei dati, si procederà tramite clusterizzazione per indentificare pattern ricorrenti all'interno dello schema dei dati; successivamente gli algoritmi di classificazione, grazie alla loro natura supervisionata, consentiranno di trarre conclusioni definitive in merito al nostro obiettivo iniziale.

La Figura 1 mostra il laboratorio di prova in cui è stata installata la rete di sensori per registrare i dati.
Il laboratorio è una stanza (6 m x 4,6 m) con quattro scrivanie. La stanza ha una grande finestra. Si noti che durante l'esecuzione degli esperimenti non vi è stata climatizzazione. La rete è composta da sette nodi disposti come in Figura 1 . In questo esperimento sono stati utilizzati cinque diversi tipi di sensori non intrusivi: temperatura, illuminazione, suono, CO_2 e infrarossi passivi (PIR). I nodi sensore S1-S4, di temperatura, luce e suono, sono stati distribuiti sulle quattro scrivanie. Il nodo S5 aveva un sensore di CO_2 che è stato tenuto al centro per ottenere la migliore lettura possibile della stanza. I nodi S6 e S7 contengono solo sensori PIR (di movimento) e sono stati distribuiti sul soffitto con un'angolazione tale da massimizzare il campo visivo del sensore per il rilevamento del movimento. (Singh et al. (2018))

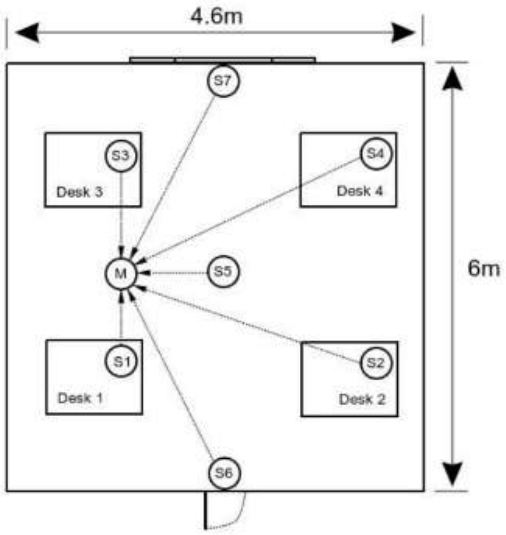

Figura 1: Laboratorio di prova.

Capitolo 1

Esplorazione dei dati

1.1 Descrizione dataset

Il dataset contiene più di 10.000 osservazioni e 19 variabili: per ogni istanza sono riportati il giorno e l'orario in cui avviene la rilevazione e le restanti carattteristiche sono dati di un particolare sensore. Ogni persona che entrava o usciva dalla stanza firmava l'ora esatta insieme al numero della scrivania in un registro.
I dati relativi alla CO_2 sembrano spiegare in maniera affidabile il numero di occupanti nella stanza, sebbene ciò avvenga con un certo ritardo. Pertanto è stata costruita una nuova caratteristica sotto forma di pendenza della CO_2. Questa è stata calcolata adottando una regressione lineare in una finestra di 25 punti per ogni istanza e calcolando la pendenza della linea. Il fattore di 25 è stato ottenuto per tentativi. Sintetizzando:

- Date: data della rilevazione
- Time: tempo della rilevazione espresso in ore/minuti/secondi
- S1-4_Temp: temperatura in gradi Celsius dei quattro sensori
- S1-4_Light: luce in 1 Lux dei quattro sensori
- S1-4_Sound: suono in 0.01V* dei quattro sensori
- S5_CO_2: CO_2 in 5ppm dei quattro sensori
- S5_CO_2_Slope: pendenza della CO_2
- S6-7_PIR: presenza-1 o assenza-0 di movimento
- Room_Occupancy_Count: numero di persone all'interno della stanza

1.2 Analisi esplorativa

Dalle analisi preliminari si è provato a intuire il legame tra alcune delle features e la variabile target. A tal proposito una possibile visualizzazione grafica utile è la seguente:

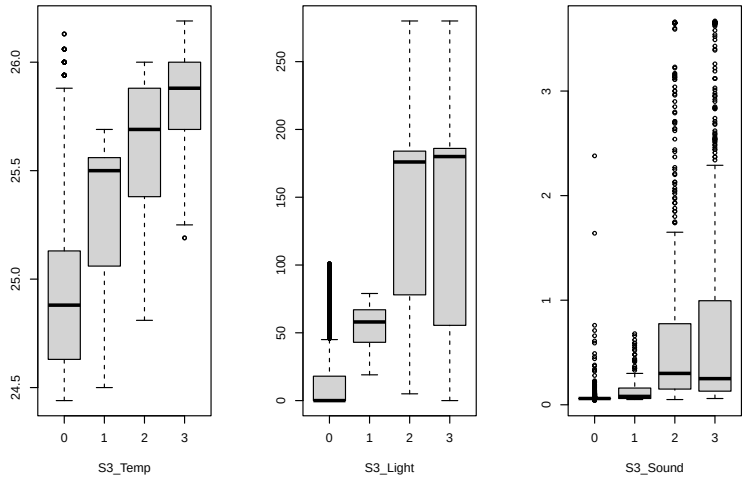

Figura 1.1: Boxplot delle variabili temperatura, luce e suono rilevate nel nodo 1 condizionatamente alle presenze in stanza.

Si può notare infatti come la crescita della temperatura sia legata al numero di soggetti nella stanza. Tipicamente si riscontra un livello di rumore più elevato nel caso vi siano più persone in stanza sebbene questa relazione sia più debole. Significativo è il range di variazione della luce in presenza di 2 o più persone: questo potrebbe essere giustificato dal movimento delle persone all'interno della stanza generando zone di luce e di ombra.
Per quanto riguarda la variabile PIR si è costruita la seguente heatmap:

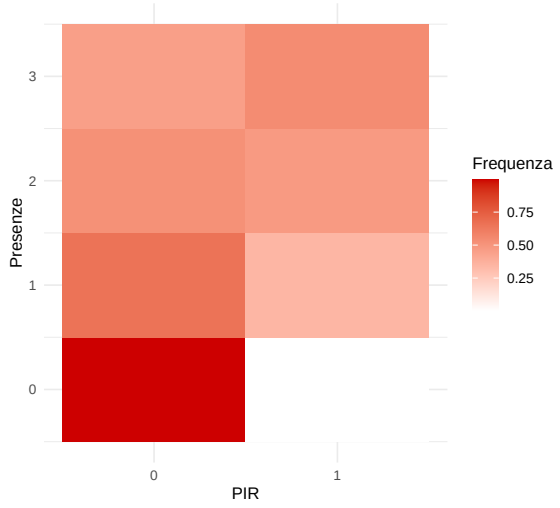

Figura 1.2: Heatmap rilevatori movimento condizionatamente alle presenze in stanza.

che sottolinea come al crescere delle presenze i sensori rilevino una maggiore quantità di movimenti in linea con quanto ci si aspetterebbe. La Figura 1.2 riporta esclusivamente i valori rilevati nel nodo 6, infatti per quanto riguarda il 7, sebbene la relazione continui a sussistere, essa è più leggera probabilmente per il fatto che quest'ultimo è posizionato nei pressi della finestra (lontano dall'ingresso), mentre quello considerato è vicino alla porta.
A titolo di esempio, si è considerato un periodo di circa 10 ore, relativo al 23/12/2017, per mettere in luce la relazione tra le features considerate e il tasso di occupazione al variare del tempo come riporta la seguente figura:

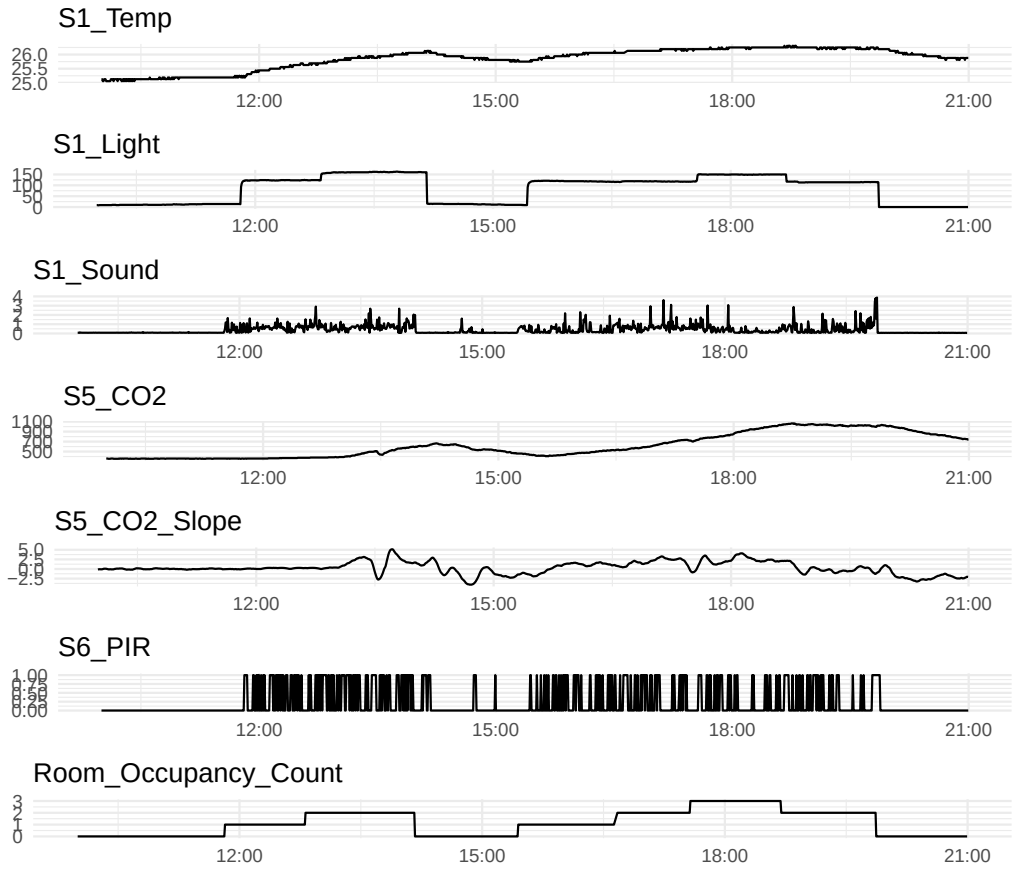

Figura 1.3: Dati da alcuni sensori rappresentativi per un periodo di circa 10 ore il 23/12/2017.

Si può notare come congiuntamente ai picchi di presenza vi siano valori elevati per le variabili temperatura, luce, suono, CO_2 e allo stesso tempo venga rilevata una grande quantità di movimento. Anche i valori della Slope CO_2 sono mediamente più elevati in corripsondenza ai picchi, sintomo di una tasso di crescita positivo del livello di CO_2 in quegli istanti.
Si sono poi calcolate le correlazioni tra le variabili per valutarne l'associazione ai fini di un eventuale selezione o riduzione di dimensionalità

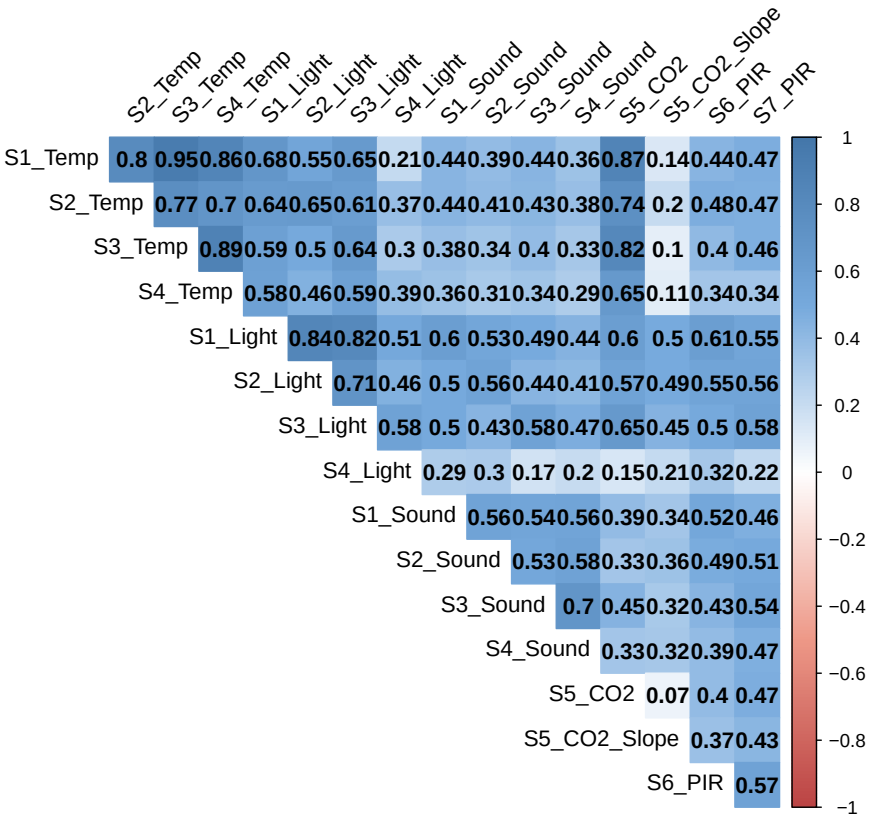

Figura 1.4: Corrplot delle variabili.

Essa evidenzia forte corrrelazione relative alle temperature registrate dai quattro sensori; questo vale anche per la variabile luce anche se in maniera meno pronunciata. A tal fine si potrebbe pensare a qualche forma di riduzione della dimensioanlità, per mantenere le informazioni e ridurre la complessità del problema. Anche il livello di CO_2 è fortemente correlato con tutte le variabili relative alla temperatura.
Un'ulteriore elemento fondamentale dell'analisi eplorativa è la rimozione dei valori anomali. Essendo inseriti in uno scenario supervisionato, sono state valutate le distanze di ogni osservazione dal proprio centroide (in base al valore della variabile target) tramite l'utilizzo della distanza di Mahalanobis. Valori anomali di distanze ci hanno suggerito la presenza di alcuni outlier che dopo una valutazione sul datset sono stati eliminati. Questi infatti erano in corrispondenza ai passaggi bruschi da tante a poche presenze e viceversa.

Capitolo 2

Algoritmi di Machine Learning

2.1 Clustering

2.1.1 Preparazione dei dati

In base a quanto ci suggerisce la matrice di correlazione riportata nella Figura 1.4 abbiamo scelto di ridurre la complessità del problema procedendo con una riduzione della dimensionalità tramite l'utilizzo della PCA (Principal Component Analysis) applicata all'intero dataset.
Abbiamo deciso, in linea con la letteratura presente, di considerare un numero di variabili tale per cui la varianza spiegata fosse superiore al 90%; di conseguenza abbiamo selezionato 8 componenti (scores).

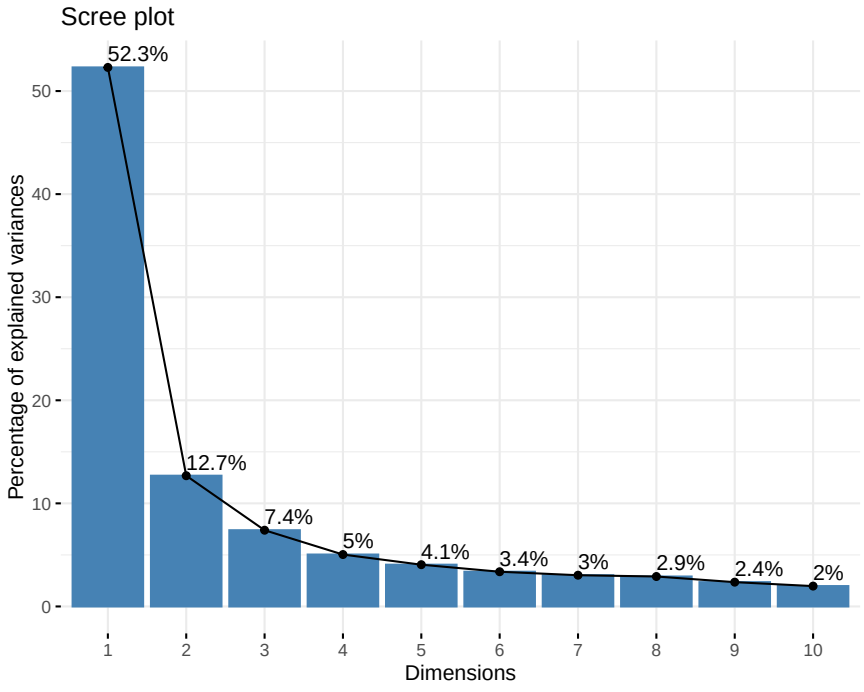

Figura 2.1: Percentuale di varianza spiegata da ogni componente.

2.1.2 Distance based Clustering

K-medie è il primo algoritmo di clustering ad essere stato applicato; si è scelto di considerare $k = 4$ sulla base della classificazione presente nei dati.
Questo algoritmo fa parte della famiglia iterative distance-based approach e in questo caso utilizza come distanza quella euclidea: la forma dei cluster che si creano è quindi ipersferica.
Confrontando i clusters ottenuti con la classificazione ground-truth in nostro possesso si ottiene la seguente suddivisione:

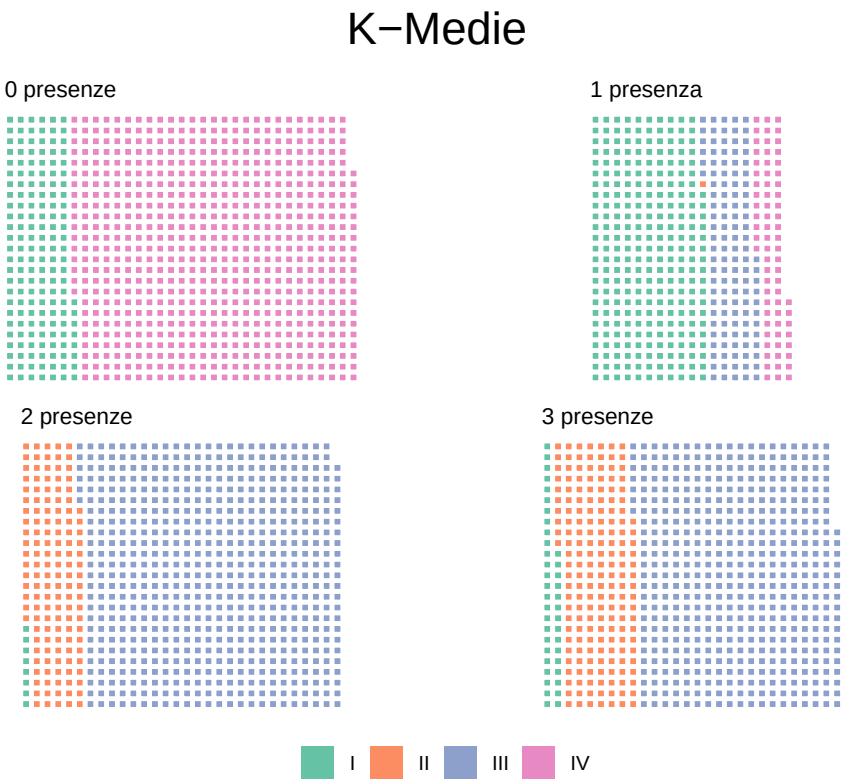

Figura 2.2: Suddivisione in clusters per ogni singola classe.

La figura 2.2 suggerisce come le osservazioni nel caso di poche presenze nella stanza vengo clusterizzate principalmente nei gruppi I e IV. Al contrario i clusters II e III contengono le istanze relative alle presenze più massicce di persone in stanza.
Inoltre risulta evidente come questo algoritmo non riesca a distinguere in maniera netta l'occupazione della stanza da parte di due o tre persone.
La suddivisione ottenuta risulta quindi essere sensata: i gruppi possono essere ben modellati da forme sferiche.
Si è provato a implementare l'algoritmo k-medoidi, che a differenza del precedente, tiene conto della presenza di outliers e fornisce clusters leggermente più irregolari. Le performance di questo algoritmo sono decisamente peggiori e per questo motivo non se ne riportano i risultati.
Infine si è provato a implementare un kernel-based k-means con kernel gaussiano; siccome esso risulta computazionalmente oneroso si è proceduto all'impostazione

dell'iperparametro tramite un'ottimizzazione bayesiana, basata sulla minimizzazione della purity, solo su un sottoiensieme del dataset. In questo modo l'iperparametro ottenuto, superiore a 50, ci indica come l'ampiezza della campana ottimale sia particolamente stretta: viene confermato, anche in questo caso, come i gruppi siano particolarmente compatti.
Tuttavia la performance del k-means tradizionale risulta essere migliore; non è quindi necessario l'introduzione di una funzione kernel per rimappare i dati.

2.1.3 Clustering gerarchico

Proseguendo, sono state applicate varie forme di clustering gerarchico: si sono provate varie distanze e diversi metodi di linkage.
L'unico modello che ha fornito suddivisioni soddisfacenti è risultato essere quello che adotta come distanza quella euclidea e come forma di linkage il legame di Ward (Ward.D2).
Questo conferma nuovamente quanto detto nel caso dell'iterative distance-based approach: sembra ancora essere sensato assumere clusters di forma sferica (Patlolla (2018)); inoltre questo tipo di legame empiricamente è solito creare gruppi sufficientemente compatti (forma globulare).

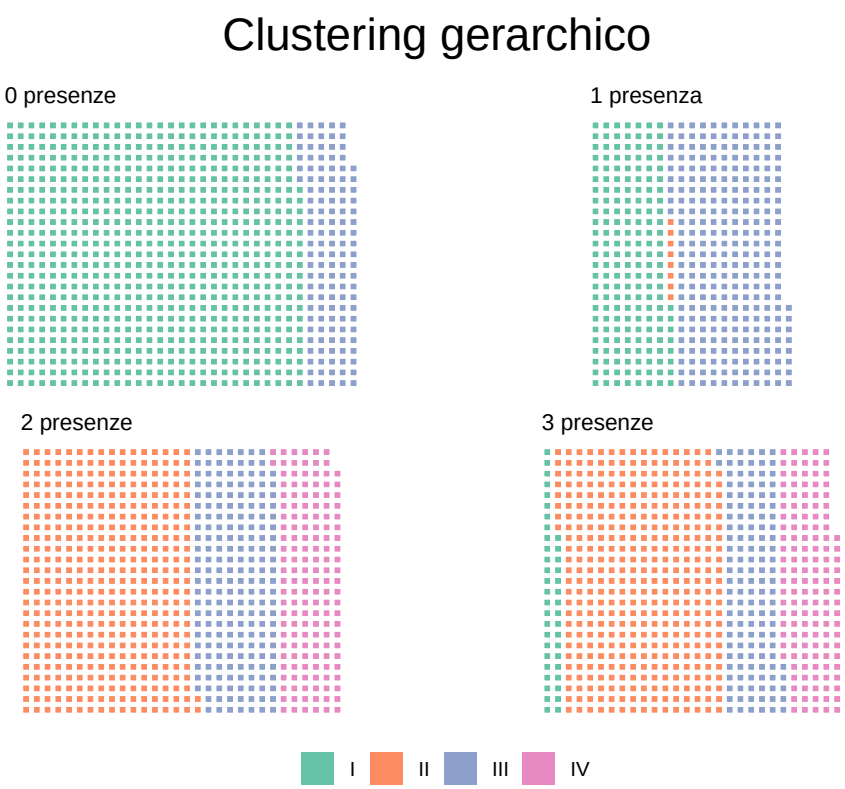

Figura 2.3: Suddivisione in clusters per ogni singola classe.

La figura sopra riportata, mostra, come anche in questo caso, l'algoritmo raggruppi separatamente le istanze relative alla classe 0 e 1, rispetto alla 2 e

3. Risulta ancora più evidente come questi due gruppi abbiano effettivamente caratteristiche diverse.
Si riporta di seguito il dendrogramma:

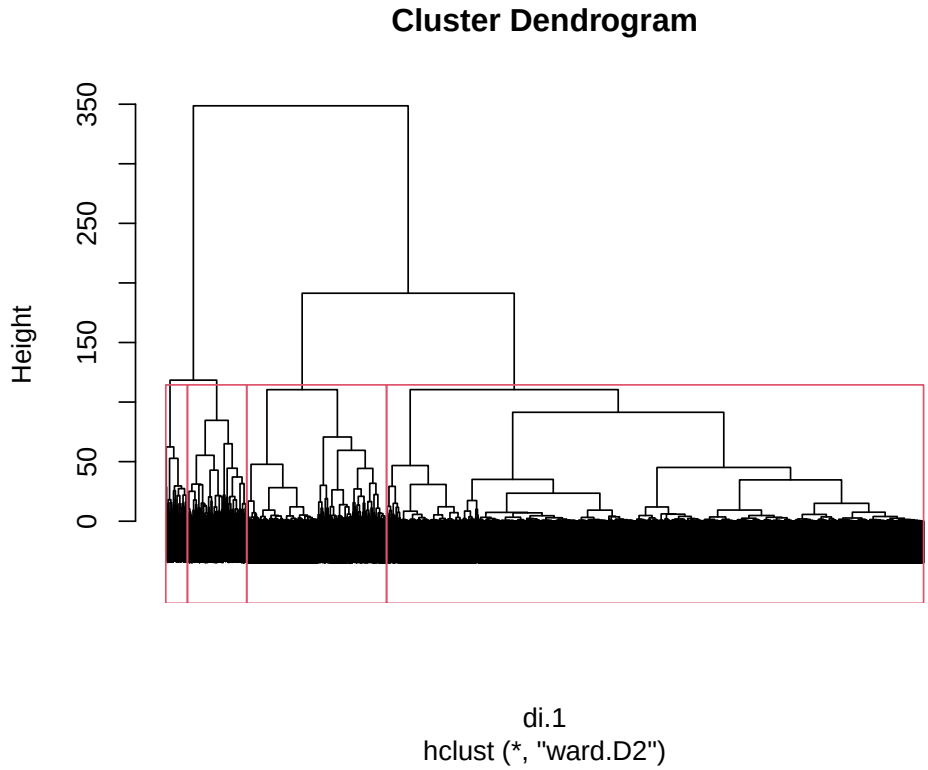

Figura 2.4: Dendogramma.

Esso conferma quanto riportato dalla figura 2.3: i clusters II e IV, che contengono osservazioni relativi alla stanza con 2 o 3 persone, sono effettivamente molto vicini.

2.1.4 Model-Based Clustering

Un ulteriore algoritmo di clustering che è stato adottato è il Model-Based Clustering. Abbiamo scelto una mistura di gaussiane con quattro componenti, data la conoscenza a priori del numero di classi.
Abbiamo considerato la possibilità di inserire delle restrizioni sulla matrice di varianza-covarianza ma i criteri considerati (BIC e ICL) hanno suggerito di utilizzare un modello senza vincoli (VVV).
I risultati ottenuti sono riportati nel grafico seguente:

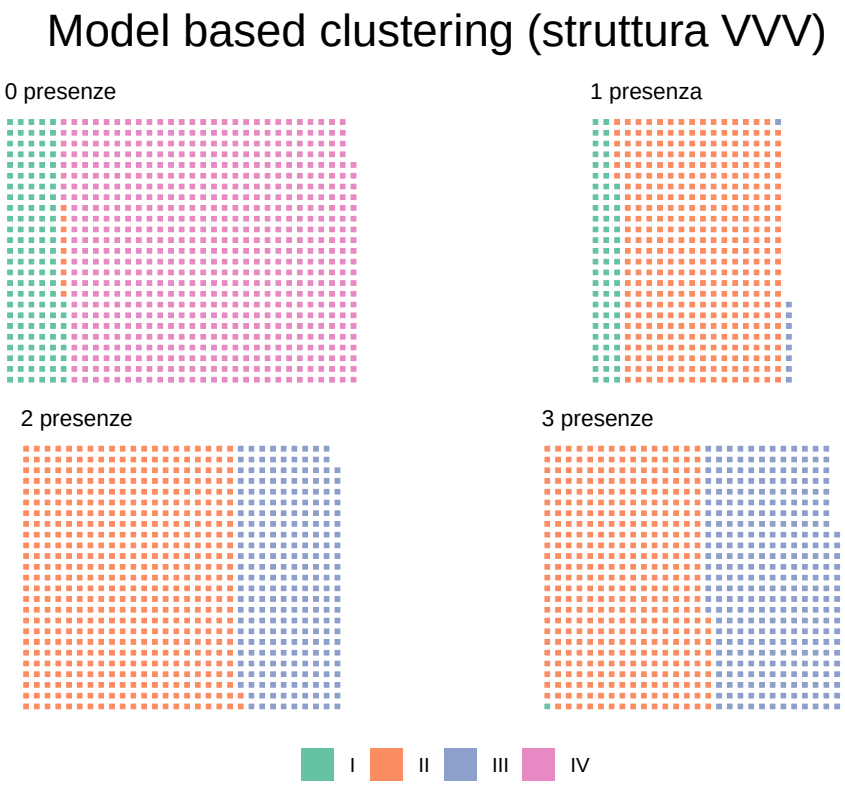

Figura 2.5: Suddivisione in clusters per ogni singola classe.

Anche in questo caso i raggruppamenti risultano sensati: i gruppi II e III sono relativi a maggiori presenze nella stanza mentre i gruppi I e IV racchiudono quasi esclusivamente stanze vuote.
La misura di bontà del clustering (Silhoutte) ha mostrato tuttavia come clustering gerarchico e k-medie risultassero più adeguati; la mistura di normali non sembra quindi il modo migliore per modellare i gruppi.

Si è deciso di non procedere con l'implementazione dell'algoritmo DBSCAN in quanto la forma dei cluster individuati finora risulta sferica e compatta.

2.2 Classificazione

2.2.1 Preparazione dei dati

Per approcciare il problema di classificazione abbiamo scelto di suddividere il dataset completo in training e test con rispettive proporzioni 80% e 20%, andando ad accertarci che non si modificasse la distribuzione della risposta.
Abbiamo scelto di ridurre la dimensionalità del dataset applicando la PCA su training e successivamente, utilizzando i risultati ottenuti, sono stati calcolati gli

scores sul test.
Abbiamo deciso di mantenere le prime otto componenti adottando lo stesso criterio proposto in precedenza.
Gli algoritmi che verranno presentati successivamente faranno ricorso alla suddivisione in k-fold cross validation con $k = 5$ per la determinazione degli eventuali iperparametri.

2.2.2 K-NN

Il primo algoritmo di classificazione sviluppato è il KNN. Si è proceduto tramite metodi di ottimizzazione automatica bayesiana degli iperparametri per quanto riguarda il numero di vicini (k).
Si è deciso di massimizzare la media delle F1 (misura riconducibile a una sintesi di precision e recall) delle classi per gestire il problema delle classi sbilanciate.
Il valore ottimale per k è risultato essere pari a 5. Esso è molto inferiori alla numerosità dei dati, di conseguenza i confini decionali potrebbero essere irregolari. Tuttavia l'errore sul training (0.006) e quello sul test (0.012) sono prossimi in valore assoluto pertanto non sembra esserci overfitting. Riportiamo di seguito i risultati ottenuti per il modello allenato sul training e valutato sul test con $k = 5$:

Metrica	Classe 0	Classe 1	Classe 2	Classe 3	Media
F1	0.997	0.951	0.944	0.940	0.958
BA	0.994	0.980	0.970	0.9725	0.979

Tabella 2.1: Risultati di F1 e Balanced Accuracy (BA)

Si osserva che in generale il metodo ha ottime performance. In particolare il modello riesce a individuare in maniera quasi totale le unità statistiche della classe 0, mentre le capacità dell'algoritmo calano lievemente per le altre classi (benchè siano sempre superiori al 90%).
La confusion matrix conferma il fatto che le unità a rischio missclassificazione sono quelle contigue: ad esempio le unità nella classe 2 vengono a volte classificate nella classe 3 e allo stesso modo vale per 0 e 1. Questo ci suggerisce appunto come stanze con un numero di persone vicino siano effettivamente più difficili da discriminare.

2.2.3 Support Vector Machine

Il secondo algoritmo di classificazione implementato è SVM soft margin. Anche in questo caso si è sviluppata una procedura auto-ML per l'identificazione degli iperparametri ottimali: tipo di kernel (lineare o gaussiano), costo e gamma (nel caso kernel sia gaussiano). Si è utilizzata come metrica di valutazione la misura F1 nel caso multiclasse e i risultati ottenuti sono stati: kernel radiale, costo =

27.89 e un γ pari a 0.014.
Il valore del costo ottenuto non è particolarmente elevato perciò la penalizzazione che si da ai punti che cadono nel sottospazio non corretto non è eccessiva, non c'è rischio overfitting. Infatti il numero di support vectors è 384 e siccome il training contiene più di 8000 osservazioni non risulta in alcun modo problematico. Un'ulteriore conferma è fornita dal training error (0.004) e dal test error (0.010) che risultano entrambi molto piccoli e vicini.
La tabella seguente mostra le misure per il modello allenato sul training e valutato sul test con iperparametri ottimali:

Metrica	Classe 0	Classe 1	Classe 2	Classe 3	Media
F1	0.998	0.975	0.948	0.936	0.964
BA	0.995	0.981	0.974	0.967	0.979

Tabella 2.2: Risultati di F1 e Balanced Accuracy (BA)

Si osserva che in generale le performance dell'algoritmo sono molto simili a quelle del precedente. Vi sono però alcune lievi differenze: SVM performa ancora meglio per la classe 1 mentre soffre maggiormente per la classe 3 rispetto al KNN. In generale vi sono più missclassificazioni tra 0 e 1.
Questo algoritmo conferma i risultati ottenuti precedentemente riguardo alle unità con 0 presenze nella stanza: esse rimangono le più separabili.

2.2.4 Random Forest

Un ulteriore algoritmo di classificazione utilizzato è il Random Forest. E' stata implementata una procedura auto-ML di ottimizzazione bayesiana per la scelta degli iperparametri: il numero di alberi e il numero di covariate selezionate casualmente in ogni albero. Si mantiene come metrica da massimizzare la F1. Il modello ottimale è costituito da 355 alberi e considera 2 features.
I risultati relativi al modello otttimale sul test sono i seguenti:

Metrica	Classe 0	Classe 1	Classe 2	Classe 3	Media
F1	0.997	0.938	0.931	0.937	0.951
BA	0.994	0.962	0.963	0.976	0.974

Tabella 2.3: Risultati di F1 e Balanced Accuracy (BA)

Benchè la classe 0 continui ad essere individuata adeguatamente, l'algoritmo sembra soffrire maggiormente nei casi di presenza di persone nella stanza. In questo caso le performance sono meno interessanti rispetto a quanto già visto: il grosso calo di performance si ha nella classe 1. L'algoritmo non soffre di overfitting infatti training error (0.010) e test error (0.013) sono pressochè identitici.

2.2.5 Rete Neurale

In seguito è stata implementata una rete neurale, in linea con i precedenti algoritmi, utilizzando una procedura auto-ML di ottimizzazione bayesiana per la scelta degli iperparametri. Come spazio degli iperparametri si è considerato un numero di neuroni compreso tra 2 e 16 (pari a due volte il numero di input, regola reperibile nel libro Heaton (2008)) e una costante di regolarizzazione (λ) tra 0 e 0.1 come propongono Kuhn & Johnson (2013). La metrica massimizzata è sempre la F1 sul validation. L'hidden layer della rete neurale ottimale risulta essere costituito da 16 neuroni con un valore di λ pari a 0.1. Risulta comprensibile il fatto che l'ottimizzazione degli iperparametri conduca a un valore elevato di neuroni nell'hidden layer, esso è infatti l'unico considerato. Peraltro il valore di λ ottenuto suggerisce un'influenza piuttosto elevata della regolazizzazione.
I risultati ottenute da tale modello sul test set sono i seguenti:

Metrica	Classe 0	Classe 1	Classe 2	Classe 3	Media
F1	0.999	0.981	0.955	0.953	0.972
BA	0.995	0.981	0.974	0.981	0.983

Tabella 2.4: Risultati di F1 e Balanced Accuracy (BA)

Le performance dell'algoritmo risultano molto buone, il valore dell'F1 medio è infatti il più alto tra i modelli finora implementati; tuttavia, permangono alcune lievi difficoltà nella distinzione tra le presenza di due o tre persone nella stanza, caratteteristica comune agli altri algoritmi, ma in netto miglioramento. Anche in questo caso l'errore sul training (0.003) e quello sul test (0.008) assumono valori simili, pertanto non si può parlare di overfitting.

2.2.6 Modello proportional odds

Al fine di considerare la natura ordinale della variabile risposta, abbiamo scelto di implementare un modello logistico cumulato noto anche come modello proportional odds, che sfrutta le probabilità cumulate per la classificazione.
Da una prima analisi dei risultati, si osserva che tutte le esplicative (ovvero le prime otto componenti principali) risultano significative e, pertanto, si è deciso di mantenerle nel modello finale.
Tale modello è stato utilizzato per effettuare previsioni sul test set ottenendo i seguenti risultati:

Metrica	Classe 0	Classe 1	Classe 2	Classe 3	Media
F1	0.982	0.389	0.705	0.652	0.682
BA	0.925	0.664	0.829	0.805	0.806

Tabella 2.5: Risultati di F1 e Balanced Accuracy (BA)

L'accuratezza delle previsioni, come comprensibile, è di gran lunga inferiore ai modelli più complessi considerati finora. Tuttavia si può notare come il modello sia in grado di distinguere bene classi lontane tra loro dal momento che prende in considerazione un ordine tra le modalità della variabile risposta. Al contrario, fatica a distinguere le classi adiacenti: in particolare la classe 0 dalla classe 1 e la 2 dalla 3.

Capitolo 3

Conclusioni

3.1 Efficacia

Per concludere abbiamo scelto di confrontare l'efficacia dei vari algoritmi.
Abbiamo considerato per il clustering due indici di sintesi in grado di valutare la bontà del modello. Tuttavia le nostre valutazioni non si sono basate esclusivamente su questi numeri, ma anche sull'effettiva composizione dei clusters ottenuta.

Algoritmo	Purity	Silhouette Media
K-medie	0.838	0.460
K-medoidi	0.713	0.328
Kernel K-medie	0.730	-0.260
Gerarchico	0.828	0.469
Model-based	0.839	0.414

Tabella 3.1: Valori di Purity e Silhouette per vari algoritmi.

Risulta evidente come due algoritmi non siano adeguati, K-medoidi e Kernel K-medie. Inoltre dalla composizione dei cluster vista in precedenza, si può notare come i gruppi individuati dal Model-based, benchè registrino metriche elevate, riescano a distinguere esclusivamente l'assenza o la presenza di persone in stanza. Di conseguenza considerando congiuntamente le metriche e i risultati ottenuti nelle rappresentazioni grafiche, si può concludere come gli algoritmi K-medie e gerarchico siano piuttosto solidi. Il valore della silhoutte media osservata intorno a un valore di 0.5 suggerisce infatti cluster ragionevoli, benchè non eccellenti.

Per quanto riguarda la classificazione si è proceduto valutando la sensitivity per la classe 0 e la media per le restanti classi; poichè il nostro obiettivo è la minimizzazione degli sprechi, ciò che ci interessa maggiormente è la massimizzazione della sensitivity per la classe 0. Infatti essa rappresenta la porzione di unità realmente nella classe 0 che sono state correttamente classificate: in questo modo si minimizza il rischio di considerare stanze effettivamente vuote come occupate evitando quindi sprechi inutili (aria condizionata, luce, riscaldamento...). Si può

notare come tutti gli algoritmi implementati riescano otttimamente nell'obiettivo, perciò abbaimo anche considerato la media delle sensitivity per le altre classi: questo è un indice che ci permette di capire se il meccanismo di individuazione di persone garantisca un certo livello di 'comfort'. Infatti il comfort è massimo quando l'effettiva presenza delle persone in stanza è correttamente individuata e tutti i servizi necessari vengono erogati.
Riportiamo in tabella le metriche citate:

Algoritmo	Sensitivity Classe 0	Media sensitivity altre classi
KNN	0.996	0.948
Support Vector Machine	0.998	0.951
Random Forest	0.996	0.938
Rete neurale	0.999	0.960
Proportional Odds	0.993	0.550

Tabella 3.2: Valori di Sensitivity per i vari algoritmi.

Il miglior algoritmo a conciliare riduzione degli sprechi e comfort risulta essere la rete neurale, anche se, tutti gli algoritmi, ad eccezione del proportional odds, hanno performance simili. La confusion matrix osservata sul test per la rete neurale è la seguente:

		Actual			
		0	**1**	**2**	**3**
Predict	**0**	1680	1	0	2
	1	0	78	0	0
	2	2	2	138	2
	3	0	0	7	112

Tabella 3.3: Confusion Matrix con etichette "Actual" e "Predict".

3.2 Efficienza

Infine abbiamo utilizzato la libreria 'tictoc' di R per effettuare una valutazione sull'efficienza degli algoritmi. In particolare tale pacchetto consente di misurare il tempo macchina impiegato per l'addestramento dei modelli. Di seguito è riportata la tabella con i rispettivi valori ottenuti a parità di potenza della macchina avendo già impostato gli iperparametri:

Algoritmo	Tempo
K-medie	0.2 s
K-medoidi	35.66 s
Kernel K-medie	603.88 s
Gerarchico	5.06 s
Model-based	2.78 s

Tabella 3.4: Algoritmi di classificazione e tempo computazionale

Algoritmo	Tempo
KNN	1.48 s
Support Vector Machine	0.63 s
Random Forest	1.73 s
Rete neurale	4.04 s
Proportional Odds	1.83 s

Tabella 3.5: Algoritmi di classificazione e tempo computazionale

Per quanto riguarda gli algoritmi di clusterizzazione osserviamo che a parità di efficacia, K-medie risulta nettamente più efficiente rispetto al gerarchico; si può allora concludere che l'algoritmo che combina al meglio efficienza ed efficacia è K-medie.
Per quanto concerne il problema di classificazione si verifica invece un trade off tra tempo macchina ed efficacia: le reti neaurali, che hanno performance migliori, richiedono un maggior tempo computazionale, mentre più rapide risultano essere le Support Vector Machine. Si nota, in ogni caso, che le tempistiche sono piuttosto contenute.

Bibliografia

Heaton, J. (2008). *Introduction to Neural Networks for Java*. Heaton Research.

Kuhn, M. & Johnson, K. (2013). *Applied Predictive Modeling*. Springer New York.

Patlolla, C. R. (2018). *https://towardsdatascience.com/understanding-the-concept-of-hierarchical-clustering-technique-c6e8243758ec*.

Singh, A., Jain, V., Chaudhari, S., Kraemer, F., Werner, S. & Garg, V. (2018). Machine learning-based occupancy estimation using multivariate sensor nodes. *https://archive.ics.uci.edu/* .